IACBOX Documentation

Release 1.0

IACBOX Team

May 15, 2019

Last update: May 15, 2019

CHAPTER

ONE

FIRST STEPS

In order to simplify the configuration process of the IACBOX this document will explain a wide array of basic settings step by step.

Hint:

- It is highly recommended to read and understand this document prior to installing the IACBOX.

1.1 Hardware Requirements

Before anything else it is important to verify the hardware which the IACBOX will be installed on. There are certain important requirements which must be fulfilled. The list of *Hardware Requirements* can be found on our homepage or by clicking on this link.

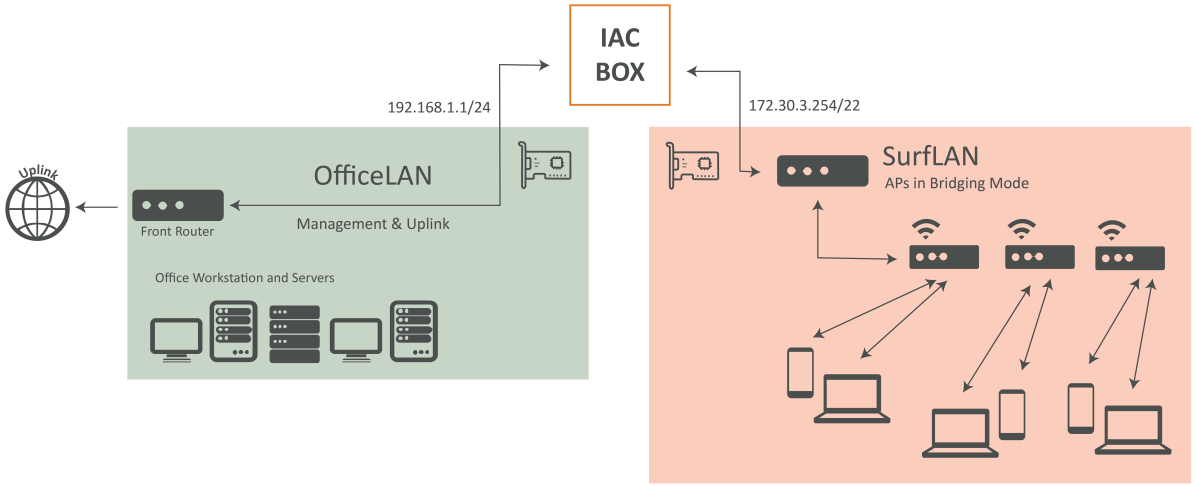
1.2 Basic Network Integration

The IACBOX network setup consists of at least 2 networks, the **Office-LAN** and the **Surf-LAN**. The Office-LAN grants the connection to the front router or firewall which will be used for the basic internet access of both, the IACBOX itself and guest devices. On the other hand the Surf-LAN is the network for guest devices. Traffic from the Surf-LAN will be managed by the IACBOX. Simplified this means that depending on the settings on the IACBOX a guest device can access the internet over the IACBOX with or without further restrictions.

The Surf-LAN always has to be bridged (except in Routing Mode (page 98)). This means that devices like Access Points and WLAN Controllers are **not allowed** to manipulate networktraffic from Surf-LAN devices. Services like DHCP, Proxy-ARP or Proxy-DHCP **must be disabled**.

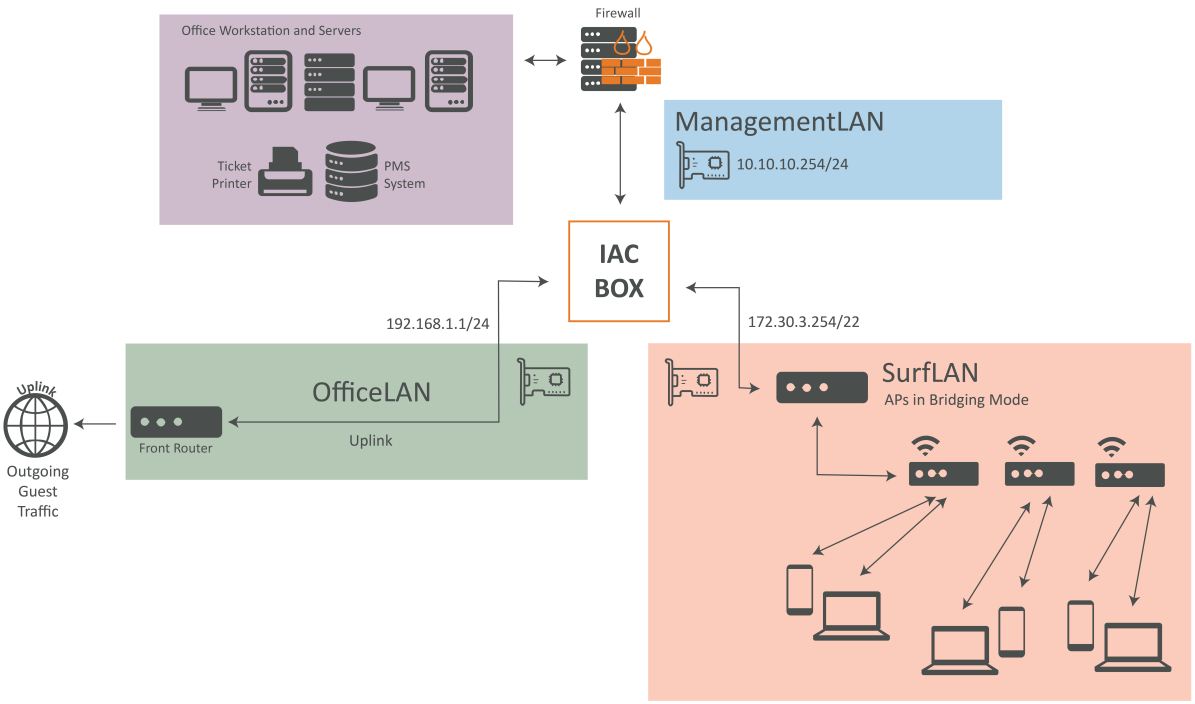
Attention:

- Best practise is to isolate the whole *Access Point management* with a custom VLAN, so that AP controllers can communicate with Access Points separated from Surf-LAN clients. This way it is also possible for Access Points to bypass the IACBOX in order to gather updates or communicate with external services (cloud configurations etc.).



1.3 With or without Management-LAN

Sometimes network environments do not permitt to add new devices into an existing and complex infrastructure. For exactly this problem the IACBOX can make use of an **optional** *third management interface*, the **Management-LAN**. So if the current network infrastructure does not allow you to add your ticket printers or PMS systems you can move them into the Management-LAN network. Note that the Management-LAN network does require a **third physical network card** in the IACBOX.



The Management-LAN can be activated with the first installation of the IACBOX and also later on. The exact process is explained in the according documentation page, the Management-LAN Activation (page 87).

1.4 Preparing the Installation

At first it is required to adjust the **BIOS settings** of the hardware which will be used for the IACBOX installation.

- The SATA controller must be set to AHCI

- All kinds of network boot options should be disabled
- Both UEFI and legacy BIOS mode are supported
- On HP servers ILO should be disabled

In order to install the IACBOX on hardware or in virtual environments (page 41) an **installation medium** is required. We do offer **ISO and USB images** for every new major release of the IACBOX, which can be downloaded from our homepage. IACBOX partners and system builders may obtain the ISO or USB images via the my.IACBOX customer portal.

The ISO image can simply be burned on an empty CD. In order for USB sticks to work it is required to mount the USB installation medium with a tool. The creation process of a bootable USB sticks is explained on this manual page (page 37). After the USB stick was created it is also possible to modify or create new **default installation profiles**, which will be described in the next section of this manual, the installation.

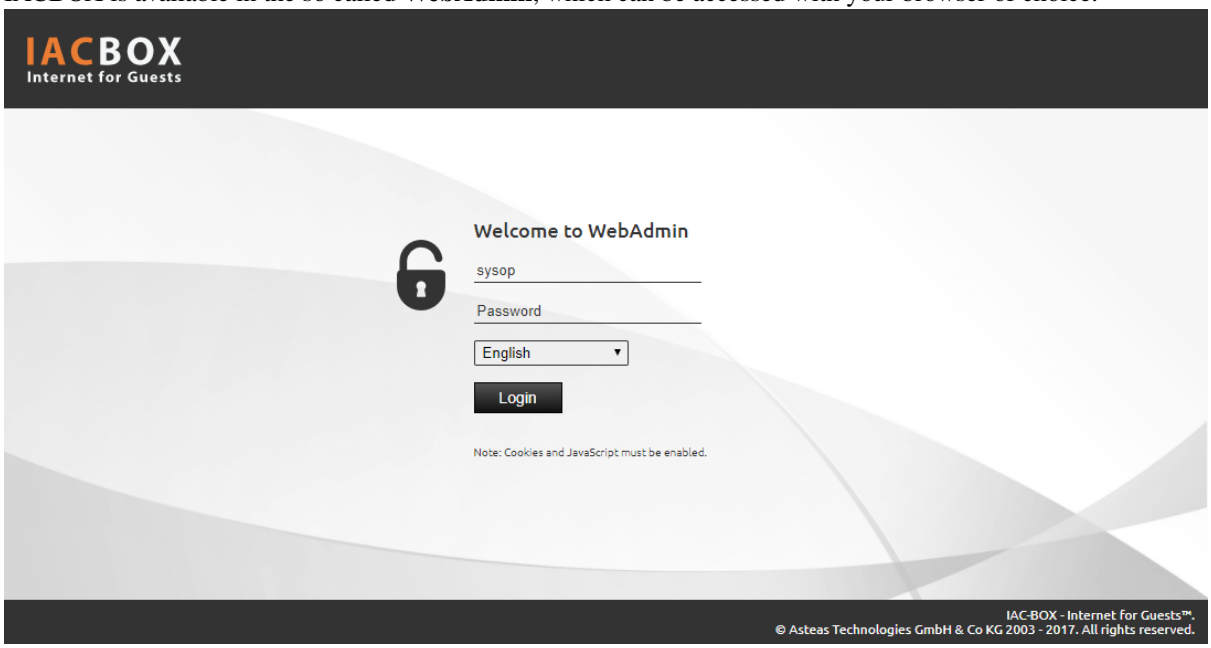
1.5 Installation

Now the installation medium can be used on the new server to install the IACBOX. Note that some systems, for example with existing operating systems, might not boot from the CD/USB stick. To do so manually press the according button to open the **boot menu** while the hardware is booting up. Usually this works with the F9 or F10 key. Now select the USB stick or the CD-drive.

The exact installation process is being described on this manual page (page 18).

1.6 Basic configuration

If the IACBOX was installed with a pre-defined *unattended profile*, the IP address after the installation will be **192.168.1.1** which means the WebAdmin will be reachable with **https://192.168.1.1**. The configuration of the IACBOX is available in the so called **WebAdmin**, which can be accessed with your browser of choice.



Attention:

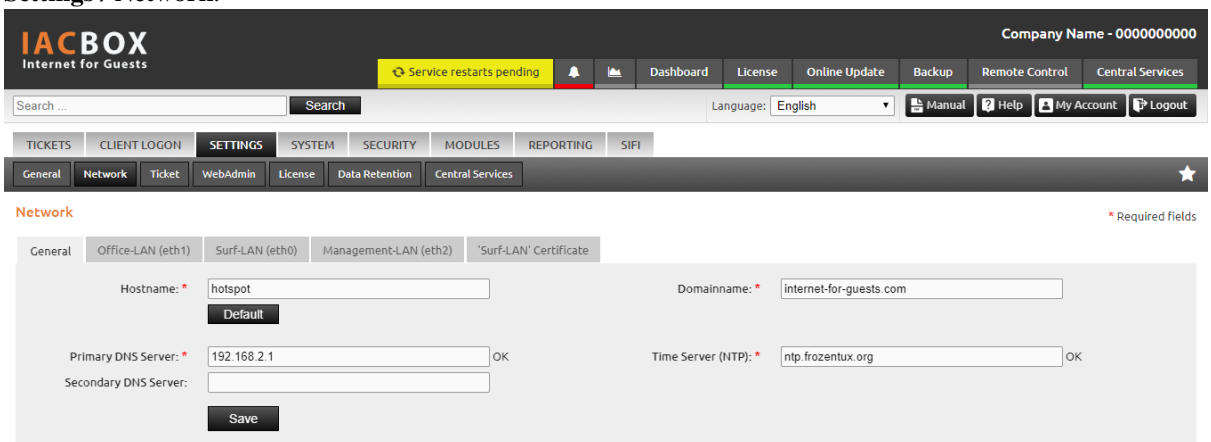
- The WebAdmin can initially only be accessed from the **Office-LAN** side of the IACBOX. In the WebAdmin itself it is possible to enable access for the *Surf-LAN* and *Management-LAN*.
- To access the WebAdmin, open **https://192.168.1.1** with your browser of choice. Note that the leading **https** is crucial.
- The default *username* and *password* for the WebAdmin login is **sysop**. The sysop password should be changed after the login by using the **My Account** button on the top right corner.

After logging in into the WebAdmin it is highly recommended to perform the **basic network configuration** and to apply the **licensing information**. Licensing information describes the **registration number** and **registration password** you've received from the **IACBOX sales department**.

Attention:

- If no **licensing information** is being applied, the IACBOX will **automatically shut down after 6 hours**. In order to apply licensing information you will need to proceed with the **network configuration** which is explained below.
- This does not only apply to the initial registration process. **The IACBOX must be connected to the internet at any given time** to verify the license.
- If this is not the case, the IACBOX will also **shut down** after 6 hours.
- **Failed online registrations mostly mean that the IACBOX could not reach the license servers. Possible reasons are lis**
  - The internet connection is down or not plugged in
  - The configured DNS servers are not resolving properly
  - Incorrect system time (and therefore failed certificate checks)
  - Missing ethernet interfaces (the IACBOX must consist of at least 2 active network interfaces)
  - In some cases, firewalls may block or try to intercept TLS/SSL encrypted traffic. The interception of SSL/TLS has to be disabled for the IACBOX to get a working licensing and update.

So before the licensing process, configure the basic network settings. Therefore navigate to the menu entry **Settings / Network**.

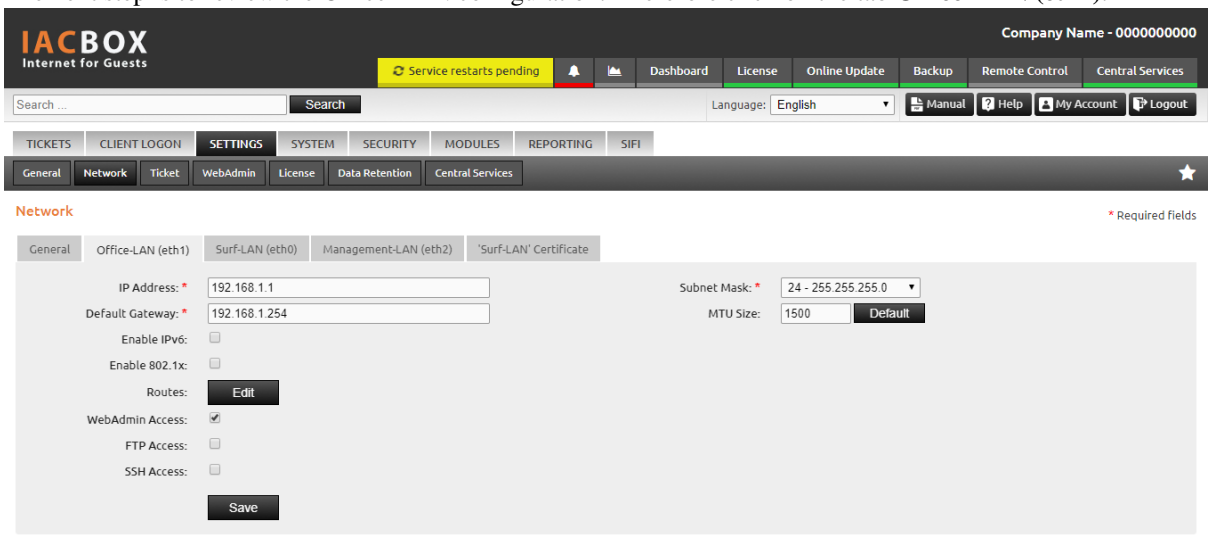


Configure the used DNS servers to your ISP's DNS servers.

Attention:

- Some public DNS servers like google's **8.8.8.8** and **8.8.4.4** use **rate limiting** which limits the DNS replies after a time. If DNS servers do not respond, guests can not use the internet access anymore.
- If the DNS servers are not reachable, the landing page for guests will take very long to load. This is due to the fact that **DNS connectivity** is being checked upon accessing the **customer logon page** (landing page).
- Changing DNS servers will require a system restart, which can be done in the WebAdmin menu **System / Services**. If multiple changes require a restart then it is enough to restart only once at the end.
- The **Hostname** and **Domainname** are associated to the installed **certificate** on the IACBOX. Do **not** touch these settings if you are not sure.
- The range **172.17.0.0 - 172.17.127.255** is reserved for internal use and can not be used in any configuration.

The next step is to review the Office-LAN configuration. Therefore click on the tab **Office-LAN (eth1)**.

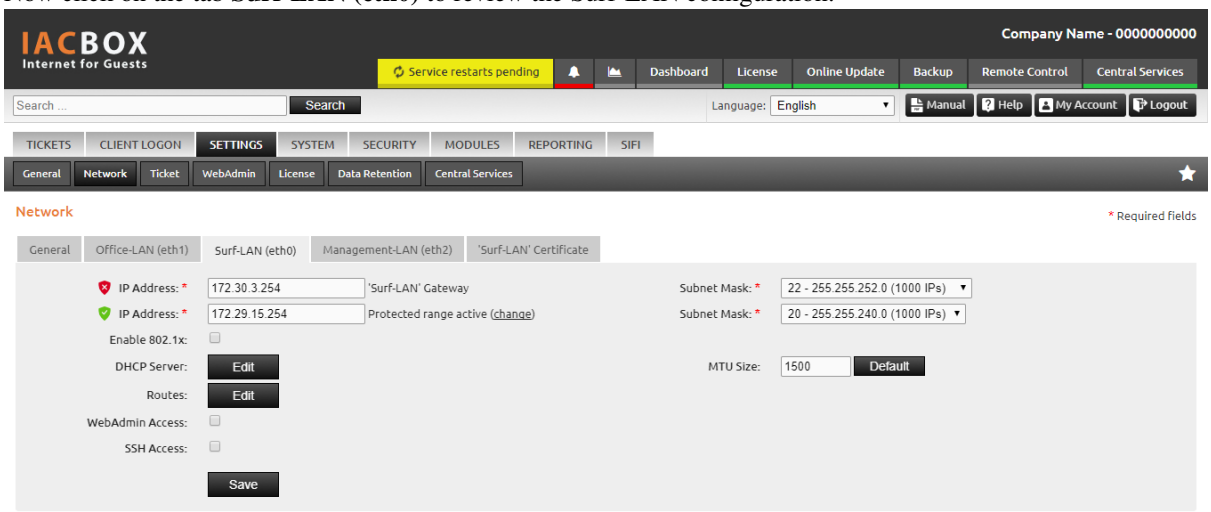


Here you can change the basic network configuration and most importantly the **Default Gateway**. This can either be a *front router* or a *firewall* and should not limit or block the connectivity for the IACBOX. If you previously installed with an *unattended profile* then you may also want to change the *IP address* in this window.

Attention:

- After changing the IP address do not forget to update existing WebAdmin bookmarks.
- As for all network environments, overlapping network ranges or duplicated IPs are not permitted.

Now click on the tab **Surf-LAN (eth0)** to review the Surf-LAN configuration.



Attention:

- The default Surf-LAN configuration is usually perfect as-is and should only be changed if it is absolutely required.
- By default the Surf-LAN uses the **Protected range**, which does put clients into a **subnet** so they can not communicate with each other. This also **avoids spoofing**, so it is recommended to keep this setting. If the **unprotected range** should be used, then you can disable the **Client/Client Protection** in the WebAdmin menu **Security / General**.
- It is not recommended to enable **WebAdmin access** for the **Surf-LAN**. For the **Management-LAN** at the other hand it is common to do so.

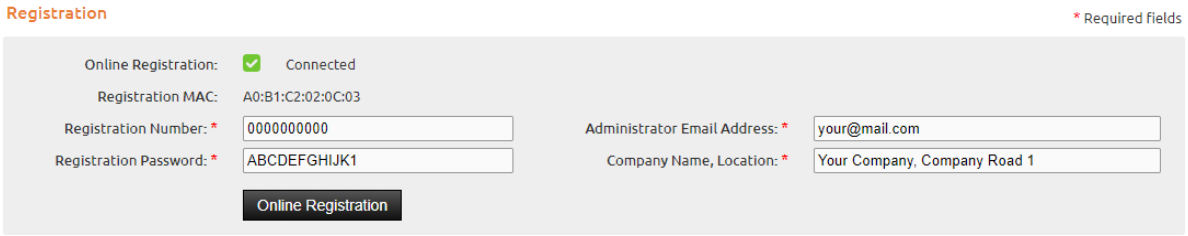
An example of the **Protected range** Surf-LAN client subnet for the default setting 172.29.15.254/20 (1000 IP addresses):

	User 1	User 2	User 3	User 4
Network Address	172.29.0.0	172.29.0.4	172.29.0.8	172.29.0.12
IP Address	172.29.0.1	172.29.0.5	172.29.0.9	172.29.0.13
Client gateway	172.29.0.2	172.29.0.6	172.29.0.10	172.29.0.14
Broadcast Address	172.29.0.3	172.29.0.7	172.29.0.11	172.29.0.15

After this is done and the IACBOX was restarted, it should now be able to connect to the internet. To test the **connectivity** open the WebAdmin and navigate to **System / Tools**. Here you can select **Ping** and perform it on a public **domain** or **IP address**, for example **8.8.8.8**. If the ping is successful proceed with the next step, otherwise review your network configuration or check your firewall gateway.

Now navigate to the WebAdmin menu **Settings / License** and fill out the required license fields at the top of the page:

- the registration number
- the associated registration password
- your administrator email address
- the company name and location



These settings will then be associated to the license. **Also the selected MAC address will be used to identify and bind the hardware onto this license**.

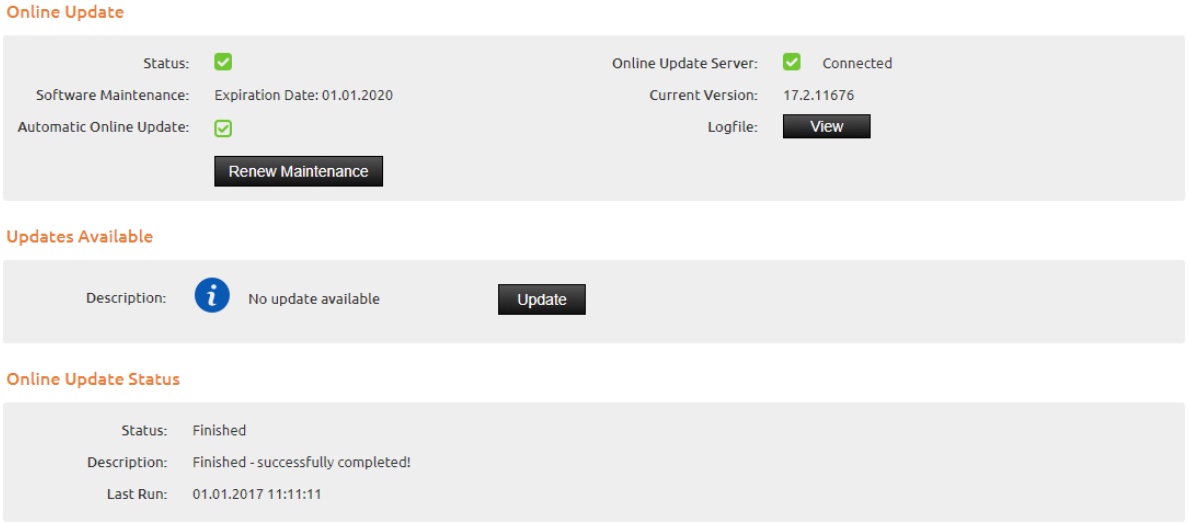
Attention:

- If the network interface cards change, then the MAC address needs to be unlocked by hand. This must be done manually by the system reseller or the IACBOX support team.
- In case the licensing does not work, check your **firewall**. The IACBOX must have unrestricted access. Also **heuristic firewall intrusion detection algorithms** can cause the online registration to fail.

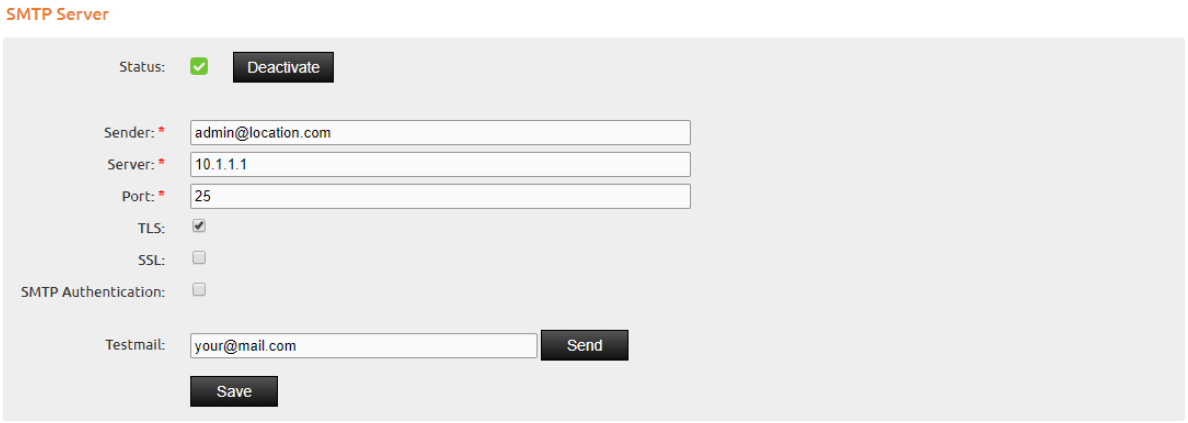
After the licensing process the IACBOX will require a *system restart*, which can be done in the WebAdmin menu **System / Services**. Now it is highly recommended to start the **Online Update** in the according WebAdmin menu **System / Online Update**, but before doing so note the following hints:

Attention:

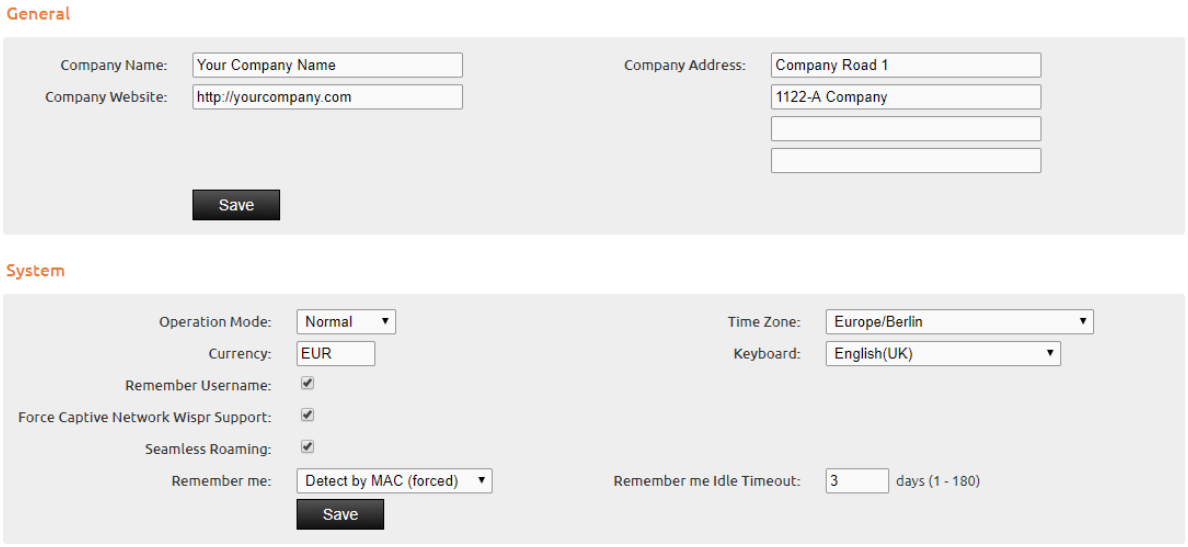
- The **Online Update** will download, extract and install all available updates **one-by-one**. The IACBOX will, if neccessary, perform a system restart, wait 10-15 minutes, and then continue to install the next update. Depending on the amount of available updates this process *can take a considerable amount of time*.
- The IACBOX will automatically search and install updates before the **weekly restart**.
- To avoid an inconsistent database and file system **the update process must never be interrupted**.



After the update process finished, it is time to continue with the basic configuration. The next step is to configure the **SMTP server** in the WebAdmin menu **Settings / Network**. Here you also have the possibility to configure a **SMTP proxy**, but usually this is not neccessary. By using the **Testmail** function you can verify your settings and send yourself an Email from within the WebAdmin.

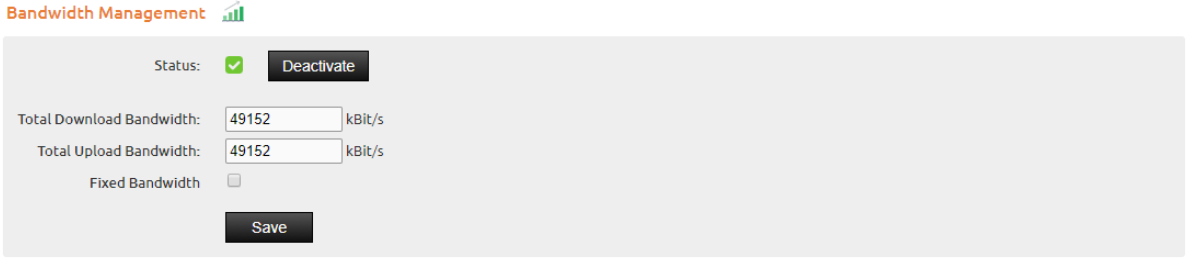


Now navigate to the WebAdmin menu Settings / General. Here you can fill out the **Company Name**, **Website** and **Address**, which will be used in some modules and also the *Customer Logon Page* (Landing Page) later on.



Note that the **Operation Mode** should not be changed from *Normal*. The available options *Free* and *Autologon* will automatically generate Surf-Tickets for client devices and log them in. These modes are **deprecated** and should therefore only be used as a **last resort**. All settings will have an according description in the **help menu** of this WebAdmin page, which can be found by clicking the **help icon** on the top right corner. If you are not sure what to configure, then it is recommended to *copy the configuration from within the screenshot*, because later on it can be applied to pretty much all use-cases.

After the IACBOX is now configured with the correct network settings, licensed and up to date, it is time to configure the **Bandwidth Management**. It is crucial to configure the **Bandwidth Management** in the WebAdmin menu **Settings / Network** according to the *available on-site bandwidth*. To do so, test the bandwidth on-site on different times of the day in order to find the best values to use for Down- and Upload.



Attention:

- If the **Bandwidth Management** does get disabled, there are *no more bandwidth regulations* which means that every client can use the maximum available bandwidth given by the **ISP**.

1.7 Guest Authentication

The next step is to figure out how to provide Internet access to guests in the Surflan network. The IACBOX offers an incredible wide array of modules and interfaces to cover the most common requirements out-of-the-box. In order for guests to access the internet, a Surf-Ticket is always required. Surf-Tickets can be generated manually beforehand (WebAdmin) or by guests (for example with the Facebook Login). Existing Surf-Tickets will always be listed in the WebAdmin menu **Tickets / Manage**. In this menu it is also possible to **log off** or **revoke** existing tickets.

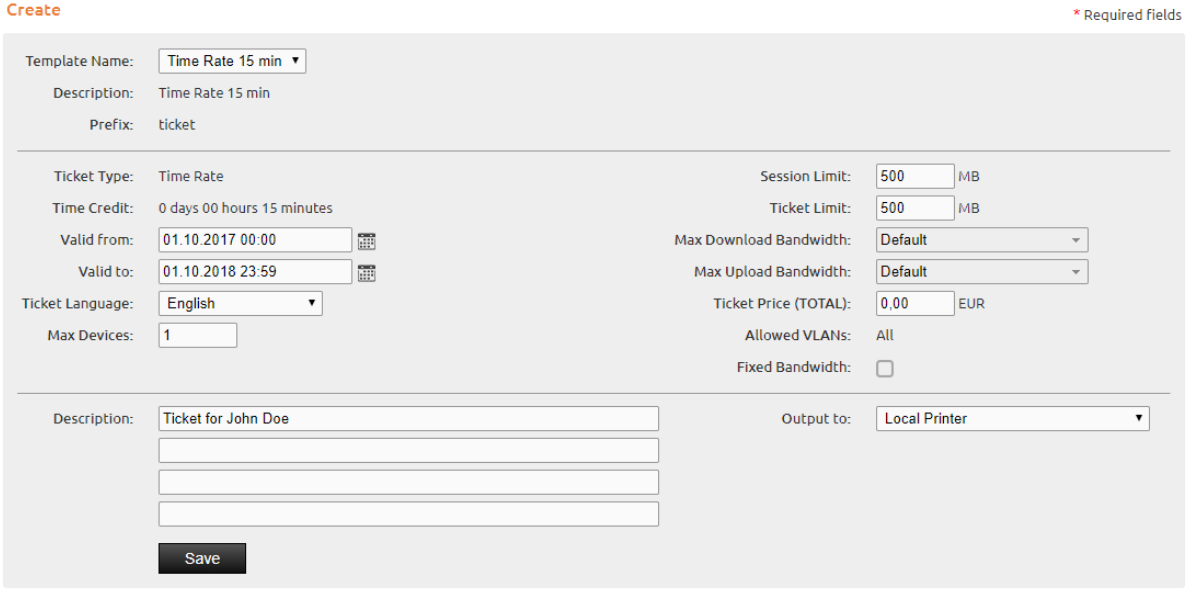
The following list contains some basic authentication possibilities of the IACBOX:

- Ticket Login with *Username and Password* or only with a *Password*
- Login with *Facebook*, *Google+* or *Microsoft Account*
- Authentication with existing *PMS Systems*

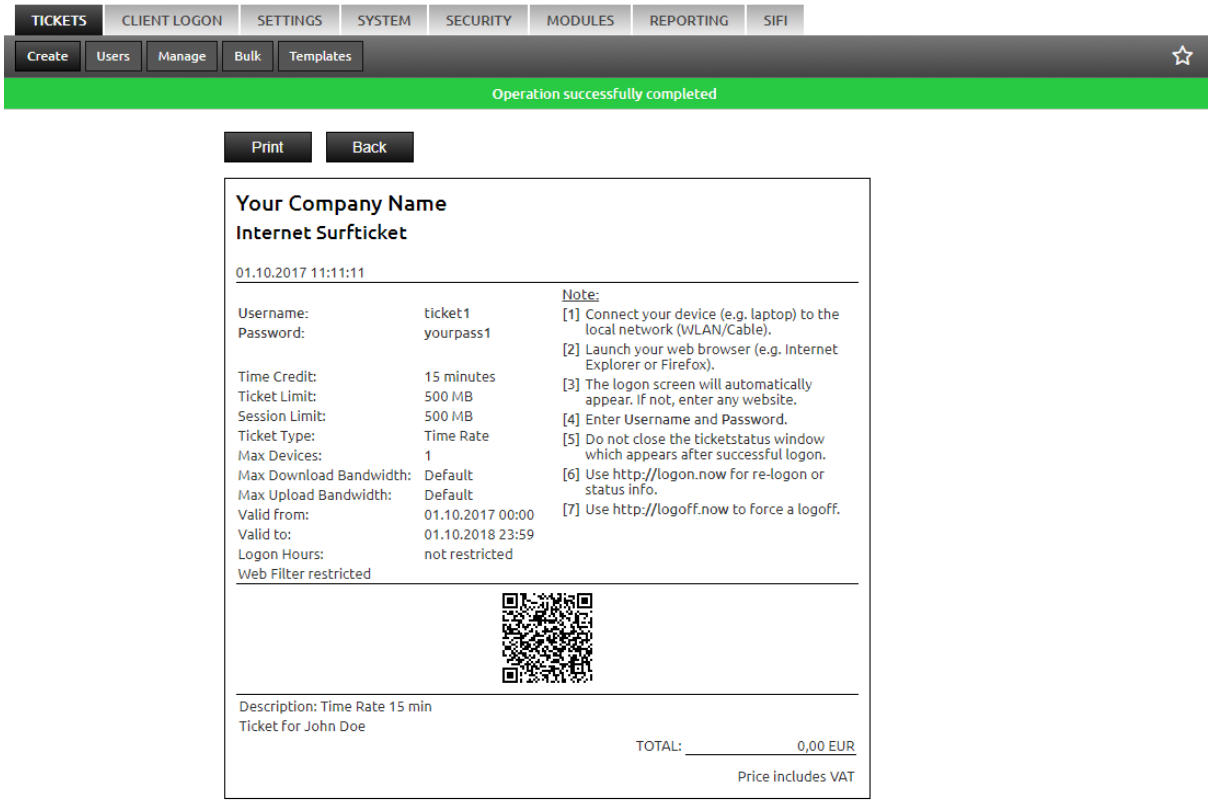
- *SMS* Login
- *Email* Login
- Buy tickets with *PayPal*
- Authentication with data from various *SQL Databases*, *AD/LDAP* and *Radius*

1.7.1 Ticket Login

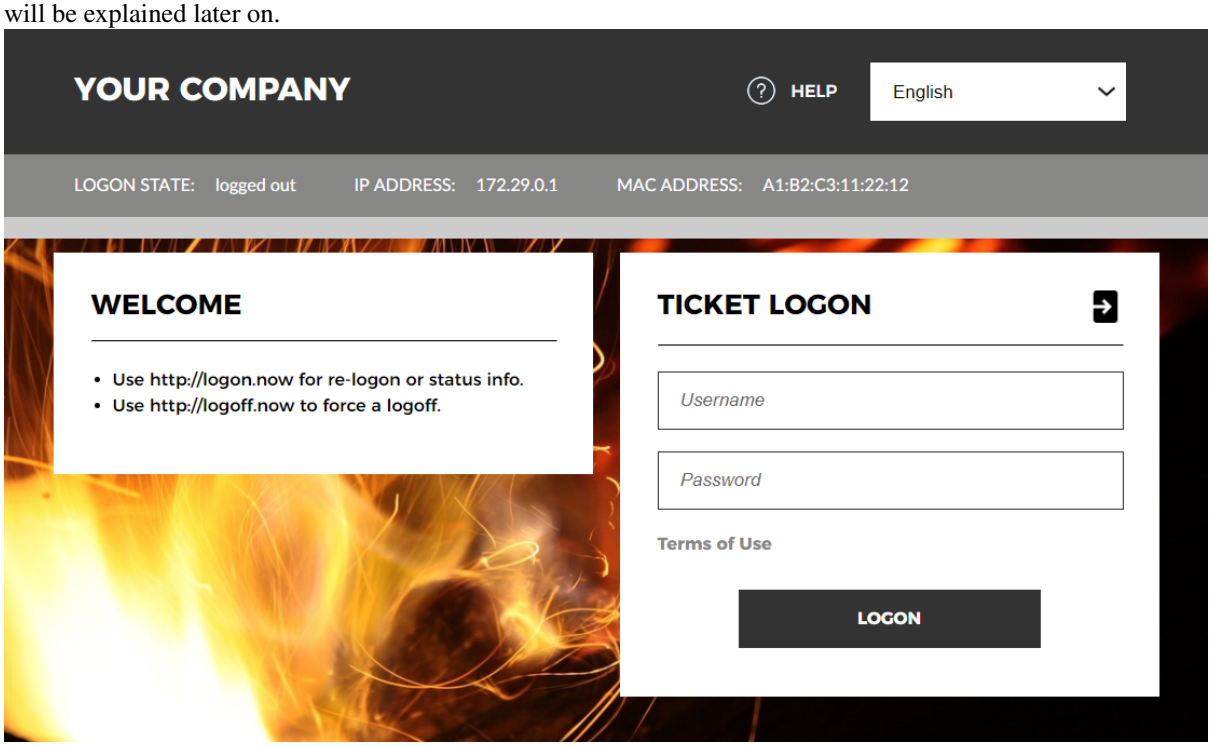
The most common used authentication method in smaller environments is the *Ticket Login*. This means that guests have to enter a combination of *Username* and *Password* - or only a *Password* (also referred to as *PIN Login*) on the *Customer Logon Page*. Tickets can be manually created by administrators in the WebAdmin of the IACBOX using the WebAdmin menu **Tickets / Create**. By using **Ticket Templates** you create tickets based on pre-defined default values, so-called templates.



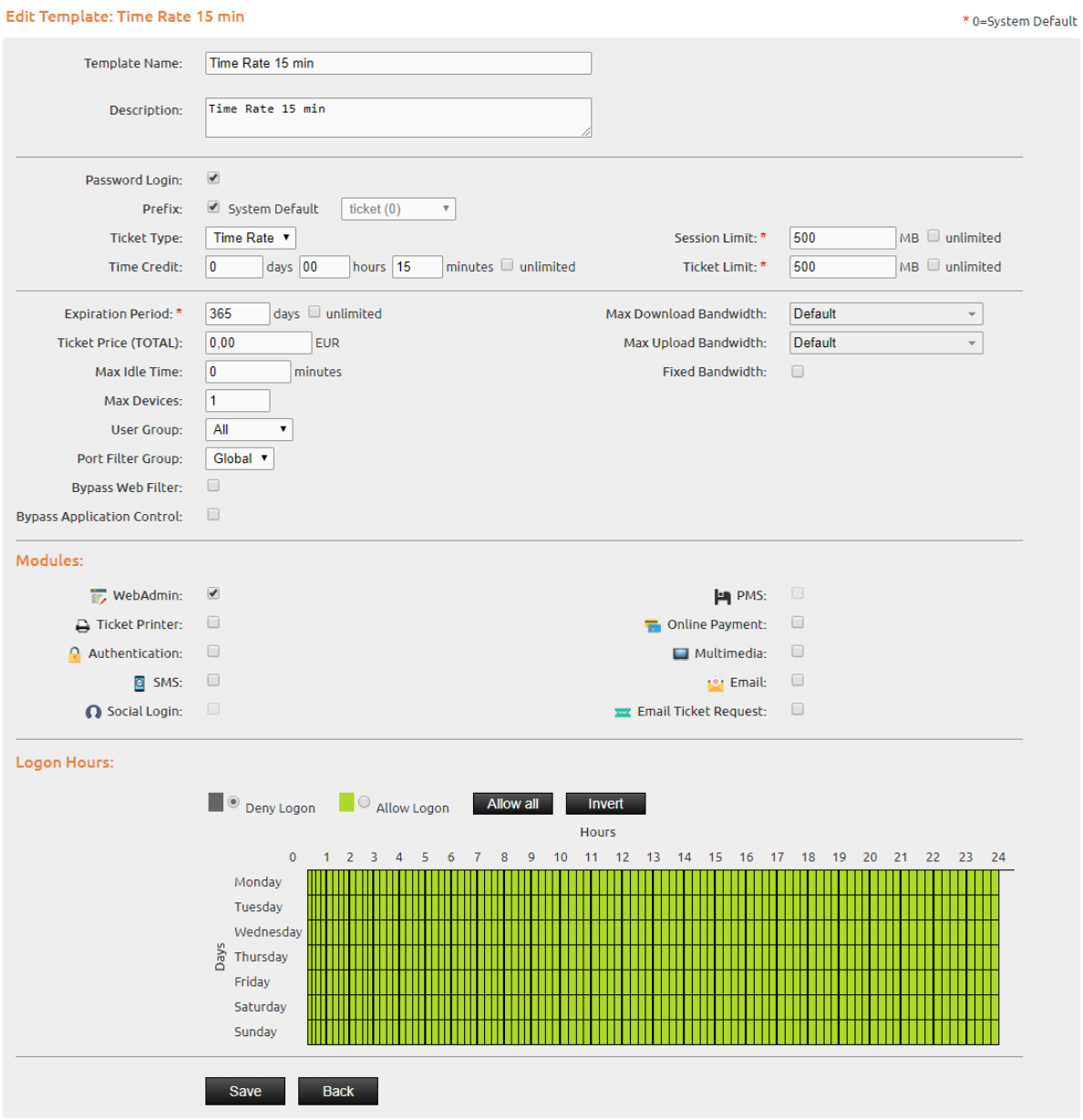
After creating the ticket, it can be printed and given to guests as a hand-out.



In the Surf-LAN network of the IACBOX guests can now log in by using the *Username* and *Password* or by scanning the *QR-Code* as shown above. Note that the customization possibilities of the **Customer Logon Page** will be explained later on.



In order to review, add and edit **Ticket Templates**, navigate to the WebAdmin menu **Tickets / Templates**. Here you can find the default ticket templates of the IACBOX.



Besides regular restrictions, a template must be enabled for *each module to use it with*, which means that if you want to use an existing or new ticket template to manually create tickets in the WebAdmin, the *checkbox for WebAdmin* must be activated.

Attention:

- In order to understand ticket values like *Time Rate*, *Flat Rate* as well as further possible combinations, it is highly recommended to take a look at the help page of the WebAdmin to explore the meaning of all **Ticket Parameters**.

1.7.2 Social Login

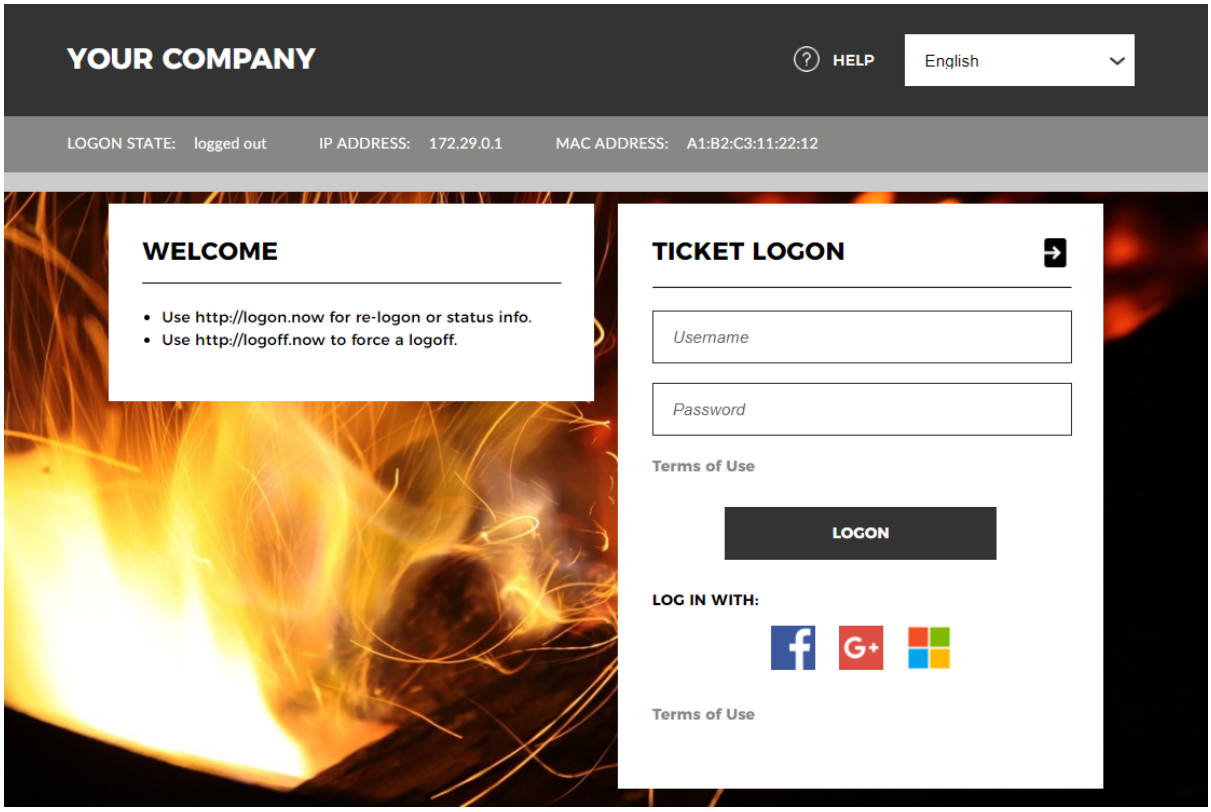
The **Social Login** is probably the most popular authentication on the IACBOX. It does allow guests to authenticate and create a Surf-Ticket by logging in with a social media account. The available Options are:

- Facebook (Manual for Facbook Configuration (page 140))
- Google+ (Manual for Google+ Configuration (page 153))
- Microsoft Account

- Twitter (Login-API only) (Manual for Twitter Configuration on the Login API (page 198))

Attention:

- In order to provide authentication for *Microsoft Accounts*, you **must** obtain and install a **custom Surf-LAN certificate** on the IACBOX. The problem with the Microsoft authentication is that hostnames can only be registred once to one single interface.

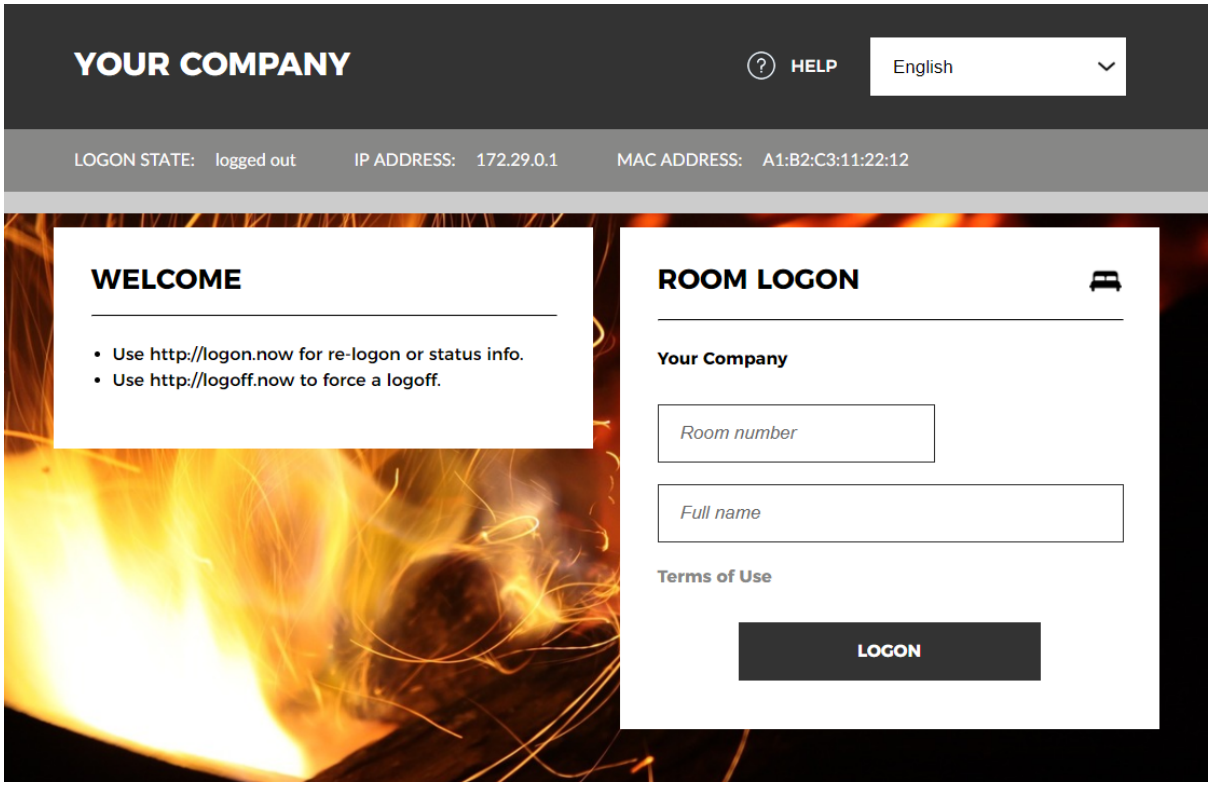


Hint:

- By November 2016 the **Social Login** module is free to use for all licenses. If you have an older License with valid maintenance navigate to **License** and click on **register**.

1.7.3 PMS Authentication

In a hotel environment often a **PMS System** is used to keep track of guest check-in's, check-out's and bookings. **Property Management Systems** or short **PMS Systems** save data like the arrival or departure date, the full name, room numbers or even the birthday of a guest. For guests this information can be used to authenticate with the IACBOX:



While the **Room Number** is always required, it is possible to combine following data fields for the authentication:

- **Name**
- **Name & Departure Date**
- **Name & Departure Date & PIN Code**
- **Name & Birthdate**
- **Name & Birthdate & PIN Code**
- **Name & Arrival Date**
- **Name & Arrival Date & PIN Code**
- **Birthdate**
- **PIN Code**

Guests then can choose between the available **Ticket Templates** which are configured for usage with the **PMS Module**. If ticket templates define a price, an according booking will be sent to the PMS system. This way guests can postpone paying tickets until checking out. The PMS manual can be found here (page 170).

1.7.4 SMS Login

The **SMS Login** enables guests to create a *Surf-Ticket* by using their mobile phone. In order to receive a SMS with the according **login credentials** (*Username* and *Password* or just *Password*), the mobile phone number has to be entered on the *IACBOX Login Page*.

Attention:

- An external SMS vendor is required to send the actual SMS. The IACBOX offers a list of supported vendors, see the :doc:'SMS configuration manual <../logon/messaging_sms>'.

Further information can be found in the according SMS configuration manual (page 157).

1.7.5 Email Login

The **Email Login** enables guests to authenticate by using an email address. To receive the **login credentials** (*Username* and *Password* or just *Password*), an email address has to be entered on the *IACBOX Login Page*, so that the IACBOX can send an email to this address.

Attention:

- After the email address was entered on the *IACBOX Login Page*, guests will have free internet access for a configured amount of time. This enables guests to access **Web-Mails** like **Gmail** or **Hotmail** without any restriction. Guests then have to log in by using the credentials in the email which has been sent by the IACBOX.
- Besides the ticket credentials the email will also contain a **hyperlink** which automatically authenticates the user with the attached credentials.

Further information can be found in the according Email configuration manual (page 132).

1.7.6 Online Payment via PayPal

Guests can also buy tickets by using the external **payment service provider PayPal** (PayPal configuration (page 165)). The payment interface on the **Login API** also offers different payment providers (although they are not fully tested yet), for example:

- SofortBanking
- Stripe
- WorldPay
- 2CheckOut
- Authorize.Net

For further explaination please refer to the Login API manual pages (page 180).

Without the *LoginAPI* and by only using the old *Login Page* you can still use **PayPal**. In order to set up PayPal you may follow the PayPal Integration manual (page 165).

1.7.7 External Authentication

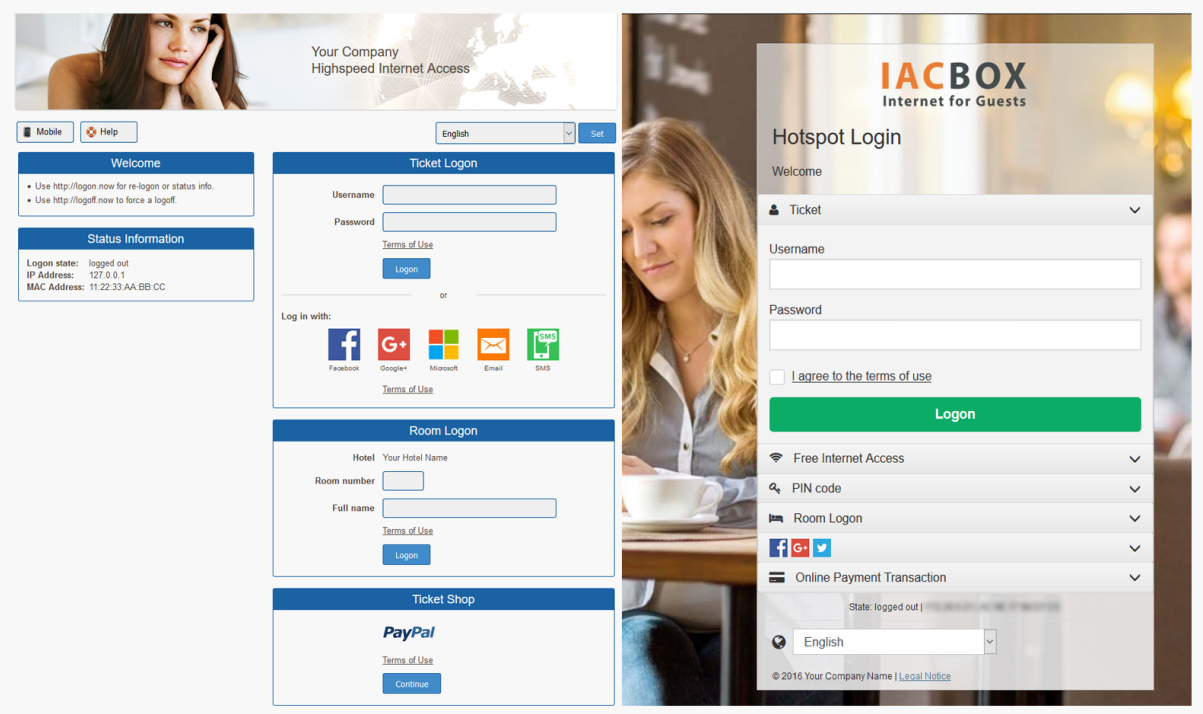
The **External Authentication** module allows you to authenticate guests on the Surf-LAN side by using existing backends:

- Active Directory/LDAP
- MSSQL/MySQL/PostgreSQL
- Radius
- iPass

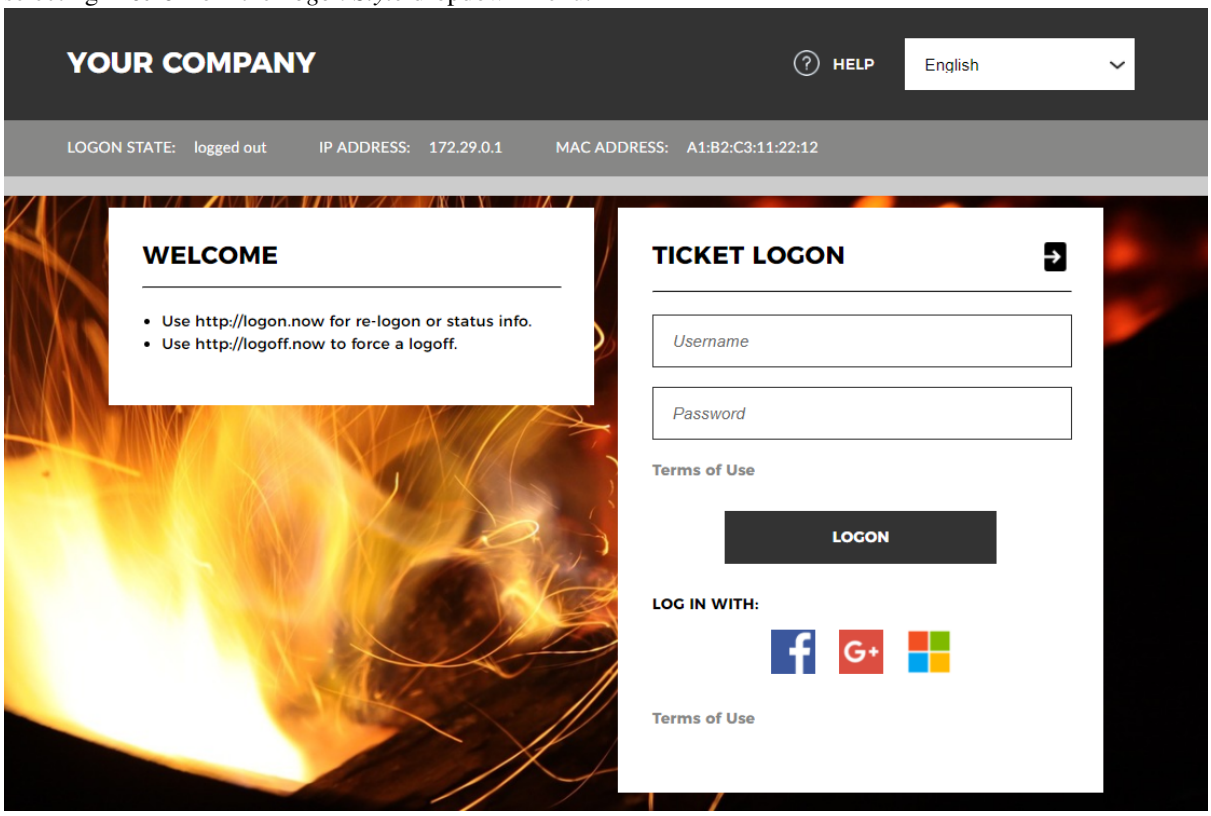
For further explaination please refer to the External Authentication manual page (page 135).

1.8 The Landing Page

The landing page is often referred to as “IACBOX Login Page” and lists all enabled authentication methods. There are two very different possibilities to customize the Landing page, the default **IACBOX Login Page** (left) and the **Login API** (right).



With IACBOX **version 17**, the new **Metro Style** was introduced. It can be activated in **Client Logon / Design** by selecting **Metro** from the *Logon Style* dropdown menu.



Customization Options for the **IACBOX Login Page** can be found in the WebAdmin menu **Client Logon / Design**. The **Login API** page is being build upon **PHP** which means that the design and functionality is completely customizable. Further information can be found in the according manual page (page 180) for the Login API.

CHAPTER

TWO

SETUP / INSTALLATION / RESCUE

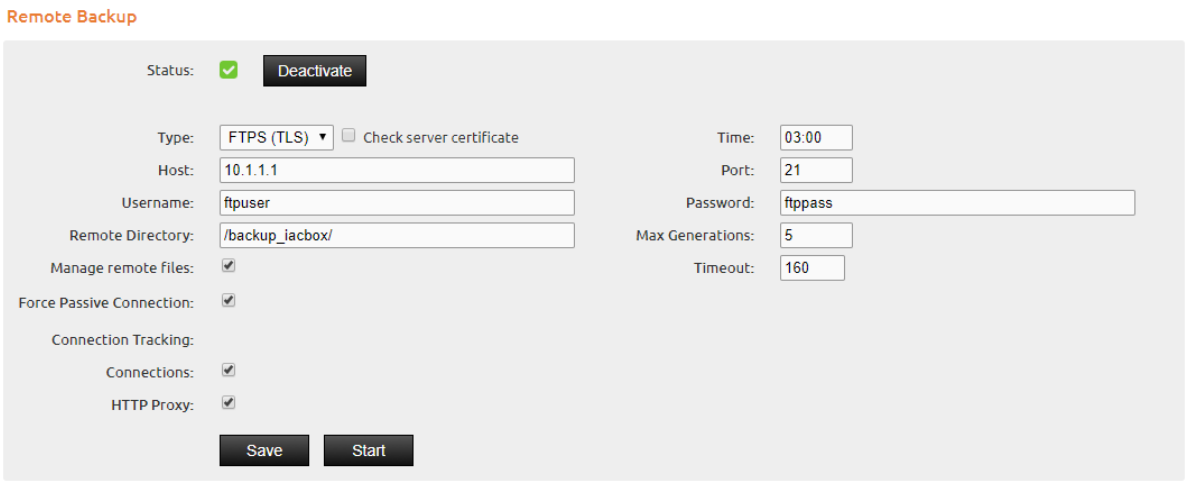
2.1 Backup

This manuals describes how you can create and restore Backups. Backups can be created manually or copied automatically via FTP/FTPS once a day.

Hint:

- Backups can only be restored on a system with the **same version**, excluding the patchlevel
- Backups which were created while or before a hardware failure could be inconsistent and should therefore not be used
- The system administrator is fully responsible to configure and create backups

2.1.1 Preview



2.1.2 Content of a backup

Following data will be saved with the backup file:

- Instances of tickets
- Ticket-Templates
- Configuration (WebAdmin)
- Logos & Designs
- License Information

2.1.3 Creating a Backup

A backup can be created in the WebAdmin-Menu **System/Backup**. Click on download to create and save a backup file to your locale computer. The filename contains the IACBOX version, the patchlevel, the date and the time of creation.

For example: `IACBOX_2010091401_20170123030014_V17.0.11166.bkp`

2.1.4 Restore a Backup

To restore a backup open the WebAdmin-Menu and navigate to **System/Backup**. At **Restore** browse for the backup-file on your local computer and hit start. Please note that the version of the backup and the target system must match.

2.1.5 Automatic FTP-Backup

In the WebAdmin-Menu at **System/Backup** activate the **Remote Backup** and enter all necessary credentials of your FTP account. With a click on Start you can verify that the creation of the backup and the file-transfer to the FTP-server is working.

- If you have configured a weekly restart in **System/Services**, the restart will be delayed until the automatic backup has finished on that day.
- It's suggested to enable the automatic backup to a time in which there is low user activity on the IACBOX.

2.2 IACBOX Installation

This manual describes how to install the IACBOX. The installation can be performed with pre-defined **unattended profiles** or **in expert mode**. For starters it is highly suggested to install the IACBOX with one of the 4 pre-defined unattended profiles.

Hint:

- The installation medium of the IACBOX can be downloaded from our homepage and is available as ISO and USB image. To create a **bootable USB stick** you may follow the instructions on the according manual page (page 37) which also explains the **unattended setup**.
- Also note the hardware requirements on our homepage.

2.2.1 Basic BIOS settings

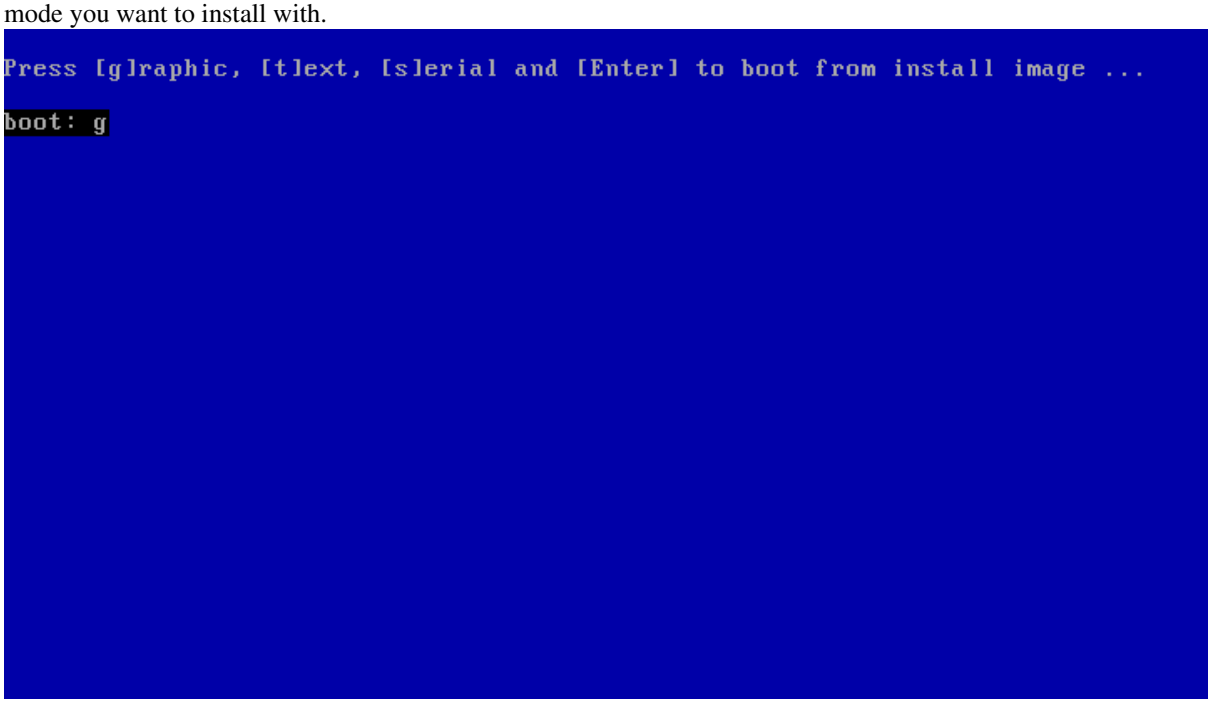
Ensure that the BIOS settings are configured according to the following points.

- The SATA controller must be set to AHCI
- All kinds of network boot options should be disabled
- The UEFI should be set to Lecacy Mode
- On HP servers ILO should be disabled

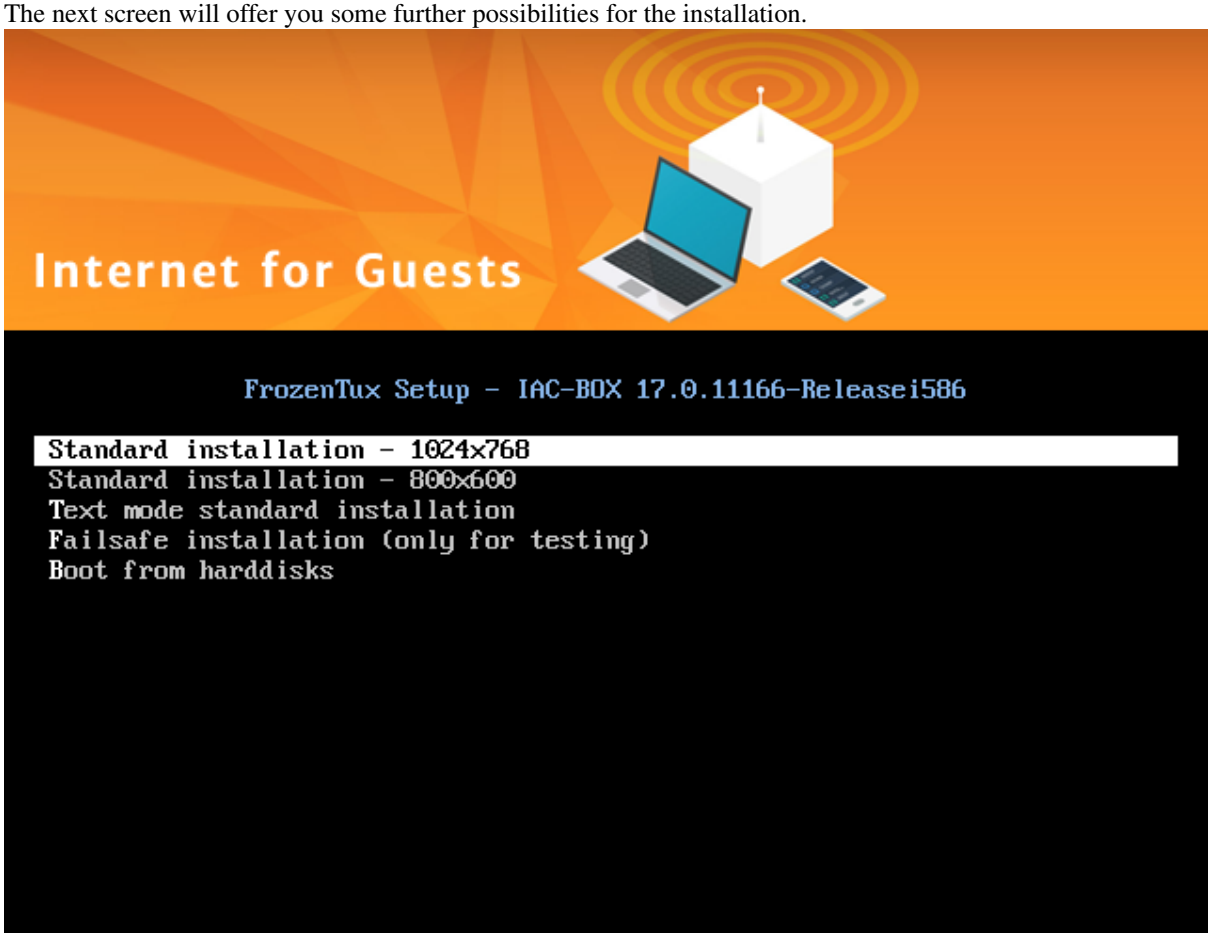
Now the installation medium can be used on the new server to install the IACBOX. Note that some systems, for example with existing operating systems, might not boot from the CD/USB stick. To do so manually press the according button to open the **boot menu** while the hardware is booting up. Usually this works with the F9 or F10 key. Now select the USB stick or the CD-drive.

2.2.2 Starting the Installation

When the systems boots from the installation medium, the first screen of the installation process will ask which mode you want to install with.

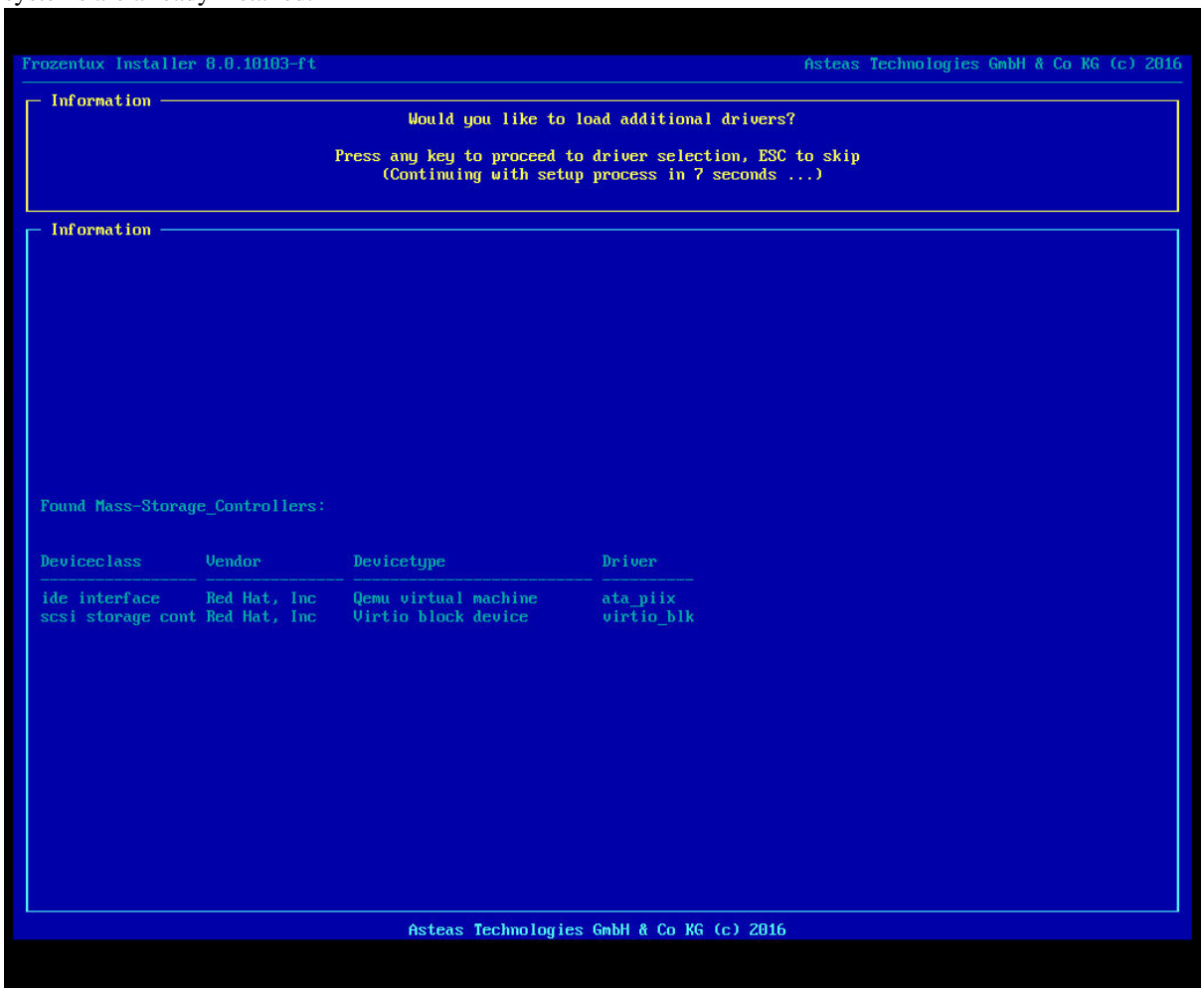


As in the screenshot above, enter **g** to start the installation in the **graphics mode** and then confirm with **ENTER**. The next screen will offer you some further possibilities for the installation.



Every *regular installation* should use the **Standard Installation - 1024x768** option. If this fails, you may try the

lower resolution alternative or even the **Text mode**. Note that the *Failsafe Mode* should never be used if you want to install a productive system. As the name suggests, it should only be used if no other mode works - and even then only to find the problem. The last option will do what it says, **Boot from harddisks** in case other operating systems are already installed.

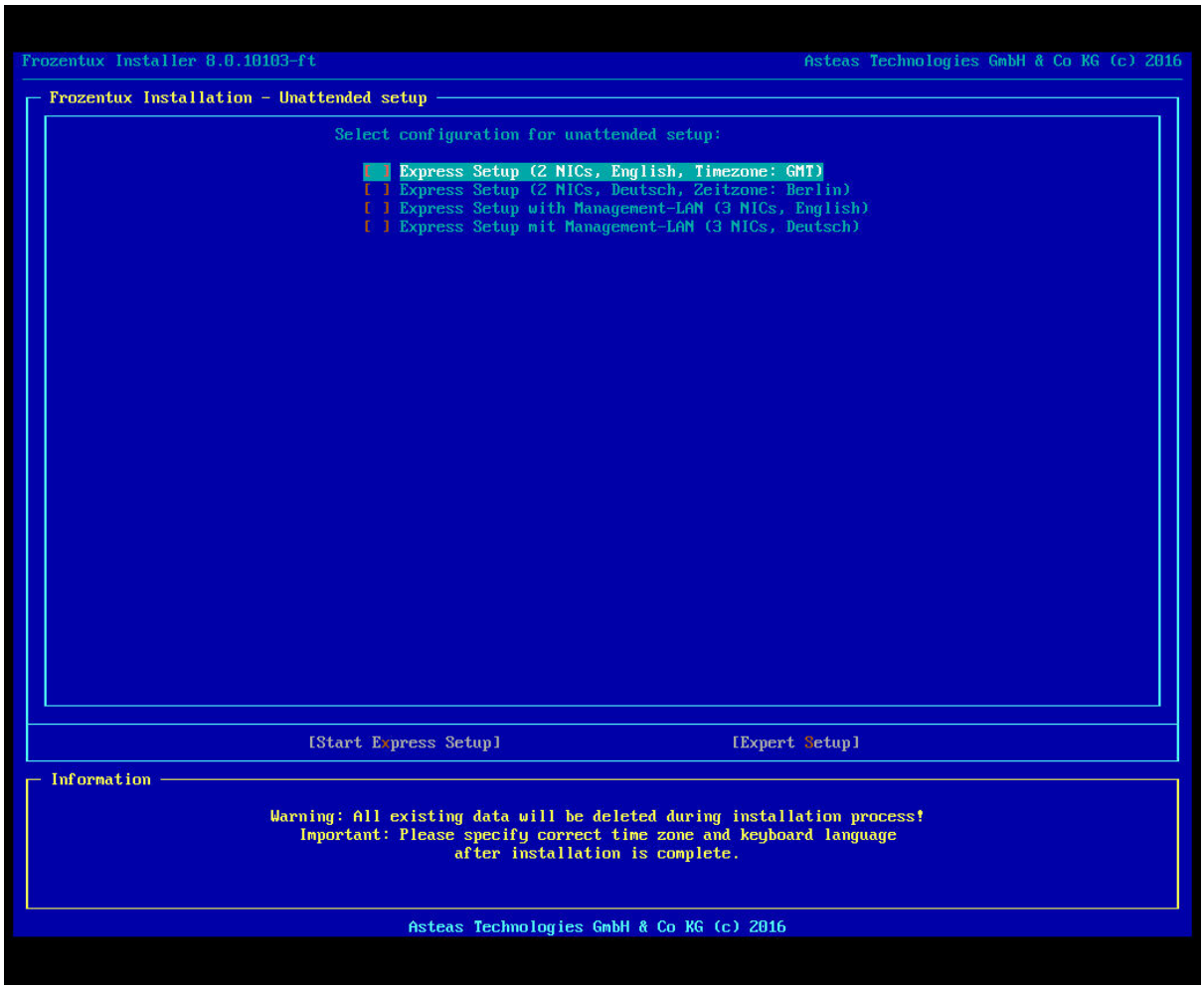


Now the *FT-Setup* will load up. While loading you will be asked if you want to install *additional drivers*. Most of the times this is **not necessary**, so if you are not sure then skip installing additional drivers. Press any key to proceed or use **ESC** to skip this screen.

Hint:

- In order to navigate in the FT-Setup use the **arrow keys** to nagivate in lists. To highlight/select an entry use the **SPACE bar**. Use **TAB** to navigate between the list entries and the save/back handlers. To proceed use the **ENTER key**.

2.2.3 Express Setup

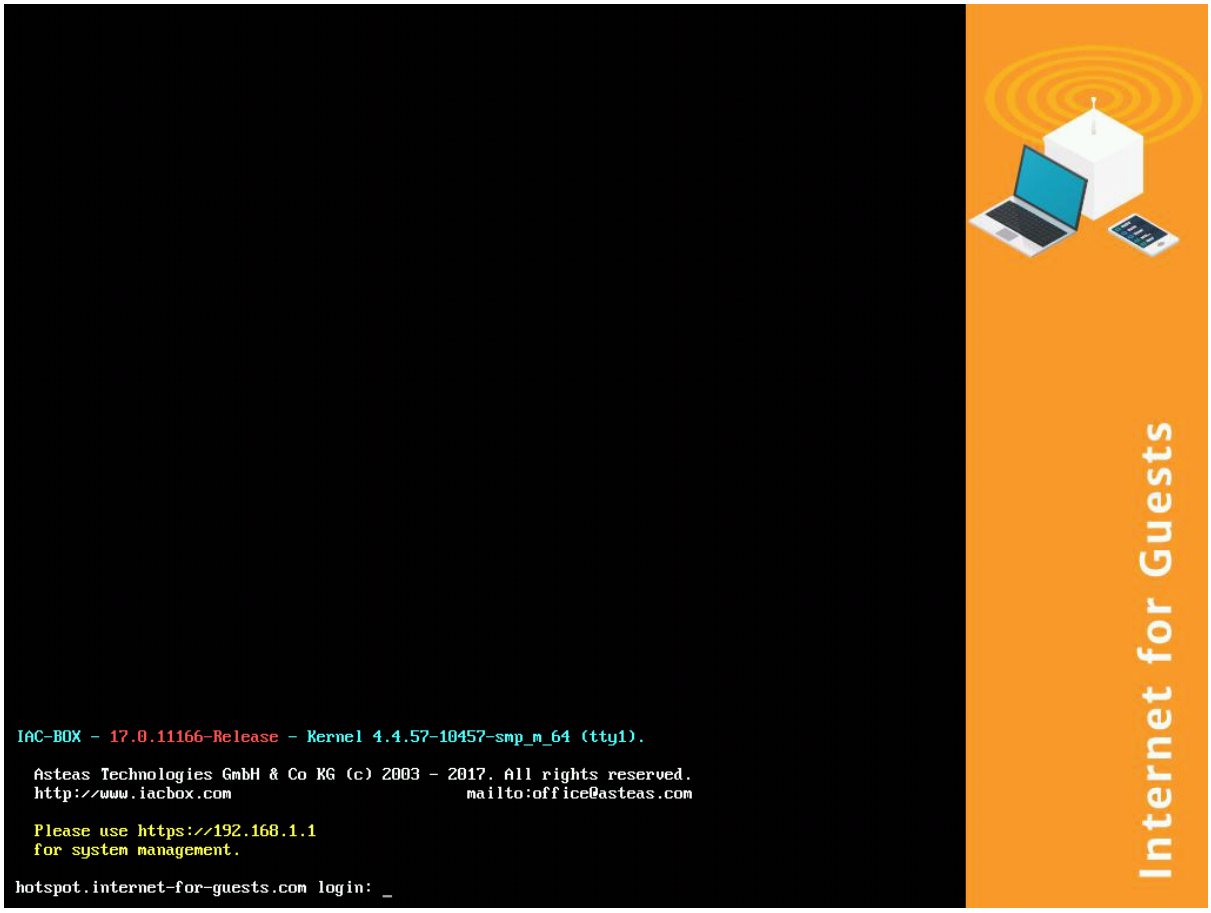


The next step is the Unattended Setup screen. Here you can select an unattended and pre-defined configuration to install the IACBOX with. For starters it is recommended to start with one of the available options, for example the **Express Setup with 2 NICs in English**. After selecting the entry with the SPACE bar, press TAB to highlight the **Start Express Setup** section and then ENTER to proceed. The IACBOX will now be installed with default settings. The settings can still be changed later on.

After the installation and the automatic restart, the following options will be available. Please note that this screen will be skipped automatically after 10 seconds.

Again you see 2 resolution based settings, the **Text Console** and the **Failsafe Mode**. The **Rescue Mode** allows you to restore defective file systems while the **Memory test** is a check of the system memory. For a better explanation you may want to review the according manual page for the Rescue Mode (page 30).

The IACBOX will now boot up to the following screen, the console is ready for login.



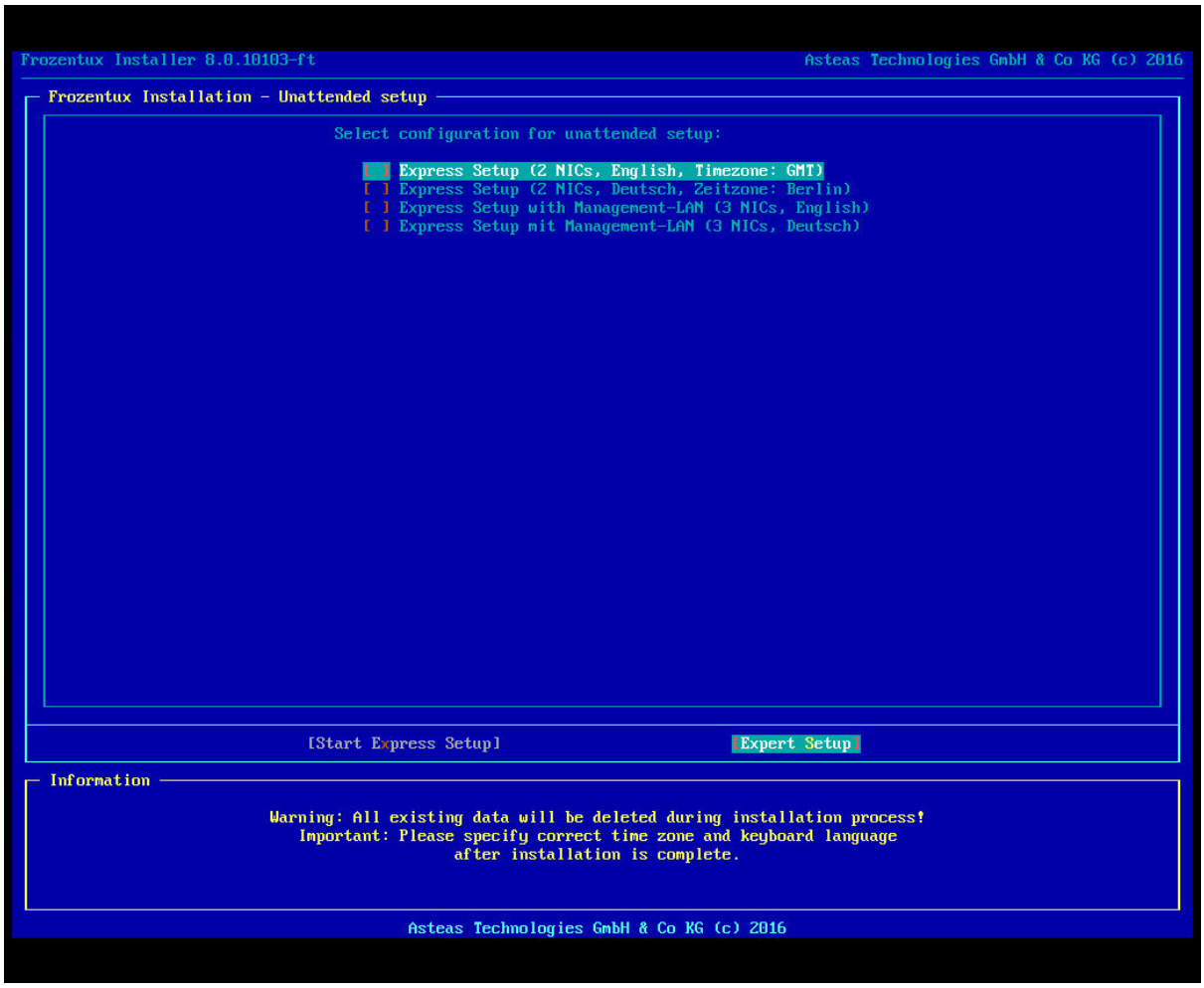
This means that the IACBOX was successfully installed. You have the IP address of the Office-LAN on-screen which can be used in order to open the WebAdmin of the IACBOX. Therefor connect your workstation/notebook to the Office-LAN and open the webpage https://192.168.1.1 with your browser of choice (replace with the IP address you've configured which is also visible on the console). Later in this manual we will present how to enable WebAdmin access via Management-LAN or Surf-LAN.

Hint:

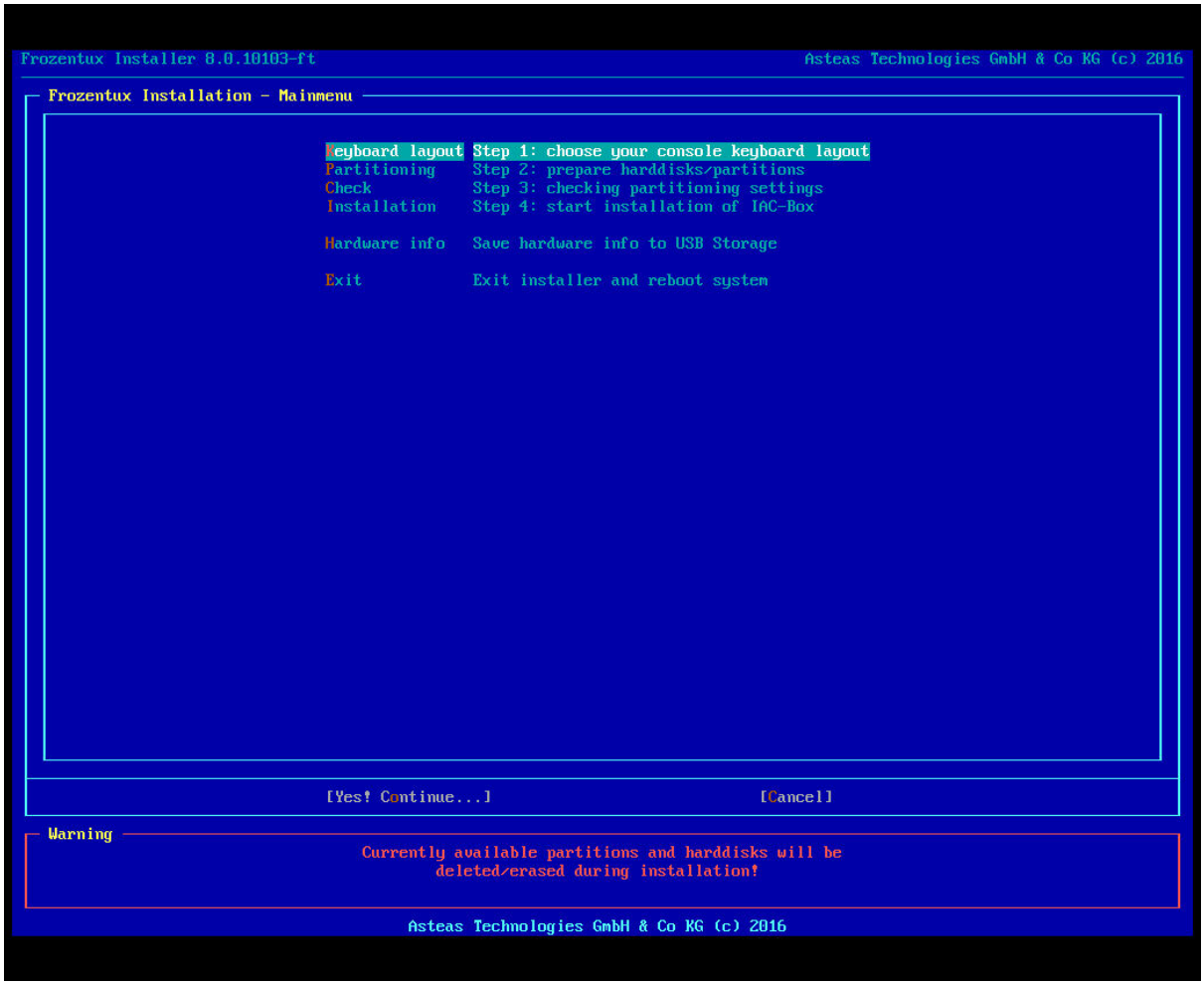
- The console enables you to change certain network settings. This can be used in case the WebAdmin is not reachable (for example due to wrong network settings). To login, you first need to change the password for the default WebAdmin user **sysop**, because the login on the console will **not work** with the default login data (sysop/sysop). In order to change the password for the **sysop** user log in into the WebAdmin (see above) and click on **My Account**.

If you log in with the sysop user on the console, you get back into the setup menu. This can be used to change the network settings without accessing the WebAdmin.

2.2.4 Expert Setup



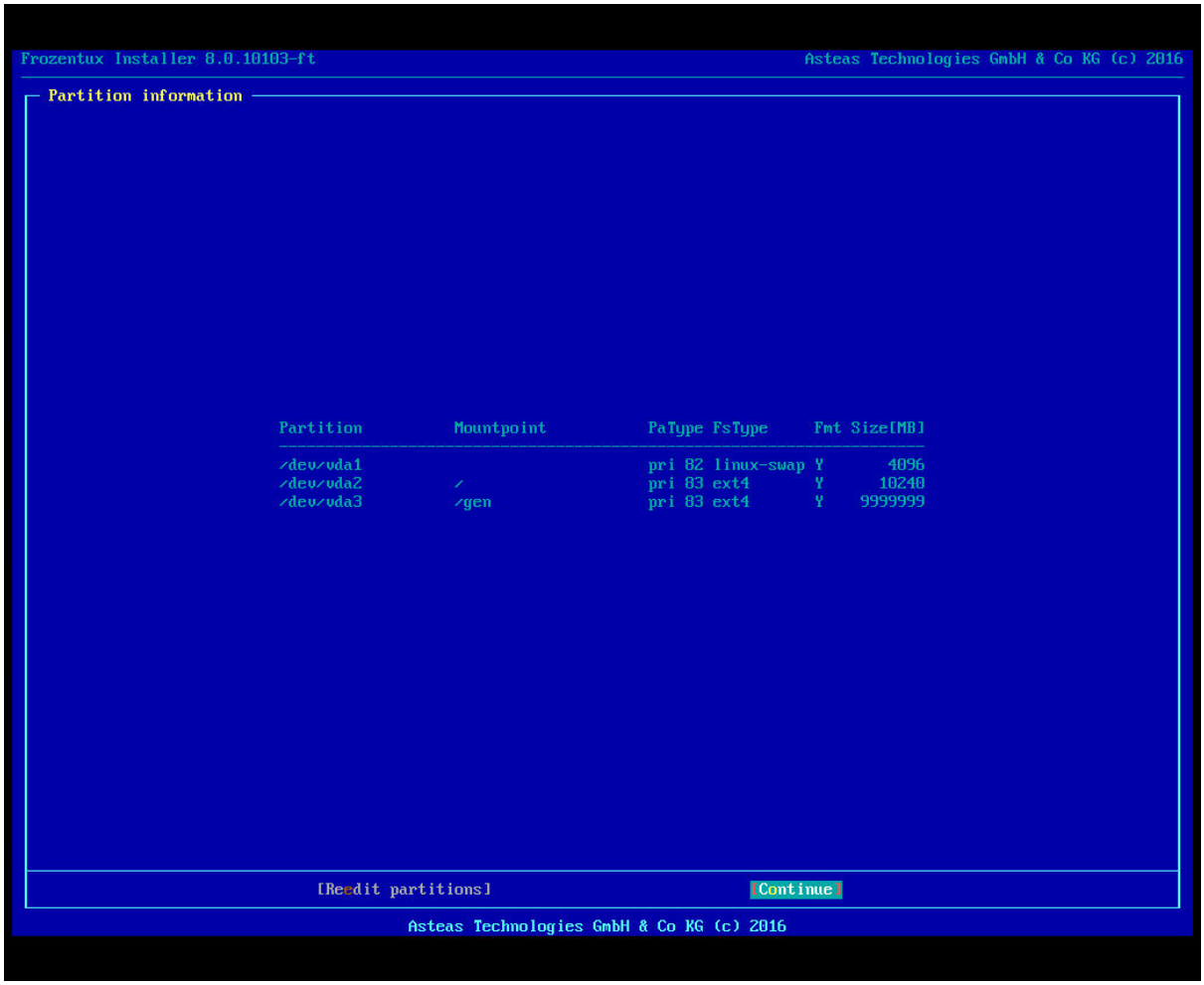
If you want to configure the interface IP addresses, the gateway and DNS servers while installing, then you may as well select the **Expert Setup**.



Hint:

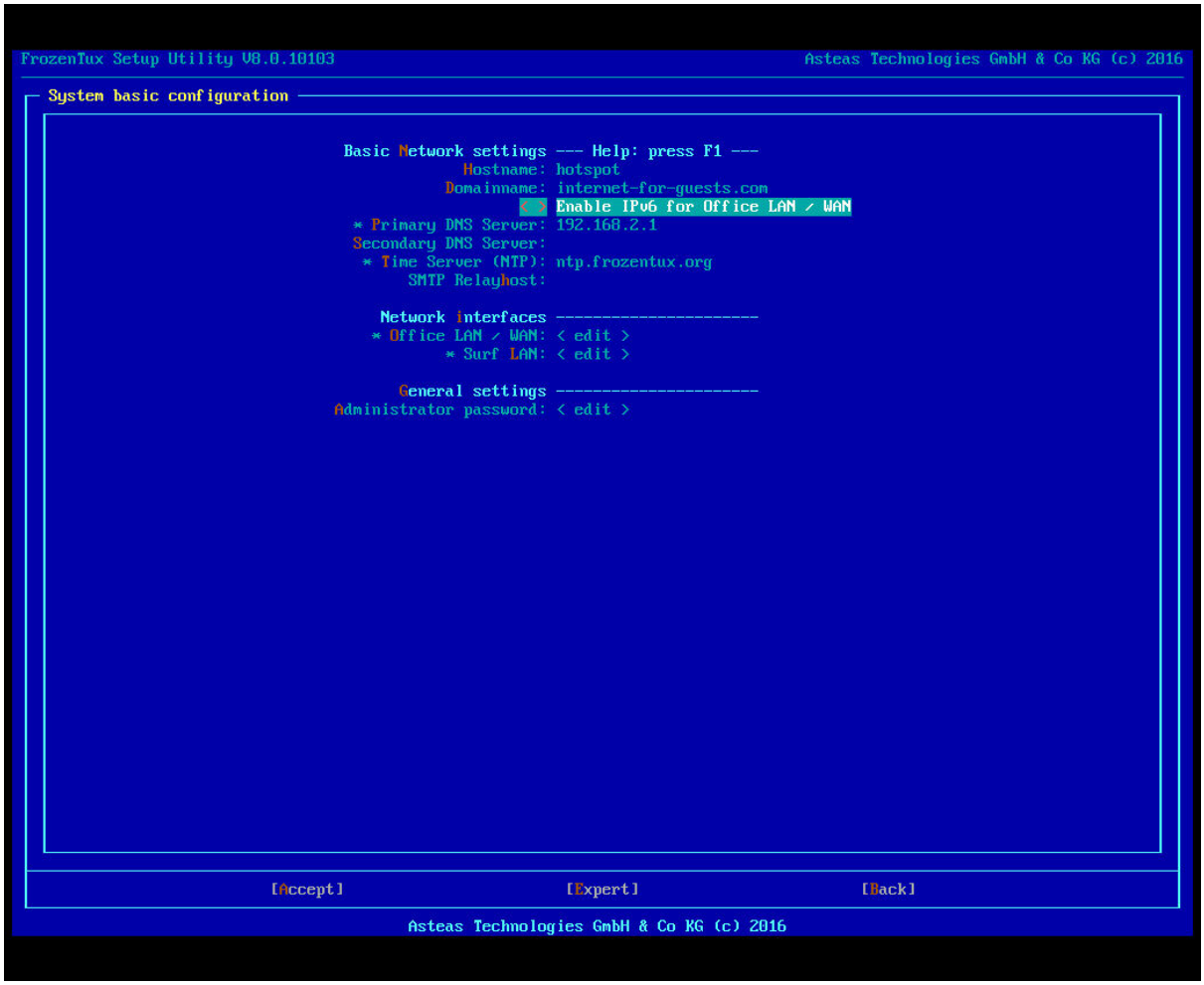
- In order to navigate in the FT-Setup use the **arrow keys** to navigate in lists. To highlight/select an entry use the **SPACE bar**. Use **TAB** to navigate between the list entries and the save/back handlers. To proceed use the **ENTER key**.

Before installing, you can change the keyboard and partitioning settings. Note that the installation requires you to **enter all important menus**. After you've confirmed the keyboard settings, navigate into the menu **Partitioning**. Here you will find the installed hard drive and it's size. To confirm the installation on this hard drive, hit **ENTER** or select **Continue**. The installation will then show a summary of the partitions which are required for the IACBOX.



Accept the summary with **Continue** to get back to the main menu. Now click on **Installation** to start the process which copies all necessary files to the selected hard drive. This can take some time.

After the copying process is finished you will see the main menu of the basic configuration. Same as with the keyboard settings, first navigate into **Timezone** and either accept or change the configuration to your liking. Now navigate to **Sys-Config**. Here you might change the used **DNS Servers**.



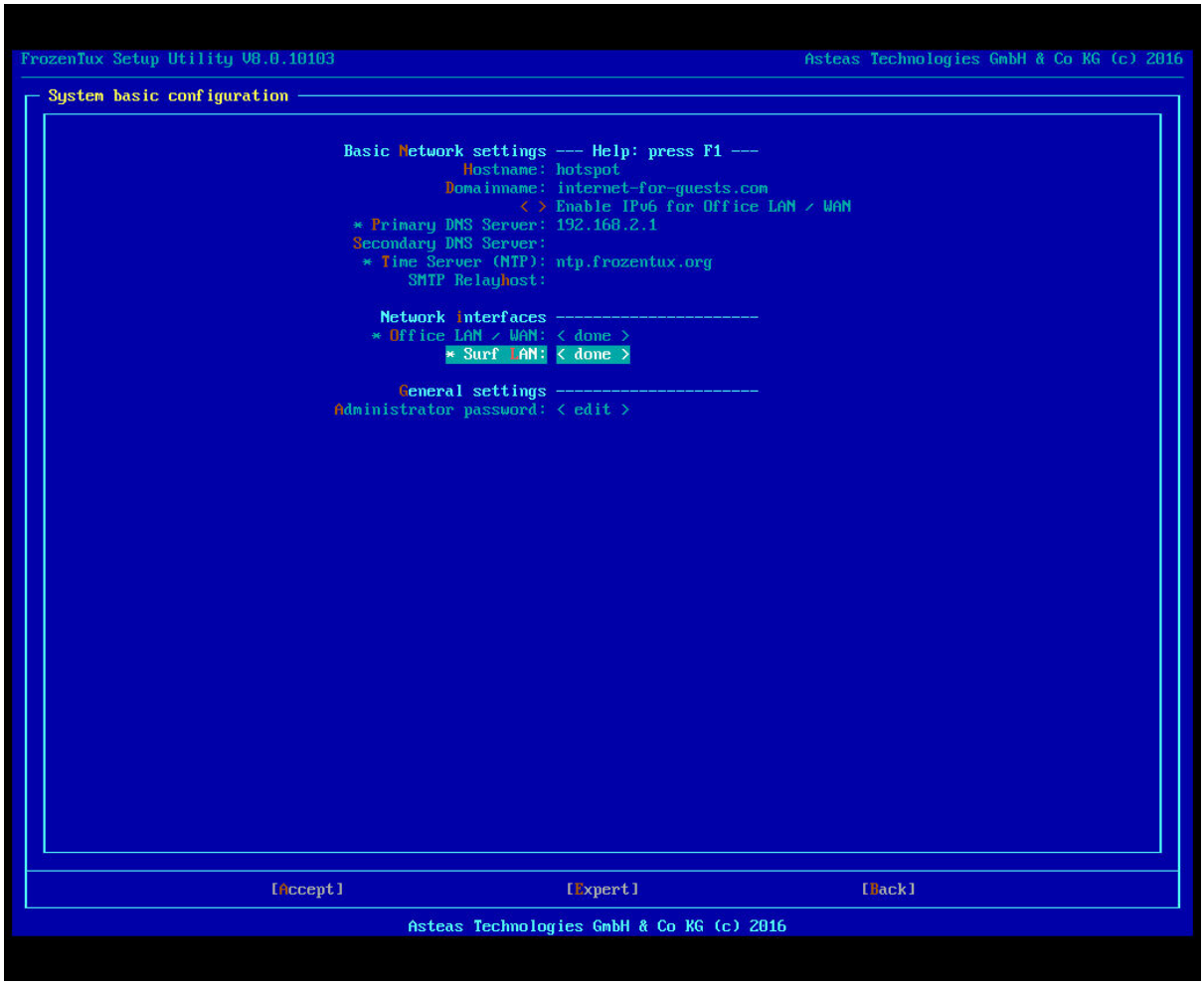
Hint:

- It is **highly suggested** to use the DNS servers of your ISP. Using local firewalls or routers as DNS servers will most likely lead to problems later on.

After configuring the DNS servers navigate into the **Office LAN / WAN** menu. Here you can change the **IP address** of the Office-LAN, it's **Subnet mask** and the **Default Gateway**. Note that this must be done by **technical staff with according network knowledge**.

Hint:

- For the Surf-LAN configuration please note that it is **highly suggested** to use the default configuration. Since the Surf-LAN is an isolated network anyway, it should not matter. The options **Enable client2client protection** and **Disable client AMC address dependency** will be explained later on.

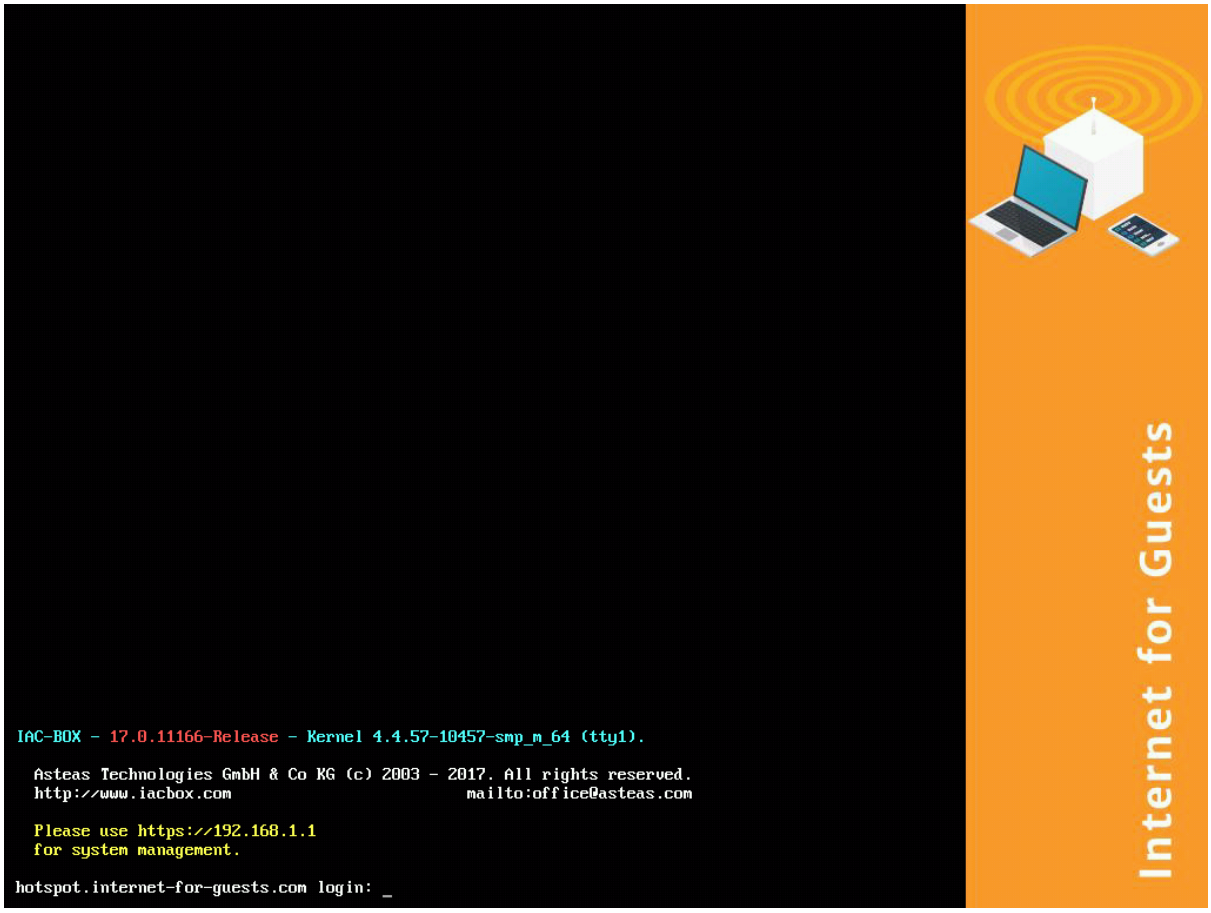


Attention:

- If you have 3 network cards installed, you will also see the **Management-LAN** menu entry. In case you want to activate and use the Management-LAN, enter the menu, select **enable management LAN** by using the **SPACE** bar. Now use the TAB key to select **Accept** and confirm it by pressing **ENTER** once. This must be done before leaving the menu, otherwise the change will not be saved.

After the Office-LAN and Surf-LAN configuration is done, you will also find a **<done>** next to the menu entries. In order to get back to the main menu use **TAB** to highlight **Back** and confirm with **ENTER**. Now navigate into the menu **Net-Auto**. This menu allows you to re-assign the installed network cards to a specific interface of the IACBOX. Even as this is not necessary, the installation requires you to enter this menu in order to verify the amount of detected network cards.

Now navigate back to the main menu, enter the **Activate** menu and confirm. The installation will now activate the configuration. Afterwards finish by choosing **End - Exit FrozentuxSetup**, this installs the kernel and prepares the system for the first boot. The system reboots and loads the IACBOX. The bootup is finished as soon as you reach the login console:



This means that the IACBOX was successfully installed. You now see the IP address of the Office-LAN which can be used in order to open the WebAdmin of the IACBOX. Connect your workstation/notebook to the Office-LAN and open the webpage https://192.168.1.1 with your browser of choice (replace with the IP address you've configured which is also visible on the console). Later on this manual will explain how to enable WebAdmin access via Management-LAN or Surf-LAN.

Hint:

- The console enables you to change certain network settings. This can be used in case the WebAdmin is not reachable (for example due to wrong network settings). To log in on the console, you first need to change the password for the default WebAdmin user **sysop**, because the login on the console will **not work** with the default login data (sysop/sysop). In order to change the password for the **sysop** user log in into the WebAdmin (see above) and click on **My Account**.

If you log in with the sysop user on the console, you get back into the same setup menu which you've seen in the installation. This can be used to change the network settings without accessing the WebAdmin.

2.3 Rescue Boot

This manual describes how to recover a corrupted file system after a power failure on an IACBOX.

Attention:

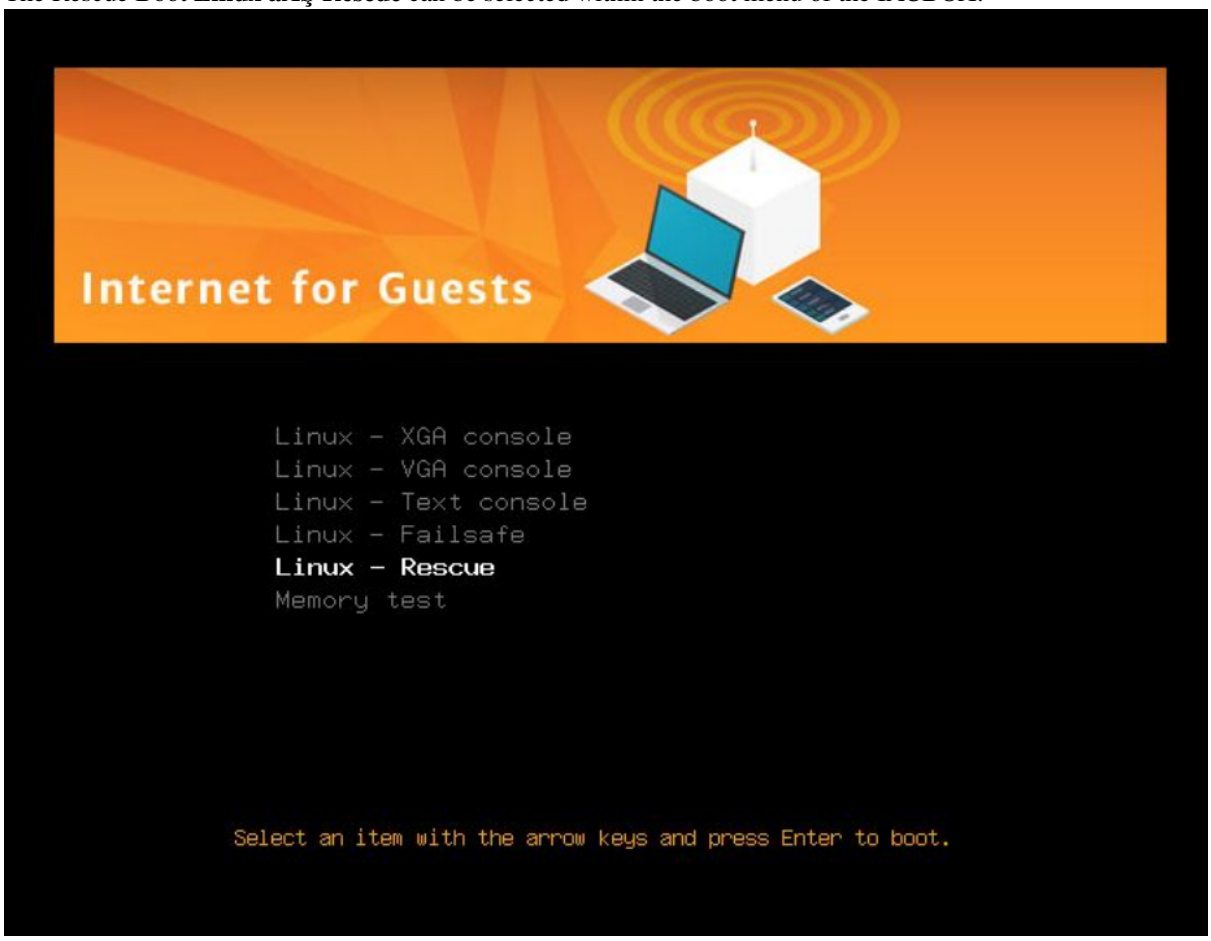
- **These options are available for IACBOX version 8 and older.** For newer IACBOX versions, skip to *the rescue boot option* (page 32).
- If none of the options listed below enables you to get into the **Rescue Menu** (e.g. if the boot selection does not appear), then you can always access it with an IACBOX installation medium (version 17.0 and newer). After booting from the installation medium, select the first entry **Standard Installation** and wait until the FT Setup is fully loaded. Then proceed by selecting **Expert Setup** and hit **F2** to unlock the according rescue menu entries.

Hint:

- A corrupt file system can be caused by a defective hard drive.
- Depending on hardware or software failures, it may not possible to repair a corrupt file system.

2.3.1 System Start

The Rescue Boot **Linux âĂŞ Rescue** can be selected within the boot menu of the IACBOX.



Now the IACBOX will launch a special console mode which will enable you to input commands in order to fix possible errors on the filesystem.

2.3.2 Check and repair file systems

First all partitions should be recognized and verified. For this purpose, they can be listed with the command.

```
fdisk -l
```

```
Loading kernel/drivers/virtio/virtio_pci.ko
virtio-pci 0000:00:03.0: found PCI INT A -> IRQ 11
virtio-pci 0000:00:04.0: found PCI INT A -> IRQ 11
virtio-pci 0000:00:06.0: found PCI INT A -> IRQ 10
virtio-pci 0000:00:06.0: sharing IRQ 10 with 0000:00:02.0
Loading kernel/drivers/block/virtio_blk.ko
 vda: vda1 vda2 vda3
Loading kernel/fs/jbd2/jbd2.ko
Loading kernel/lib/crc16.ko
Loading kernel/fs/ext4/ext4.ko
Waiting for device /dev/vda2 to appear: . ok
rootfs:  major=253 minor=2 devn=64770
bash: no job control in this shell
bash-3.00# fdisk -l

Disk /dev/vda: 21.4 GB, 21474836480 bytes
16 heads, 63 sectors/track, 41610 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks  Id System
/dev/vda1               1        8323     4194760+ 82 Linux swap
/dev/vda2   *        8324       29129    10486224  83 Linux
/dev/vda3           29130       41610     6290424  83 Linux
bash-3.00# random: nonblocking pool is initialized
```

Usually the partitions are named **sda1**, **sda2** and **sda3** but the naming can vary in different environments (for example virtualized). First of all you can check these partitions for errors and eventually try to repair them.

2.3.3 Check for errors

In order to **check** for errors, the following command can be used with all listed partitions.

```
fsck.ext4 -n /dev/<partition>
```

2.3.4 Repair

If any errors are found while checking the partitions, the follwing command may be able to repair them. This command should be executed for all partitions.

```
fsck.ext4 -p /dev/<partition>
```

2.3.5 Exit the Rescue Boot

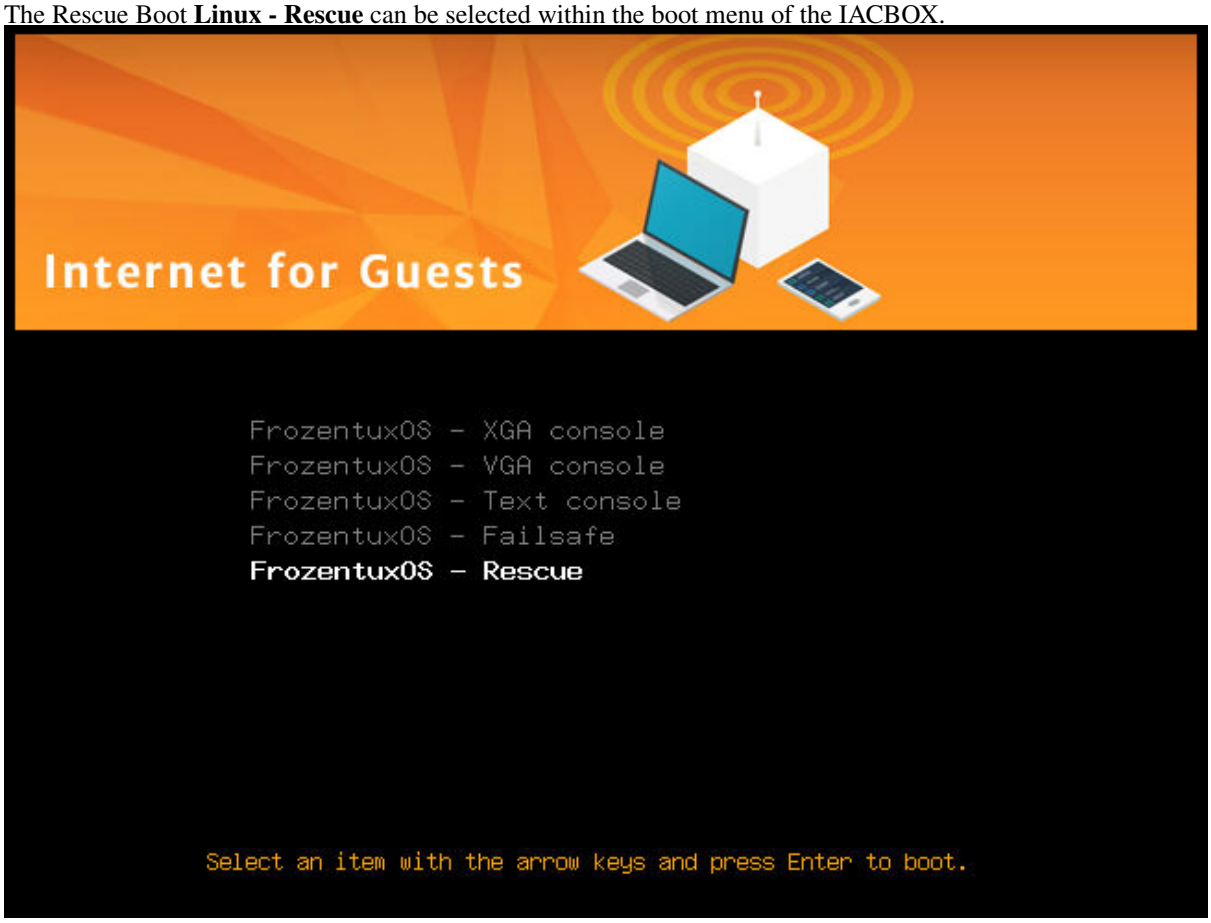
After all errors were repaired, use the following command to exit the rescue mode.

```
exit
```

2.3.6 Rescue Boot for newer IACBOX versions

Attention:
- **The following section will handle the Rescue Boot on systems starting with version 17**.

The Rescue Boot **Linux - Rescue** can be selected within the boot menu of the IACBOX.



Now the IACBOX will launch the FT-Setup with a new menu order which allows you to do the following:

- Export hardware information to an USB storage device.
- Start networking (either via DHCP or by a manual configuration) in order to start the **remote control**.
- Repair faulty file systems with the **Repair discs** menu entry.
- Set the **Date and Time**.

Repair file systems

In order to repair a faulty IACBOX file system, select the menu entry **Repair discs**. In the next screen select **Scan partitions** and proceed with **Repair partitions**. In **Repair partitions** select each partition manually and continue with **OK**. This will attempt to automatically find and fix errors. After the process is finished, you are prompted to continue with **ENTER**.

Start the remote control

The new **Rescue Boot** menu will also allow you to start the remote control. Navigate to **Networking** and either obtain an IP address via DHCP or set up the network interface manually. After the configuration is done, continue with **Start remote control** and enter the data you've got from the IACBOX support team.

2.4 Serial installation

This manual describes how to install the IACBOX software via serial interface.

Attention: Please note that there are known problems with third party tools such as PuTTY. For creating a serial connection, we do recommend any linux distribution. In this example **openSUSE** was used with an USB to serial adapter.

2.4.1 Installing a serial communication programm (terminal emulator)

On Linux a handful of terminal emulation programs are available, e.g. **minicom** or **screen**.

This manual references to the popular software **screen**.

In openSUSE open the terminal and execute the following command to obtain the most current version of minicom:

```
sudo zypper install screen
```

For the Debian/Ubuntu distribution you can use the following command:

```
sudo apt-get install screen
```

For RedHat/CentOS/Fedora you can use the following command:

```
sudo yum install screen
```

2.4.2 Configuration of Screen

First you need to check which interface is used to connect to the IACBOX. In this example, a USB to serial adapter cable was used. With the following command you can list serial devices:

```
sudo dmesg | grep -i tty
```

The output shows, that the port *ttyUSB0* must be used. If the output is too or shows too many different devices, disconnect all USB devices, perform a system restart and retry the command from above:

```
% sudo dmesg | grep -i tty
[    0.000000] console [tty0] enabled
[    2.203312] 00:06: ttyS0 at I/O 0x3f8 (irq = 4, base_baud = 115200) is a 16550A
[ 3340.415857] usb 4-1: pl2303 converter now attached to ttyUSB0
```

2.4.3 Using screen

Now execute the following command to start up **screen**:

```
% sudo screen  /dev/ttyUSB0 115200 -T xterm
```

Before installing the IACBOX, make sure the hardware is connected. Insert the required boot media (CD or USB stick) and then start the device on which you want to install the IACBOX.

After the BIOS screen disappears, you will be asked for the installation mode. Type in **s** for serial installation and continue with **ENTER**.

Now the IACBOX setup will start in serial mode.

Hint:

Useful keyboard combinations for screenn

Quit (kill) minicom CTRL+A, CTRL+K - confirm killing with [y]

Detach minicom (put it in background) CTRL+A, CTRL+D

Resume the screen session screen -r

2.5 System Migration

This manual describes how to migrate an active IACBOX system onto new system.

Hint:

- With the exception of version **17.2.11676** (see special case), backups can only be applied to systems with the same base-version.

2.5.1 Creating a backup

Before migrating a system, a backup must be downloaded. This can be done at any given time in the WebAdmin menu **System/Backup**. The backup contains following data:

- Instances of tickets
- Ticket-Templates
- Configuration (WebAdmin)
- Logos & Designs
- License Information

Now that this data has been secured, the backup can be uploaded on a new IACBOX system by using the same WebAdmin menu, **System/Backup** - **Restore**.

2.5.2 Migrating on new hardware

When migrating to new hardware, the same IACBOX main-version from the base system must also be installed on the new hardware. As an example: If the base system is running on version **17.0.11166 (p11450)**, then version **17.0.11166** must be installed on the new hardware. This will ensure **compatibility** in between the backups from either system.

After the same version was installed on the new hardware, a backup from the base system can now be applied on the new system.

Attention:

- Backups will also restore **license information** from the base system - if both systems are running simultaniously with the same license, then the license may becomes invalid.
- After changing hardware, the IACBOX license must be **unlocked** in the **my.iacbox partner portal**. Therefor find the according license in the list of all your licenses and **right click** on it to find the **Unlock License** menu entry. After unlocking a license, it can be activated on new hardware.

2.5.3 Migrating on a new IACBOX version

There are a handful of reasons why the IACBOX must be reinstalled:

- The kernel of the system is simply too old to support any newer updates
- The initial installation was performed with a 32-bit release and does not receive any new updates
- The system administrator wants to migrate to a brand new IACBOX version which is not available via *Online Update* yet

After the installation image (ISO, USB) of the new version was obtained from our homepage or the *my.iacbox portal* and a bootable medium (page 37) has been created, download a **backup** of the existing system. Now install the new version of the IACBOX according to the IACBOX Installation Manual (page 18).

After the installation has finished, navigate into the WebAdmin menu **System/Backup** of the freshly installed system and **restore** the backup which was created on the old system. After the backup has been applied, the IACBOX will perform an automatic reboot.

2.5.4 Migrating to version 17.2.11676

The brandnew IACBOX version **17.2.11676** was designed to be compatible with backups from version **17.0.11166 (p11450)**. System administrators can install the new version **17.2.11676** and simply **restore** a backup from a *fully updated version 17.0*, which then will work out of the box.

Attention:

- This will only work with backups from fully updated version **17.0.11166** systems with patchlevel **p11450**. Backups from earlier versions can not be used.

2.6 Unattended Setup

This manual describes how to use the unattended setup of the IACBOX which allows you to install the system with pre-defined settings.

Hint:

- The unattended setup only works when installing from USB stick

---

2.6.1 Preparation

Starting with IACBOX version 4.0.6790 there are 4 pre-defined profiles for the unattended installation included on both, the USB and ISO image.

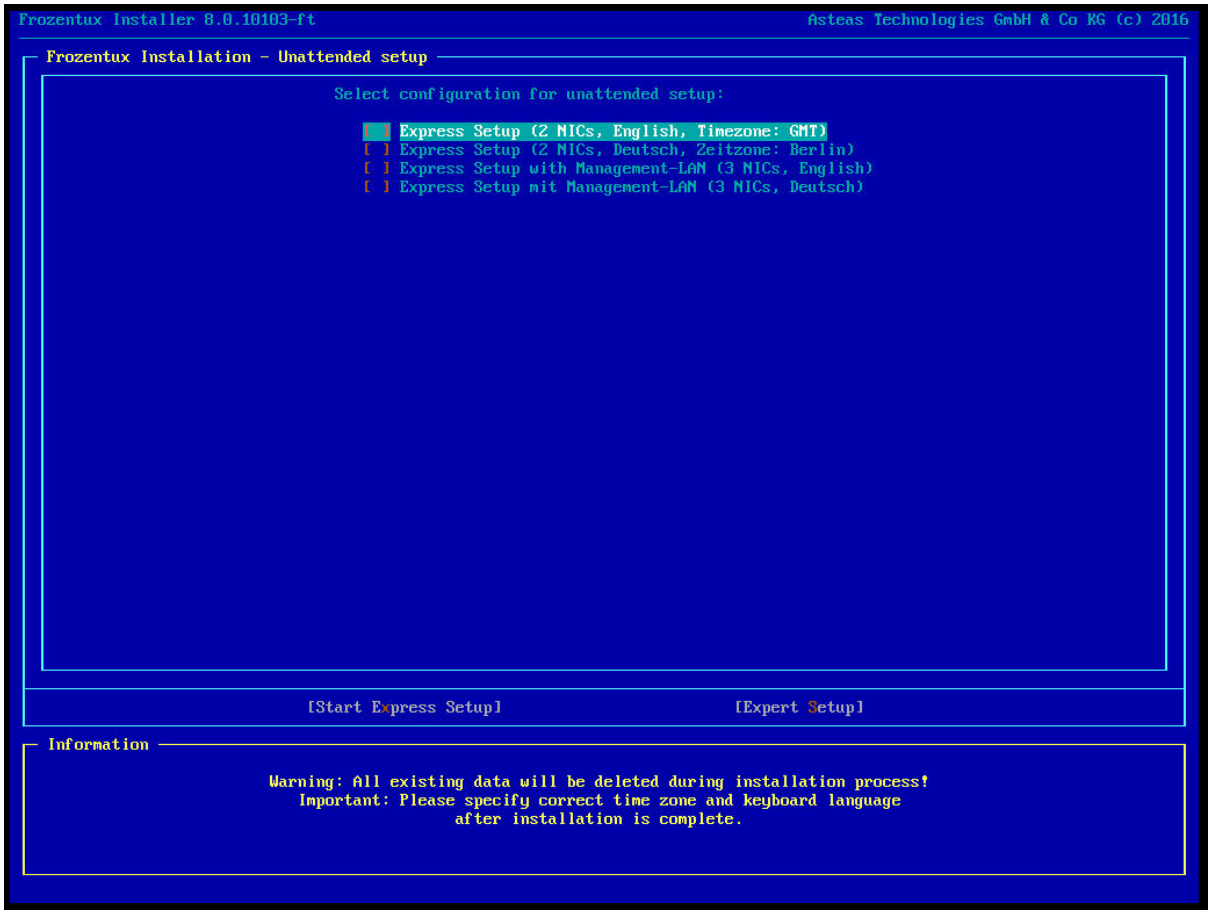
- **01_uas2nic_en.uas** Installation with 2 network cards, english language
- **02_uas2nic_de.uas:** Installation with 2 network cards, german language
- **03_uas3nic_en.uas:** Installation with 3 network cards, english language
- **04_uas4nic_de.uas:** Installation with 3 network cards, german language

If you are installing multiple systems with the same basic settings, then you can create *your own unattended setup file*. In order to get started simply copy one of the files (.uas) from above and perform the according changes with an advanced text editor. Please note that text editors like Windows NotePad can potentially break the file formatting. Notepad++/Kate is suggested.

After opening the file with an according text editor you can directly edit all basic settings for this profile. Please note that the profile shown in the FT-Setup will be named after the file name, so it is suggested to use an easy to memorize filename. Also the **file extension** must be named **.uas** in order for this profile to be recognized. After this is done the file can be copied directly on the USB stick.

2.6.2 Installation

By starting the IACBOX installation from an USB stick you now will find your new UAS profile in the Unattended Setup selection screen.



Here you can select one of the UAS profiles with the **SPACE BAR** and get started by selecting **Start Express Setup** with **TAB**.

2.7 USB Stick Creation

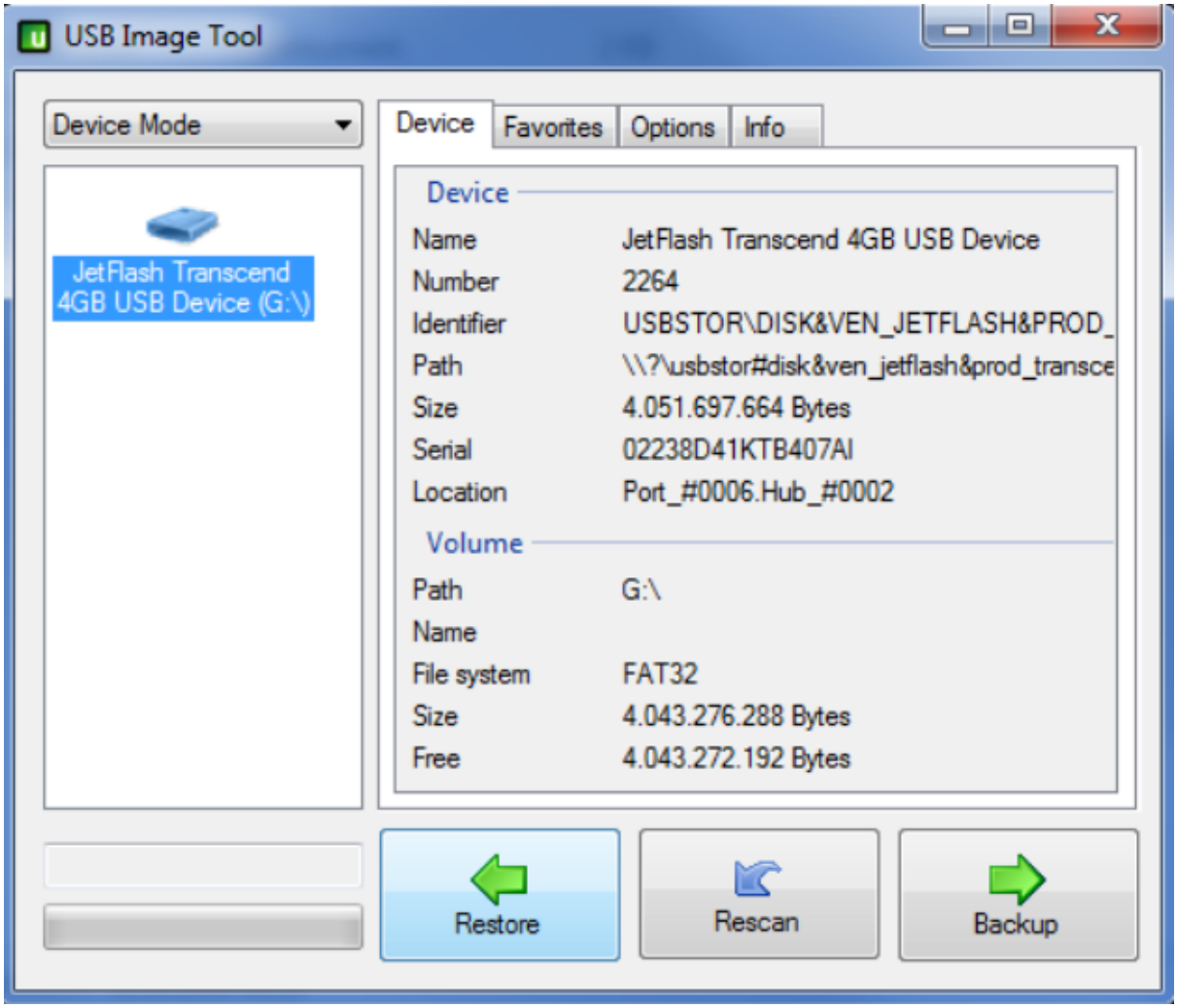
This manual describes how to prepare and create a bootable USB stick from the downloadable IAC- BOX **USB image file** on windows and linux based operating systems.

Hint:

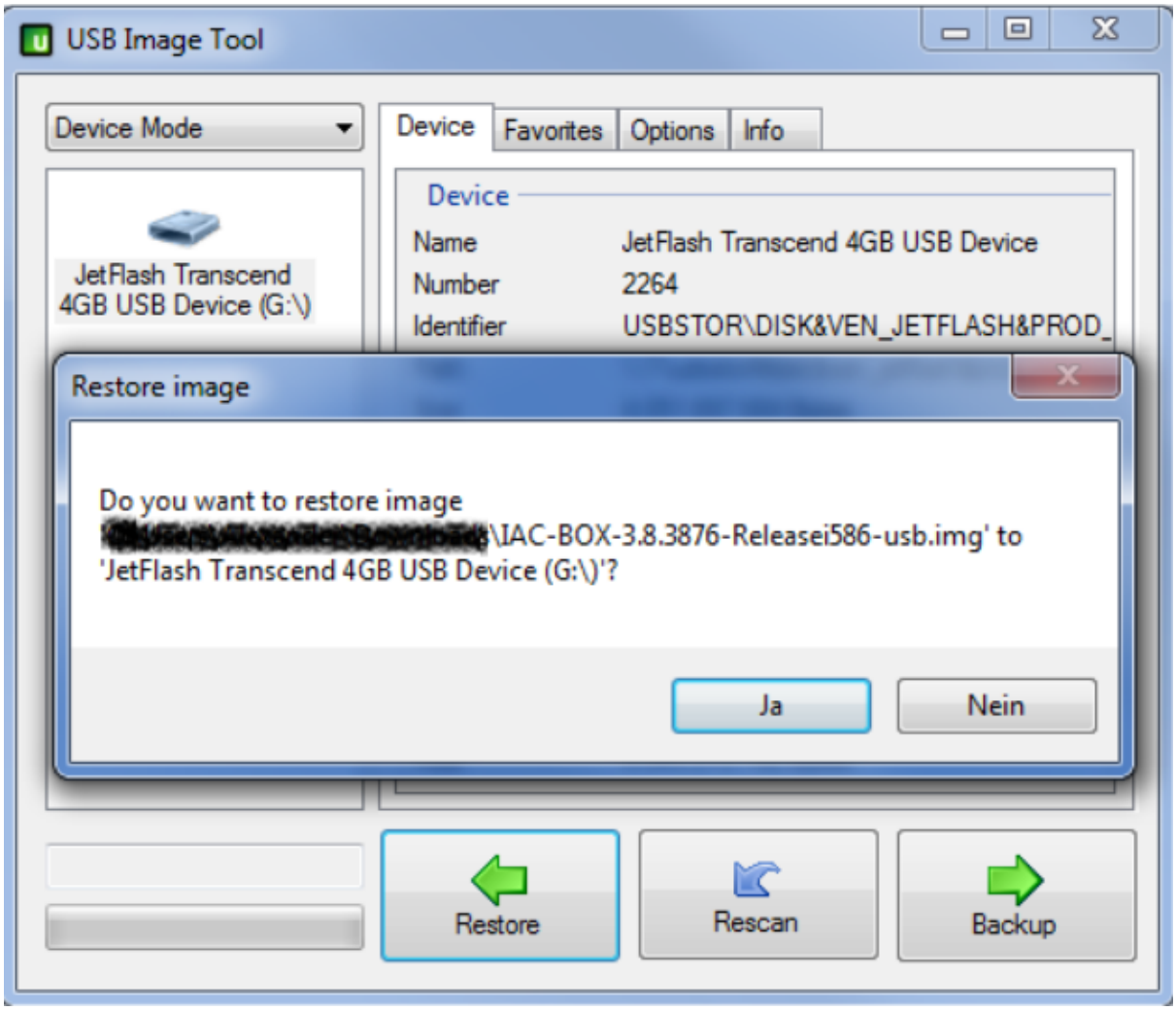
- All existing data on the USB stick will be deleted during this operation!

2.7.1 USB Stick Creation for Windows

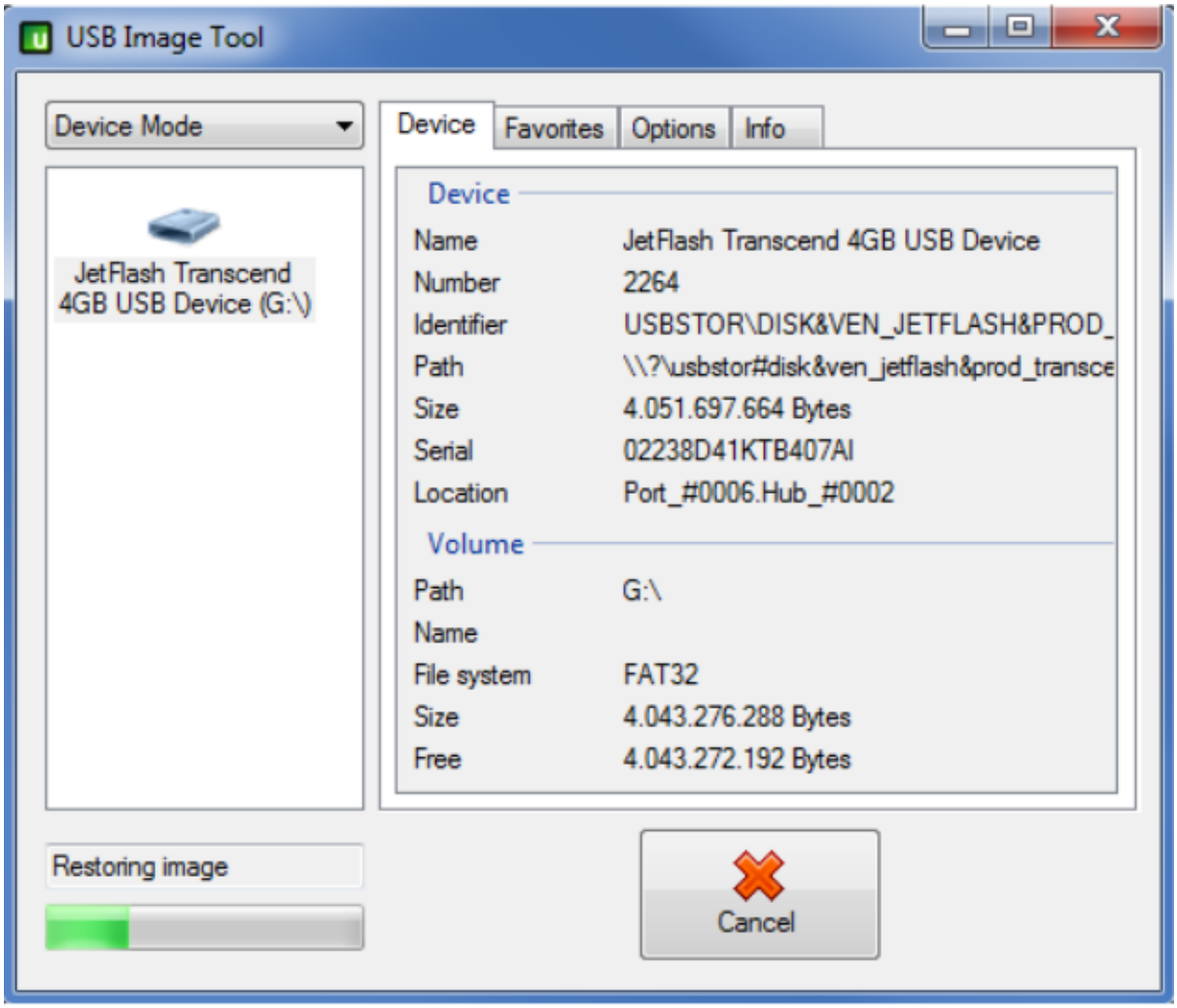
To prepare the USB stick with Microsoft Windows operating system it is recommended to use the **USB Image Tool**, which can be downloaded for free from alexpage.de. Now start the USB Image tool so that the following window opens.



Click on the USB device you want to prepare for USB installation and then on **Restore**. Now select the IACBOX USB image you want to use for Installation.



Confirm this dialog by clicking on **Ja (Yes)** so that the tool will start the copying process.



Now the USB Image gets restored to the USB device as you can see on the lower left corner in the screenshot above. Once this process is complete, the preparation of the USB stick for the USB installation is done. To avoid writing errors do not access the USB device during the copying process! If the copy process is completed you can safely remove the USB stick from your computer.

2.7.2 USB Stick Creation for Linux

If you are using a linux based operating system we recommend using the dd command-line utility. After the USB device was verified with **fdisk -l** you may start the copying process.

```
dd if=IACBOX-x.x.xxxx-aaaaReleasei586-usb.img of=/dev/sdX bs=1M
```

Attention:

- Typing mistakes and wrong usage of device nodes, or other usage errors, for example using the wrong volume (device), can cause irreparable damage to the system!

CHAPTER

THREE

VIRTUALIZATION

3.1 Installation on Microsoft Hyper-V

Attention: This manual was updated in May 2019 to resolve performance problems based on the old guidelines. If you created a Hyper-V VM based on said old guidelines, it is highly recommended to re-create the VM with the updated ones.

This manual describes the steps to configure and prepare **Microsoft Hyper-V** in order to install the IACBOX.

Hint:

- A 64-bit host-system is required.
- It is strongly recommended to use a dedicated physical network interface card for the **Surf-LAN**.
- The system must be online at any time in order to synchronize necessary IACBOX registration data with the licensing server.
- This manual describes the installation of the IACBOX on HyperV, not the HyperV installation itself.
- The IACBOX is a realtime system. Therefore it is **critical** to only assign and use ressources which are capable of working within the same operational context. For example: If a VM cluster is used, only the CPU cores of one socket may be used for the IACBOX to avoid rapid context switching between multiple sockets, which would lead to intolerable delays and possible system faults.

Please note the minimum hardware requirements.

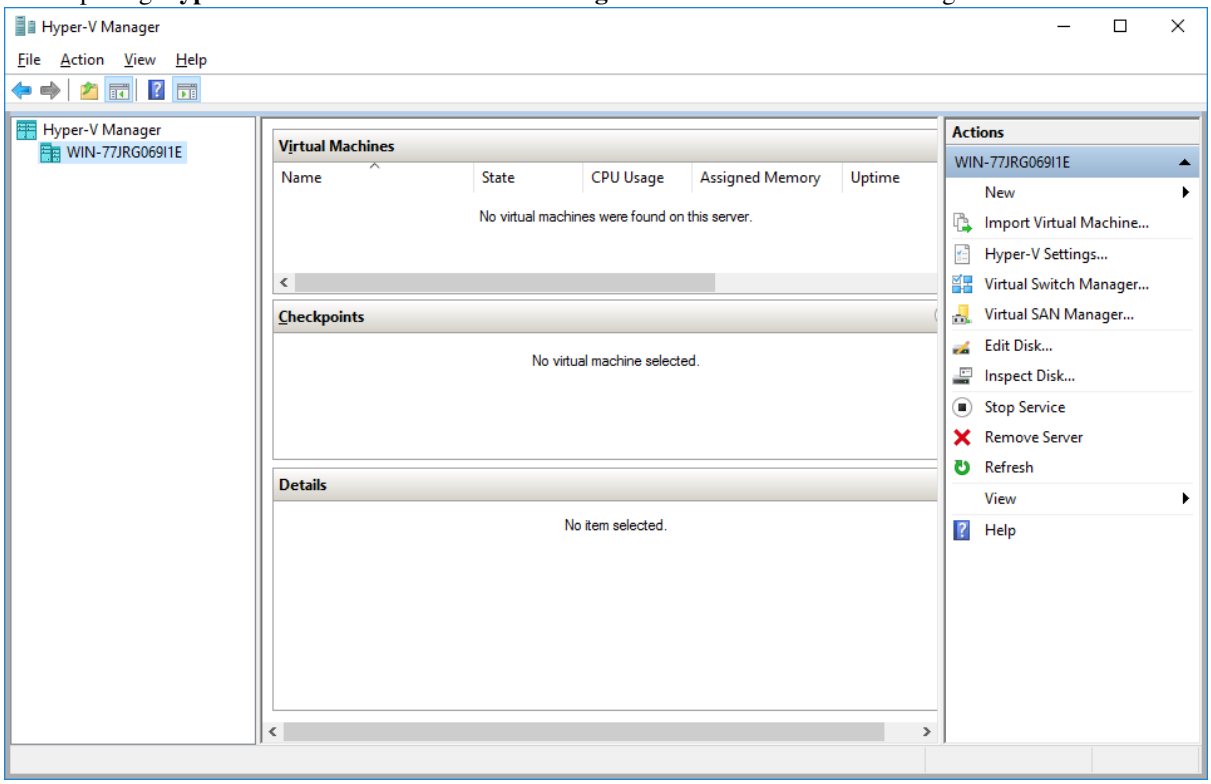
Users / Devices	CPU Cores	RAM	HDD Capacity
10 users	1	2 GB	60 GB
25 users	1	2 GB	60 GB
50 users	2	2 GB	60 GB
75 users	2	2 GB	60 GB
100 users	2	2 GB	60 GB
175 users	2	2 GB	60 GB
250 users	2	4 GB	1 TB
375 users	4	4 GB	1 TB
500 users	4	4 GB	1 TB
750 users	4	4 GB	1 TB
1000 users	4	8 GB	1 TB
1500 users	4	8 GB	1 TB
2000 users	8	8 GB	1 TB
3000 users	8	16 GB	1 TB
Unlimited users	8	16 GB	1 TB

Attention:

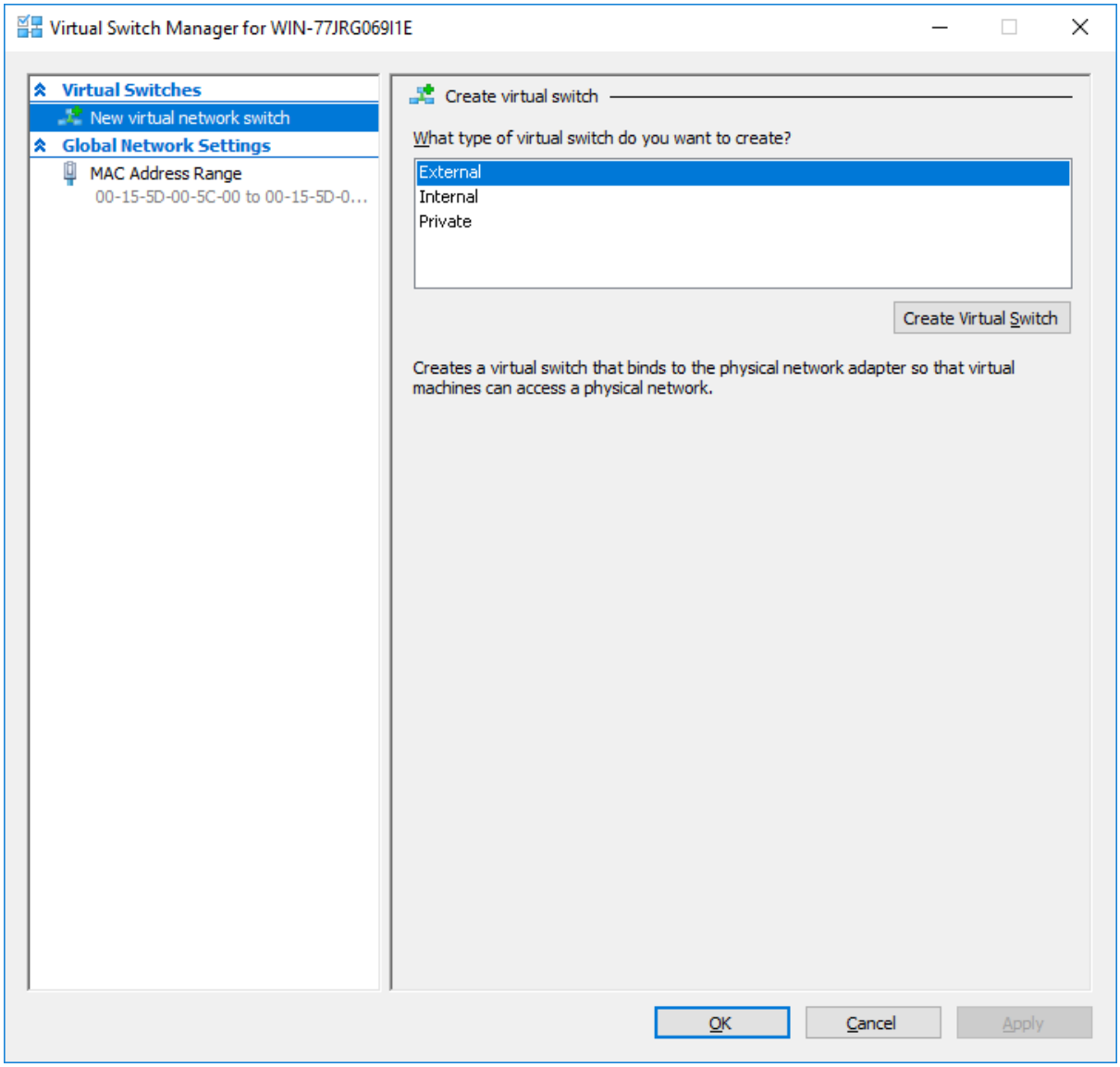
- Starting from 250 users a processor with at least 2,50 Ghz or better is required
- Virtualized environments generally need more ressources due to the nature of virtualization
- Functions like the **Advanced Web Filter**, the **Application Control** or the **Connection Tracking** are very CPU-intensive and should therefore be used with caution
- All specifications are subject to change without notice. The most recent version of this list can always be found here: http://www.iacbox.com/en/support/hardware-requirements/
- Some versions of Hyper-V may not support **VLANs**!
- In order to use the new *DNS based Web Filter* of IACBOX version 17.2, at least 4GB of internal memory must be available.

3.1.1 Configuration

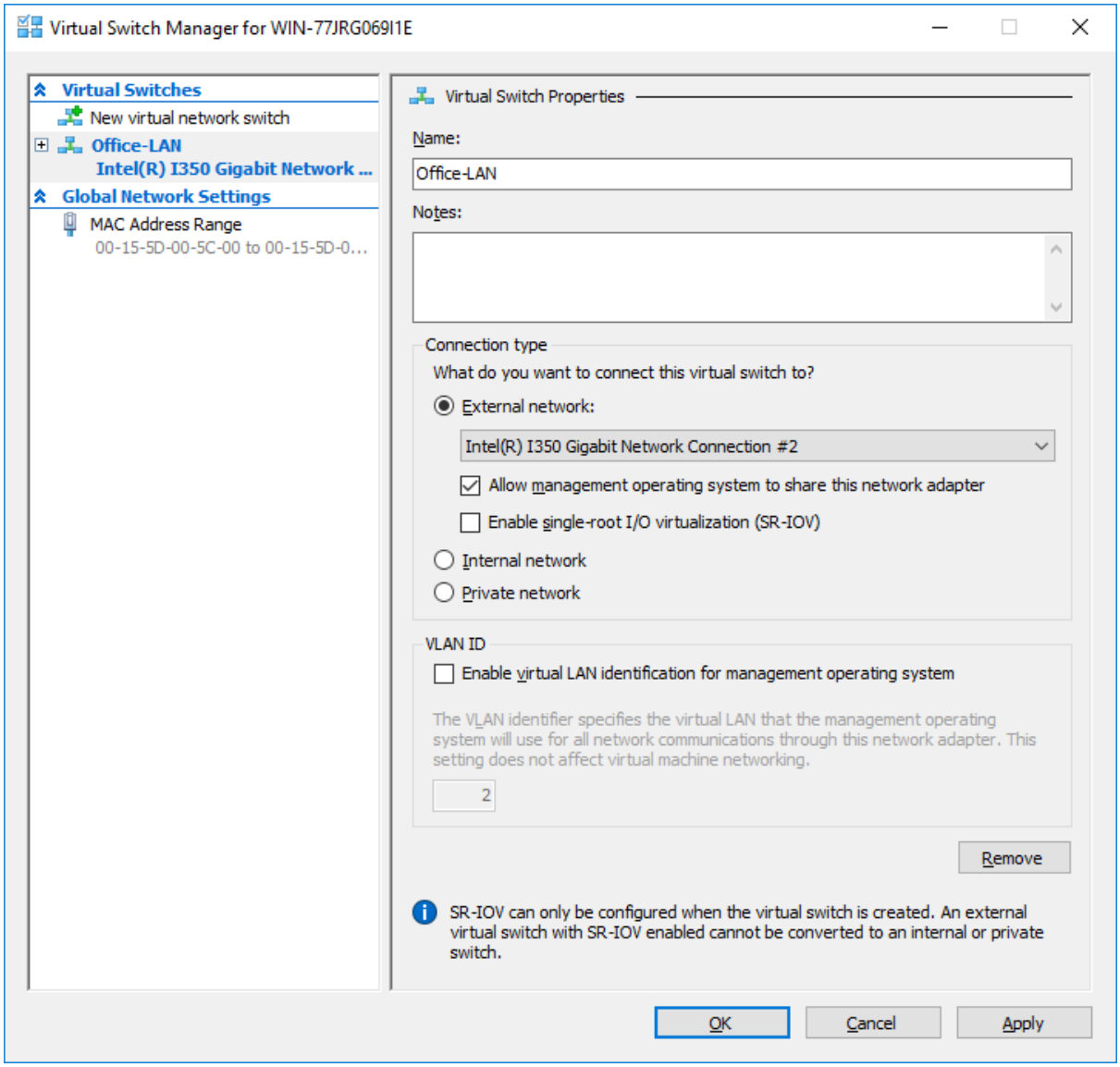
After opening **Hyper-V** click on **Virtual Switch Manager** which can be found on the right side of the screen.



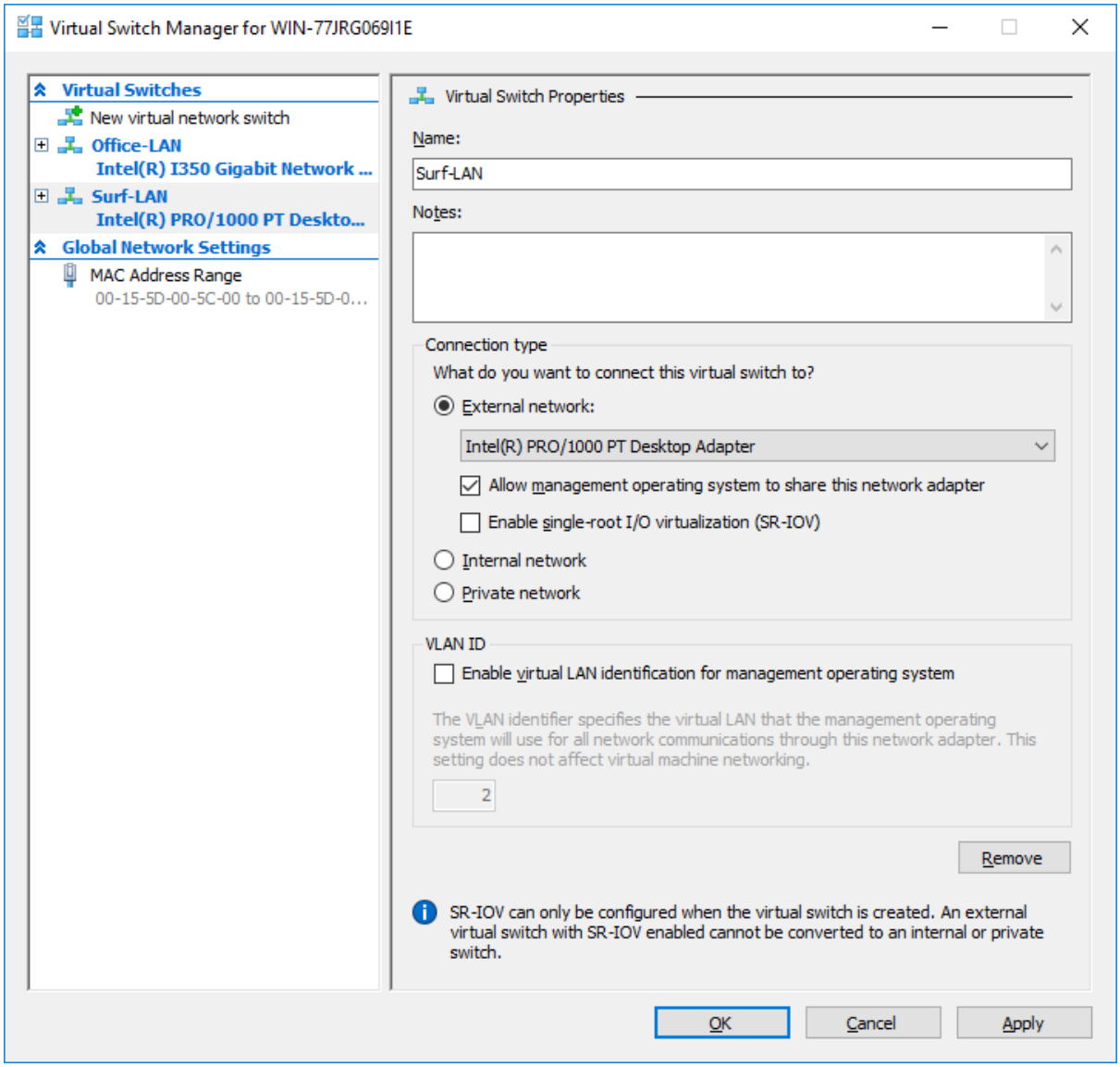
The IACBOX does **require** at least 2 network interfaces, the **Office-LAN** for the uplink and management and the **Surf-LAN** for guest devices. Therefore in the **Virtual Switch Manager** click on **New virtual network switch** and then select **External**. Continue with clicking on **Create Virtual Switch**.



In the next step name this interface **Office-LAN**. As **External Network** select your primary physical network adapter which is connected to the internet.

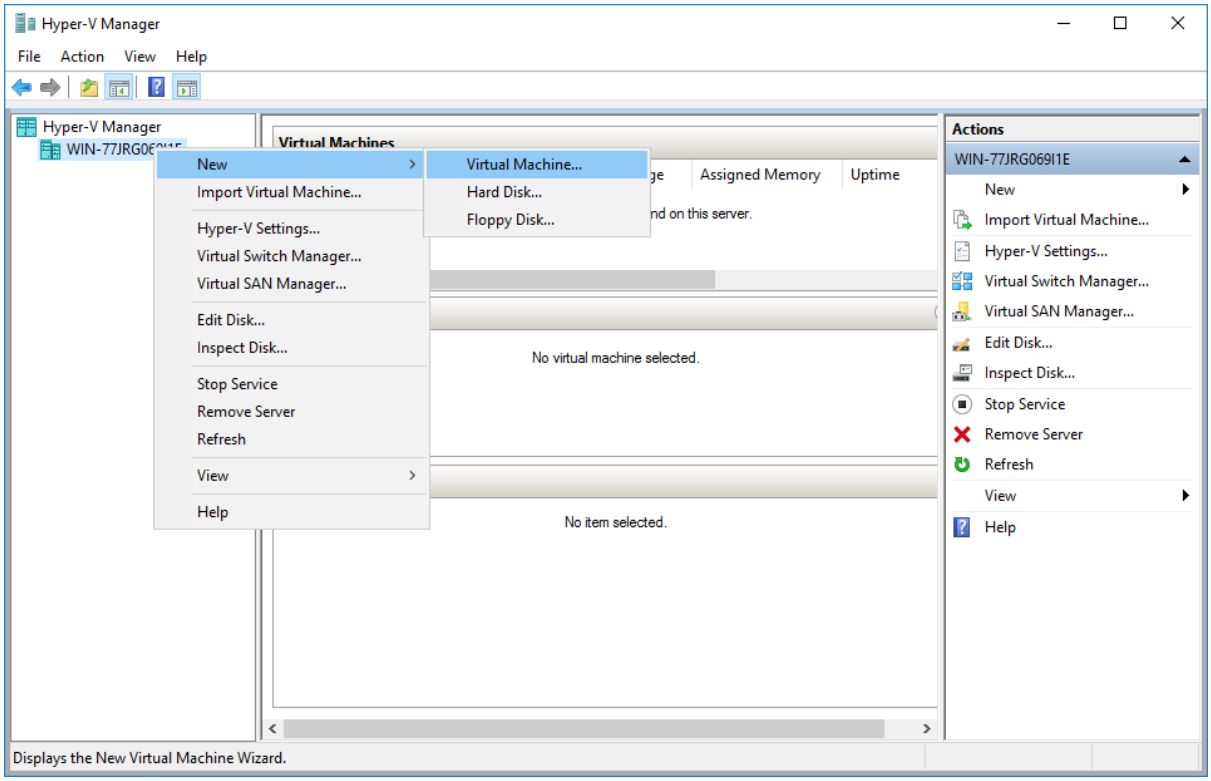


Now repeat this process with another **virtual network switch** and call it **Surf-LAN**. For this virtual interface, select the physical interface which will be connected to the Surf-LAN.

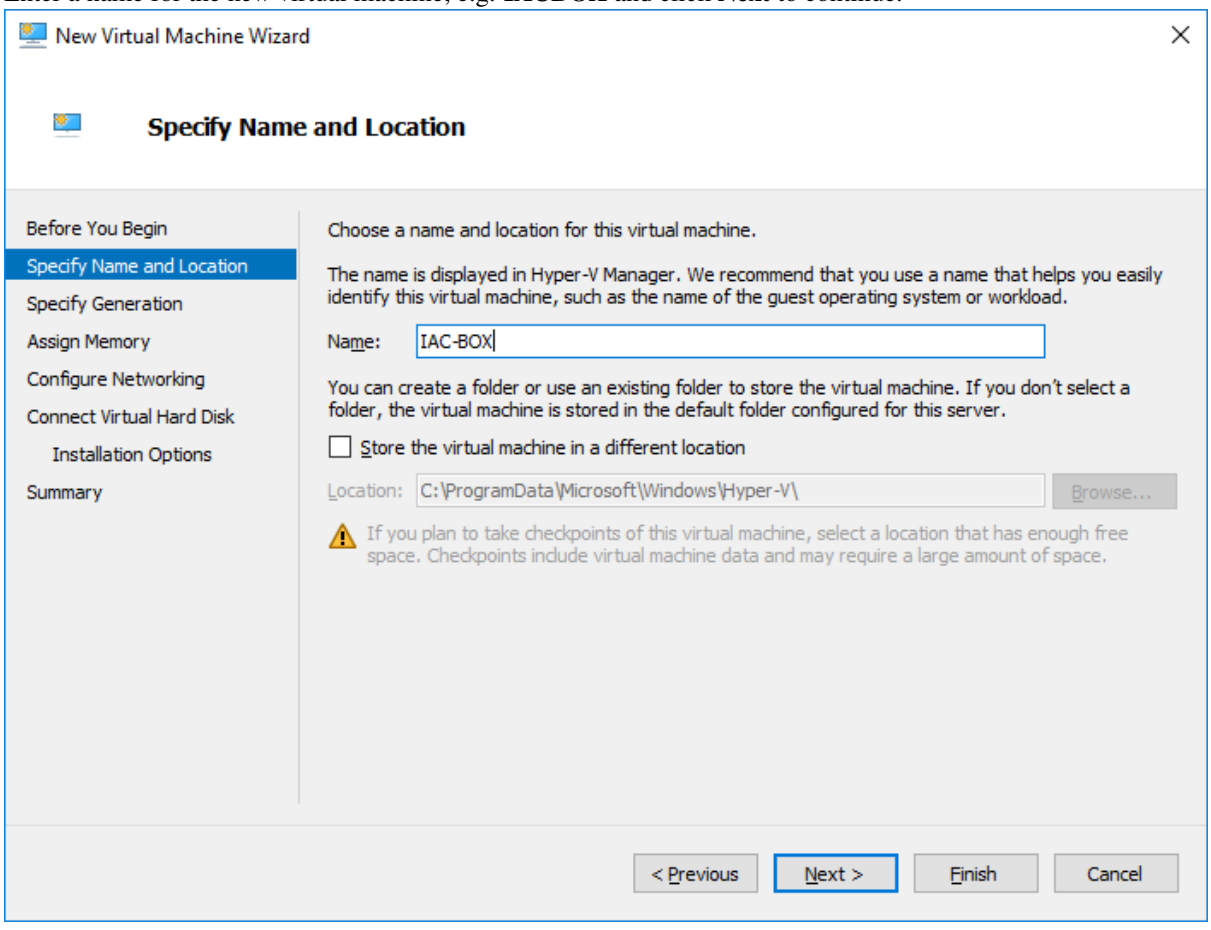


3.1.2 Creating the Virtual Machine

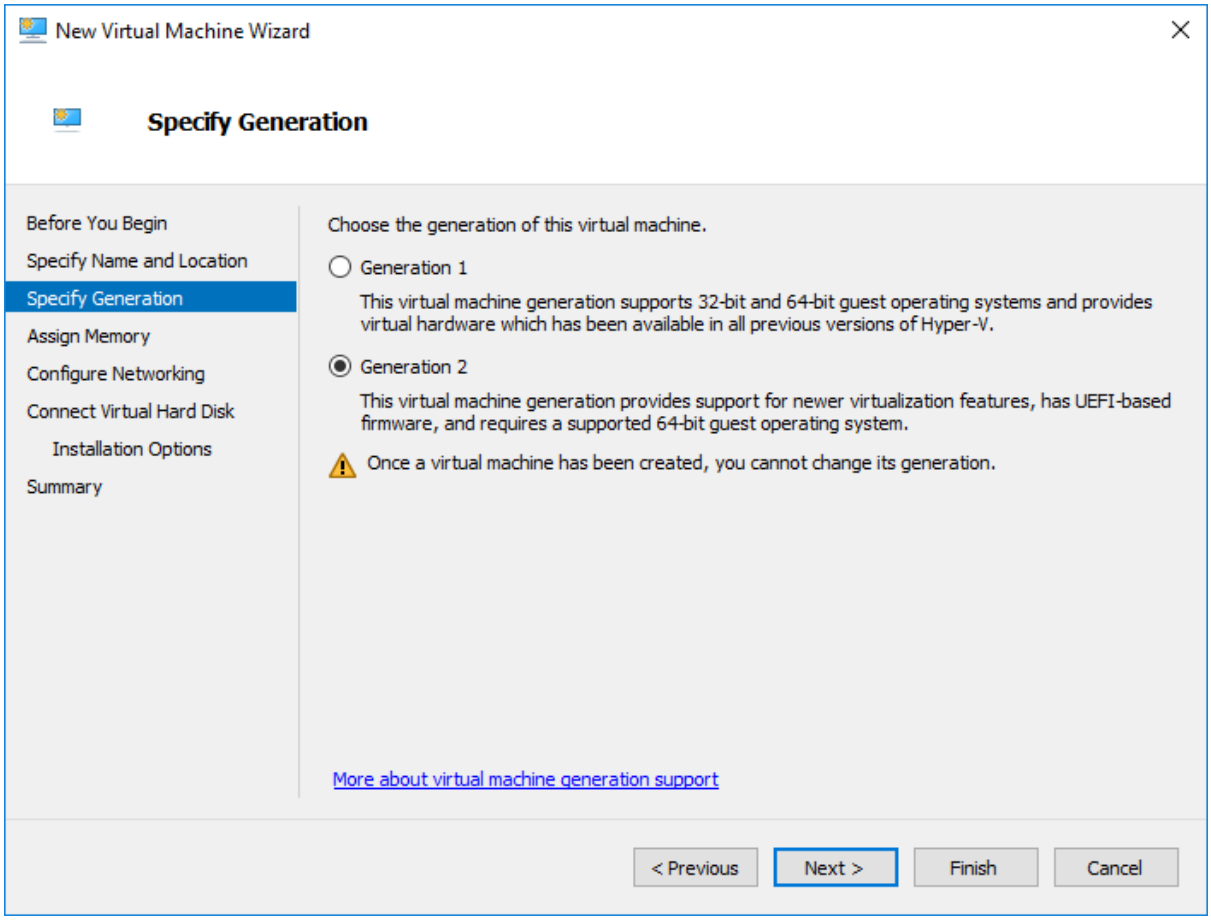
After the configuration of the virtual network switches is done, right-click on the **Hyper-V** server in the left list and choose **New** and **Virtual machine**. This will open up a new window and show you the **Before You Begin** hints, proceed with **Next**.



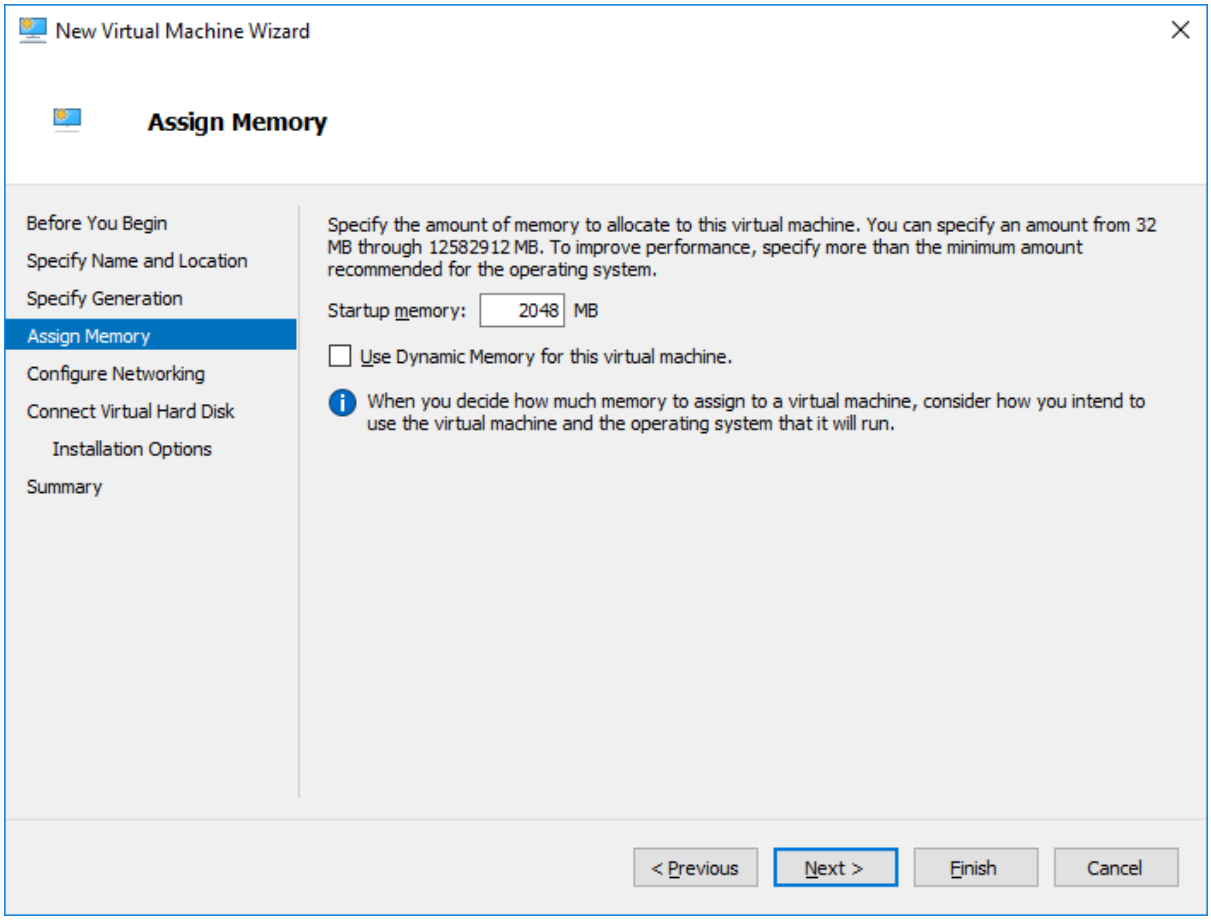
Enter a name for the new virtual machine, e.g. **IACBOX** and click **Next** to continue.



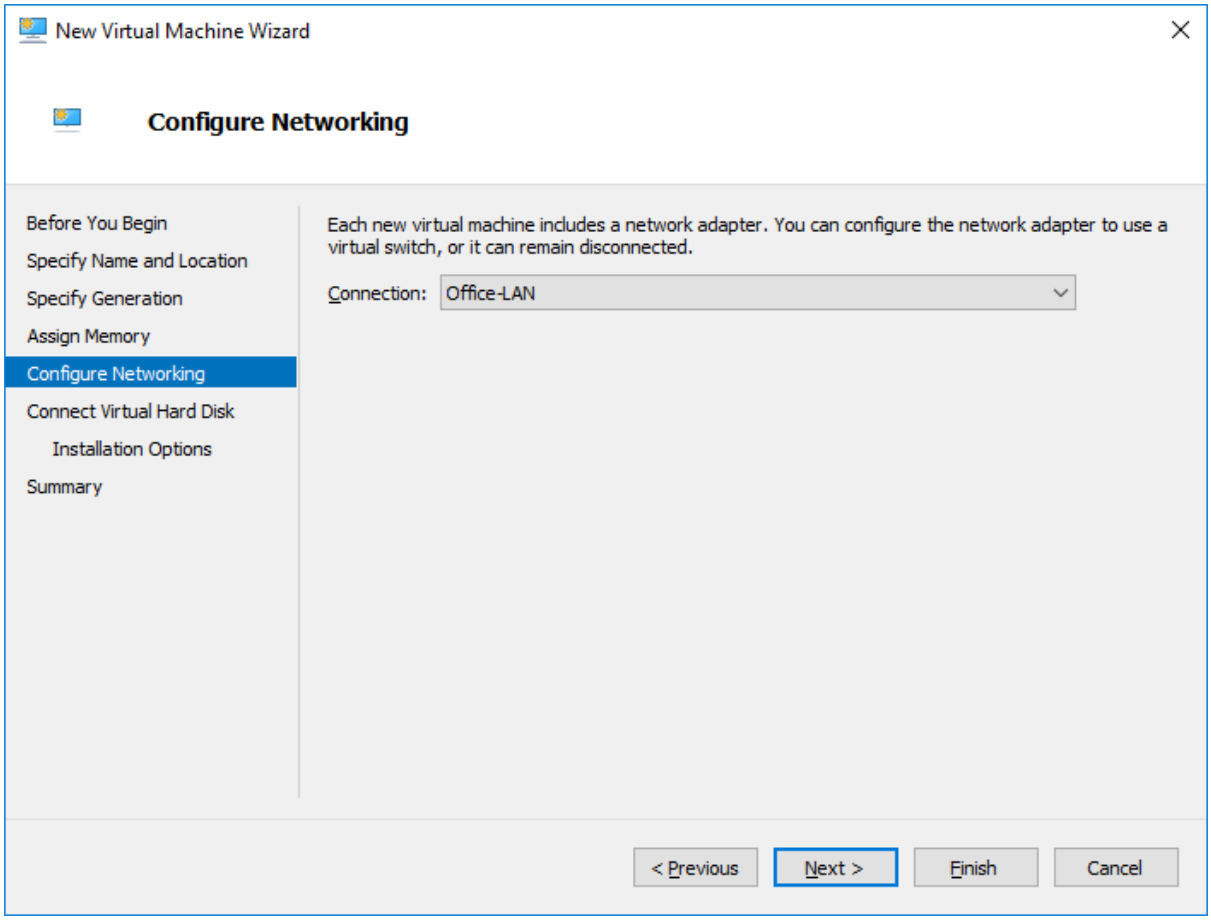
Choose **Generation 2** and continue with **Next**.



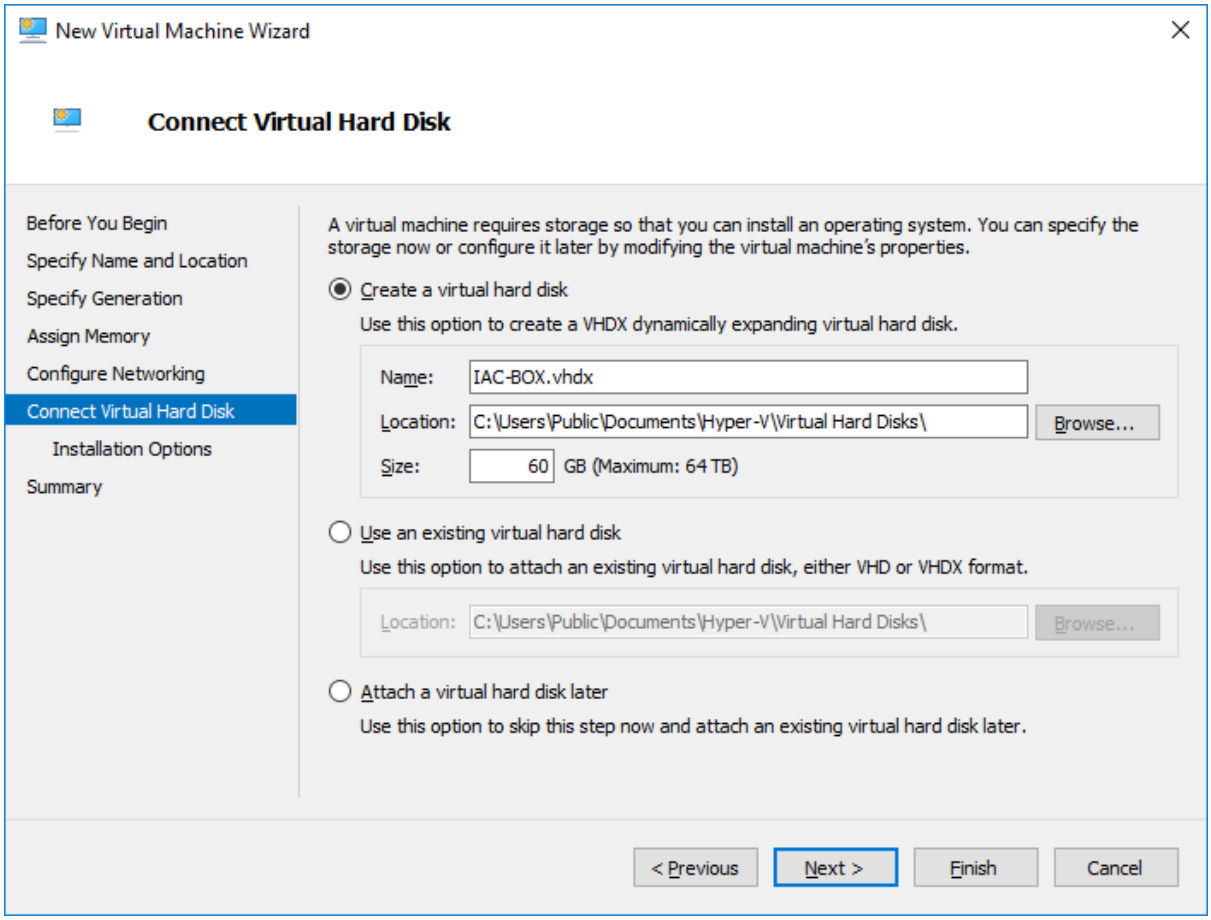
Now select the amount of memory you want to allocate to this virtual machine. Note the **minimum hardware requirements** displayed on top of this manual.



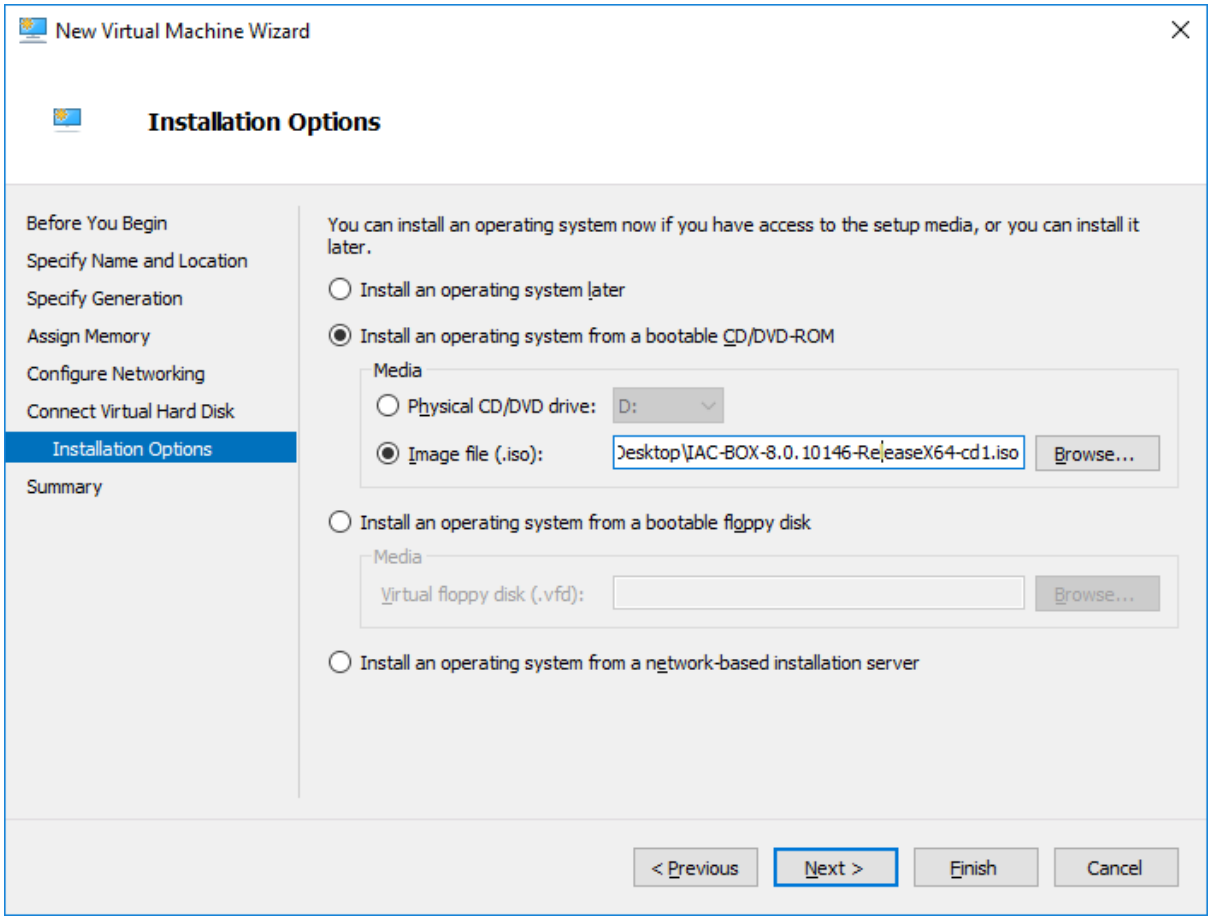
In the following window select the **Office-LAN** as connection and proceed by clicking on **Next**.



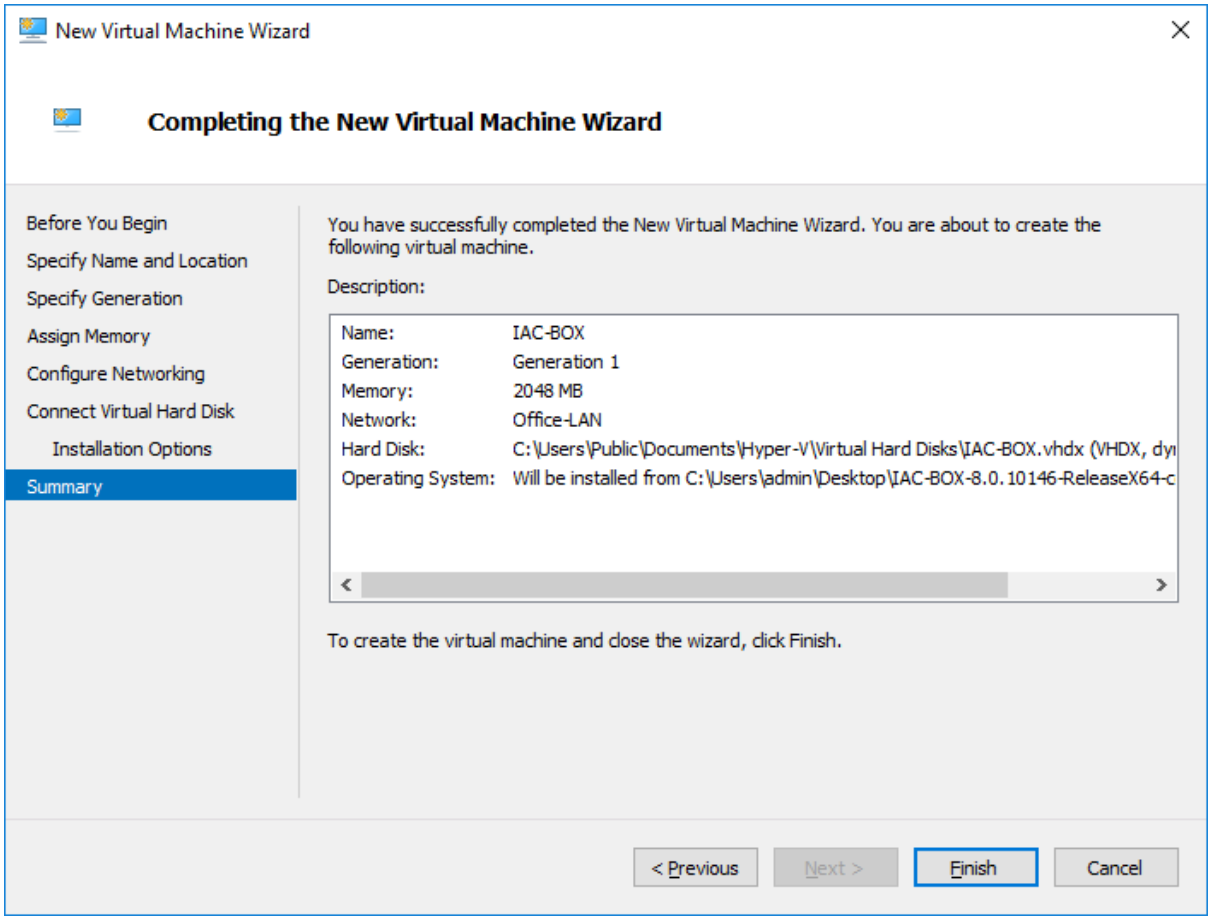
Now a **virtual hard drive** must be configured. Enter a name and configure the size according to the **minimum hardware requirements**.



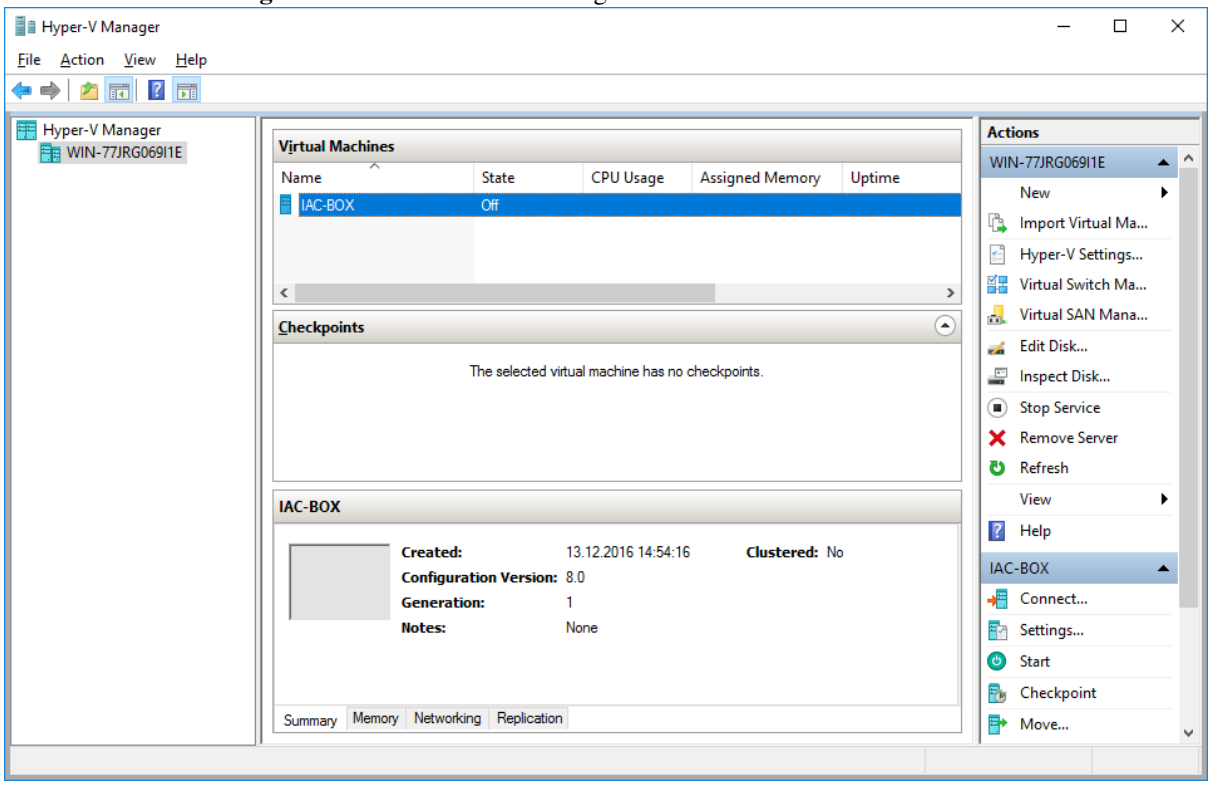
In the next screen you can decide how to include the installation medium. Usually in virtualized environments this is done by **ISO images**, but if the host system has a CD-ROM drive, then a CD can also be used.



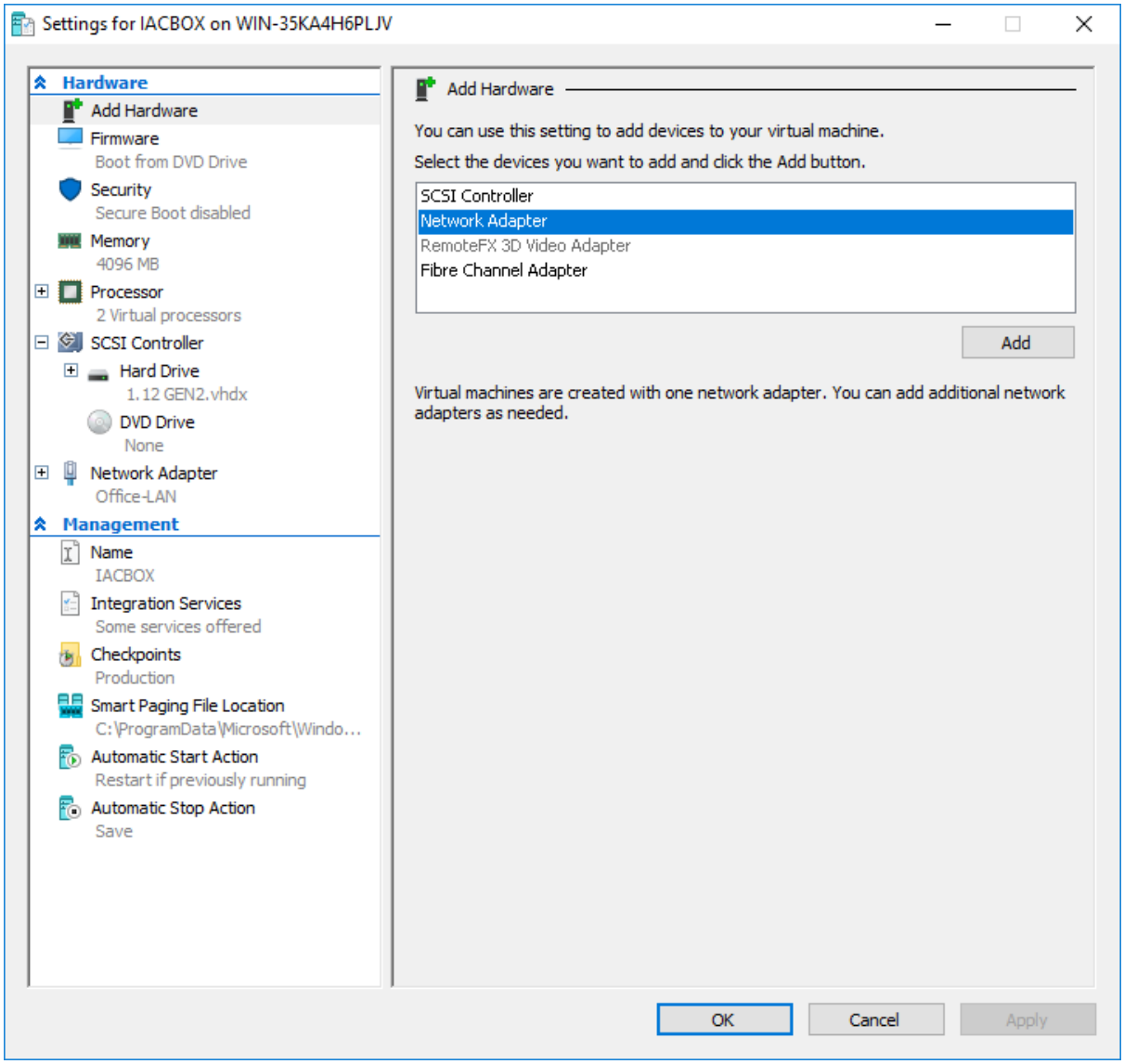
Confirm the settings summary for this virtual machine and click **Finish** to create it.



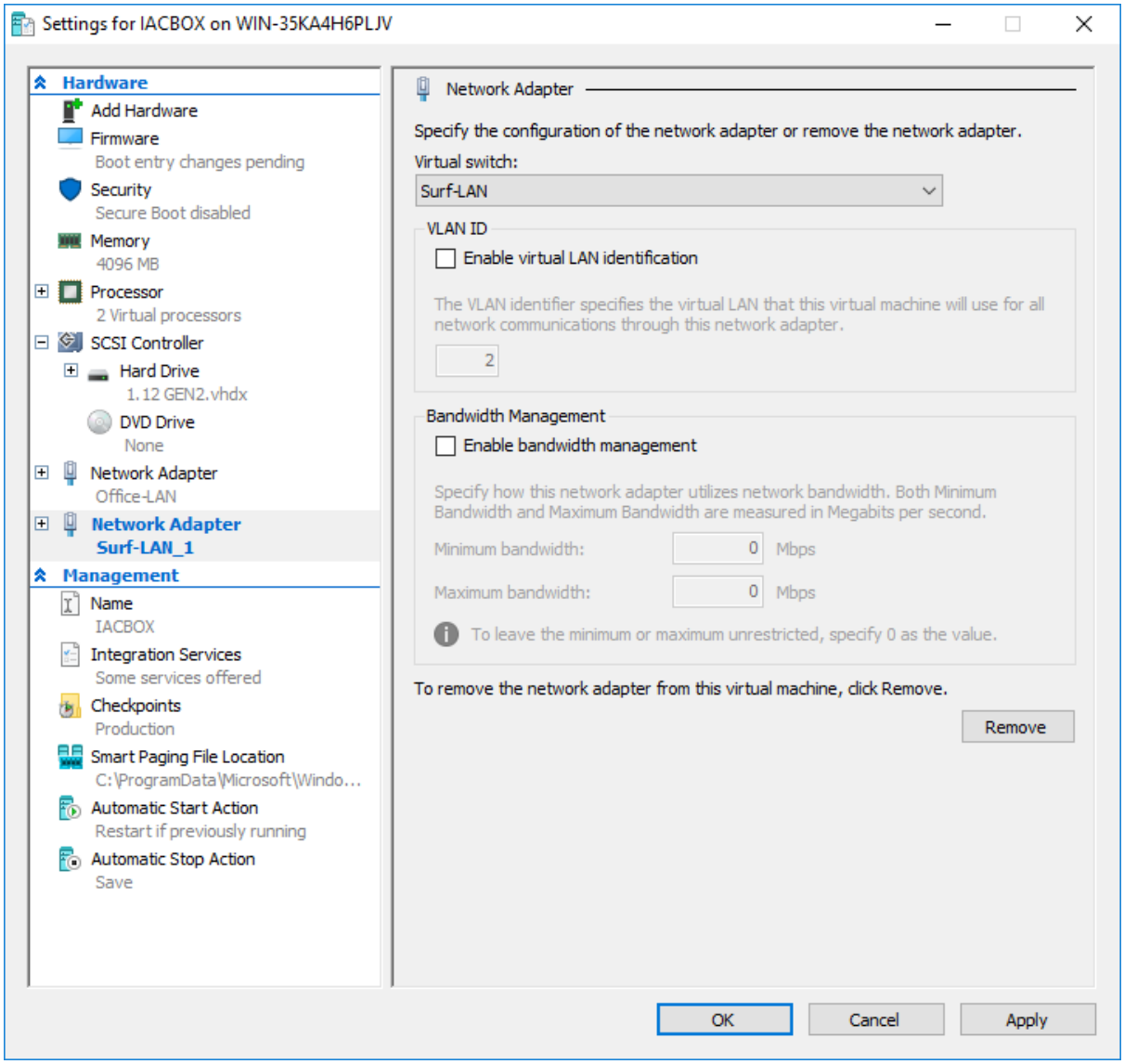
Now the newly configured virtual machine will show up in the main list of *Hyper-V*. Select the new virtual machine and then click on **Settings** which can be found on the right side.



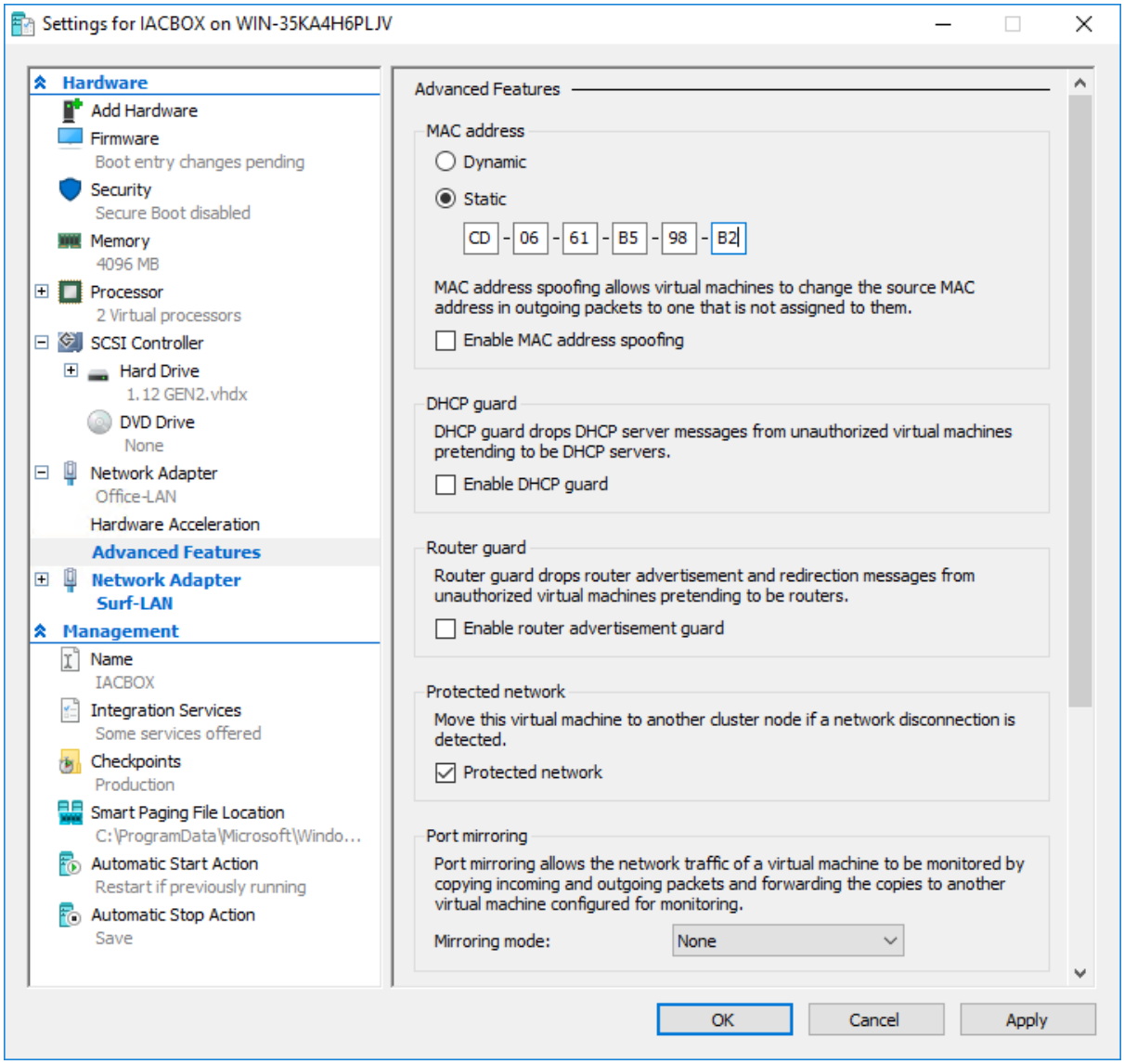
After the initial Creation, the VM will only consist of one network interface. Click on *Add Hardware* and select **Network Adapter** to add a second Interface for the Surf-LAN.



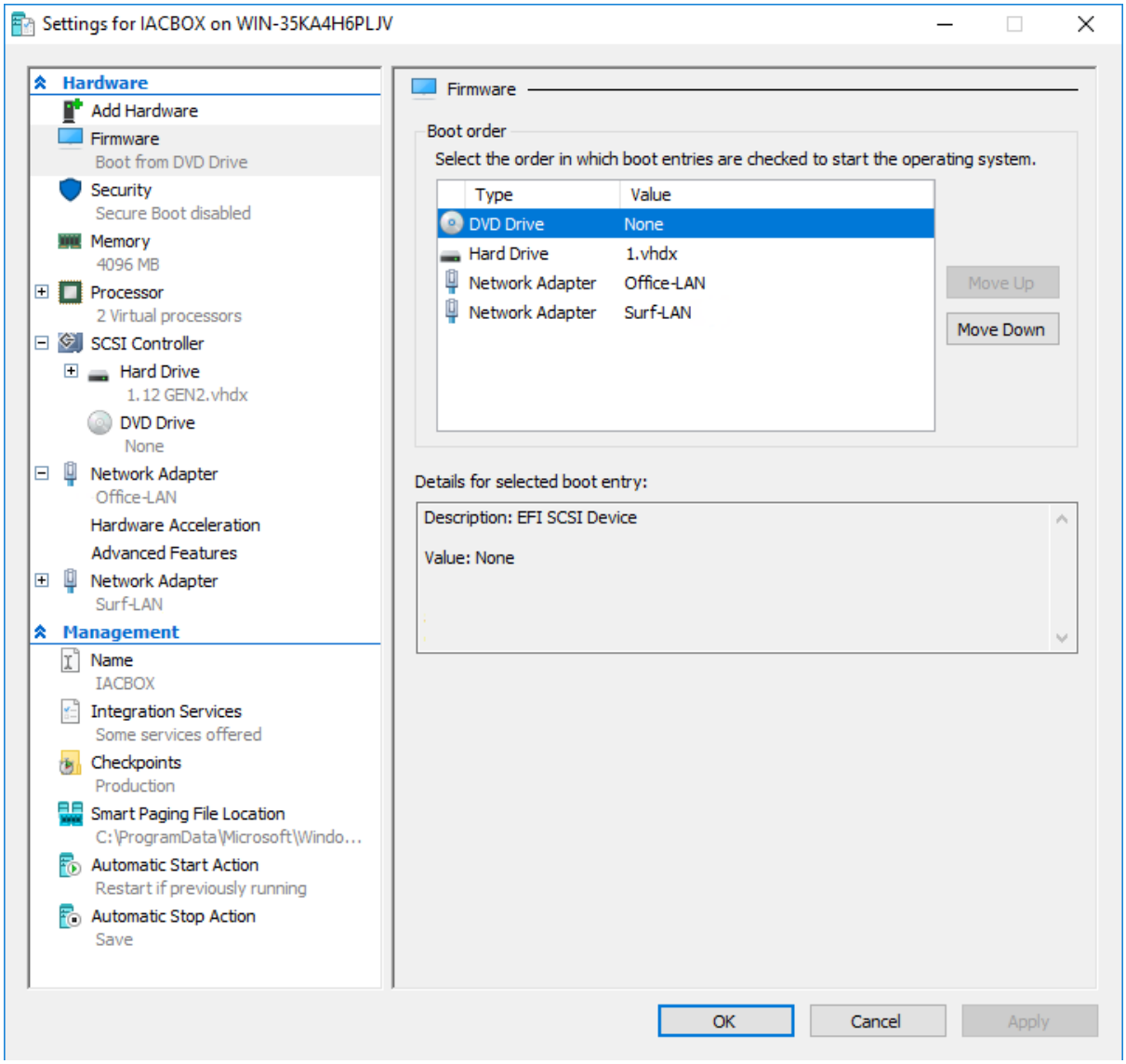
Click on the newly added **Network Adapter** and assign its Virtual switch to the previously created **Surf-LAN vswitch**.



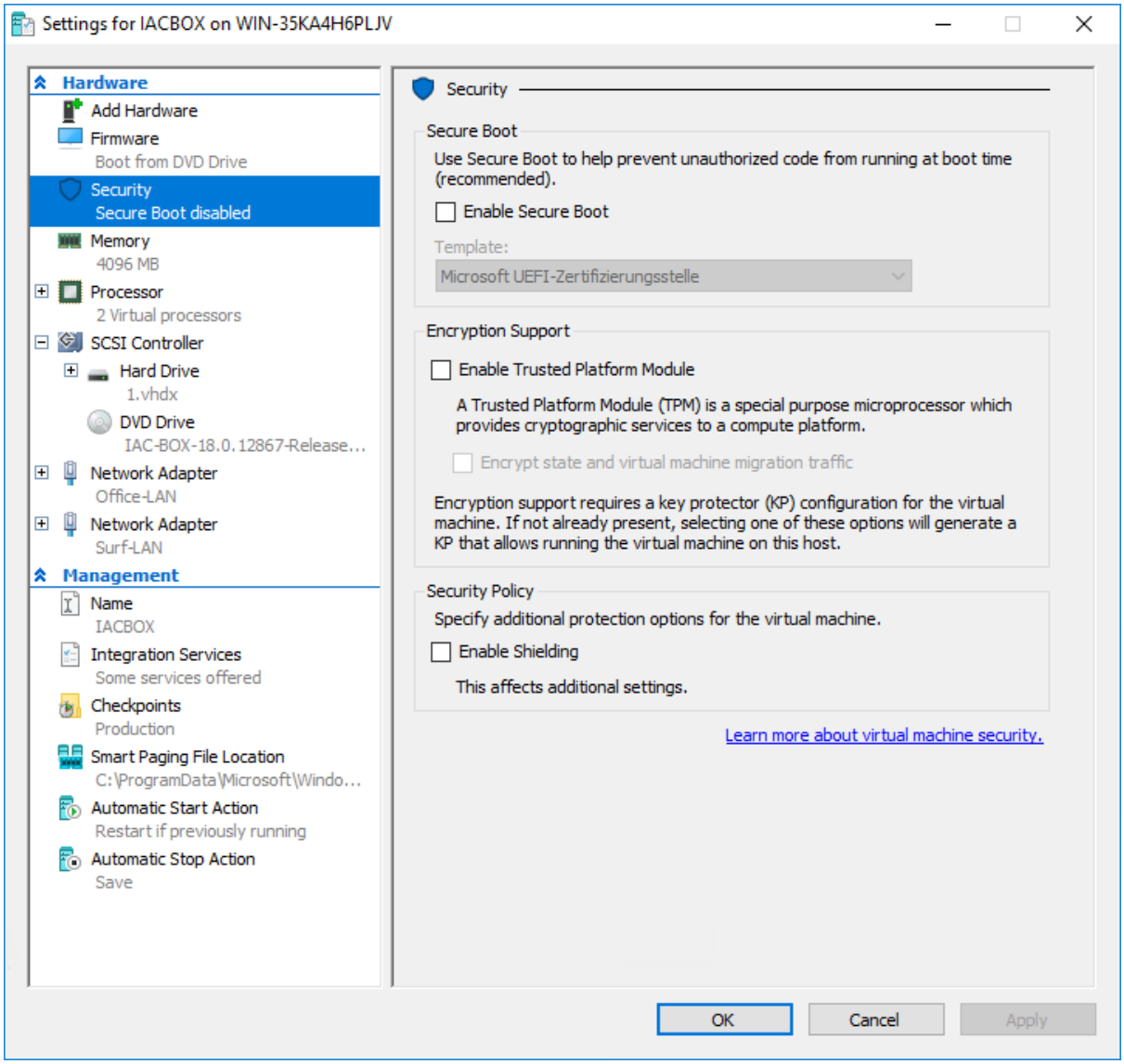
Now both Interfaces should be almost done. The last missing step is to ensure, that they will consist of a **static MAC address**. **Expand** the Network Interface settings by clicking on the **+ Symbol**, select **Advanced Features** and assign a **Static MAC Address** to both, the **Office-LAN** and **Surf-LAN** Network Interface.



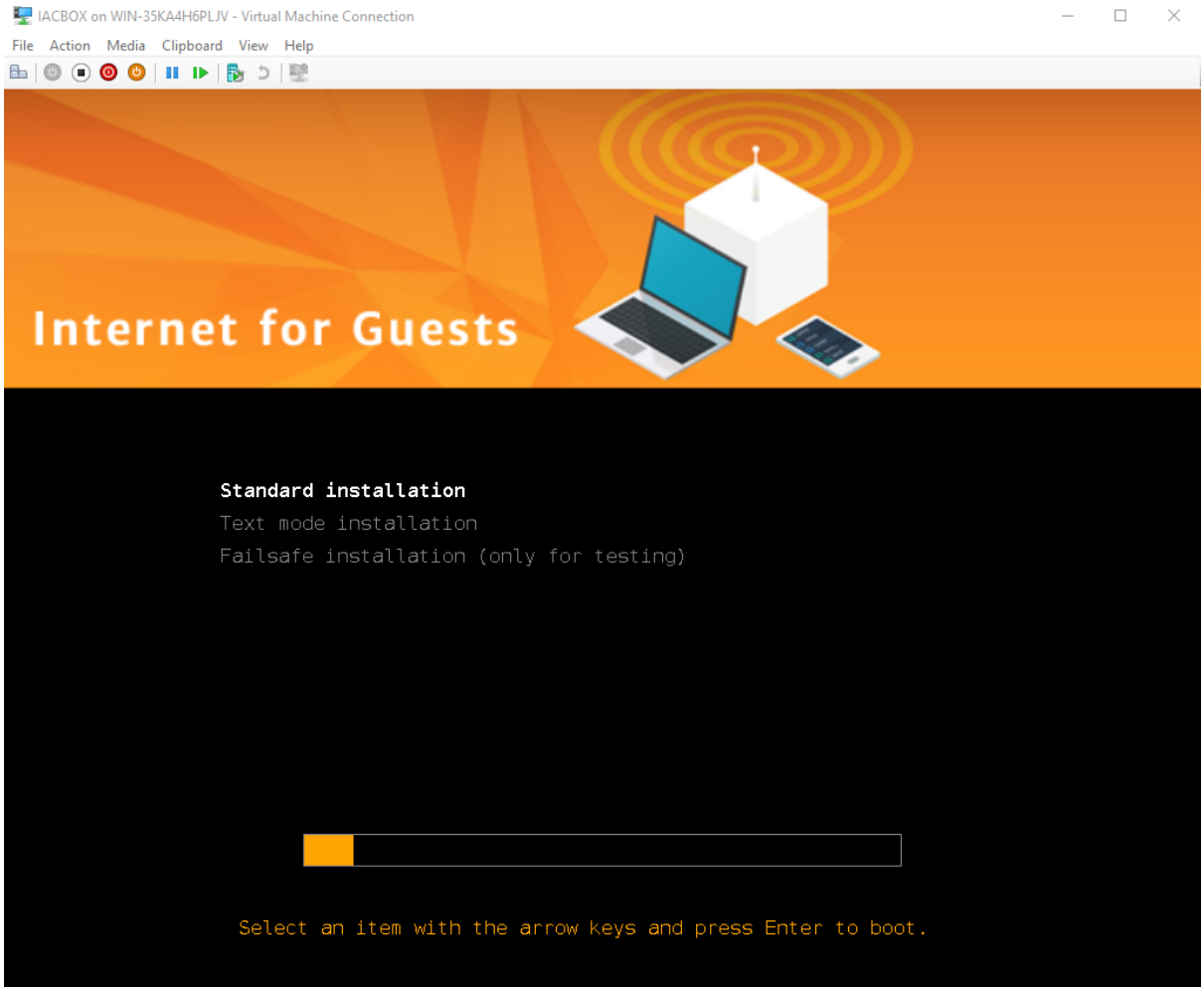
After that, navigate to the entry **Firmware**. This page lists the **Boot Order** of all connected components. Here it is important to move the Network Interfaces **Down**, to avoid an endless UEFI network boot loop. Since this VM will be used for the IACBOX installation next, a virtual DVD-ROM drive should be on top and secondly, it should boot from the hard drive.



Next, open up the entry **Security** and **disable all checkboxes**.



Now Return to the **Hyper-V Manager**, select the **IACBOX virtual machine** and click on **Connect**. In the new **Virtual Machine Connection** window click on **Action** and then **Start** to start up the new IACBOX virtual machine. If the configuration of the virtual machine has been done right, it will start to boot from the IACBOX ISO image or from the IACBOX CD/DVD.



Now you can go on with the installation of IACBOX.

Hint: Please note that in case the setup does get stuck while initializing, you can press **ESC** to continue.

3.2 Installation on KVM

This manual describes how to install the IACBOX in a KVM-based environment. For demonstration purposes **Proxmox VE** is used in this manual. Competing products such as **Redhat Enterprise Server / CentOS**, **SuSE Enterprise Server / OpenSUSE** are also supported and tested.

Attention: The kernel version needs to be > 3.2 which is provided since IACBOX version 4.0. If an older version is used, please reinstall with a current install image.

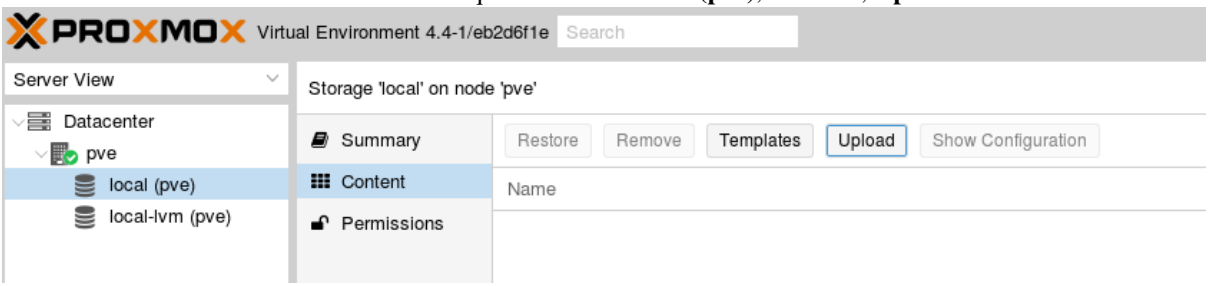
Hint:

- This Proxmox VE demonstration uses **Proxmox VE Version 4.4**.
- It is crucial to utilize the **virtio-block** device as a base for virtualized IACBOX hard drives, best performance results can be achieved with it.
- It is recommended to use the most recent version of the according VM software.
- The IACBOX is a realtime system. Therefore it is **critical** to only assign and use ressources which are capable of working within the same operational context. For example: If a VM cluster is used, only the

CPU cores of one socket may be used for the IACBOX to avoid rapid context switching between multiple sockets, which would lead to intolerable delays and possible system faults.

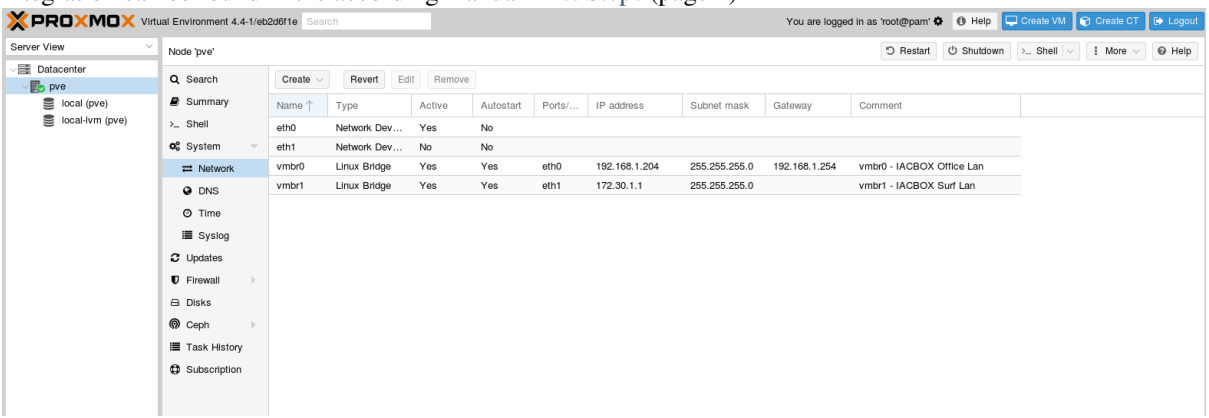
3.2.1 Uploading the Installation Medium

For Proxmox a ISO/CD-based medium is required. Choose **local (pve)**, **Content**, **Upload**



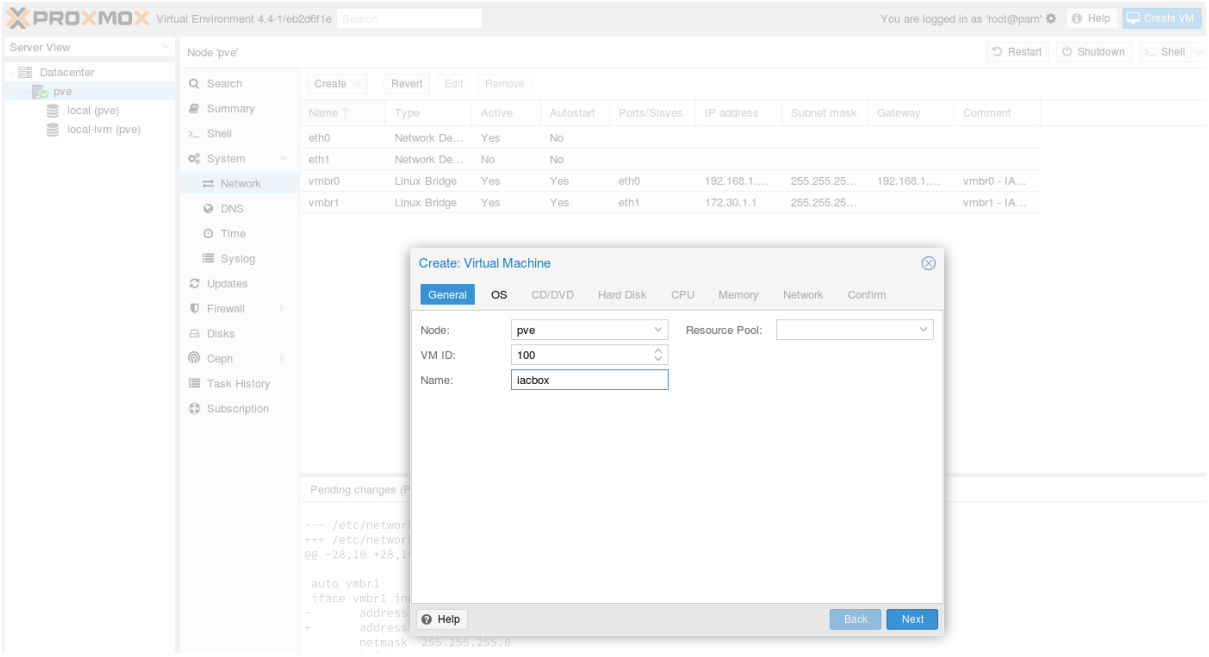
3.2.2 Network setup

The network architecture of the VM host is essential to provide a good integration of *Office-LAN* and *Surf-LAN*. Therefore pay attention to the network integration of this installation. Further information regarding network integration can be found in the according manual First Steps (page 2)

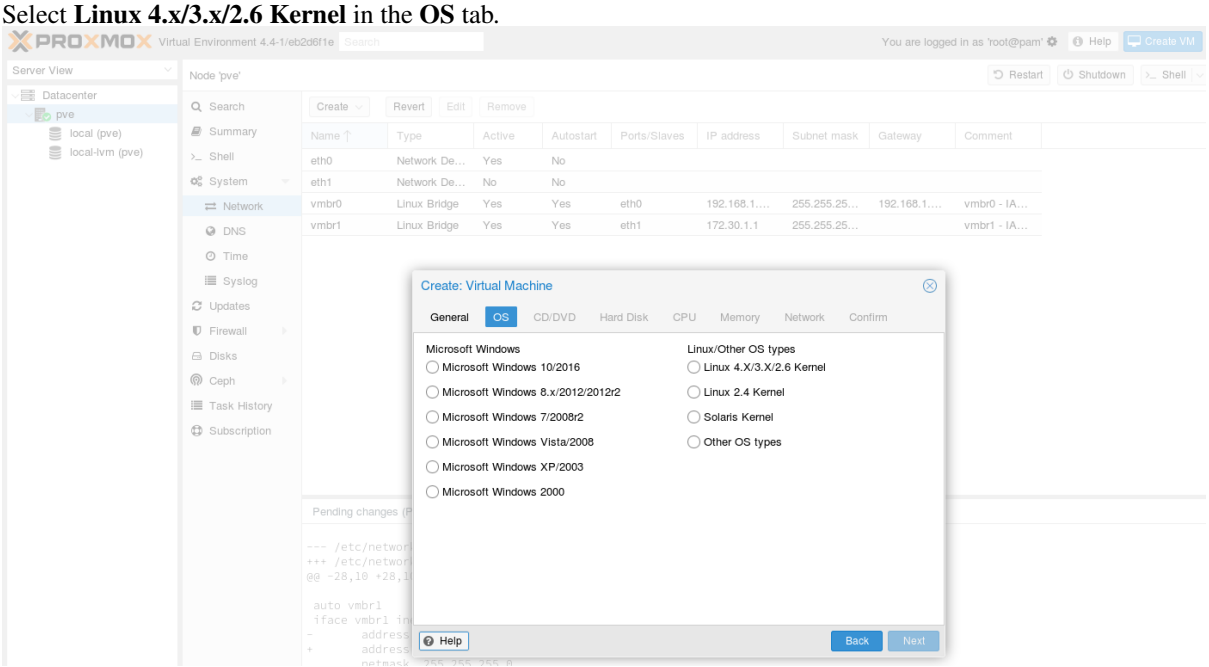


3.2.3 Installation

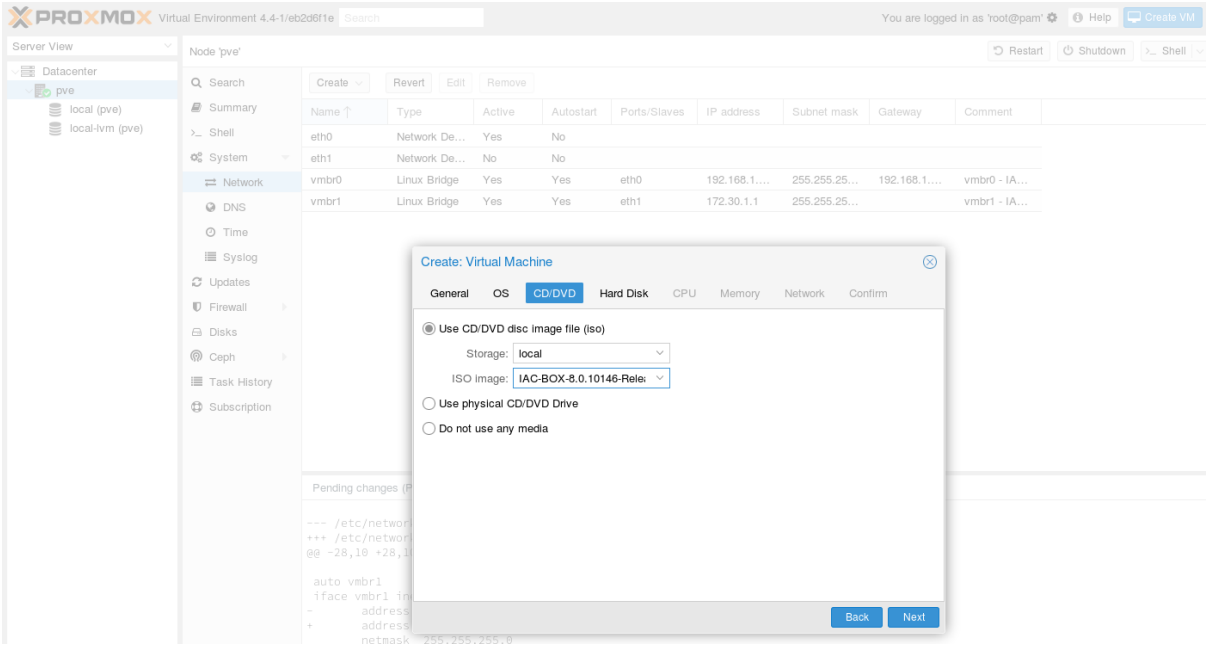
Select **Create VM** and configure it with the following defaults.



Select **Linux 4.x/3.x/2.6 Kernel** in the **OS** tab.

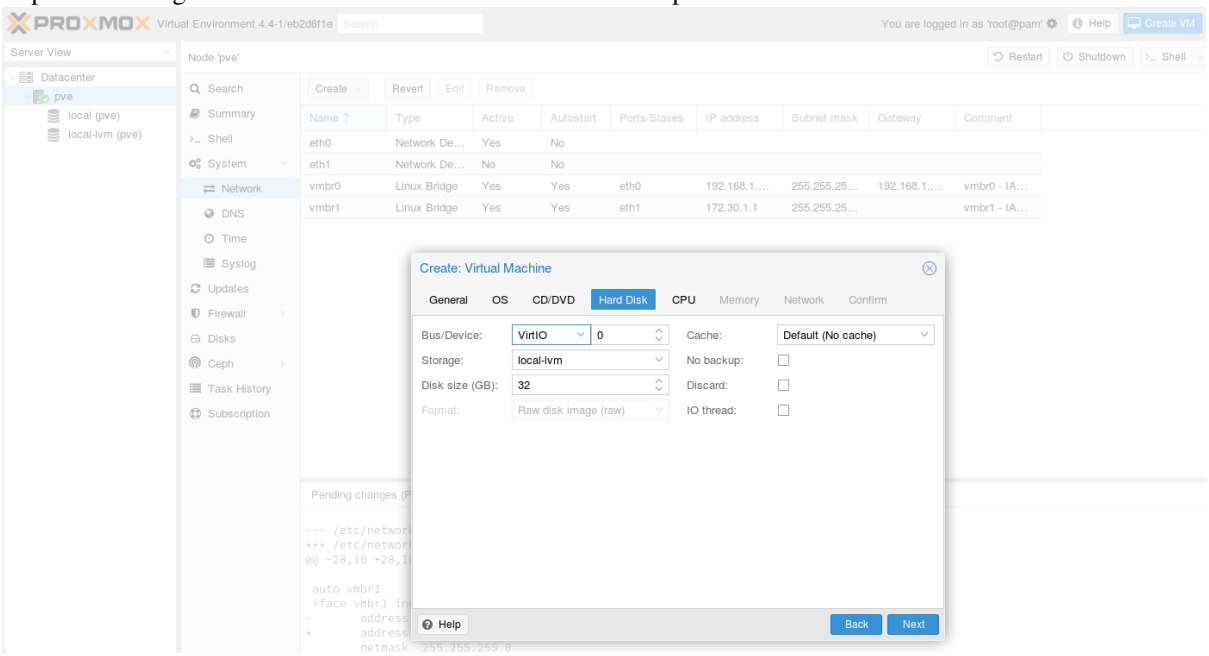


In tab **CD/DVD** select the installation medium which was mentioned in the first step of this manual.

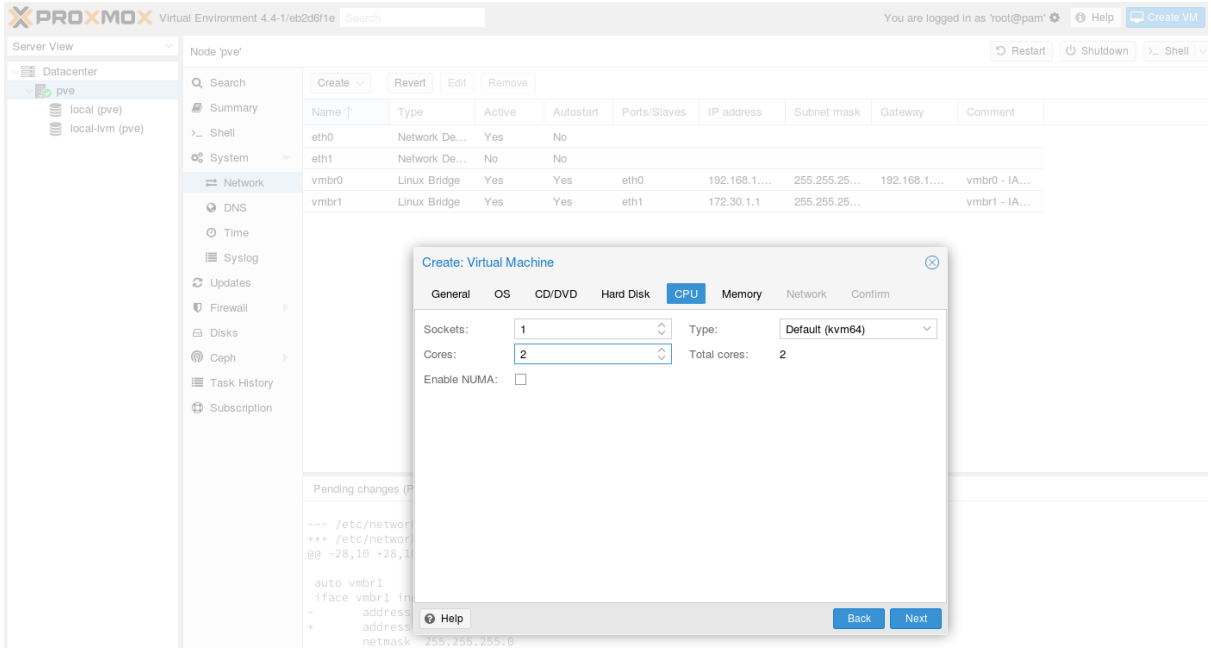


In the tab **Hard Disk** you can create a new disk for the IACBOX.

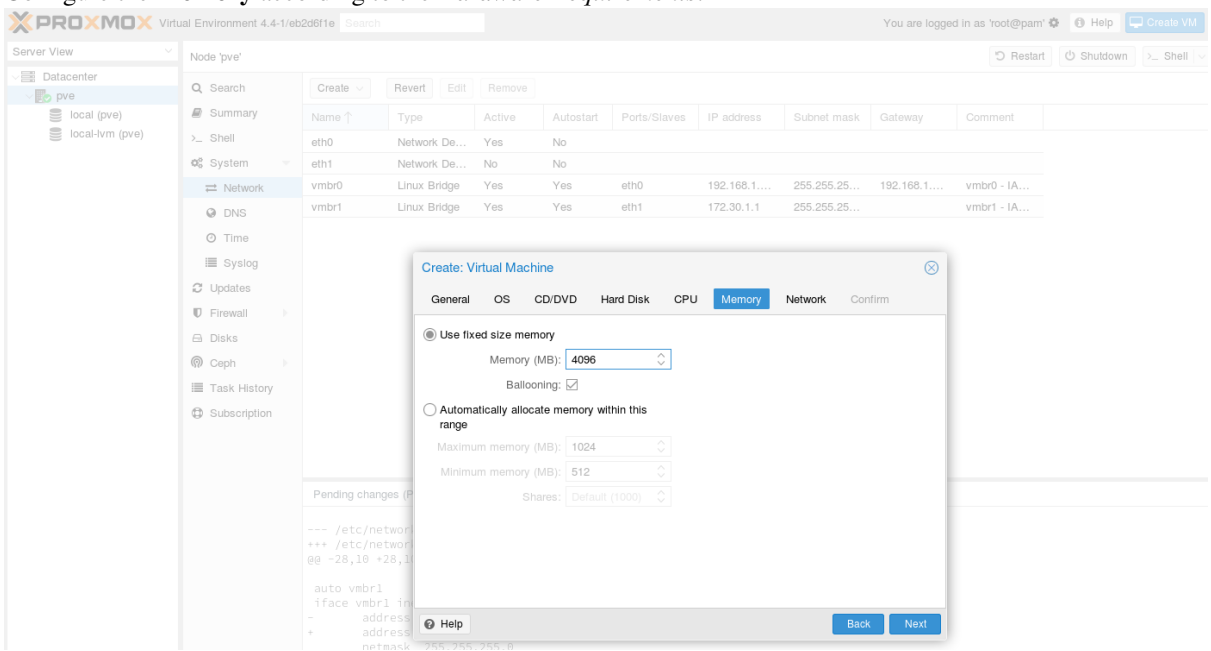
Important: Using the **VirtIO** bus is essential as described on top of this manual.



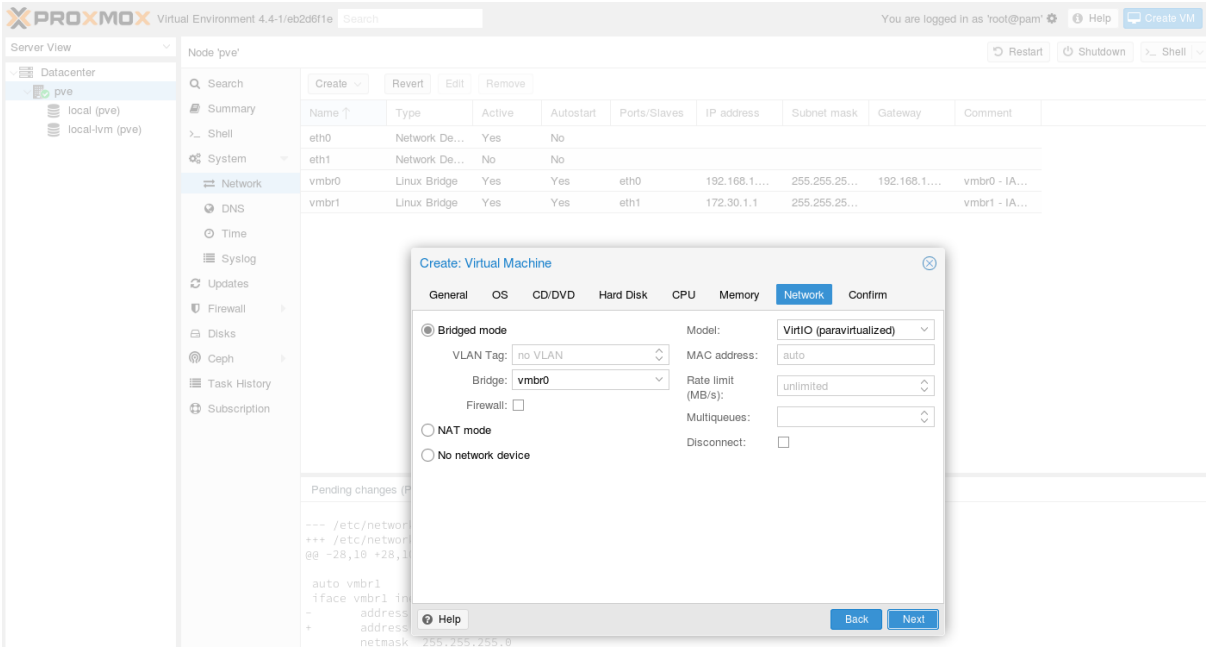
The **CPU** and **Memory** settings must be configured according to the minimum **hardware requirements** which can be found on top of this document - or **better**. The *Hardware Requirements* can be found here: Hardware Requirements page.



Configure the **Memory** according to the *Hardware Requirements*.



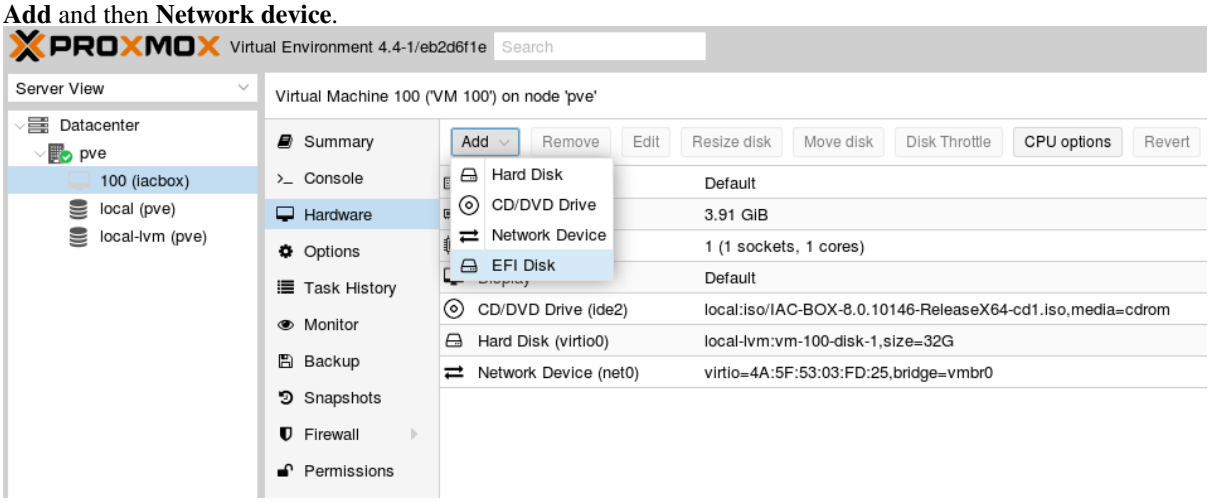
In the **Network** tab configure the virtual interfaces according to your requirements. Note that the IACBOX will require two interfaces, the *Office-LAN* and the *Surf-LAN*. While the *Office-LAN* is bridged to the uplink of the host system (according to the screenshot), the *Surf-LAN* can be configured as an isolated interface later on.



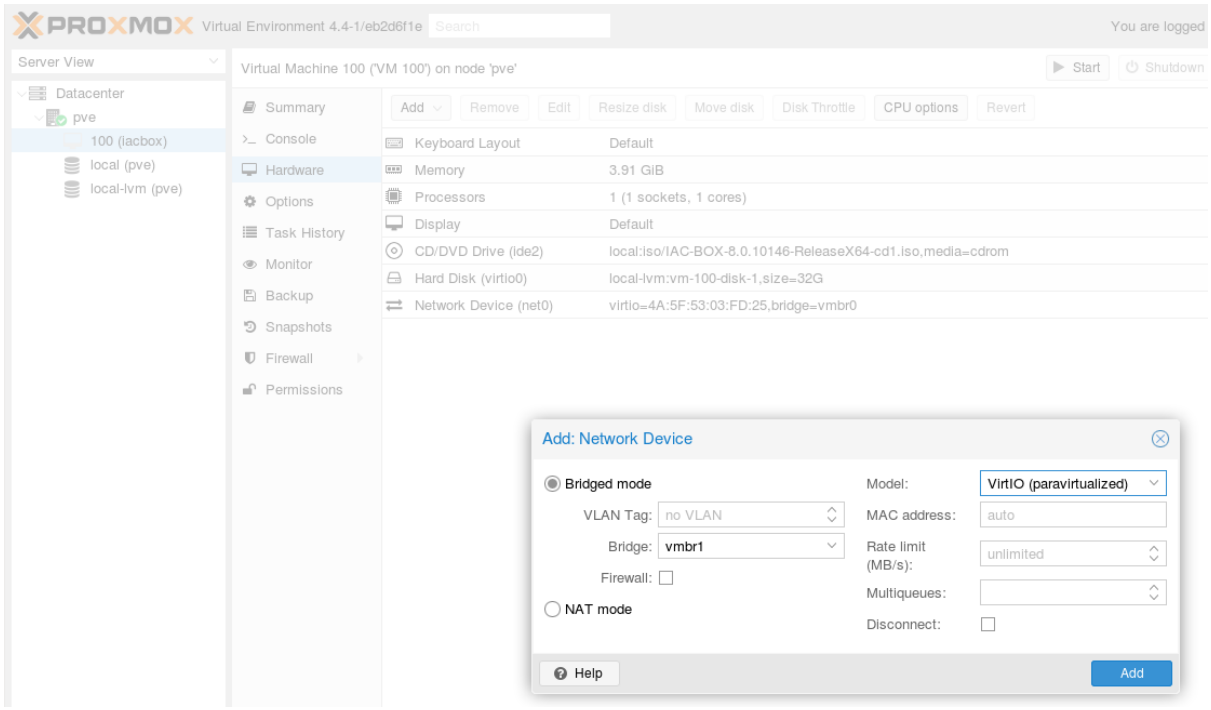
The last step is a summary of all the entered data above. Confirm it with **Finish**.

3.2.4 Surf LAN Interface

At last a second adapter must be added for the Surf-LAN. Therefore select the *VM* and then click on **Hardware**, **Add** and then **Network device**.



The bridge interface is called **vmbr1** and the network device model is **VirtIO**.



3.3 Installation on VMware ESXi

This manual describes the steps to configure and prepare **VMware ESXi Version 5.5** or newer in order to install the IACBOX.

Hint:

- A 64-bit host-system is required.
- It is strongly recommended to use a dedicated physical network interface card for the **Surf-LAN**.
- The system must be online at any time in order to synchronize necessary IACBOX registration data with the licensing server.
- This manual describes the installation of the IACBOX on ESXi, not the ESXi installation itself.
- The IACBOX is a realtime system. Therefore it is **critical** to only assign and use ressources which are capable of working within the same operational context. For example: If a VM cluster is used, only the CPU cores of one socket may be used for the IACBOX to avoid rapid context switching between multiple sockets, which would lead to intolerable delays and possible system faults.
- The IACBOX is a **closed system**, therefore utilities like VMware Tools, which would need full access to the system, can not be installed.
- The IAC-BOX is a realtime system. Therefore it is **critical** to only assign and use ressources which are capable of working within the same operational context. For example: If a VM cluster is used, only the CPU cores of one socket may be used for the IAC-BOX to avoid rapid context switching between multiple sockets, which would lead to intolerable delays and possible system faults.
- The IAC-BOX is a **closed system**, therefore utilities like VMware Tools, which would need full access to the system, can not be installed.

Please note the minimum hardware requirements.

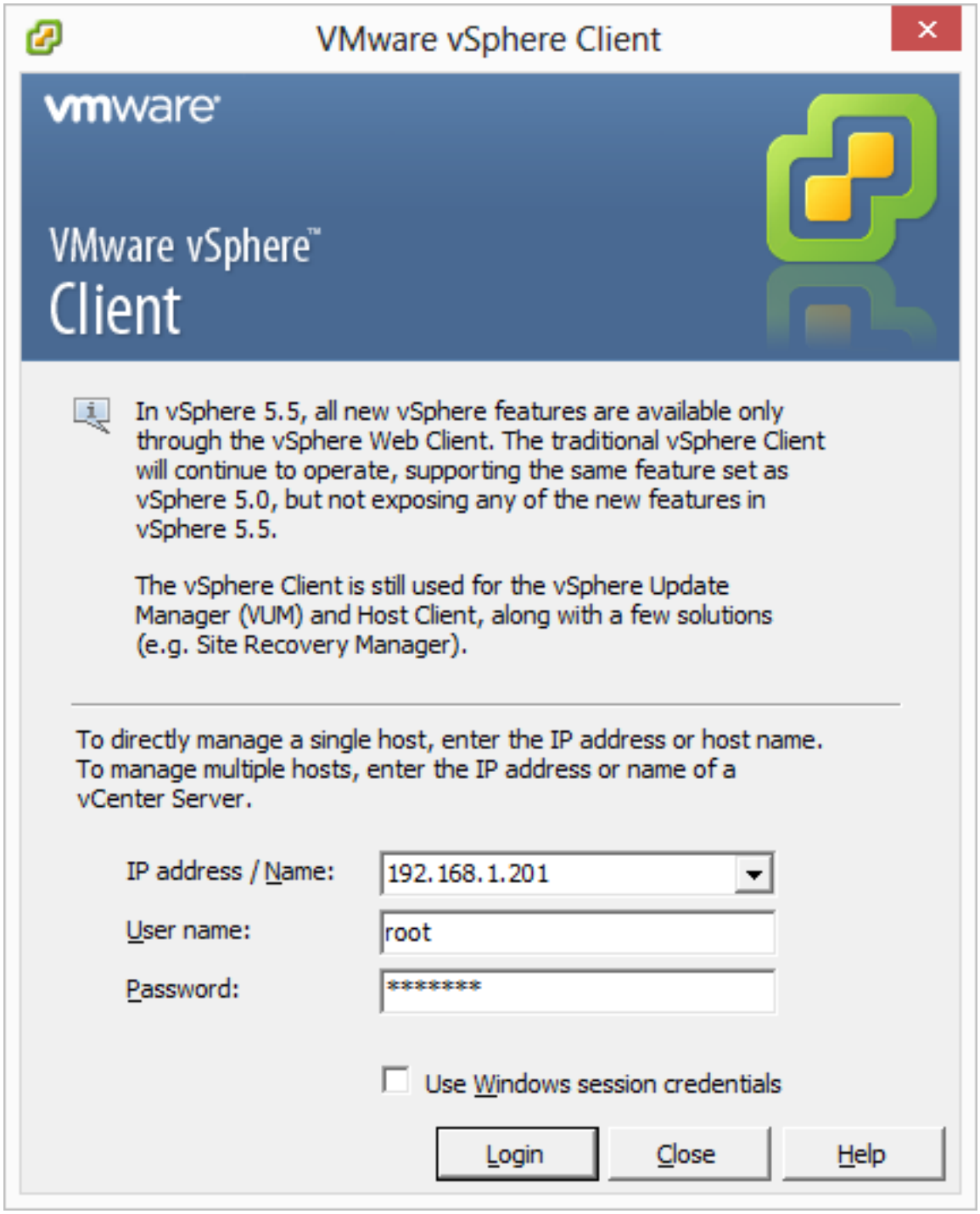
Users / Devices	CPU Cores	RAM	HDD Capacity
10 users	1	2 GB	60 GB
25 users	1	2 GB	60 GB
50 users	2	2 GB	60 GB
75 users	2	2 GB	60 GB
100 users	2	2 GB	60 GB
175 users	2	2 GB	60 GB
250 users	2	4 GB	1 TB
375 users	4	4 GB	1 TB
500 users	4	4 GB	1 TB
750 users	4	4 GB	1 TB
1000 users	4	8 GB	1 TB
1500 users	4	8 GB	1 TB
2000 users	8	8 GB	1 TB
3000 users	8	16 GB	1 TB
Unlimited users	8	16 GB	1 TB

Attention:

- Starting from 250 users a processor with at least 2,50 Ghz or better is required.
- Virtualized environments generally need more ressources due to the nature of virtualization.
- Functions like the **Advanced Web Filter**, the **Application Control** or the **Connection Tracking** are very CPU-intensive and should therefore be used with caution.
- All specifications are subject to change without notice. The most recent version of this list can always be found here: http://www.iacbox.com/en/support/hardware-requirements/
- In order to use the new *DNS based Web Filter* of IACBOX version 17.2, at least 4GB of internal memory must be available.

3.3.1 Preparation

Use the **VMware vSphere** client software to log in on your ESXi server. The client software can be obtained on the VMware homepage by using the following link: http://www.vmware.com/products/vsphere.

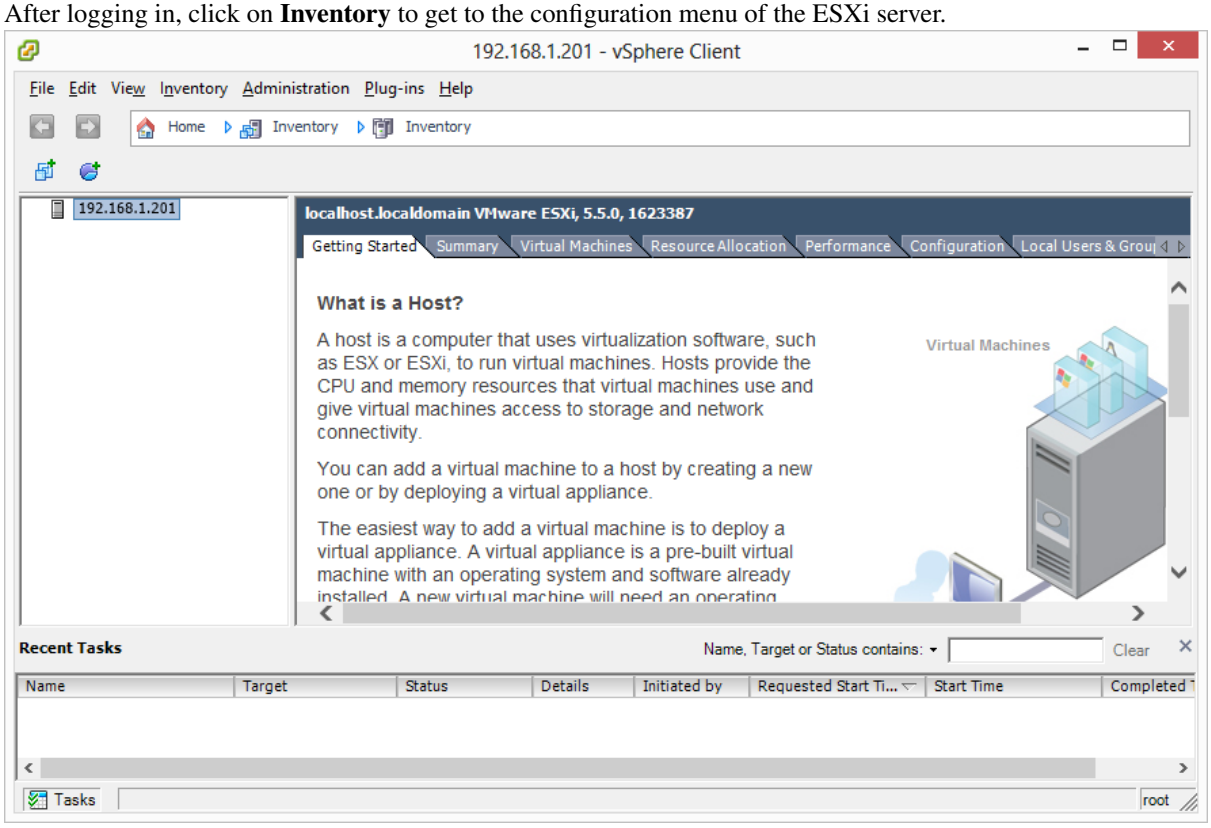


This manual was created with and for ESXi version 5.5. Other versions may differ slightly from what is demonstrated in this manual. While logging in the first time you may face a certificate warning.

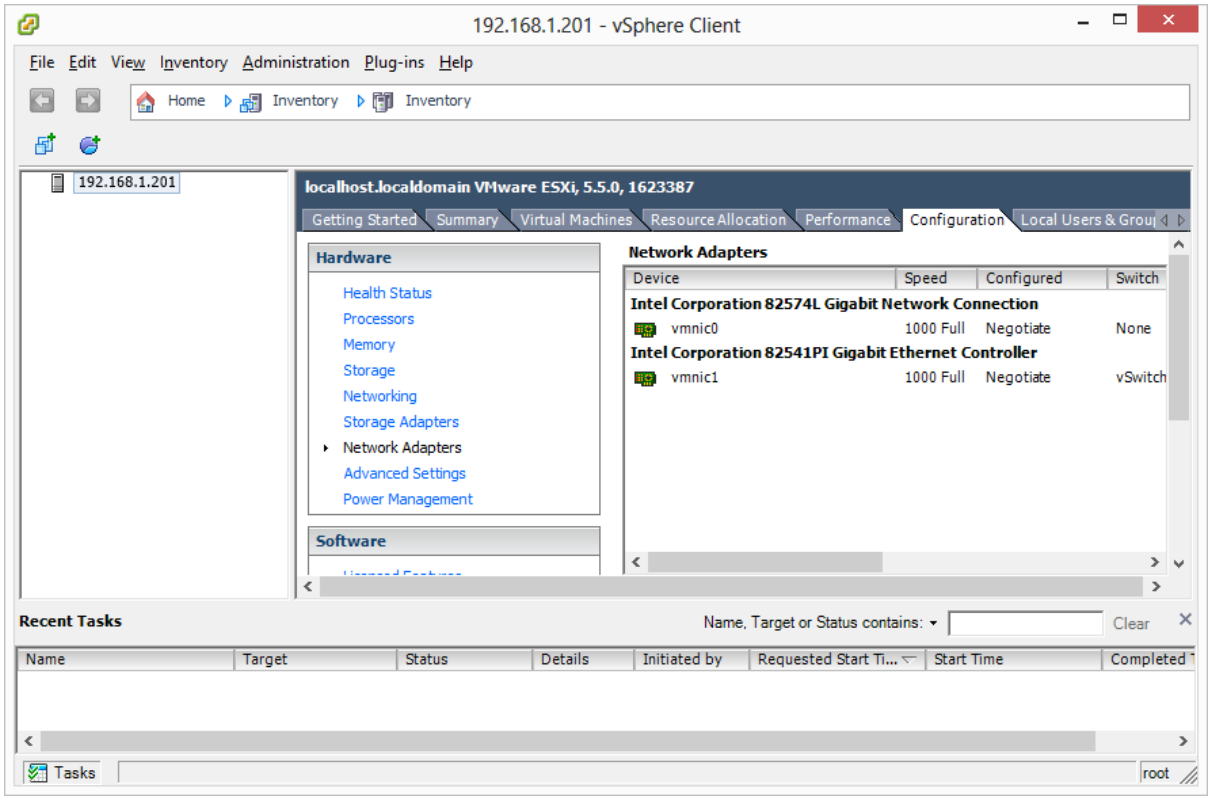
Hint:

- This certificate warning is normal and not critical upon initial usage.
- If you face this warning while you've already installed the certificate and did not change the ESXi server configuration, then it might be break-in attempt.

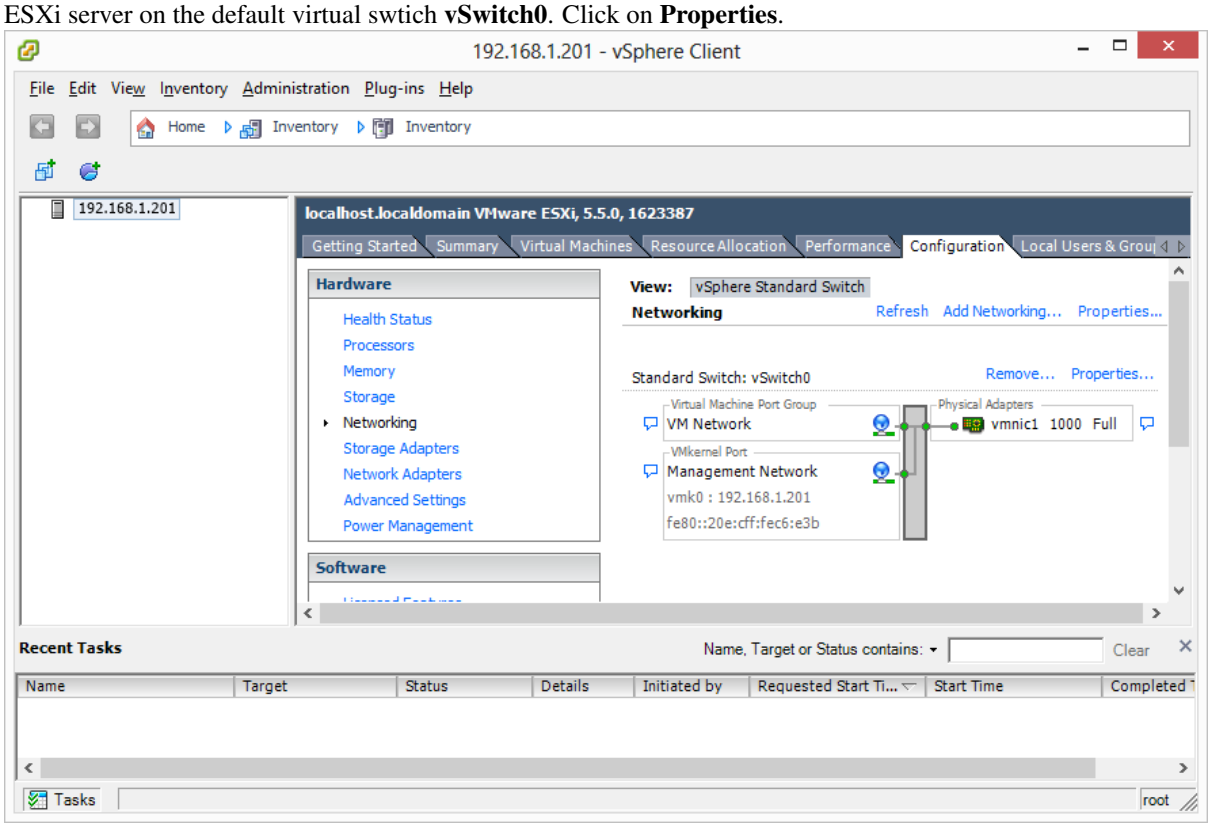
After logging in, click on **Inventory** to get to the configuration menu of the ESXi server.



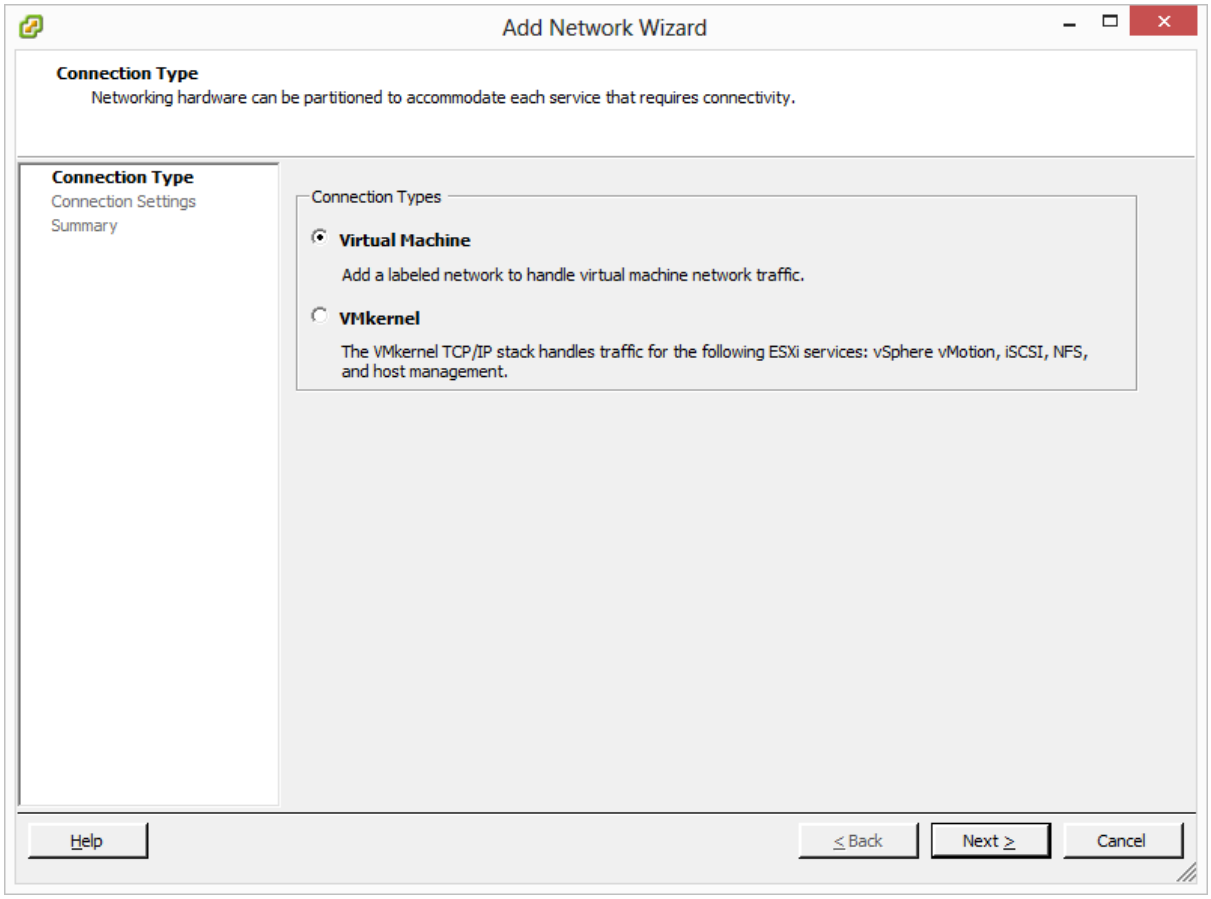
Then navigate to **Configuration / Network Adapters**. This listing shows the mapping of the virtual/physical network interfaces.



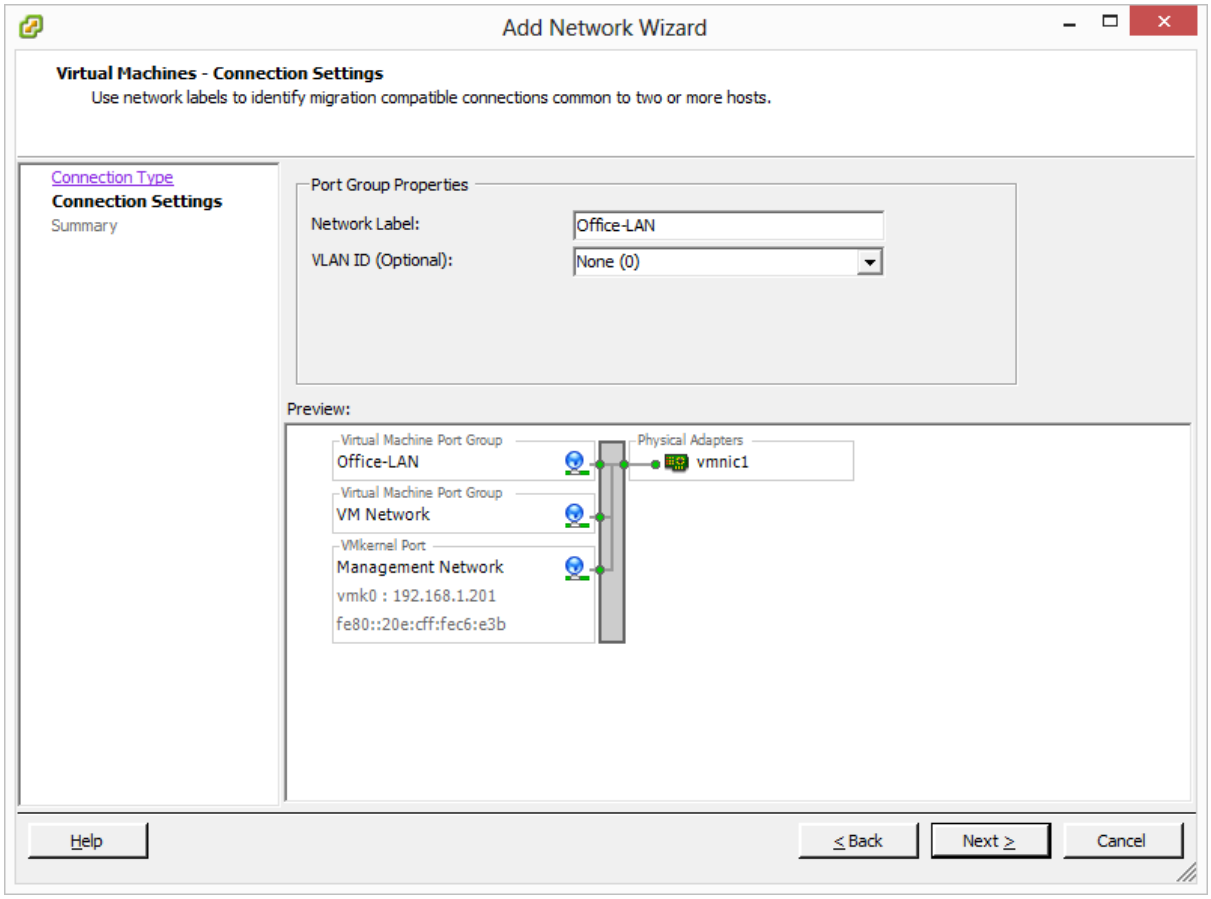
Now click on **Networking** which can be found in the menu on the left side. Here you can see the interfaces of the ESXi server on the default virtual swtich **vSwitch0**. Click on **Properties**.



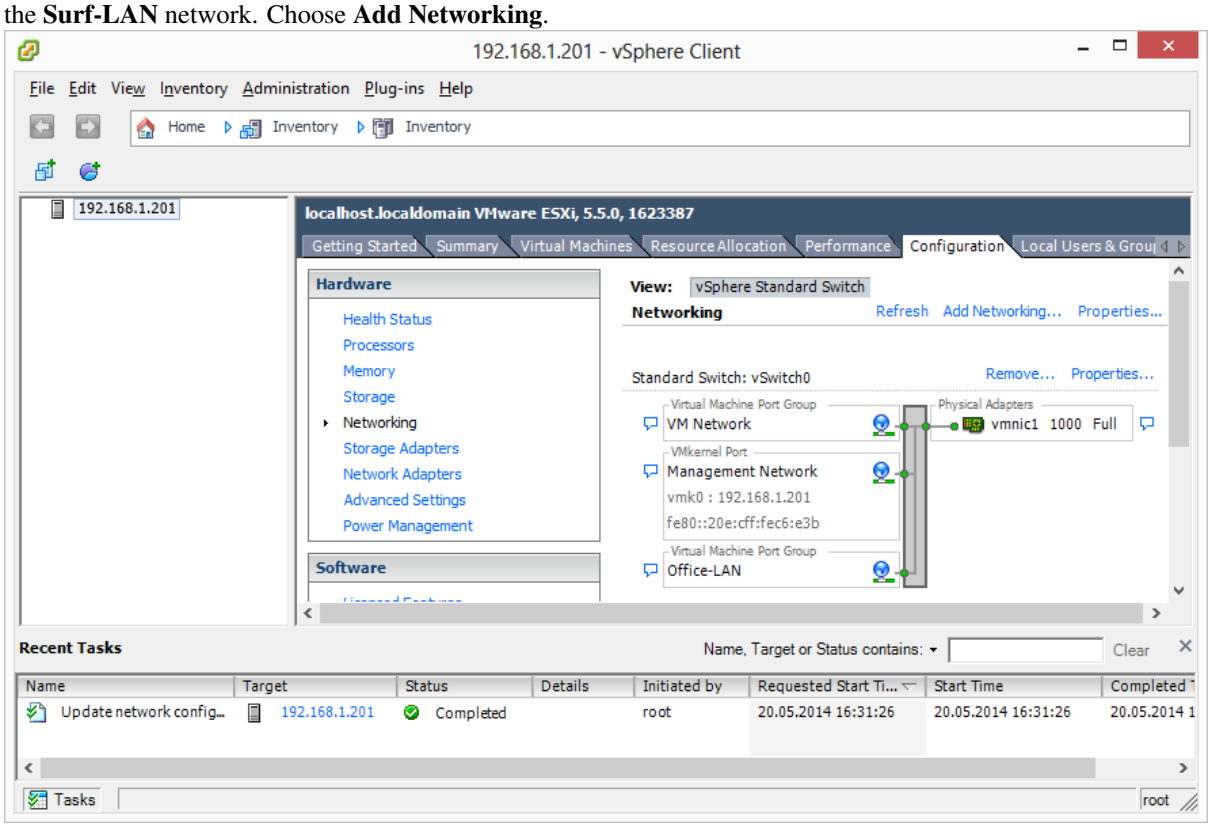
In the next window (*vSwitch0 Properties*) click on **Add**, then add a **Virtual Machine** and click **Next**.



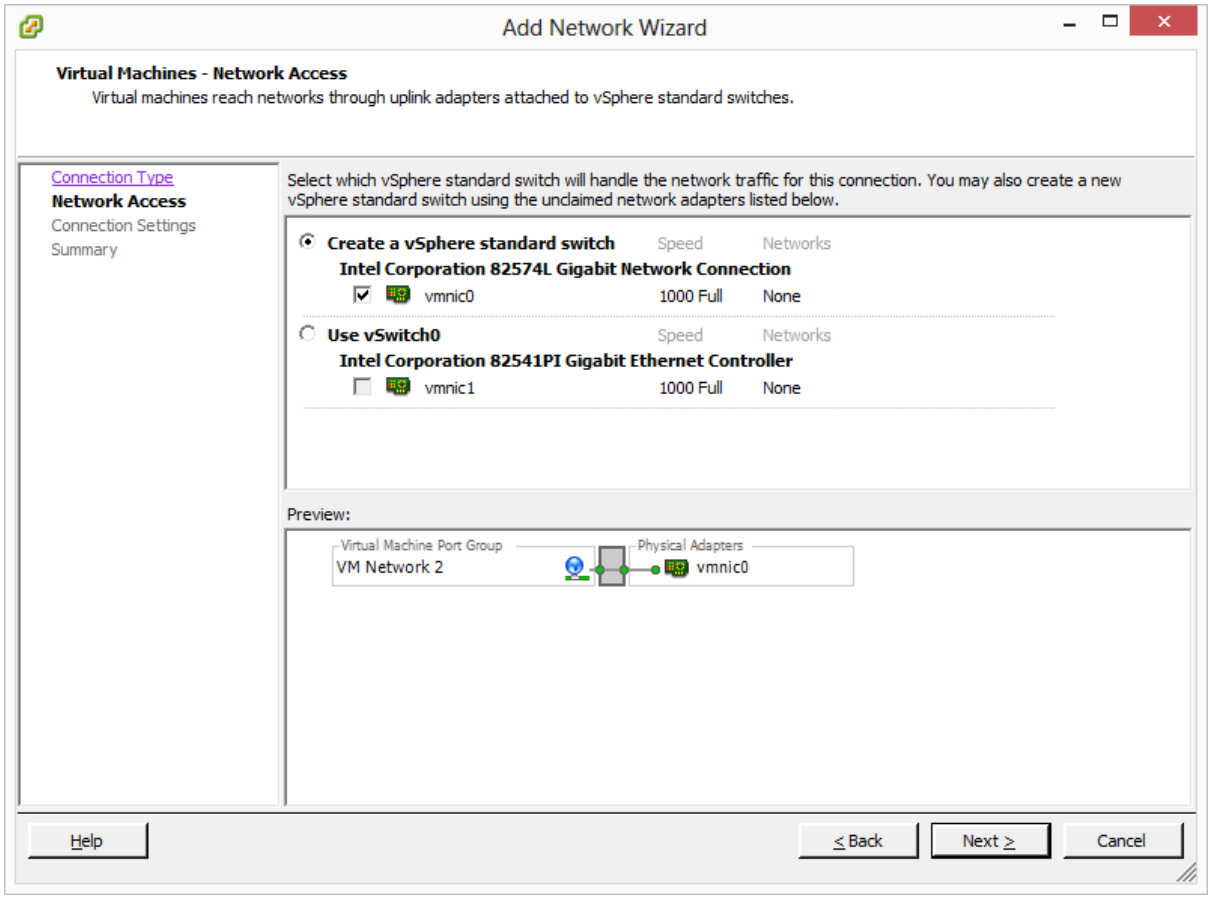
Now type in a name for your **Office-LAN** connection and hit **Next**. Here you see that the **Office-LAN** and **Management-Network** do share the same physical network card.



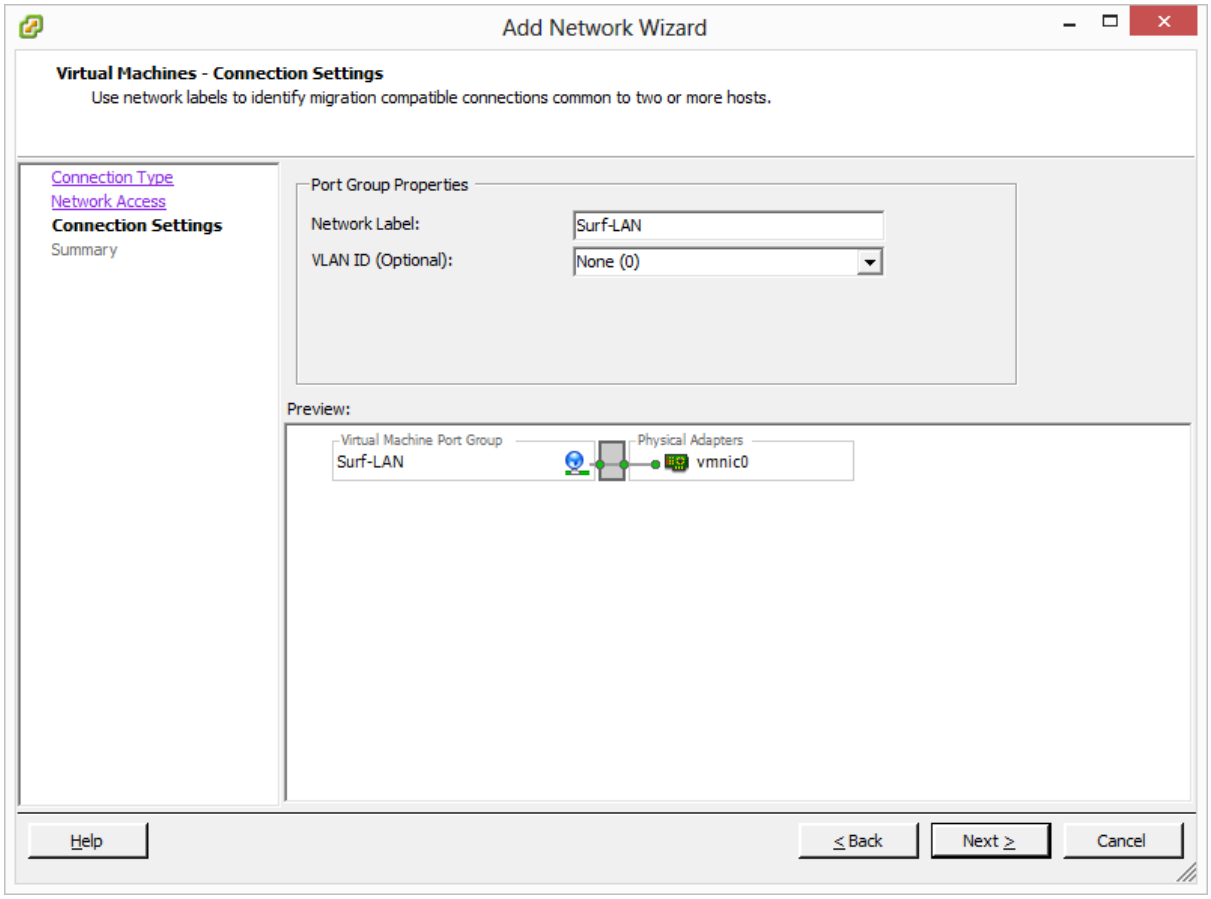
Now click on **Next** and confirm the summary of the changes with **Finish**. The next step is about configuration of the **Surf-LAN** network. Choose **Add Networking**.



As **Connection Type** choose **Virtual Machine** and then click on **Next**. In the next window choose **Create vSphere standard switch** and continue with **Next**.



In the **Connection Settings** enter the name of the **Surf-LAN** network and click on **Next**. After that, confirm the summary screen in the next window and accept by clicking on **Finish**.

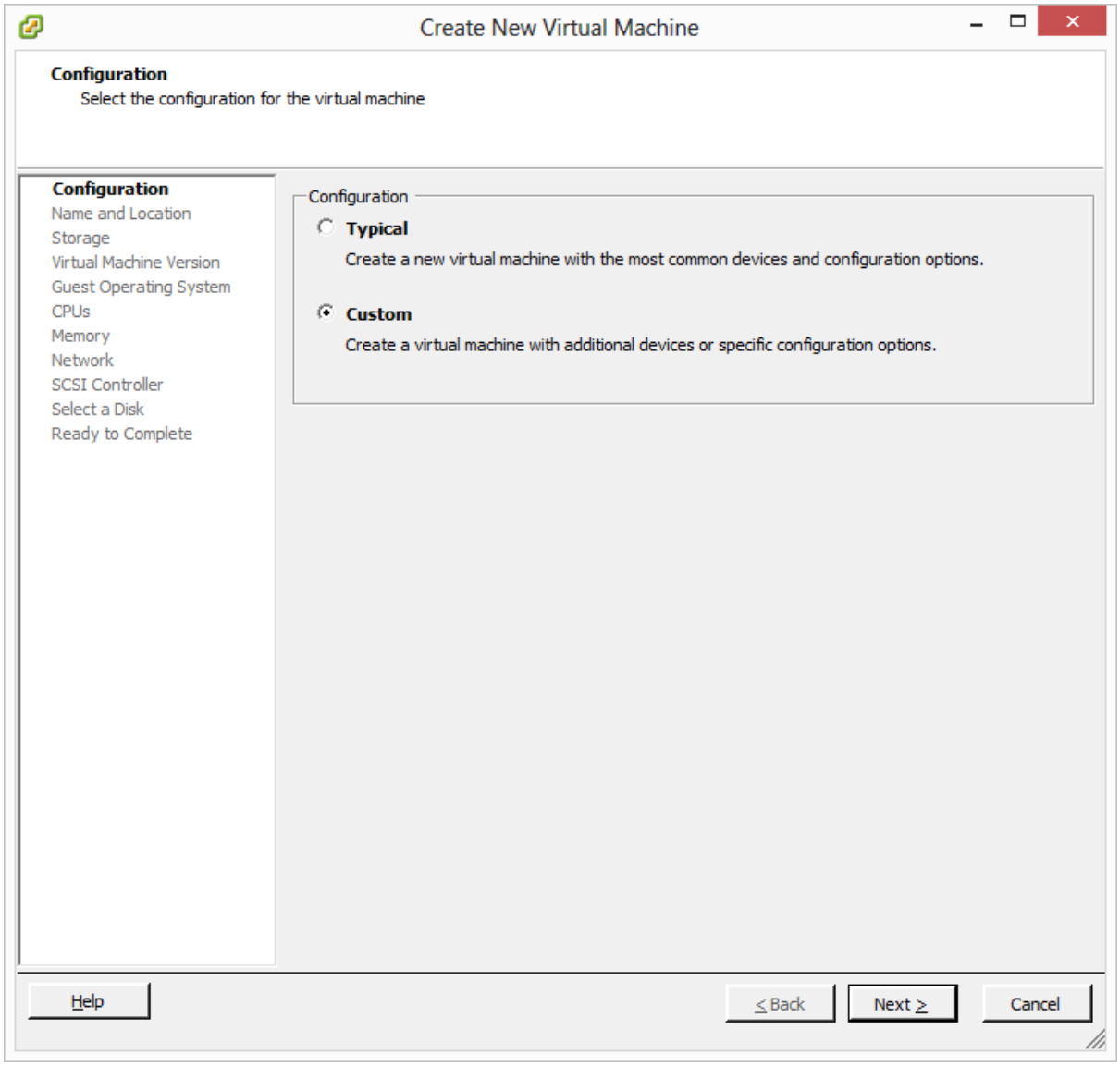


Summary of the configured interfaces:

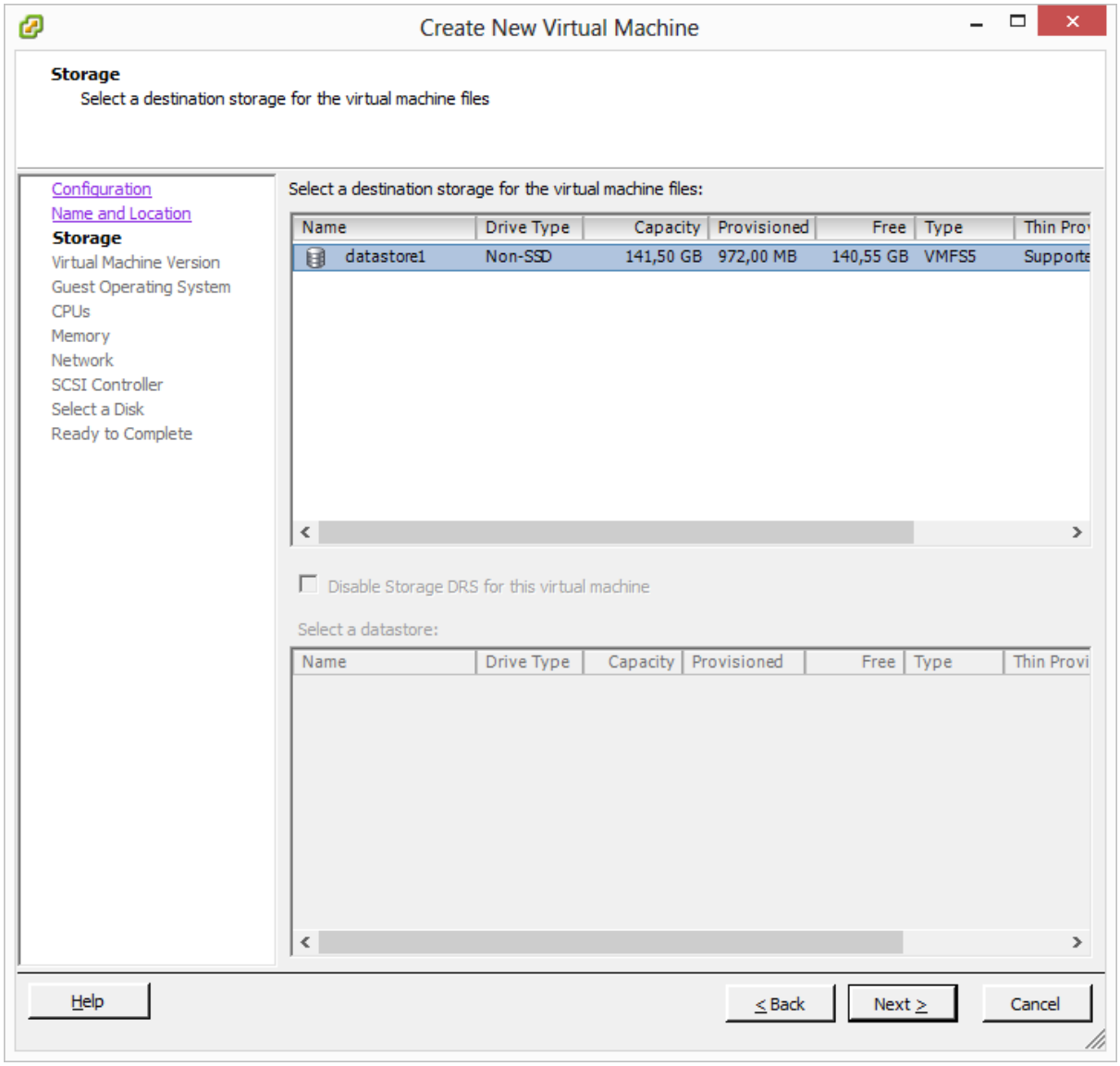
NIC #1		Management Network (vSphere) and Office-LAN
NIC #2		Surf-LAN (Dedicated)

3.3.2 Creating the virtual machine

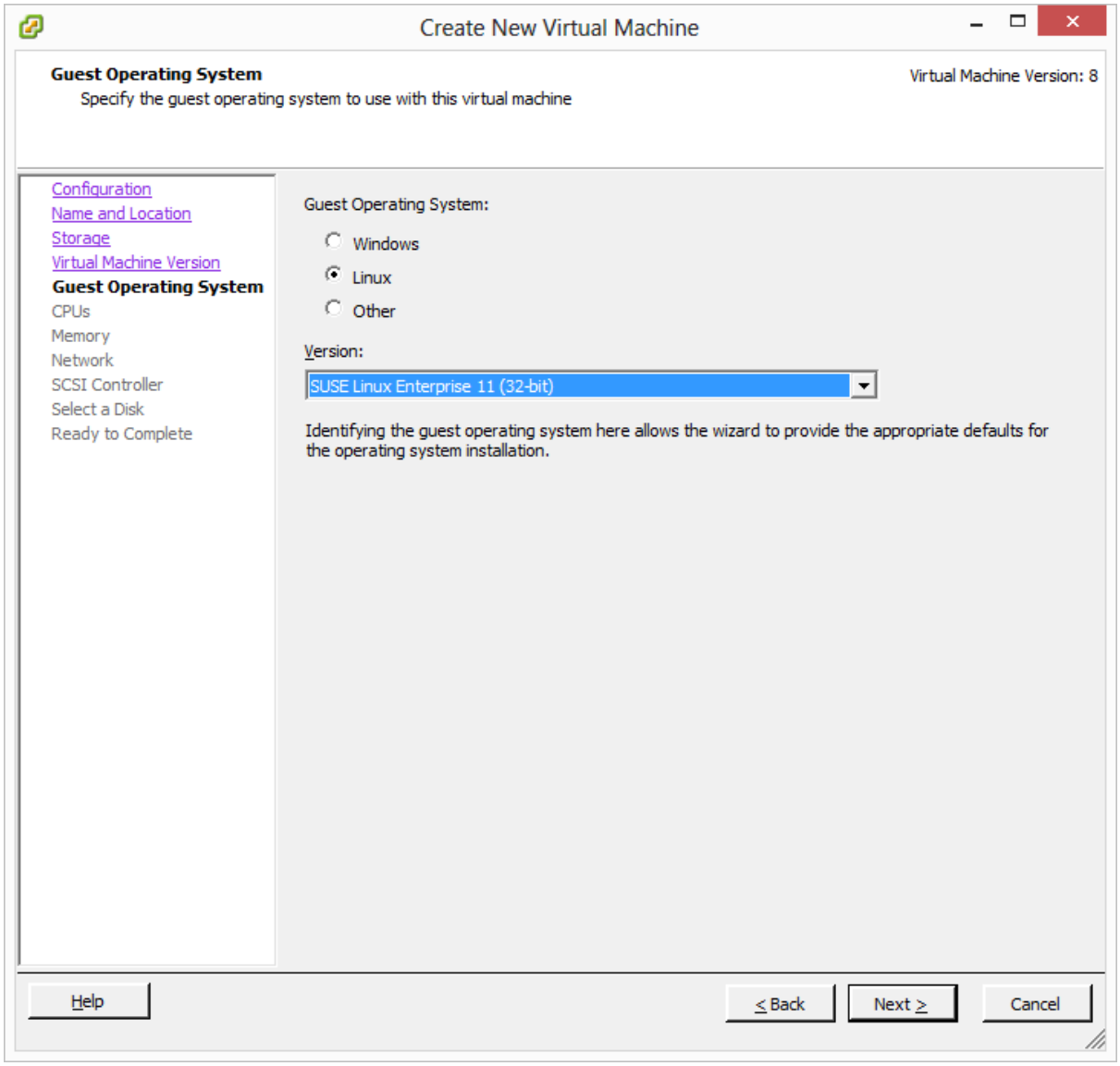
To create the virtual machine click on **File**, **New** and then **Virtual Machine**. A configuration window will open, choose **Custom** and continue with **Next**.



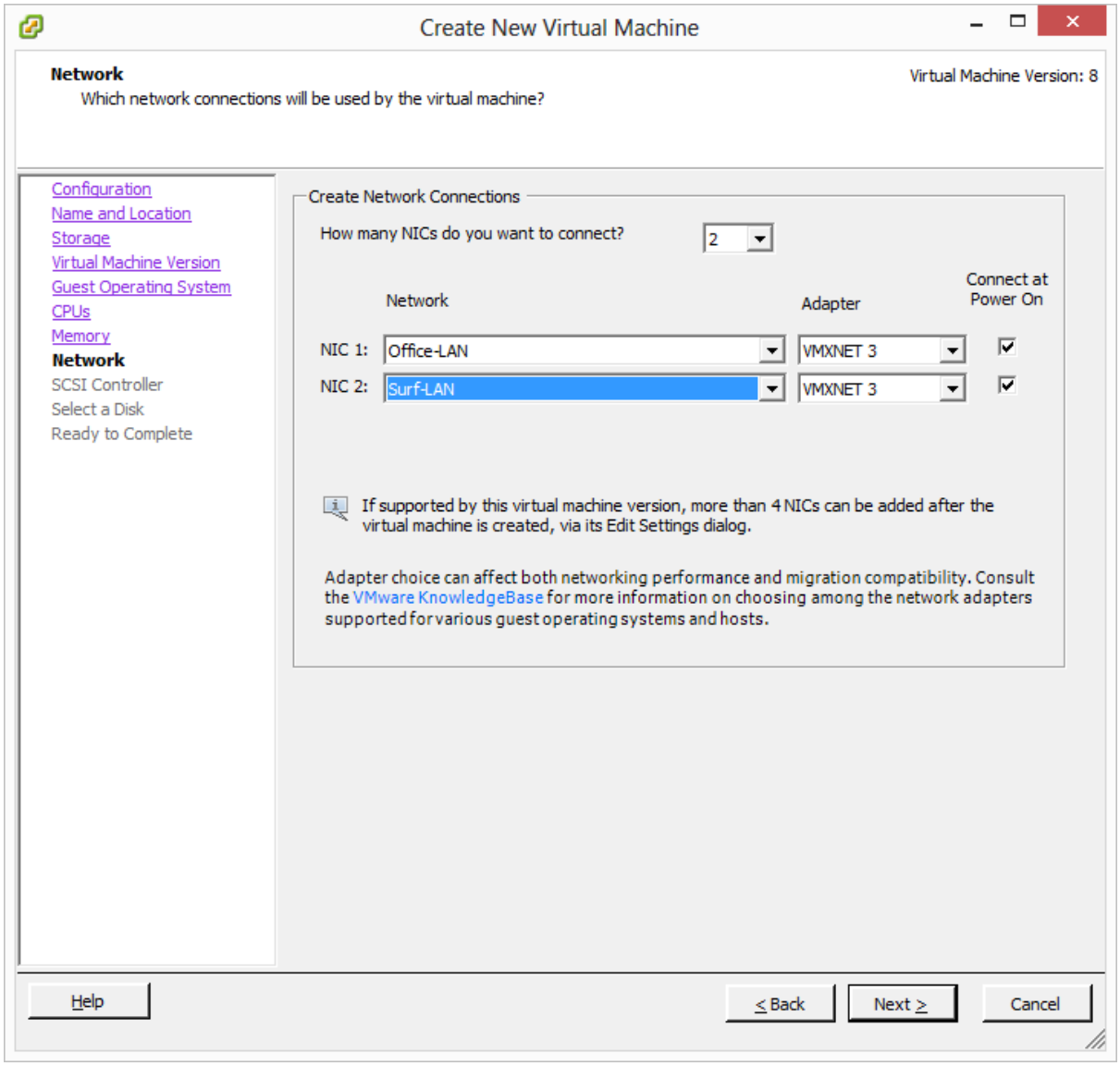
In the next step choose a name for the virtual machine and confirm. Now you need to assign the **Destination Storage**. This setting does not yet allocate any space on the destination storage.



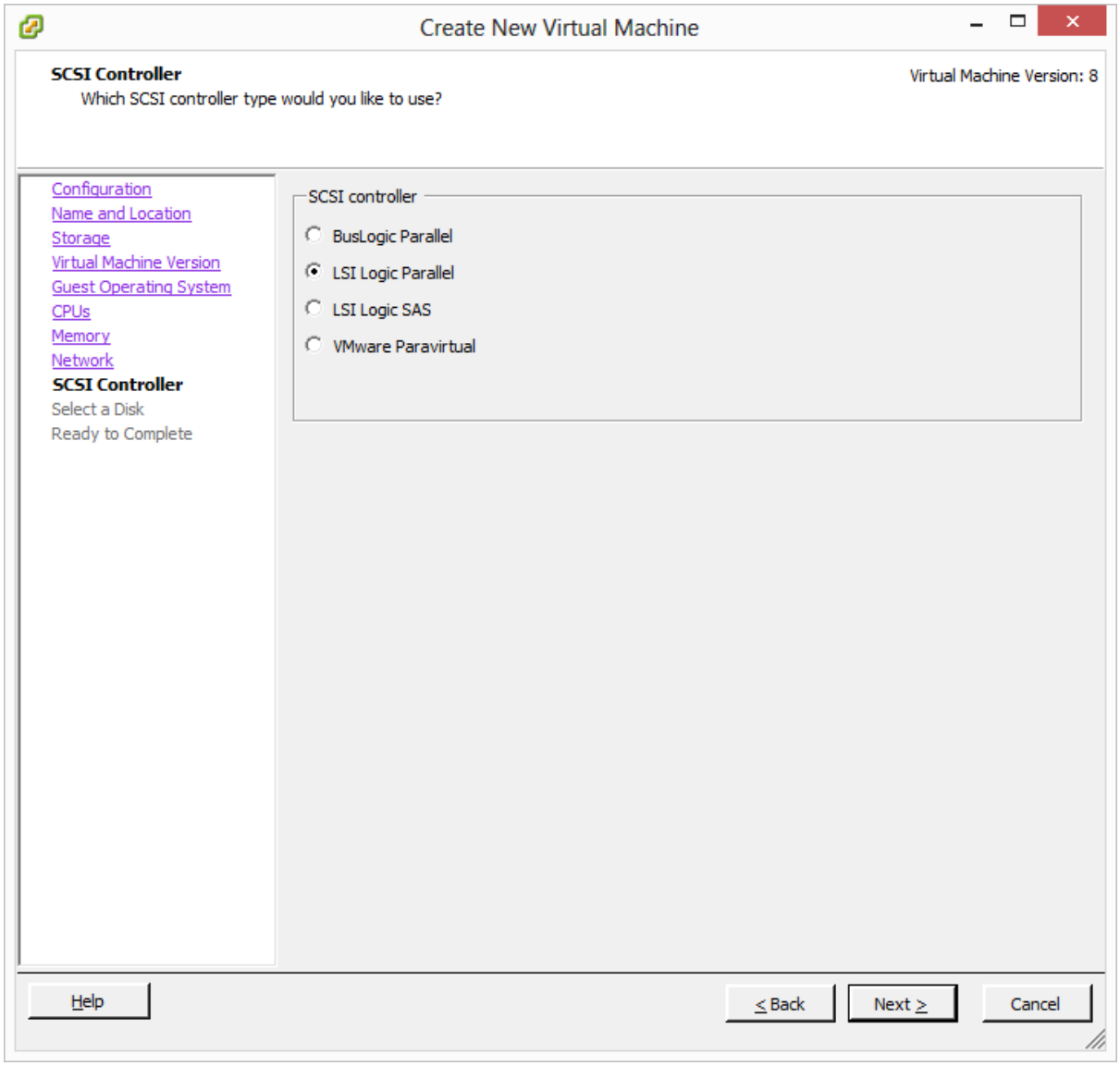
In the next menu, select **version 8 or higher** as **Virtual Machine Version**. In the **Guest Operating System** menu choose **Linux** and then select **SUSE Linux Enterprise 11** either as **(32-bit)** or **(64-bit)**, depending on your IACBOX installation medium.



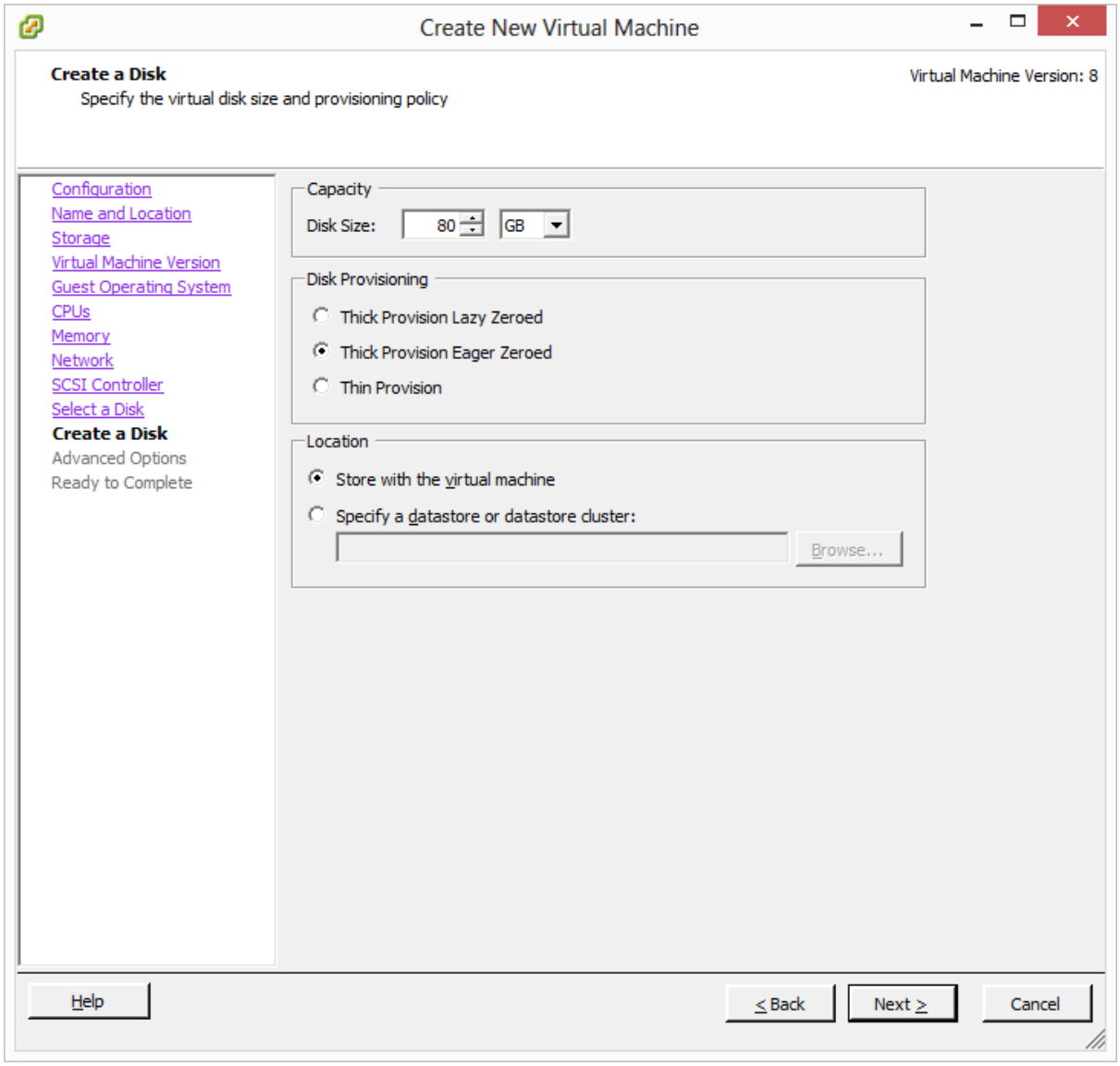
The **CPU** and **Memory** settings must be configured according to the minimum **hardware requirements** which can be found on top of this document - or **better**. For the **Network** settings ensure that you select **VMXNET 3** as **Adapter** type and then enable the **Connect at Power On** for both network interfaces.



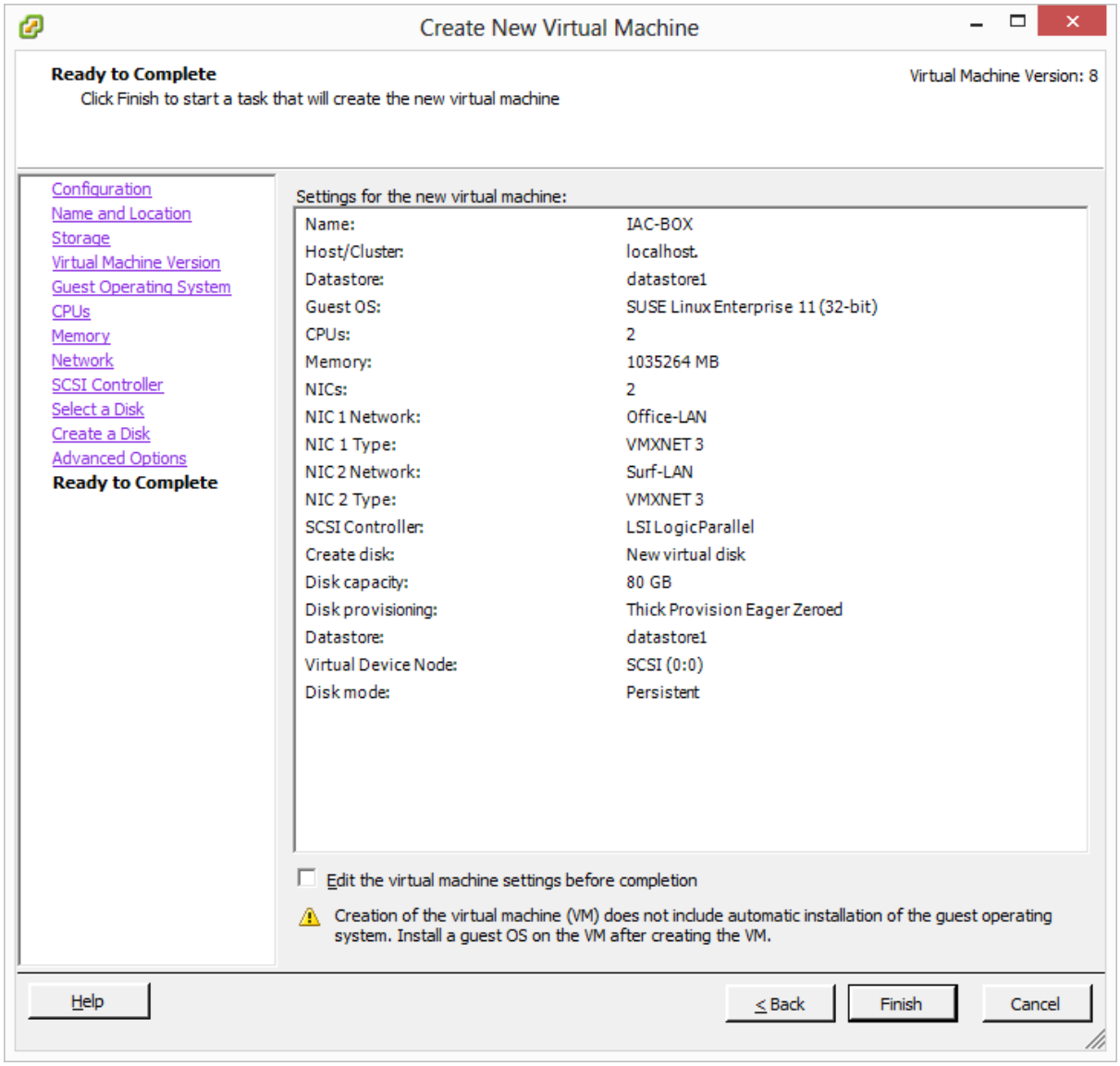
In **SCSI Controller** settings select **VMware Paravirtual**. This will add two aditional configurations to the setup list on the left, so the next step will become **Select a Disk**. Here just select **Create a new virtual disk** and continue with **Next**. If the option for **VMware Paravirtual** is not available in your ESXi, then select **LSI Logic Parallel**.



For the **Create a Disk** configuration again note the minimum hardware requirements. Also enable the option **Thick Provision Eager Zeroed**.

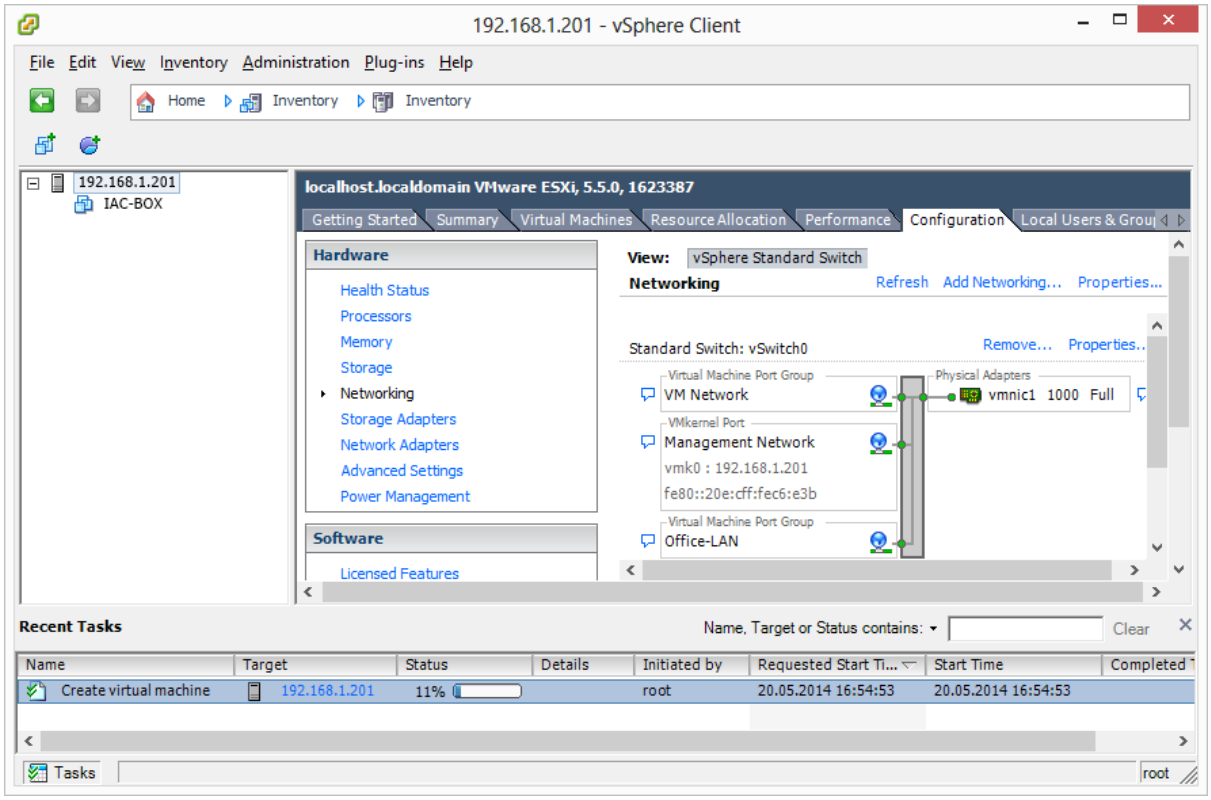


The settings in **Advanced Options** are usually fine by default, continue to **Ready to Complete**. Verify your configuration and then finish the process by clicking on **Finish**.



Hint:

- Note that the creation of the actual virtual machine can take some time.



After the virtual machine creation was done, *right click* on the new virtual machine and select **Properties**. Here you can decide how to include the installation medium. Usually in virtualized environments this is done by **ISO files**, but if the host system has a CD-ROM drive, then a CD can also be used.

Now you can proceed with the installation of the IACBOX. The detailed installation process is described in the manual IACBOX Installation.

3.4 vSphere High Availability

The content of this document is based on test results of *Unify Deutschland GmbH & Co. KG* and demonstrates how to set up an HA environment with the IACBOX and **VMware vSphere**.

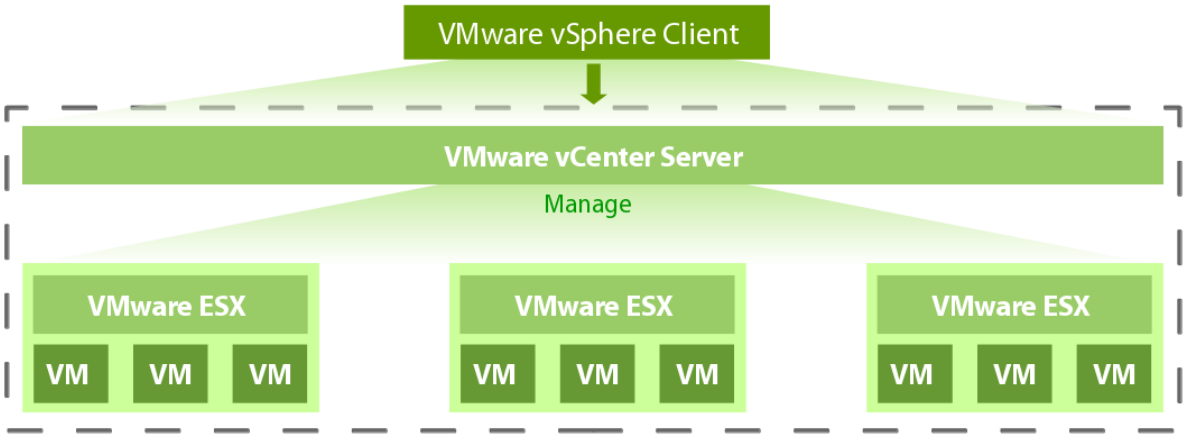
Hint:

- This document illustrates the possibility of setting up the IACBOX as High Available in an virtualized environment. If you require assistance with VMware products, contact according consulting services.

3.4.1 Product Information

VMware vSphere

The Software **vSphere** is the virtualization platform of **VMware** which can be used to virtualize and manage servers. **VMware ESX** serves as operating system to assign virtual machines to a target system. The administration of the virtual machines and the ESX servers is handled by the **vCenter** server, which can be accessed with the **vSphere** client. This management interface, together with shared network- and storage resources, enables a wide variety of possibilities. For example vMotion, which can be used to move and reallocate virtual machines to other servers on the fly.

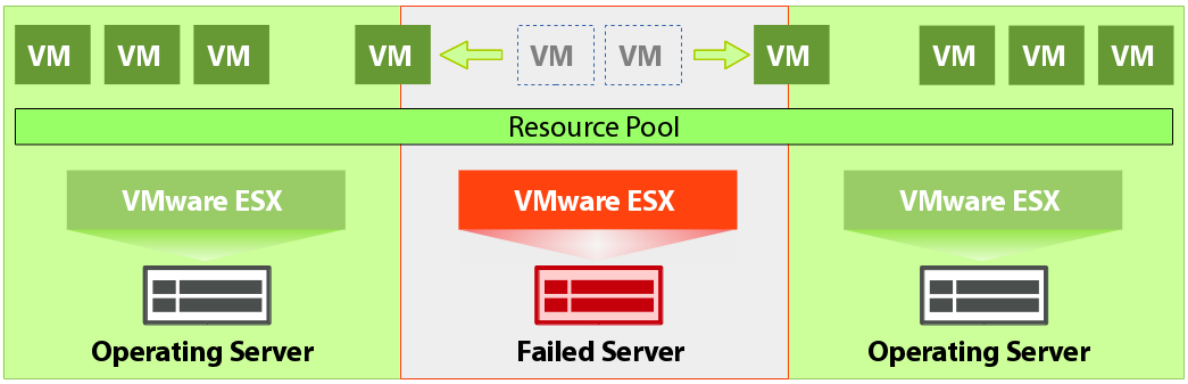


3.4.2 Procedure

With **VMware** solutions there are several possibilities to monitor systems and react in case of system failures. Therefore the differences between **HA** (**High Availability**) and **FT** (**Fault Tolerance**) must be considered.

vSphere High Availability (HA)

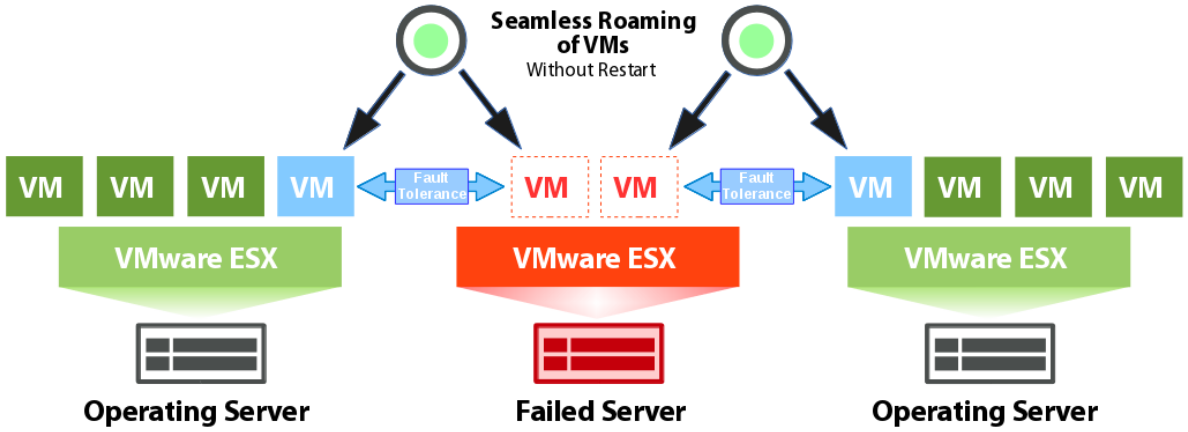
The HA feature of vSphere can react to failures and automatically restart the affected VM in the cluster on another ESX server.



In case of failure of a server the virtual machines will be moved on to another server and restarted. The downtime is limited to the restart of the virtual machine. It should be noted that a virtual machine failure can result in data loss. To use *vSphere High Availability*, a centralized SAN or NFS storage is required, so it can be used on all ESX servers. Further, a management network connection between all ESX servers, as well as an identical network configuration for the virtual machines on all ESX servers in the cluster is required. More informationen about this can be found on the VMware website: http://www.vmware.com/products/vsphere/features/availability.html

vSphere Fault Tolerance (FT)

With *vSphere Fault Tolerance* virtual machines along with their memory content will be replicated to other servers continuously, so that a hardware failure can, in the best case, be compensated without any downtime.



For each running virtual machine, an exact copy is being replicated while both, the original and the copy share the same memory and data. In case of failure, the replicated system becomes active and enables a seamless transition without the loss of data or downtime.

CHAPTER

FOUR

NETWORK

4.1 802.1X - IEEE-802 authentication

This howto describes the configuration and use of the **IEEE 802.1X** authentication for Surf-LAN clients in combination with an IACBOX.

> **Attention:** Do not confuse this with the 802.1x settings found on the network settings page - this is used when the IACBOX has to authenticate itself at a switch.

Hint:

- The network devices used (WiFi access point, router, switches) must support IEEE 802.1X
- Client devices also need to have this type of authentication implemented (for WiFi this is often called *WPA Enterprise*)
- The IACBOX currently supports the EAP-TTLS and PEAP variants.

4.1.1 How does 802.1X work?

IEEE 802.1x is a **network security procedure** which forces client devices to **authenticate** themselves before they get access to a local network. The protocol used is **EAP (Extensible Authentication Protocol)** and is the core of IEEE 802.1x and allows the exchange of authentication messages on layer 2.

Components of an IEEE 802.1x network are the **supplicant devices**, **authenticator devices** and the **authentication server**.

The authentication server validates the request of the supplicant devices and notifies its decision to the authenticator. Based on this, the authenticator grants or denies access to the local network for the supplicant device.

- Supplicant Device - WiFi Clients, LAN-Stations
- Authenticator Devices - WiFi-Access-Points, Router, Switches
- Authentication Server - Radius-Server, LDAP-Gateway/Server

The communication between **supplicant and authenticator** is done with **EAP** and for the communication between **authenticator and authentication server**, **EAP packets** are **encapsulated** in radius packets. Since the original EAP is not very safe, there are *advanced EAP variants* that provide additional security. For example with EAP-TTLS and PEAP (protected EAP), an own tunnel from each supplicant device to the local network is established.

4.1.2 IEEE 802.1X and IACBOX

If IEEE 802.1x is used in combination with the **IACBOX**, the IACBOX is used as **authentication server**. Thereby the IACBOX works as radius-server for the authentication clients.

Configuration on the IACBOX

You can activate the IEEE 802.1x authentication in the WebAdmin menu **Security/General** of the IACBOX.

First of all, you need to select the networks (Office-LAN, Surf-LAN, Management-LAN) where authenticator clients are accepted from.

Next, all authenticator clients which should be connected to IACBOX need to be defined in the tab **IEEE 802.1x Authenticator Clients**. Please note, that the **secret** needs to coincide with the configured secret on the authenticator client. Otherwise there is no communication possible between both devices. The mandatory IP-address range means that only supplicant devices (client devices) with an IP-address within this range are allowed to authenticate.

In addition, you can add **allowed supplicant devices** and **general 802.1x users** manually. You can add individual supplicant devices with their MAC- addresses. This means the userdata defined can only be used by the supplicant device with the associated MAC-address. If the **MAC-address field** is left **blank** (general 802.1x user), all supplicant devices can authenticate with the defined user data.

All other supplicant devices (client devices) which are not defined manually, can **authenticate via** 802.1X by using **WebAdmin Tickets** of the IACBOX. The **authentication** with **Ext-Auth users** (MySQL, PostgreSQL, MSSQL) and **local users** is also possible, but associated with limitations.

Example Configuration 1 - Two-stage logon

In the WebAdmin of the IACBOX, manually define a general 802.1X user for all supplicant devices (leave the MAC-address field blank). Thus, supplicant devices have access to the network only after authenticating with the manually defined 802.1X user. This means, that after 802.1x authentication, devices get to the IACBOX logon page and need to authenticate again in order to get online.

Example Configuration 2 - Instant logon

Instant logon means that the supplicant device (client device) does not need to authenticate twice, once for the 802.1x procteded network and once at the IACBOX, in order to go online. Instead, with entering the user data (username & password) at the 802.1x authentication, the device will also be taken online immediately by the IACBOX. This can be used with **multiple logon methods** of the IACBOX:

- **WebAdmin Tickets:** Use the ticket username and password for the 802.1x authentication.
- **WebAdmin Local Users:** Use the defined username and password of the local user for the 802.1x authentication.
- **WebAdmin External Authentication:** Any supported external authentication method that stores the **password in clear-text (necessary)** can be used for the 802.1x authentication. This includes SQL databases and Radius servers that are used for external authentication by the IACBOX.

4.2 Custom TLS/SSL Certificate

Hint:

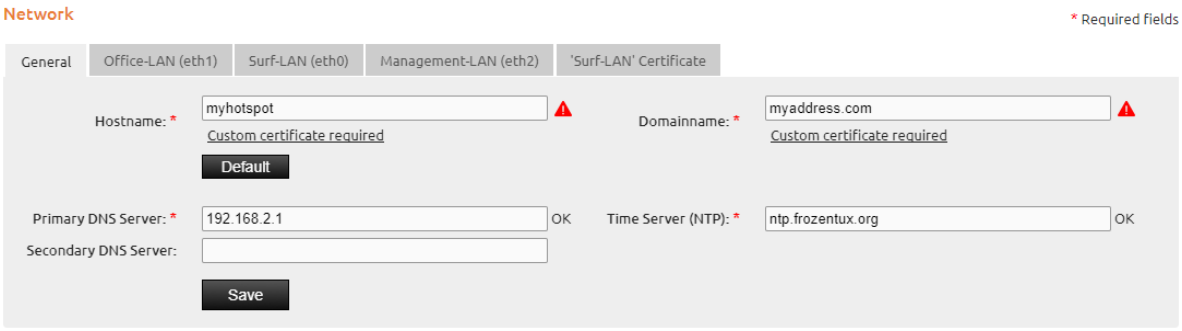
- **TLS** (the successor of **SSL**) is the only secure protocol that is used, but in combination with certificates the term **SSL** is still used very often.
- **Basic knowledge about TLS-certificates is required**, this document expects a certain level of familiarity with TLS and X.509.
- The IACBOX does only support **PEM** certificates, **DER** certificates have to be converted.
- The key file must not be password protected.
- Intermediate certificates have to be appended to the ca-file.

- System Administrators are in charge to backup the key-files and store them securely.

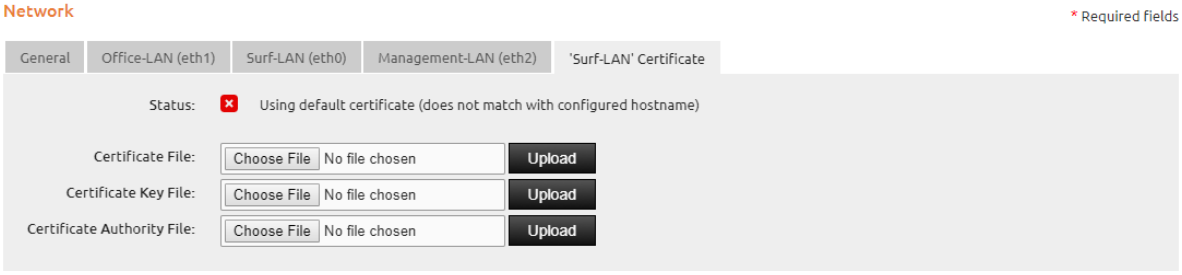
4.2.1 Using a custom certificate

You can use any **PEM** type certificate on the IACBOX. To enable the option to upload your certificate navigate to **Settings / Network / General** and change the **hostname** and **domainname** according to your certificate.

Now click on **Save**. After the settings are saved, the IACBOX will now recognize that you require a custom certificate.



Navigate to the tab **Surf-LAN Certificate**. Now you can upload your certificate files.



Attention: After you uploaded your files, you have to navigate back to General and click on **Save**.

4.2.2 CSR Generator

With the CSR Generator you can generate your own **certificate signing request** on the IACBOX. Make sure that you save all the provided data. After generating the CSR request it has to be signed by a **CA** (certificate authority).

You will then receive your new certificate which you can upload as shown above.

4.3 Fixed Bandwidth

This manual explains the use and configuration of the fixed bandwidth. This feature is especially useful for a guaranted bandwidth/access speed for guests of the IACBOX.

Hint:

- **The fixed bandwidth is meant to be used with exclusive VIP tickets only and should** never be used on multiple tickets at the same time.
- After enabling/disabling this feature the IACBOX needs to be restarted.

4.3.1 Usage

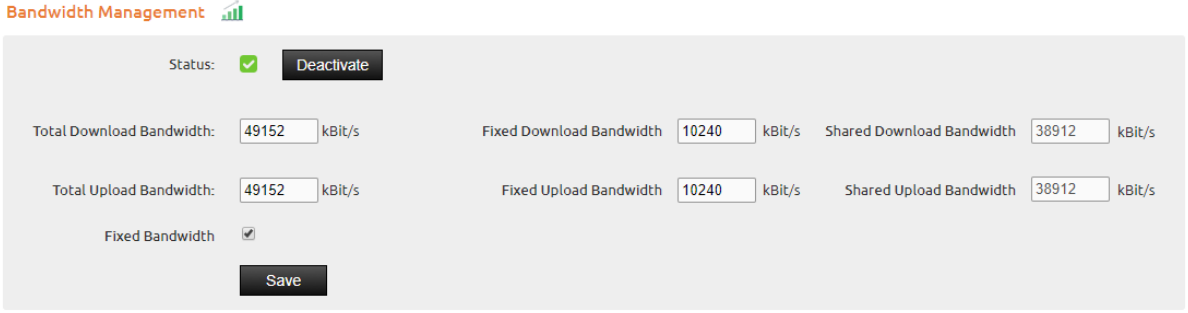
The **fixed bandwidth** should only be used for a single VIP ticket. Although the configured fixed bandwidth will not be reserved permanently, it can cause huge disturbances within the network stability when used incorrectly.

Tickets with fixed bandwidth **should only be created in the WebAdmin** of the IACBOX.

This feature is using **bandwidth shaping**, which means that the required bandwidth will be allocated as soon as the ticket with fixed bandwidth is online and does require it.

4.3.2 Enabling the fixed bandwidth

In the WebAdmin menu of the IACBOX navigate to **Settings/Network** and activate the fixed bandwidth within the **Bandwidth Management**.



Here you can define the amount of the fixed bandwidth. Please note that you must have enough bandwidth available for regular tickets within the **Shared Download/Upload Bandwidth**.

Attention: Afterwards restart the IACBOX!

4.3.3 Configure a ticket template

In the WebAdmin of the IACBOX navigate to **Tickets/Templates** and create a ticket or edit an existing one.

Then enable the Fixed Bandwidth and configure as desired.

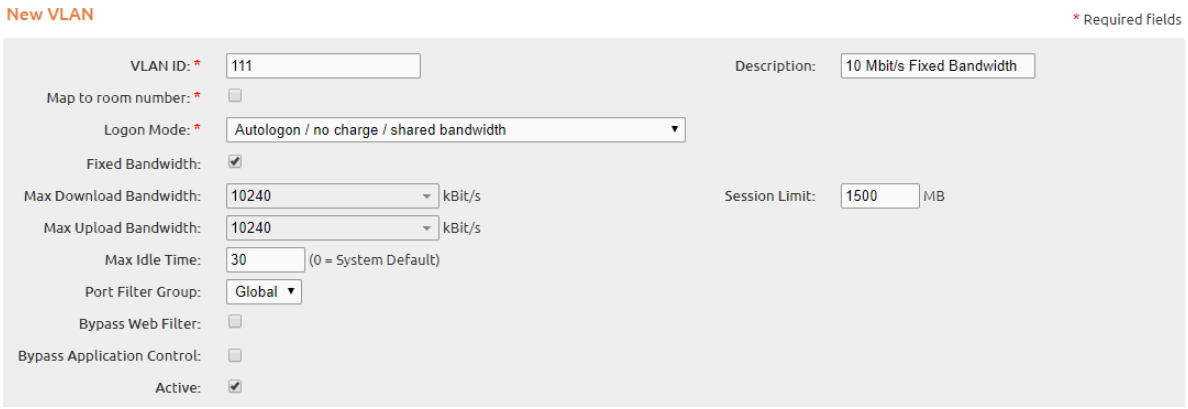
Then save the ticket template.

Please note that the automated creation of tickets with fixed bandwidth (for example via PMS) can result in system instability! The fixed bandwidth is not meant to be used with multiple tickets.

4.3.4 Activate for VLANs

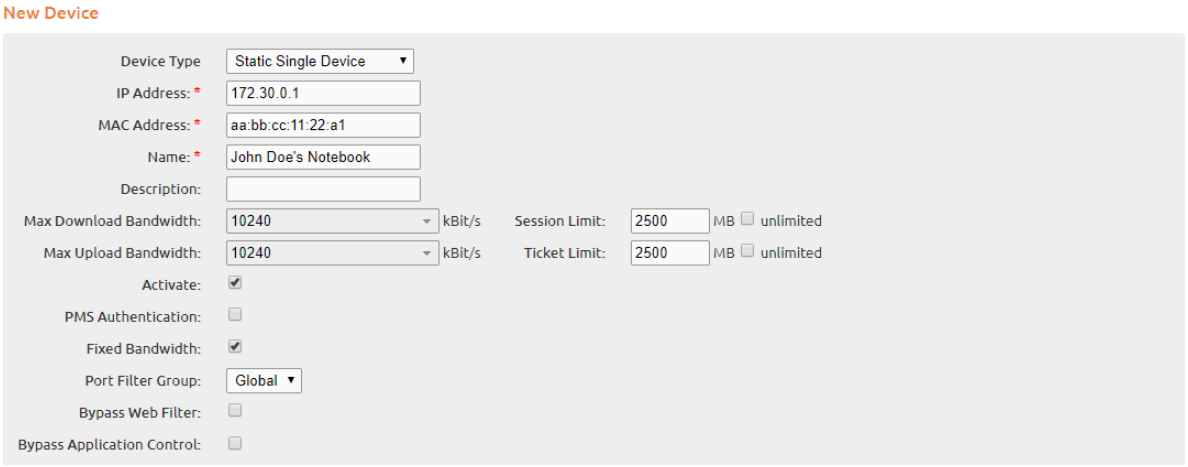
If you use as **shared bandwidth**, then you can also assign a fixed bandwidth to VLANs (page 99). This means that the whole VLAN will be using the same configured fixed bandwidth (shared).

Modification of these settings is made under **Security/VLANs**.



4.3.5 Activate for Autologon Devices

The fixed bandwidth can also be used for static autologon devices (page 131). Please keep in mind that multiple fixed bandwidth devices can cause instability in the whole network.

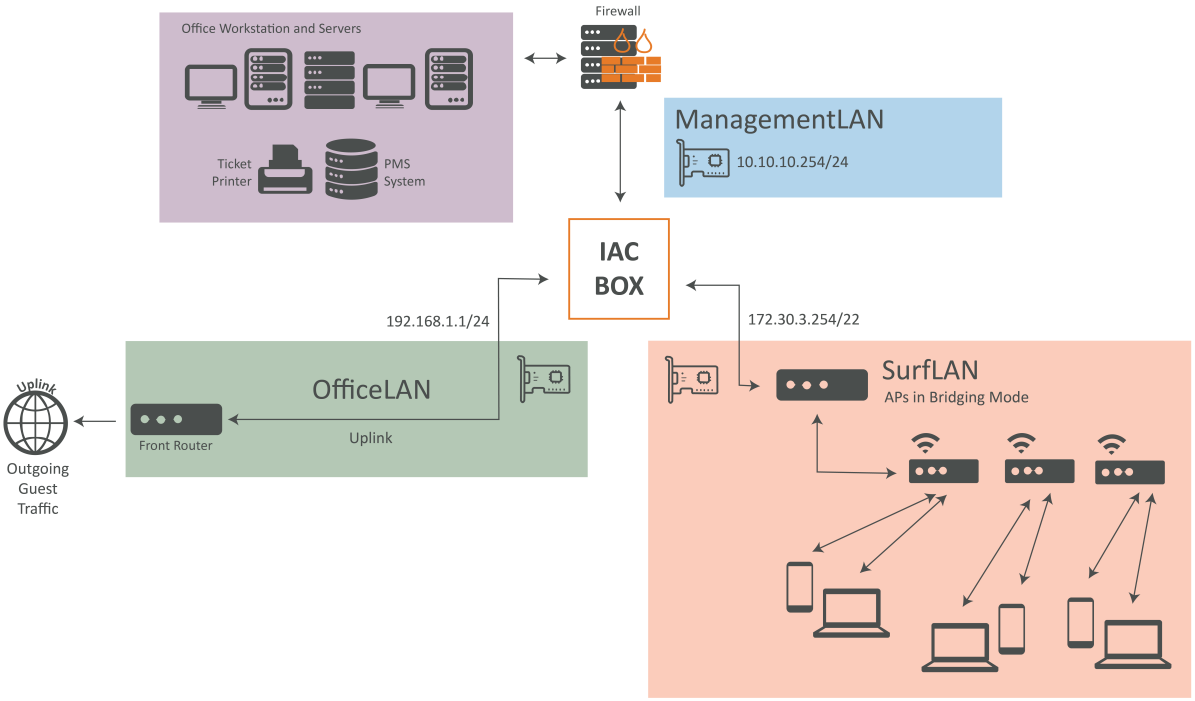


4.4 Activation of Management LAN

This manual describes how to activate the Management LAN.

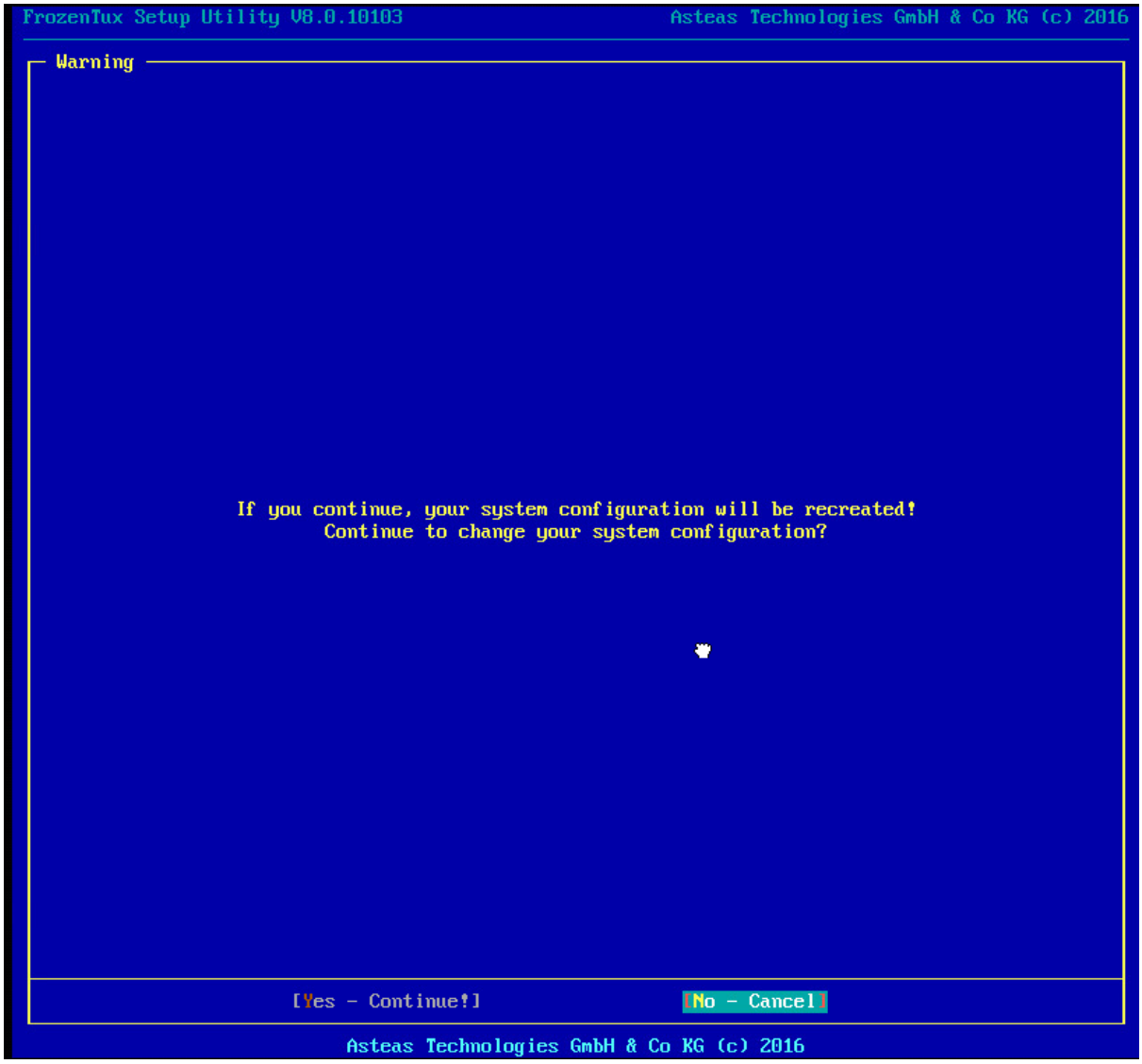
4.4.1 General

Sometimes network environments do not permit to add new devices into an existing and complex infrastructure. For exactly this problem the IACBOX can make use of an **optional** *third management interface*, the **Management-LAN**. So if the current network infrastructure does not allow you to add your ticket printers or PMS systems you can move them into the Management-LAN network. Note that the Management-LAN network does require a **third physical network card** in the IACBOX.

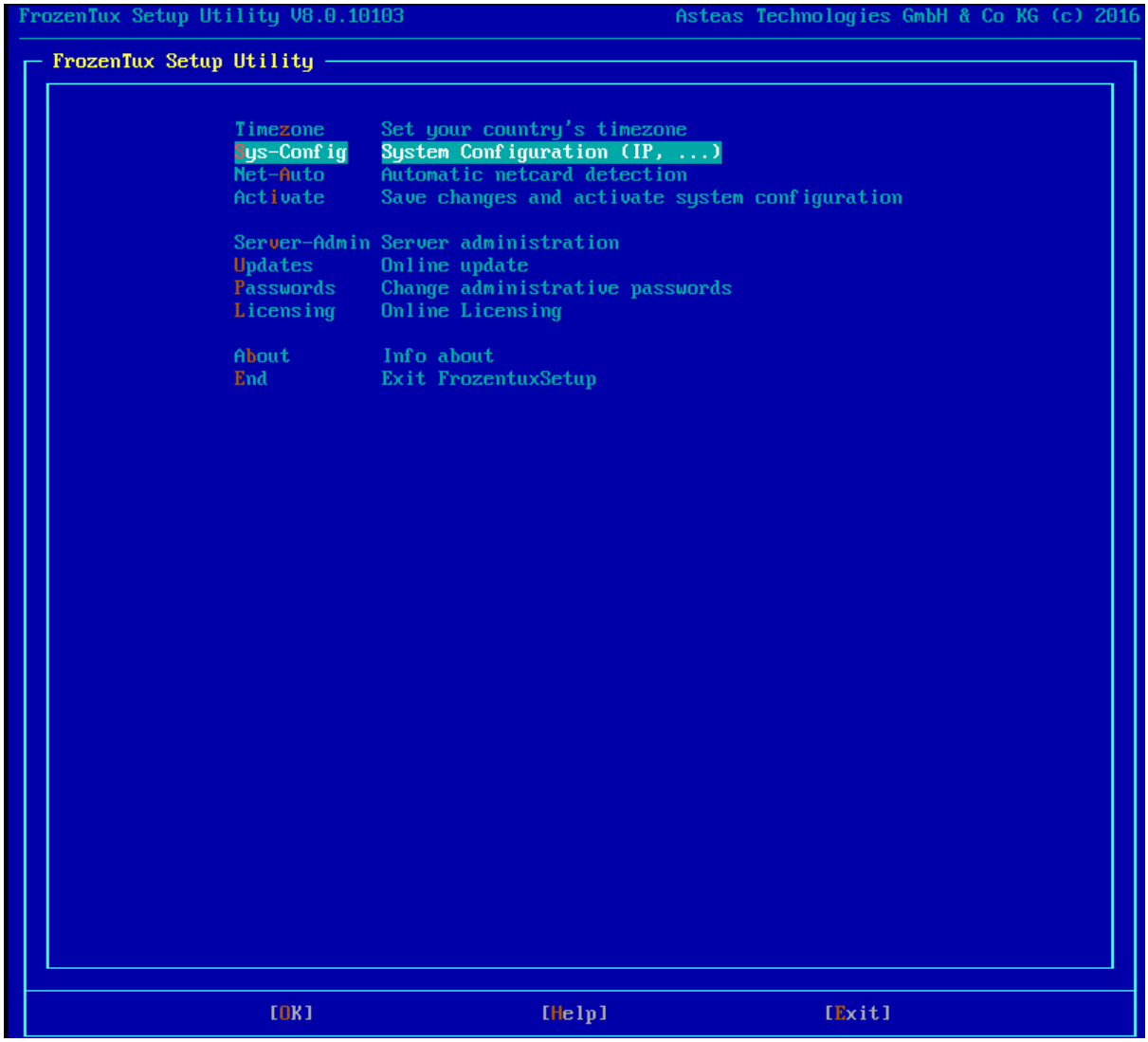


The Management-LAN can be activated **while installing** the IACBOX and also **later on**. To activate the Management-LAN while installing the IACBOX, you can also use the available **Unattended Setup** options, which by default does offer the *automated installation* with 3 network interface cards in *german* and *english*. Further information about the **Unattended Setup** can be found in the corresponding manual page (page 36).

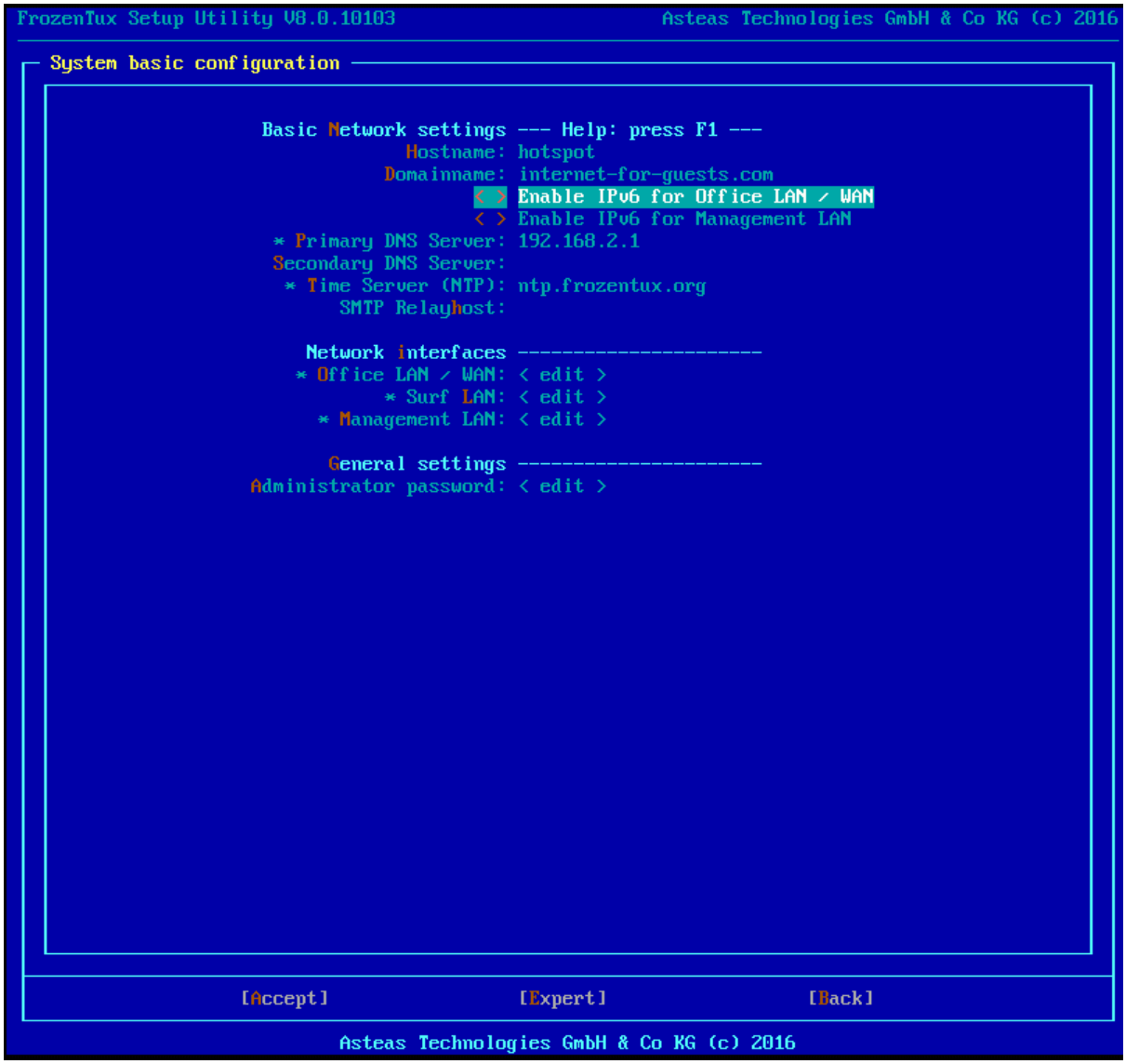
You can also activate the Management-LAN after the IACBOX was already installed. For this variant log on to the console with your “sysop” user. The following setup dialog will show up.



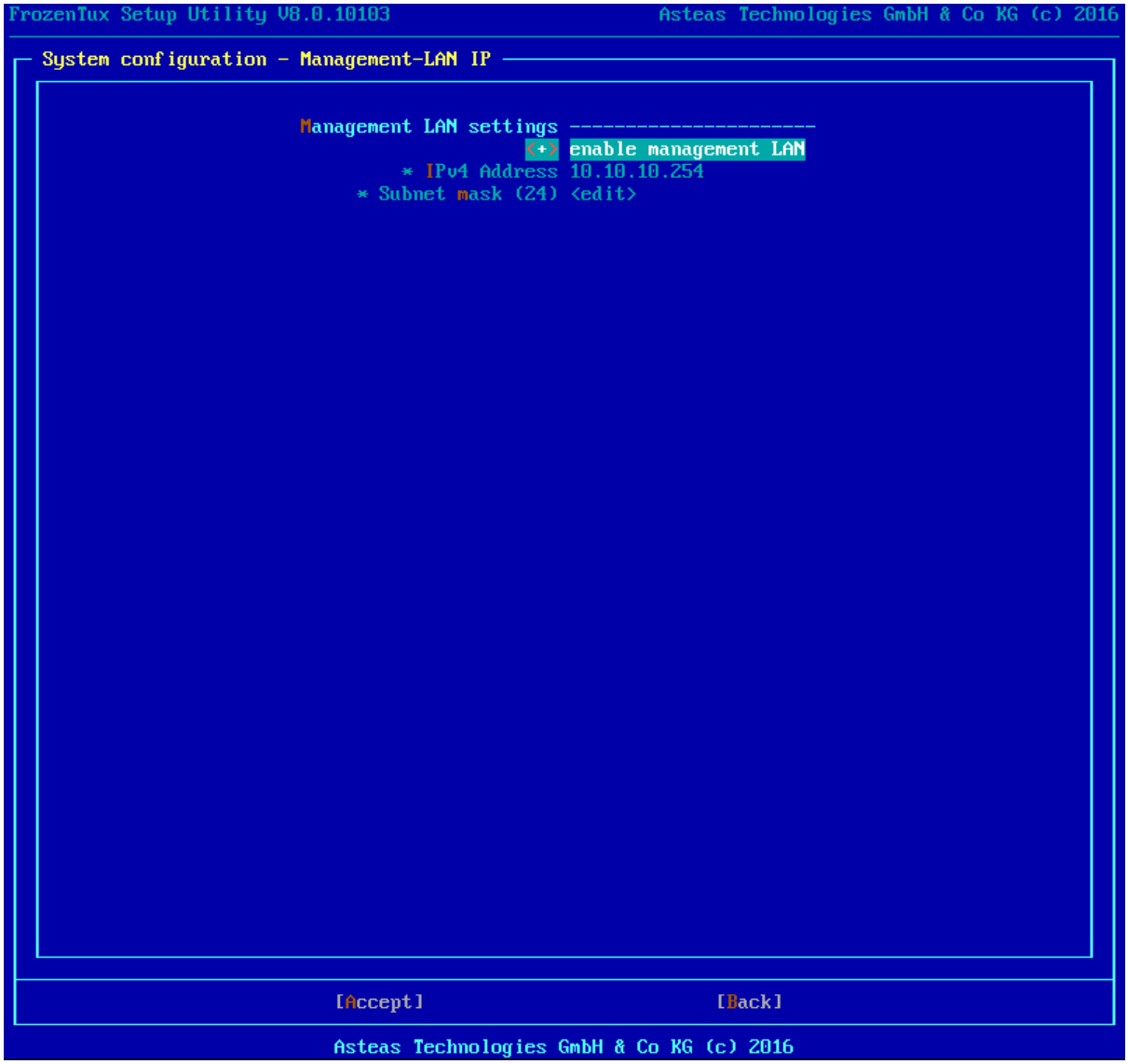
Confirm the dialog with **Yes - Continue!** to enter the setup menu. First switch to the menu **Sys-Config** to configure the network settings.



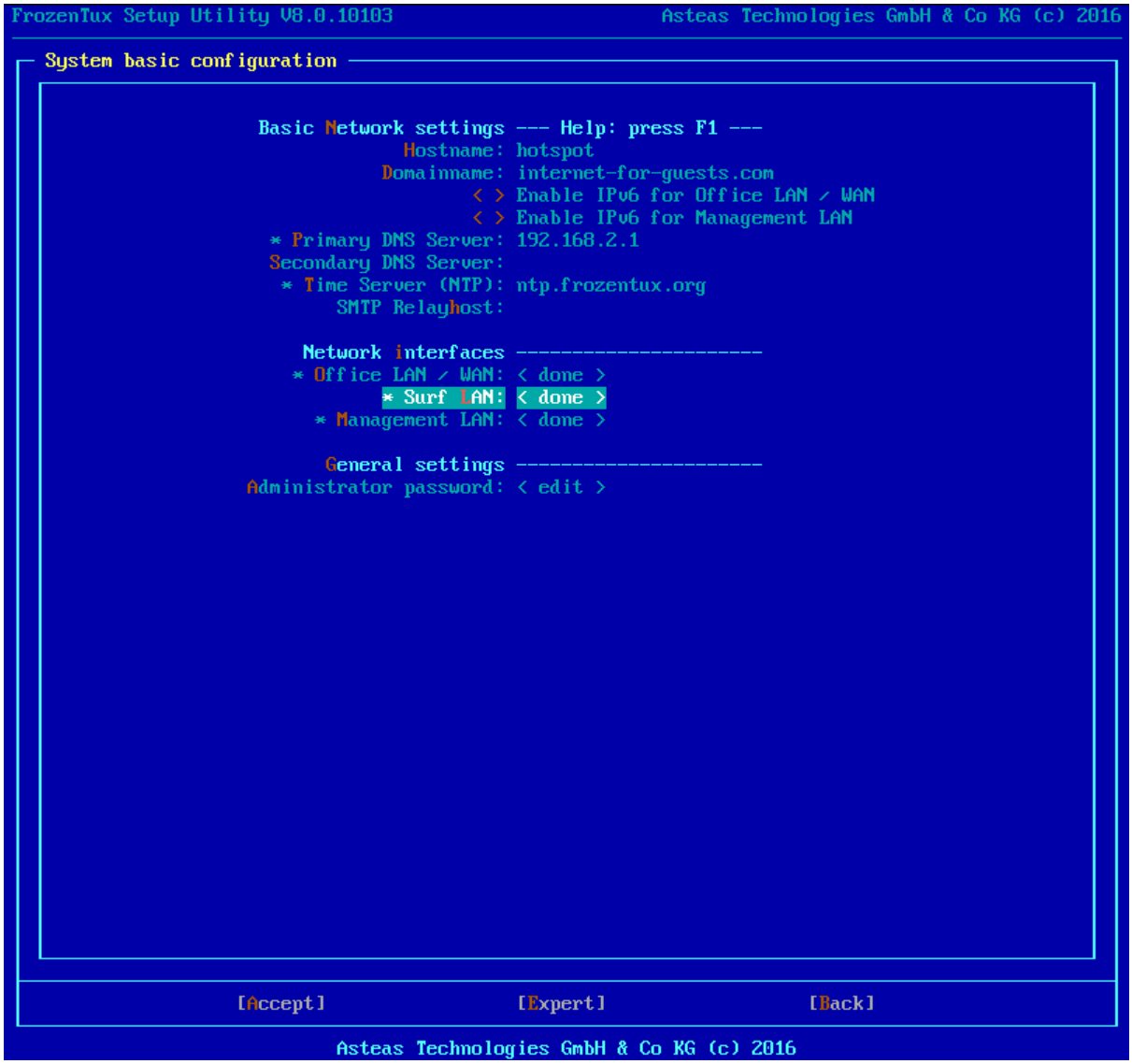
You have to edit all network interfaces in order to activate the third network card at the next step.



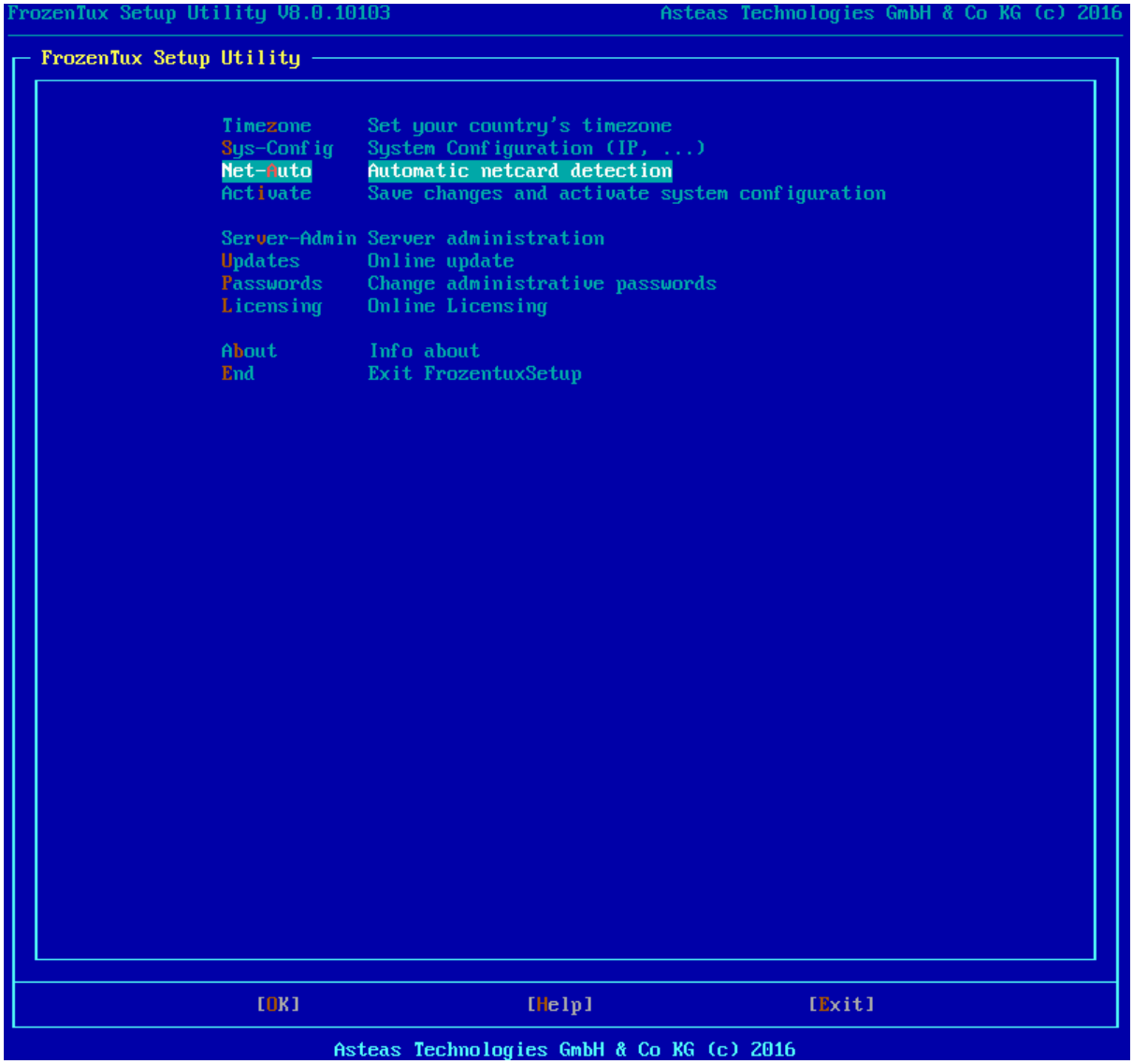
Now the Management-LAN can be configured and activated.



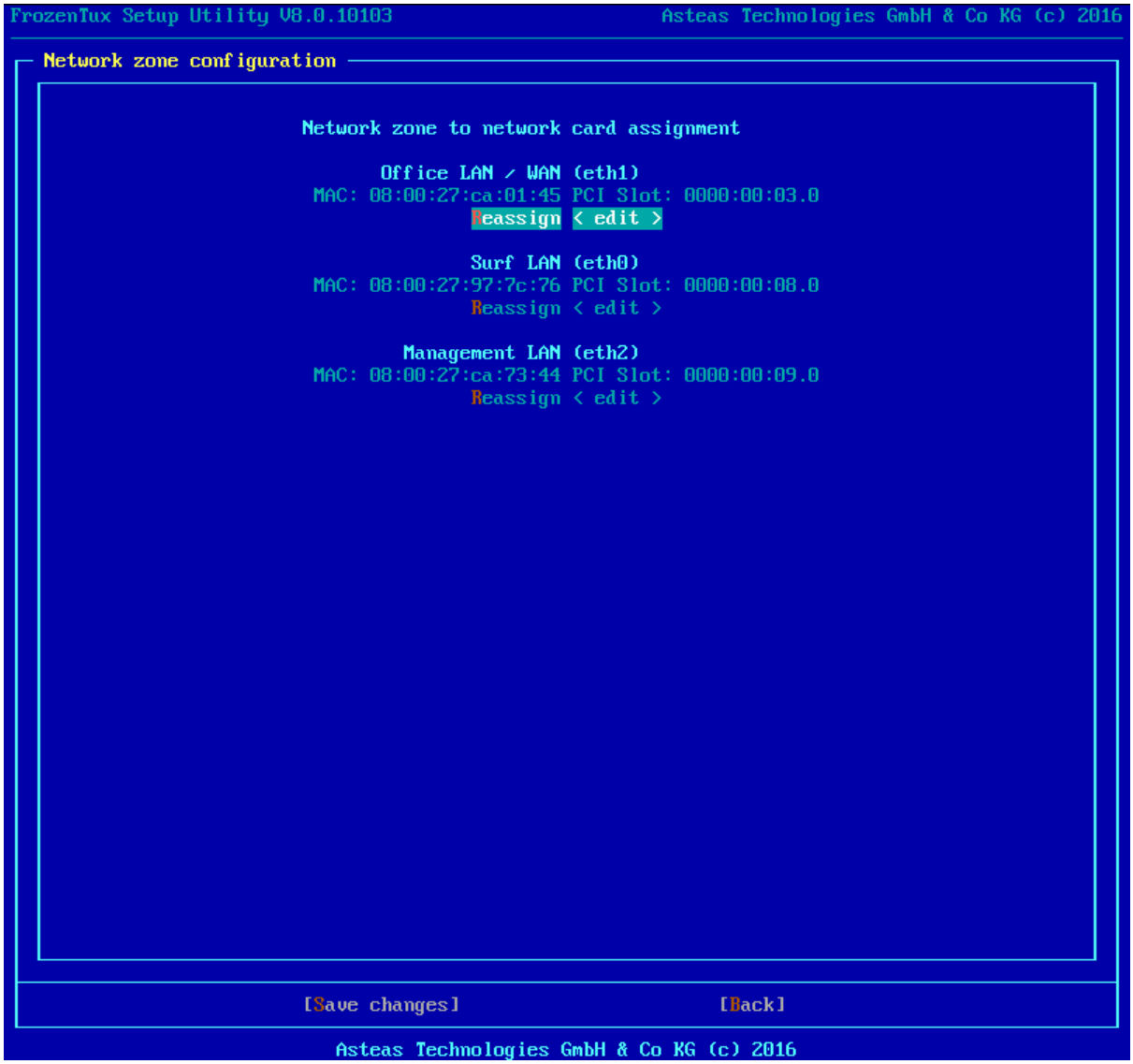
If those steps were carried out as shown in the pictures above, the status of all three network interfaces should have the status *done*



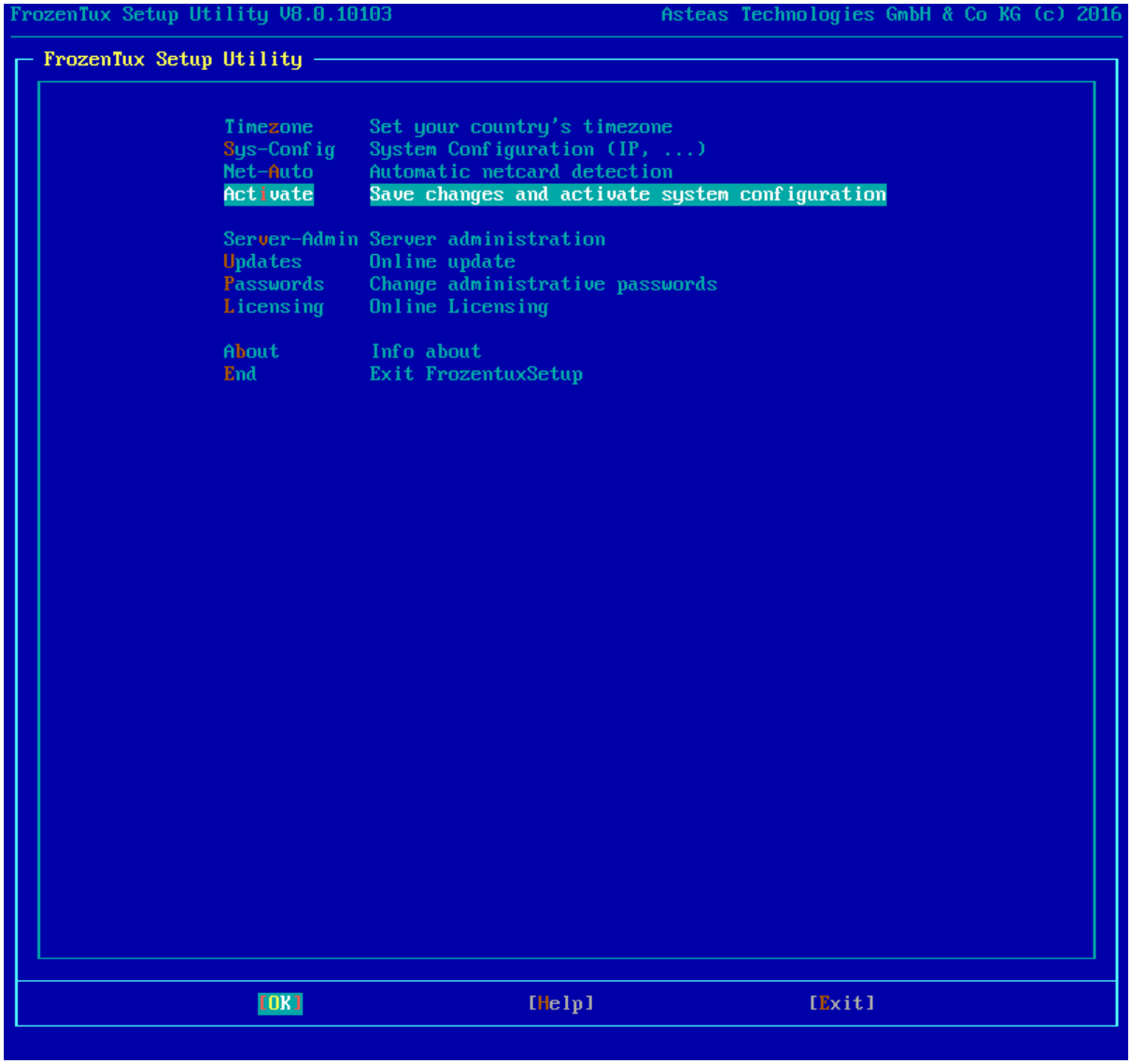
In order to activate the settings, switch back to the main menu and select the menu **Net-Auto**.



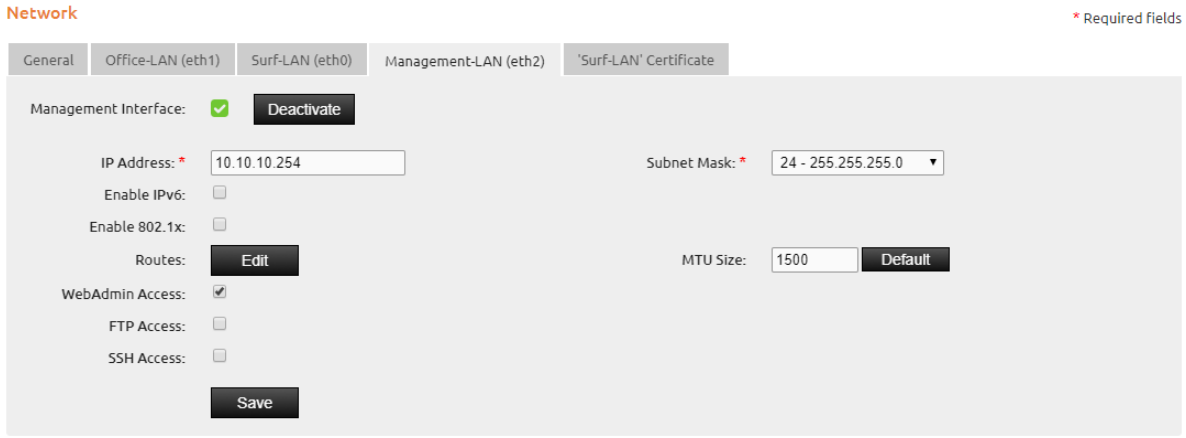
This listing shows the assignment of the network interfaces to the 3 different network zones (Office LAN, Surf LAN, Management LAN). Use *Save changes* to commit any made changes or use *Back* to return to the Main Menu.



Now activate the changes made, exit the setup menu and reboot the IACBOX.



After system reboot the Management-LAN can be configured now.



For the Management-LAN network you can also activate WebAdmin Access, FTP access and SSH access to the IACBOX. The IACBOX must be rebooted in order to make the changes take effect. Please note that the network interfaces of the IACBOX can change when a third interface is added. For example, it may occur that the Office-LAN (eth1) interface transforms into the Surf-LAN (eth0) interface after activating the third interface. Thats why it's strongly recommended to check the network interfaces for changes after the third interface has been activated.

4.5 Using Routes

Attention:

- The range **172.17.0.0 - 172.17.127.255** is reserved for internal use and can not be used in any configuration.

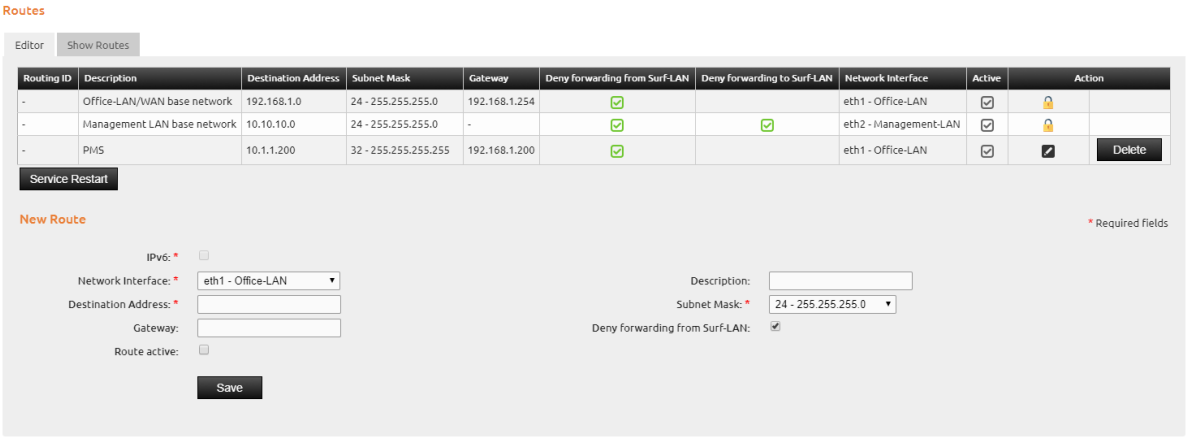
4.5.1 General

If you want to deploy in an existing network environment which does not allow adding devices in existing IP range, chances are that you must set up a custom network in the **Office-LAN** management range for this case. Since the IACBOX naturally will only see devices within the configured **Office-LAN** network, devices from other local networks can be connected via **Routes**. This manual briefly describes the configuration.

By **Default** there are 2 basic *Routes* for the **Office-LAN** and the **Surf-LAN** on the IACBOX. **Surf-LAN Routing** is only possible by using the so called **Routing Mode** of the IACBOX. Before downgrading to the *Routing Mode* please verify that this is necessary for your requirements, the according manual can be found by following this link: Routing Mode.

4.5.2 Example of Routes

Routes can be added in the WebAdmin menu **Security / Routes**. In order to specify a route, you must know the destination network and the gateway for it. The following screenshot shows an example of a network system (**PMS** reachable via **gateway 192.168.1.210**) which can not be reached by the IACBOX by default, because it is outside the **192.168.1.0/24** *Office-LAN* network and outside the reachable scope of the default route.



In order to make the PMS-System accessible, you can add a *Route* on the IACBOX with following parameters:

- Network Interface **eth1 - Office-LAN**
- Destination Address: 192.168.100.101 (**the PMS System**)
- Gateway: 192.168.1.2 (**the Router/Firewall**)
- Subnet Mask: 255.255.255.255 (**/32 as host route**)
- Deny forwarding from Surf-LAN **enabled**
- Route active **enabled**

If you want to access **all devices** in the target network, then simply change the **Destination Address** to the target network and adjust the **Subnet Mask** according to the accessible range.

Attention:
- After adding or editing an entry, a **Service Restart** is **required**.
- The *Route* in the example above must also be added on the **PMS-System**, so that the IACBOX can receive an answer from the target system.

4.6 Routing Mode

This manual describes how to activate and configure the **Routing Mode** on the IACBOX.

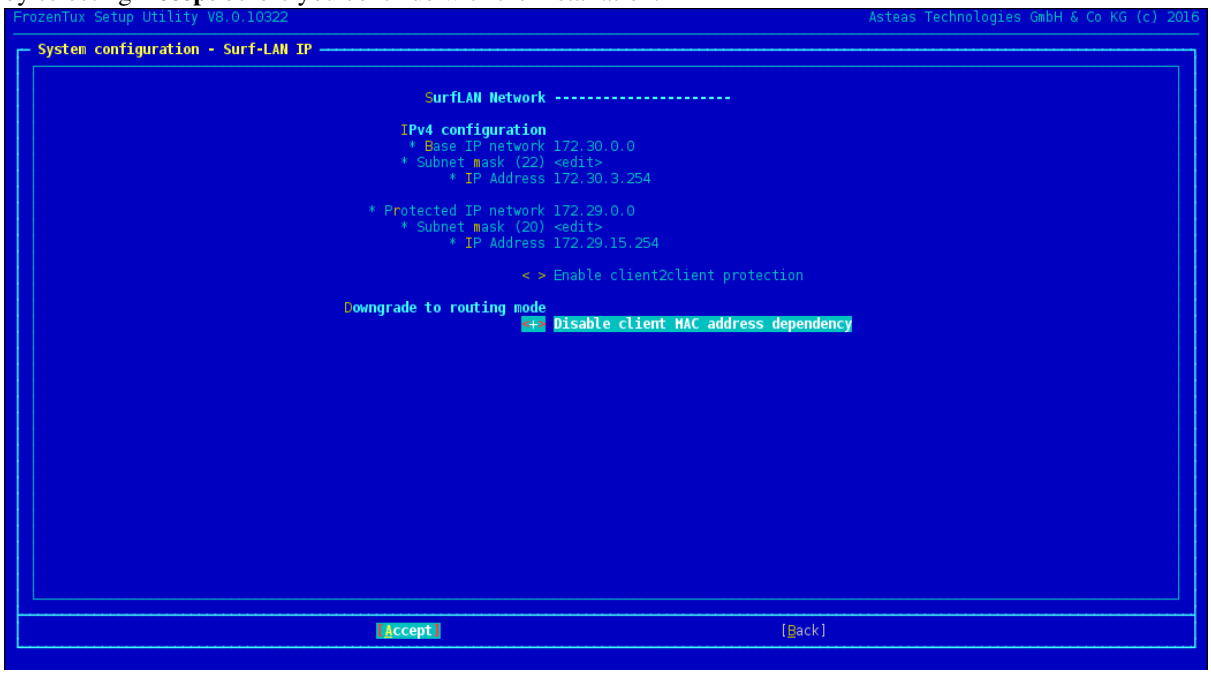
Hint:

- Clients on the IACBOX can only be identified by their IP addresses.
- The **Plug & Play** will only work in the local broadcast domain of the IACBOX, not via routes.
- DHCP must be done by the management network devices in the *Surf-LAN*, e.g. routers or access points.
- This enables you to create different DHCP ranges in the *Surf-LAN*, but therefore routes must be added on the IACBOX and on the management network devices in the *Surf-LAN*.

4.6.1 General

While the *Routing Mode* is mostly used in centralized enterprise environments, customer requirements for different networks with DHCP ranges can also be implemented with it.

The **Routing Mode** can be activated during the installation of the IACBOX. This can be done while configuring the **Surf-LAN** network settings. Therefore select the option **Downgrade to routing mode** and confirm this change by selecting **Accept** before you continue with the installation.



After successful installation, the routes for the *Surf-LAN* must be configured manually. Switch to the *WebAdmin* menu **Security/Routes** and activate the **Advanced Routing**. Now routes for the *Surf-LAN* can be configured - an example is shown below.

Routes

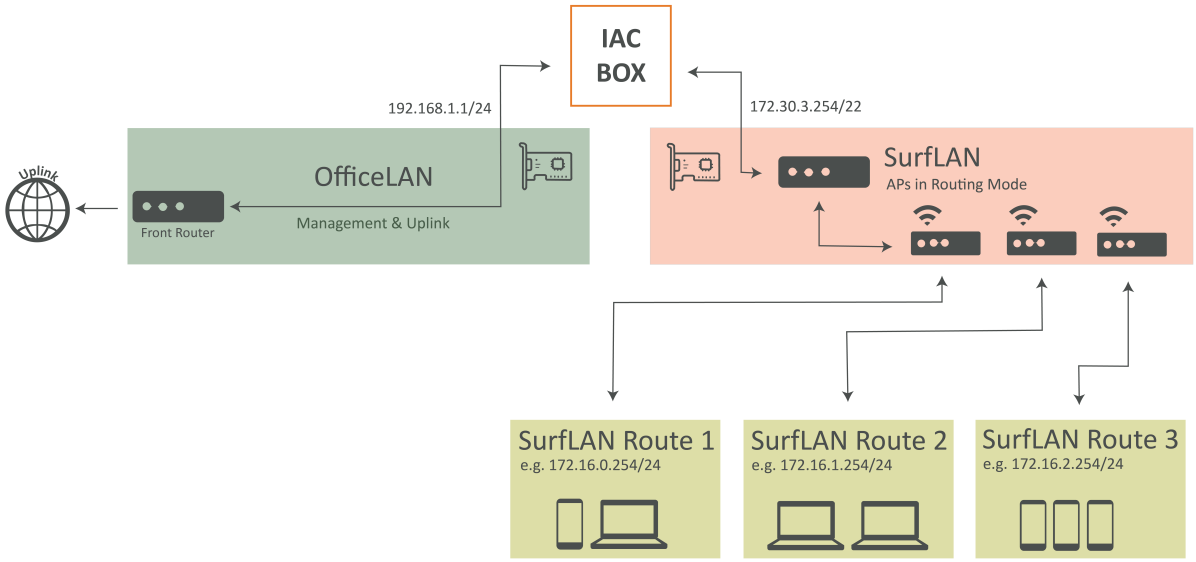
Editor | Show Routes

Routing ID	Description	Destination Address	Subnet Mask	Gateway	Deny forwarding from Surf-LAN	Deny forwarding to Surf-LAN	Network Interface	Active	Action	
-	Office-LAN/WAN base network	192.168.1.0	24 - 255.255.255.0	192.168.1.254	☑		eth1 - Office-LAN	☑	🔒	
-	Management LAN base	10.10.10.0	24 - 255.255.255.0	-	☑	☑	eth2 - Management-LAN	☑	🔒	
1	Surf-LAN base network	172.30.0.0	22 - 255.255.252.0	172.30.3.254			eth0 - Surf-LAN	☑	✎	
2	Surf-LAN Route 1	172.16.0.0	24 - 255.255.255.0	172.30.3.240			eth0 - Surf-LAN	☑	✎	Delete
3	Surf-LAN Route 2	172.16.1.0	24 - 255.255.255.0	172.30.3.241			eth0 - Surf-LAN	☑	✎	Delete
4	Surf-LAN Route 3	172.16.2.0	24 - 255.255.255.0	172.30.3.242			eth0 - Surf-LAN	☑	✎	Delete

Service Restart

Here you can see three *Surf-LAN* routes that have been created. For each route certain settings like **Logon Mode**, **Free Logon**, **Web Filter** Settings, etc. can be performed. In addition, individual ticket templates can be enabled or disabled for certain routes by editing the corresponding ticket templates in the *WebAdmin* menu **Tickets/Templates**.

The following example shows a network infrastructure with 2 different *Surf-LAN* routes. Each route must be configured on the IACBOX **and on the according access point**, a *Route* back to the IACBOX. Each access point routes to an own network (172.16.0.0/24, 172.16.1.0/24). Since the DHCP of the IACBOX can not reach clients befind the access points, the DHCP handshake must be performed by them.



4.7 Configuration of VLANs

With VLANs it is possible to divide the Surf-LAN network into different scopes, while each scope can be configured individually.

Hint:

- **VLANs** can only be configured on the **Surf-LAN** side of the IACBOX.
- If VLANs are active, the incoming traffic from the Surf-LAN has to be **tagged**.
- All network devices (access points, managed switches, etc.) in the *Surf-LAN* must be configured properly.

4.7.1 VLAN Configuration

In the WebAdmin menu at **Security / VLAN** you can add new VLANs and then activate them. Afterwards you can assign different methods to log in for each VLAN.

- **Ticket based** The user does get redirected to the landing page. Then he needs to login with either an existing ticket, or by creating a new one.
- **Autologon/no charge** The guest will get logged in automatically with the assigned bandwidth.
- **Autologon/no charge/deny roaming** The guest will get logged in automatically with the assigned bandwidth. The created ticket is only valid in the VLAN it was created in.
- **Autologon/no charge/shared bandwidth** The guest will get logged in automatically with the assigned bandwidth. All guests in this VLAN share the configured bandwitdth.

4.7.2 VLAN Configuration with PMS

If the PMS-interface is configured in the WebAdmin menu in **Modules / Interfaces**, you can activate the option **Map to room number** in **Security / VLAN**. This will create a VLAN on the IACBOX for each room and also enables alternate logon modes:

- **Show only room number** When guests get redirected to the logon page, the room number field will already be filled.
- **Semiautomatic logon** Guests do not need to enter any data to logon and automatically get redirected to the *ticket selection*.

4.7.3 Per VLAN Redirect Before Logon

For each configured VLAN a redirect before logon can be assigned.

For example: If a guest gets online via an *Access Point* near a hotel bar, he can get redirected to a custom page with additional suggestions relating to the hotel bar.

To accomplish this, in the WebAdmin menu navigate to **Client Logon / Redirect** and fill in all the websites you want to redirect to.

Then switch to **Security / VLAN** and activate the option **Per VLAN redirect before logon**. Now you can select a redirect link for each VLAN.

Please keep in mind that you will need to redirect to the IACBOX logon-page in return, so the guest can use or create a ticket to access the internet. To establish this backlink, simply add a button or text on your custom redirect page which does point to the logon-page of the IACBOX:

- https://hotspot.internet-for-guests.com/logon/cgi/index.cgi
- https://proxy.surfnet.iacbox/logon/cgi/index.cgi (old IACBOX TLS/SSL certificate)

4.7.4 Web Filter per VLAN

If one of the both **Web Filter** options are active on the IACBOX, you can enable the option **Bypass Web Filter** in the VLAN settings. This option can be useful if you want to assign a VLAN for families/kids or in general if you want to block certain content.

4.7.5 Free Logon per VLAN

If the option **VLAN Based** is configured in the WebAdmin menu in *Tickets / Templates*, you can activate or disable the **Free Logon** per VLAN. Navigate to the WebAdmin menu **Security / VLAN**. Now you can activate or disable the option **Allow Free Logon**. In addition you can select **Yes but PMS authentication required**. This means that only users which are valid in your PMS-System can use the Free Logon.

4.7.6 Ticket Templates per VLAN

If VLANs are configured on the IACBOX you can assign single ticket templates to them. Navigate to **Tickets / Templates** and edit one. Here you can assign and limit the use with only specific VLANs.

- Example 1 - PMS Configuration

  In the WebAdmin menu **Modules / Interfaces** the PMS module is activated and configured. Additional 3 ticket templates are assigned to **PMS** and also to specific VLANs in **Tickets / Templates**

  - Ticket Template 1: assigned VLANs 10, 12
  - Ticket Template 2: assigned VLANs 10
  - Ticket Template 3: assigned VLANs 11, 12

  Depending on which VLAN guests are coming from, only the PMS tickets which are allowed for the current VLAN are being displayed.

- Example 2 - Social Login

  The Facebook login is activated and configured in the WebAdmin menu **Modules / Interfaces**. In addition in section **Tickets/Templates** one template is assigned to **Social Login**.

  - Ticket Template Social: assigned VLAN 10

  Since the ticket template was assigned to VLAN 10 only, the Facebook login is only visible if guests connect from this VLAN.

CHAPTER

FIVE

FILTERING

5.1 Application Control

This manual will explain the functionality and configuration of the module **Application Control**.

Hint:

- The module *Application Control* must be licensed separately.
- Note that the *Application Control* can cause high CPU usage and therefore requires **additional ressources**, for suggestions refer to the hardware requirements.
- It is not recommended to enable more then 20 protocols at the same time.

5.1.1 General information

The IACBOX module *Application Control* allows you to log, restrict or block **about 190 different applications and network protocols** within the Surf-LAN. This allows you to get an overview (log) of the Surf-LAN activities to then restrict (e.g. online streaming) and/or block (e.g. filesharing) different protocols and applications.

5.1.2 Differences between BASIC and PRO

The Application Protocol Module is available in 2 different versions, BASIC and PRO. The main differences will be explained below. First off the BASIC Edition, which consists of following functionality:

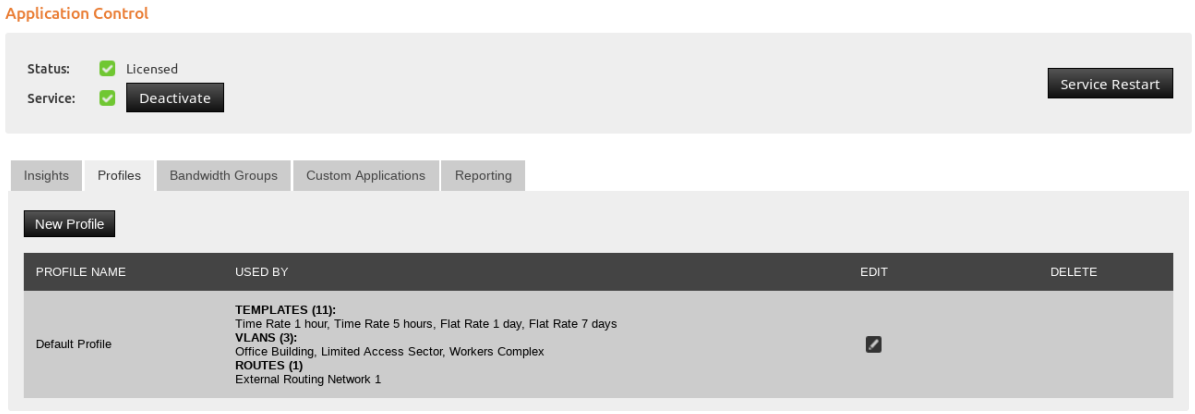
- Logging, blocking and shaping of over 190 Applications and Protocols
- Realtime Reports of current statistics
- Up to 20 independent Bandwidth Groups to shape Applications
- One global Application Control Profile

The PRO Edition expands all BASIC features with following functionality:

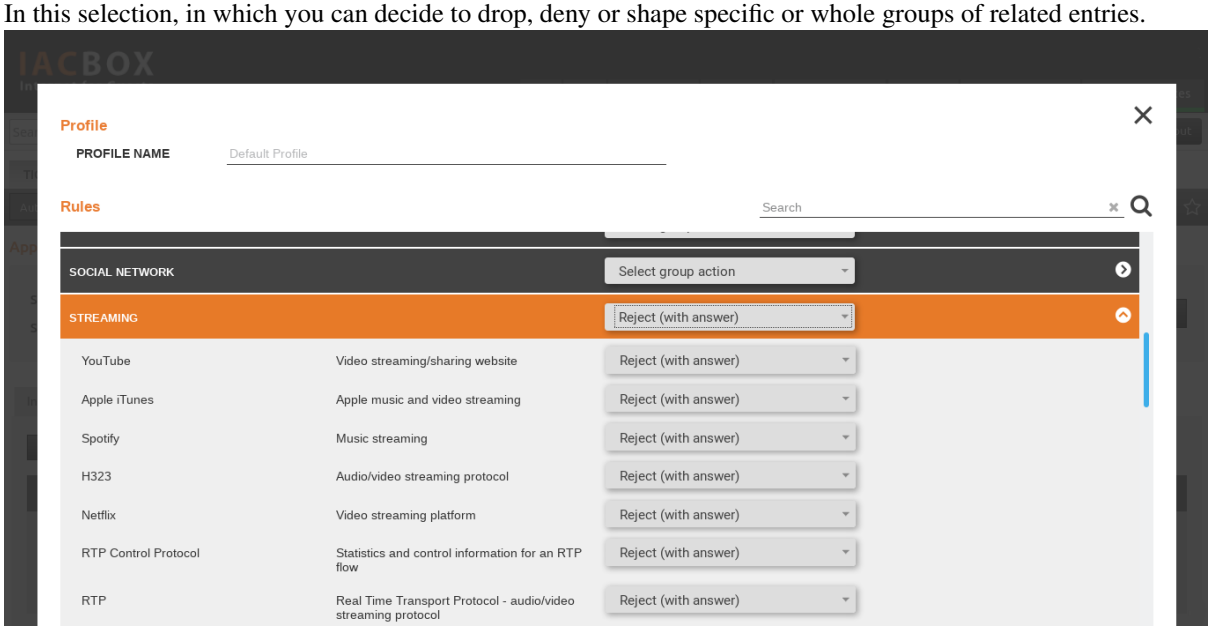
- Create unlimited Application Control Profiles and assign them to ticket Templates and VLANs/Routes
- Unlimited Bandwidth Groups
- Detailed statistics over time for detected and filtered Applications/Protocols
- Add custom Applications with your own rules

5.1.3 Configuration

After activating the Application Control in the menu **Modules / Application Control**, navigate into the **Profiles** Tab and click on the **Edit Icon** on the right side to open up the Applications/Protocols selection.



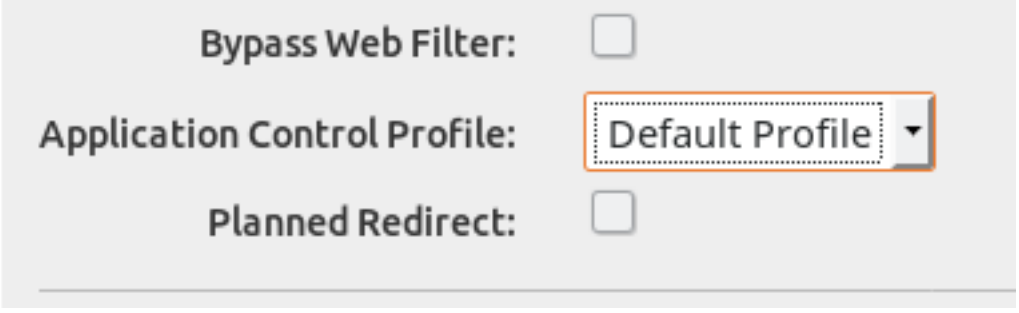
In this selection, in which you can decide to drop, deny or shape specific or whole groups of related entries.



An explaination of possible actions per Application/Protocol can be found below:

- **Drop** - Selecting "Drop" for an Application/Protocol means that it will be dropped (silent) without any answer to the requesting client or server.
- **Reject** - This Setting will actively return a Deny (e.g. a TCP Deny).
- **Shaping** - This Setting allows you to select a Bandwidth Group in order to Limit the Bandwidth of one or multiple Appliations/Protocols.

After configuring the standard or a custom profile, it still must be assigned to a Ticket Template, a VLAN or a Route to take effect. To assign Application Control profiles to a Ticket Template, navigate to the WebAdmin menu **Tickets / Templates**, edit a Template and select the favoured profile in the "Application Control Profile" dropdown menu. If you did not create a custom profile, then only the default one will be selectable here:



If you want to do this for a whole VLAN or a Route, then the same assingment is possible in either **Security /**

VLANs or **Security / Routes**. Note that for VLANs and Routes this can only be applied for for Logon Methods which are overridden, like the **Autologon** or the **Auto pass-through**, because for regular Tickets the Application Control Assignment is already handled via Ticket Templates.

5.1.4 Use-Cases

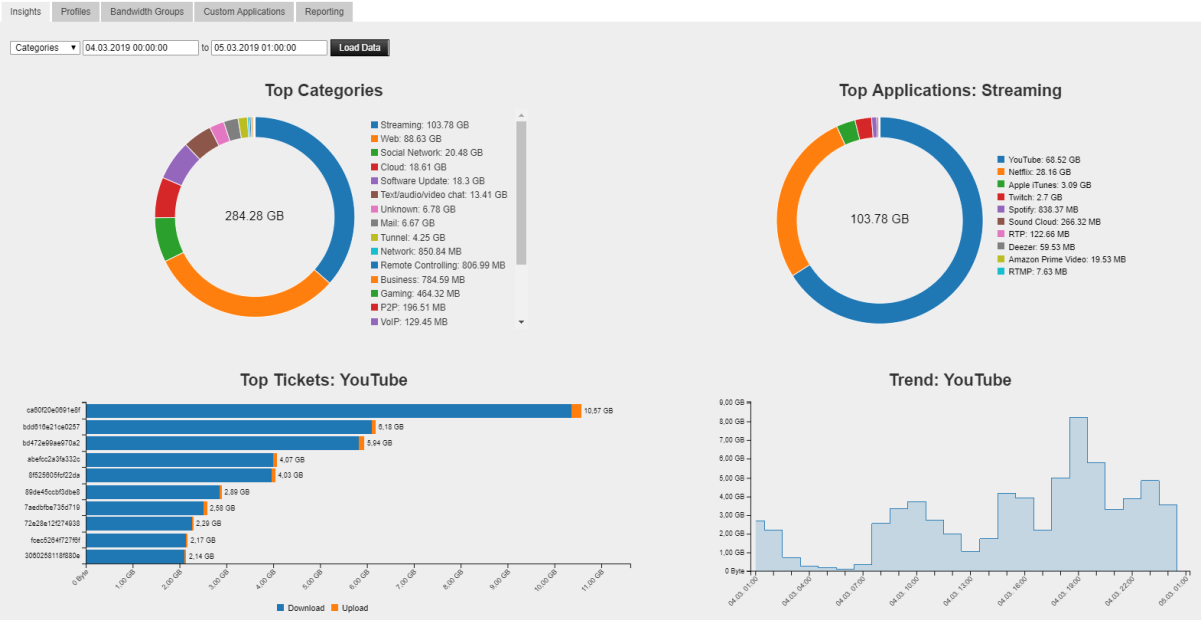
The most common use-cases are listed below:

- Restrict the Bandwidth for Streaming sites for free Tickets, while allowing undestricted access for Paid Tickets
- Prevent access to possibly illegal filesharing platforms in pulic or educational environments
- Block a variety of game launchers and social media applications to avoid distraction for children and students in educational environments
- Avoid Applications from Updating to save bandwidth on locations with limited internet connection

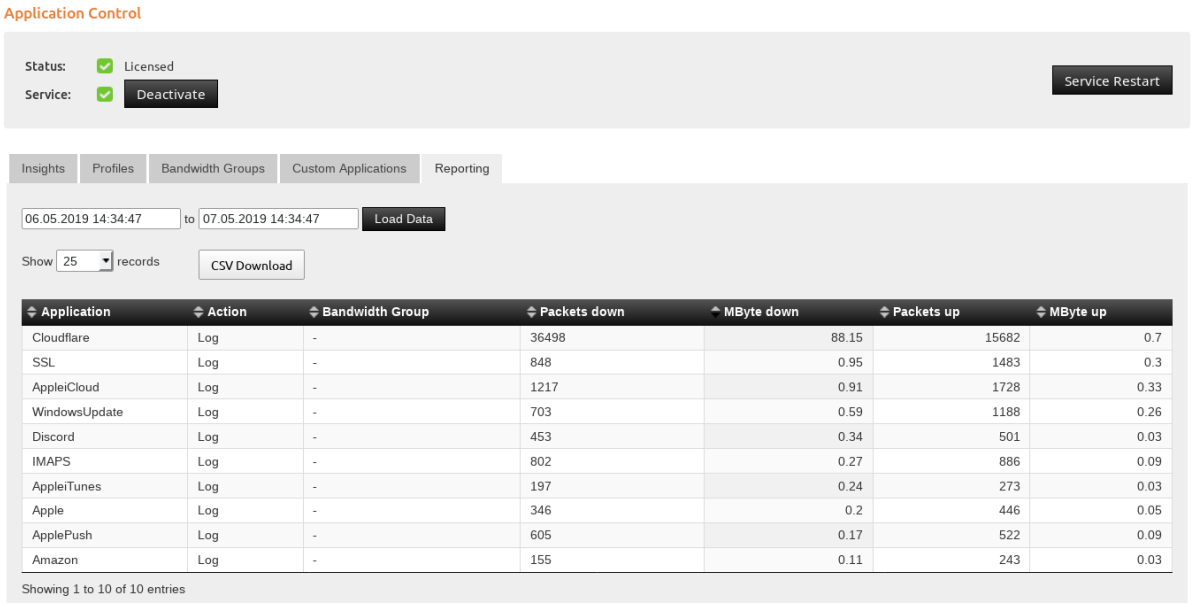
5.1.5 Statistics

There are several different statistics, depending on if you own the BASIC or PRO version of the Application Control.

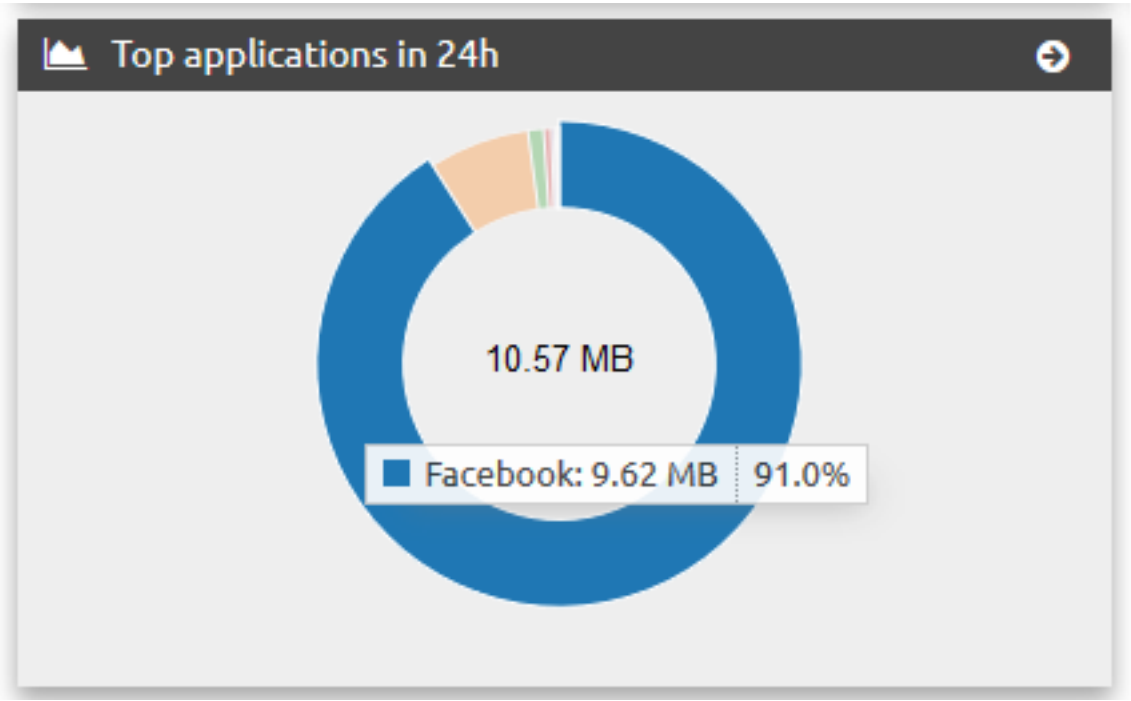
- Full, sortable graphical **Insights** can be viewed, filtered, and searched for in the Application Control front page - this is only available for the PRO version.



- An overview of what is currently being detected can be viewed on the Application Control WebAdmin page, by navigating into the Tab “Reporting”.



- A generic overview of the last 24 hours can be viewed in the WebAdmin Dashboard



5.2 Content Filter (legacy)

Attention:

- The **Content Filter** has been replaced with the **DNS Filter** as of IACBOX Version 17.2 and will only remain as **legacy** option
- It is not suggested to use the **Content Filter (legacy)** and the **DNS Filter** at the same time - if in doubt then only use the **DNS Filter**

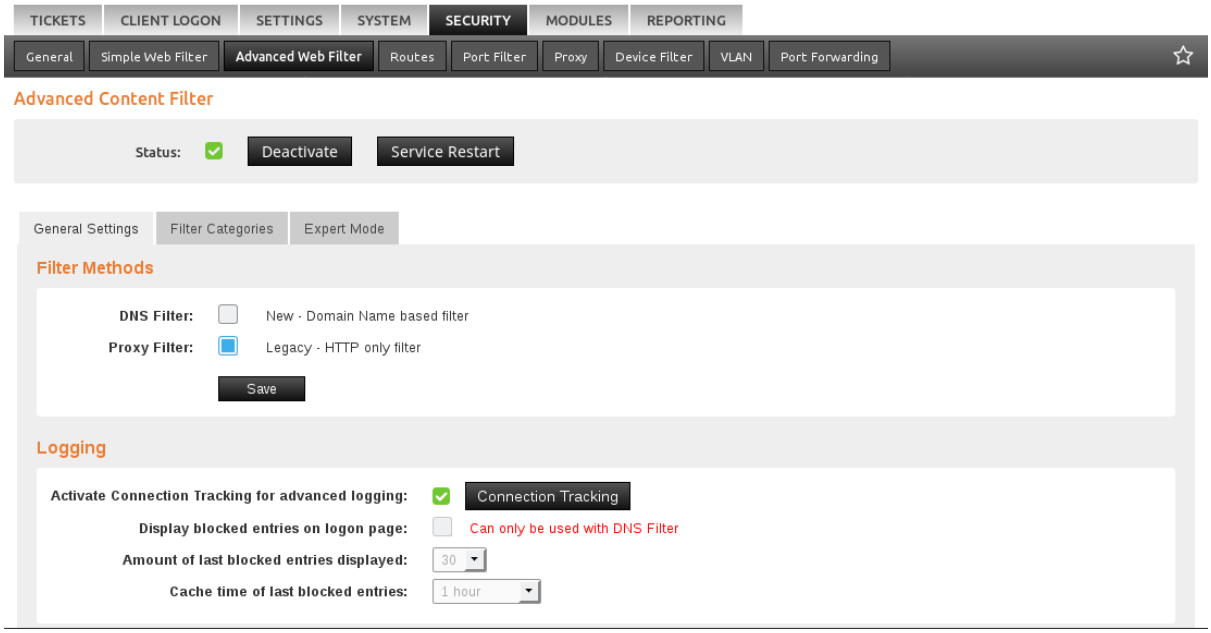
This manual describes how to configure and use the **Content Filter (legacy)**.

Hint:

- Out of the box, The **Content Filter (legacy)** can filter simple **HTTP** webpages, but not HTTPS
- Filter lists are being shared with the new **DNS Filter**, which does filter connection end-points and thus can also block **HTTPS** webpages
- In order to receive the newest and most recent Filter Lists, activate the **Content Filter (legady)** and hit **Service Restart**. From this point on, perform an **Online Update** in order to obtain the newest revision of filter lists

5.2.1 WebAdmin Configuration

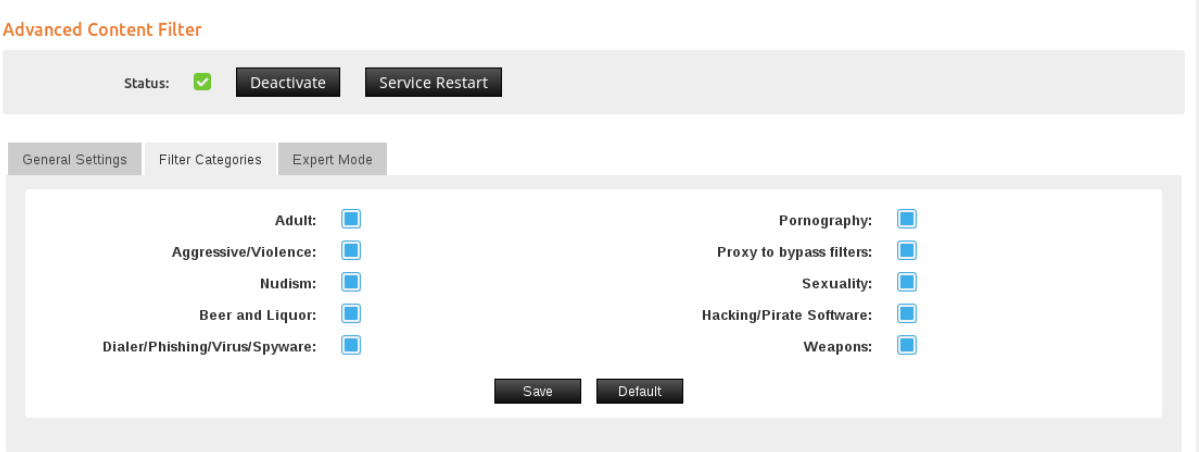
In order to activate the **Content Filter (legacy)**, open the WebAdmin and navigate to **Security / Advanced Web Filter** and first click on **Activate** to initialize its settings. This will activate the required services and display available options for it. After the service was activated, following options will be shown.



The options shown are the **DNS Filter** (new) and the **Proxy Filter**, also referred to as **Content Filter**, which is the legacy option of the two. After selecting **Proxy Filter** hit **Save** before doing any further configuration.

5.2.2 Select Filter Lists

In the next step, the **Filter Lists** can be selected in the tab **Filter Categories**. Select all desired filters and hit **Save** to continue.



5.2.3 Advanced Settings

Attention:
- If **Force HTTPS connections via Content Filter** is acivated, all active SSL connections will be forced over the proxy, which will inevitable lead to **certificate errors**. For any regular guest environments, this option will completely ruin the experience and usability.

The *Advanced Settings* allow you to to:

- Check for domains/URLs in the filter lists
- Manually add new domains and URLs to block
- Block specific file extensions and mime-types
- Work with **word lists**
- Set exceptions for all options above

A detailed description can be found in the **help menu** of this WebAdmin page. After the configuration is done, hit **Save**. This will write the configuration, but it is not yet active. In order to activate it, a **Service Restart** is required, which can be found on top of this WebAdmin page.

5.3 DNS Filter

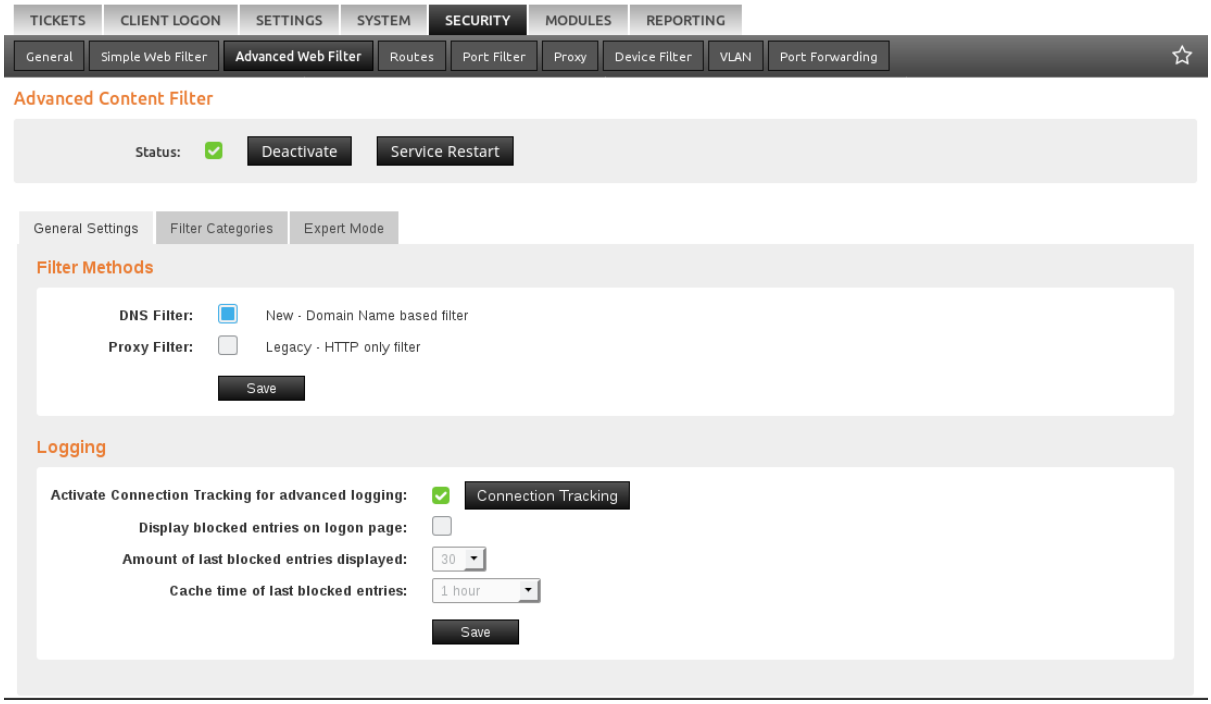
Attention:
- It is not suggested to use the **Content Filter (legacy)** and the **DNS Filter** at the same time - if in doubt then only use the **DNS Filter**

This manual describes how to configure and use the **DNS Filter**.

Hint:
- The **DNS Filter** will filter all nameserver requests, which means clients wont be able to resolve blocked domains at all. As a result, this will prevent not only HTTP/HTTPS connections to filtered domains, but all connection types.
- In order to receive the newest *Filter Lists*, activate the **DNS Filter**, hit **Service Restart** and then perform the **IACBOX Online Update**

5.3.1 WebAdmin Configuration

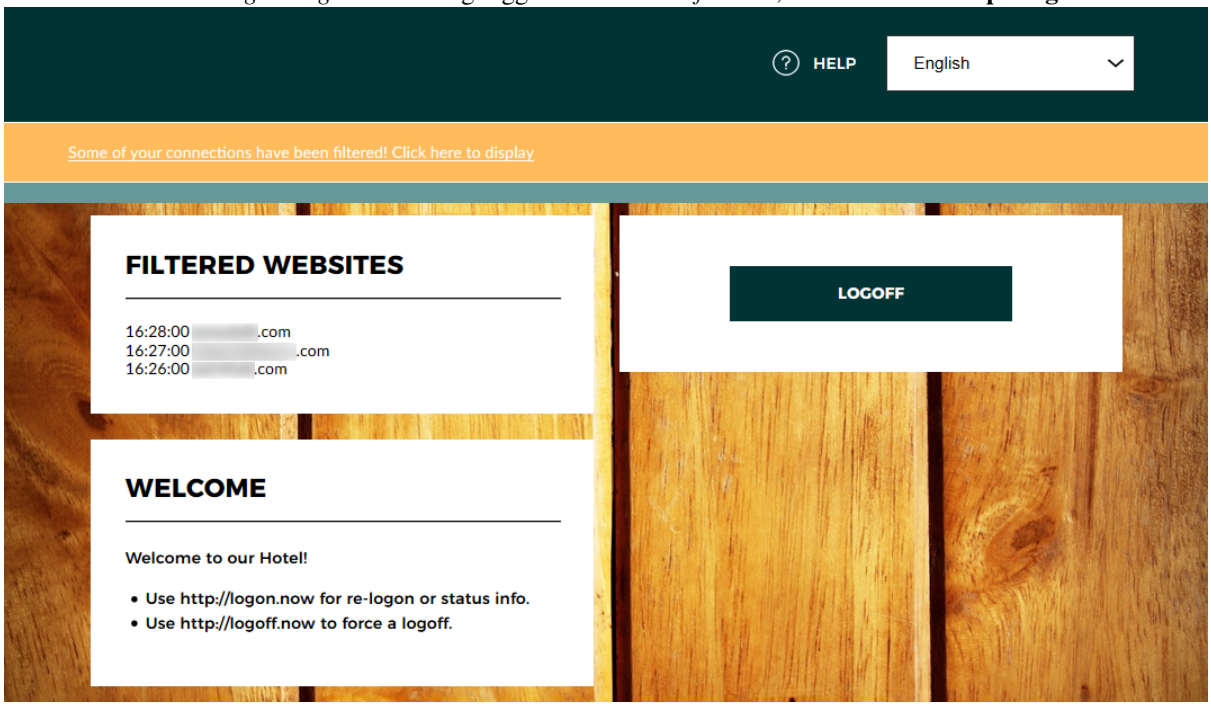
In order to activate the **DNS Filter**, open the WebAdmin and navigate to **Security / Advanced Web Filter** and first click on **Activate** to initialize its settings. This will activate the required services and display available options for it. After the service was activated, following options will be shown.



The options shown are the **DNS Filter** (new) and the **Proxy Filter**, also referred to as **Content Filter**, which is the legacy option of the two. If the IACBOX was freshly installed with version **17.2**, the **DNS Filter** will be enabled by default. Upgraded IACBOXes will not automatically switch to the **DNS Filter** - this must be done manually.

5.3.2 Special Settings for the DNS Filter

Since the **DNS Filter** blocks the name resolution of unwanted target domains, connections to these will fail. This means that, if clients try to open **HTTPS Webpages**, their browser will return a simple **The Page could not be loaded** message. The problem with these is that the IACBOX can not simply redirect clients to a local *Site has been blocked* page, because webbrowsers would detect the certificate mismatch of the target domain and the *Site has been blocked* resource on the IACBOX and eventually output an aggressive **SSL Certificate error**. The approach with the **DNS Filter** is to list blocked DNS requests on the *IACBOX Logon Page* instead. In order to access the *IACBOX Logon Page* while being logged in with a *Surf-Ticket*, clients can call **http://logon.now**.



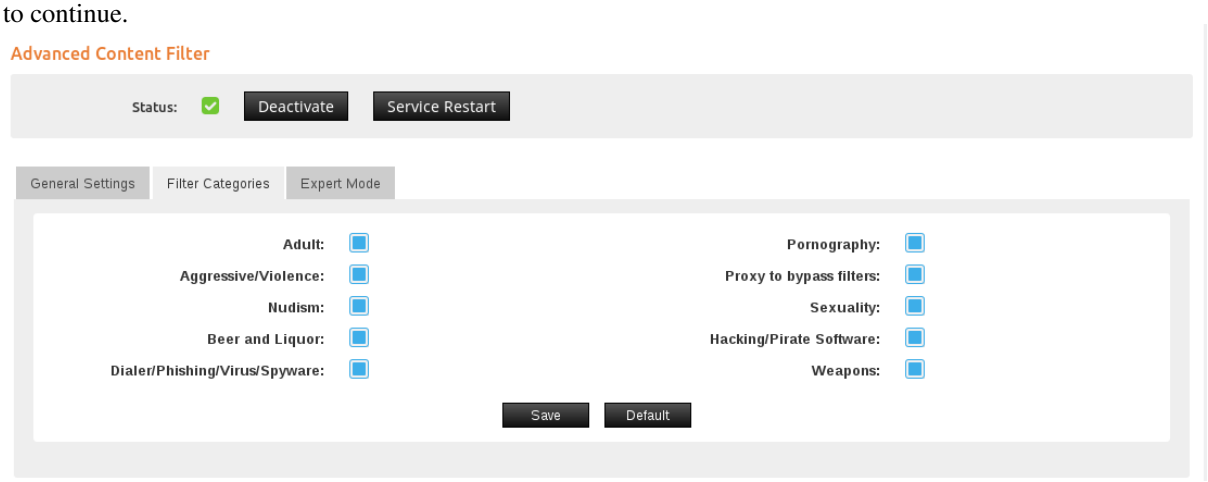
As seen in the screenshot, the *IACBOX Logon Page* displays a warning on top which states *Some of your connections have been filtered! Click here to display*. After clicking on this message, the *Filtered Websites* section will be shown above the welcome section.

Integrators can decide to configure these warnings in the Web Filter WebAdmin menu at **Security / Advanced Web Filter**. Available options are:

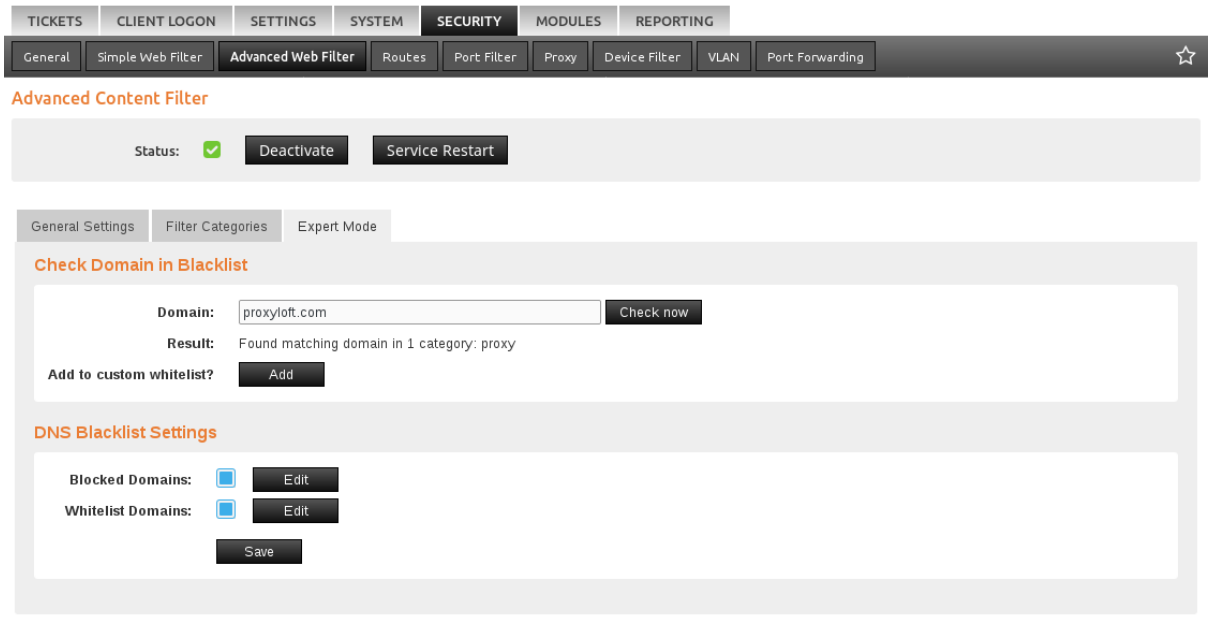
- **Activate Connection Tracking for advanced logging** - The **Connection Tracking** must be activated in order to obtain and save the DNS requests of clients. Once activated, the configuration below will become accessible.
- **Display blocked entries on logon page** - Integrators can decide to either show or hide blocked DNS entries from guests. If the **Connection Tracking** was activated before this step, administrators can review blocked connections in **Tickets / Manage** by clicking on the **Web Filer** icon to the left of the ticket name. This will open the **Connection Log** of this ticket, which includes filtered entries.
- **Amount of last blocked entries displayed** - The amount of filtered entries which should be displayed on the *Client Logon Page*.
- **Cache time of last blocked entries** - The time in which filtered entries are being shown on the *Client Logon Page*.

5.3.3 Select Filter Lists

In the next step, the **Filter Lists** can be selected in the tab **Filter Categories**. Select all desired filters and hit **Save** to continue.



5.3.4 Advanced Settings



The *Advanced Settings* allow you to to:

- Check for domains/URLs in the filter lists
- Manually add new domains to the filter- and whitelist
- Upload a **Custom Domain List** to block or whitelist alot of domains at once

A detailed description can be found in the **help menu** of this WebAdmin page.

After the configuration is done, hit **Save**. This will write the configuration, but it is not yet active. In order to activate it, a **Service Restart** is required, which can be found on top of this WebAdmin page.

CHAPTER

SIX

DATA PRIVACY

6.1 Data Privacy (GDPR/DSGVO)

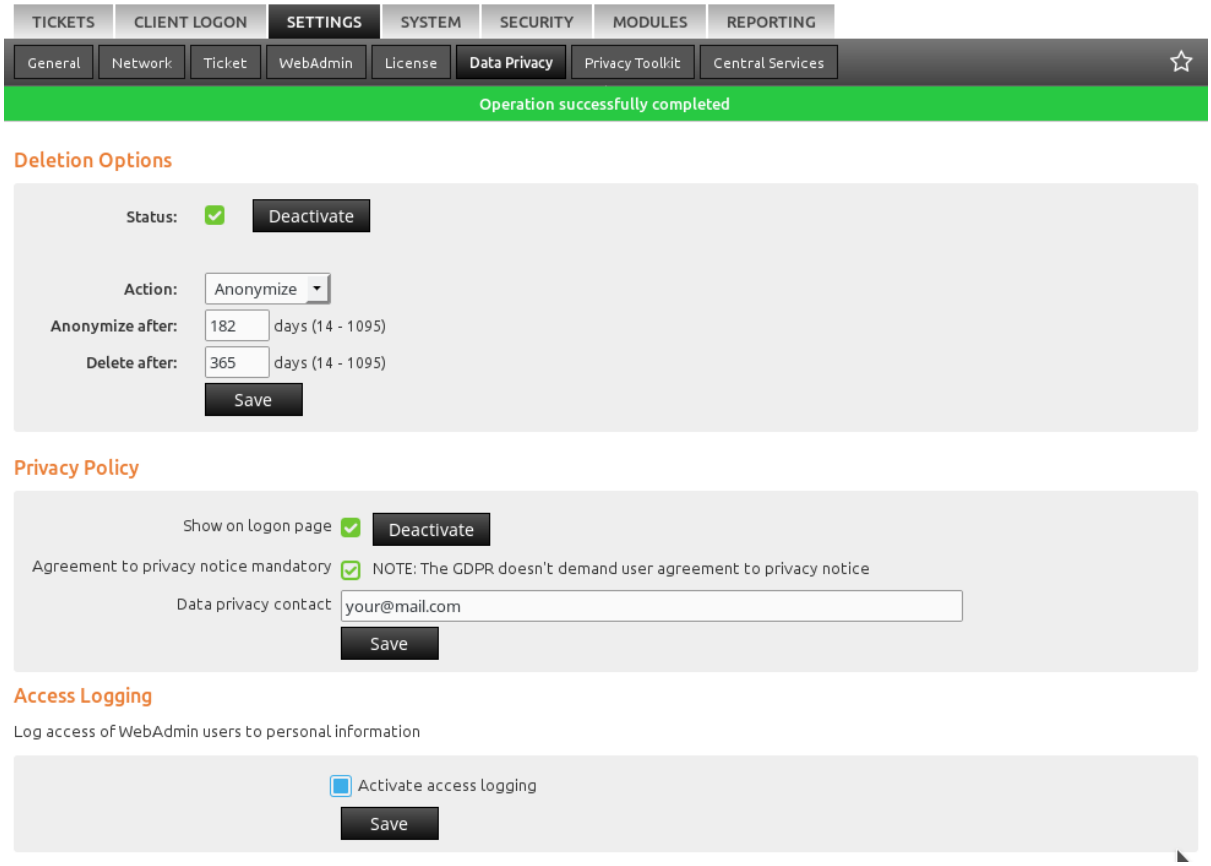
This manual describes how to configure the GDPR/DSGVO settings for the new european data regulation laws, which will become effective on **25th May 2018**. The IACBOX offers according privacy settings to comply with the new regulations.

Hint:

- The separate module **Privacy Toolkit** allows you to create and download reports automatically, depending on your current IACBOX configuration (including administrative permissions, active modules, etc.).
- The **Data Privacy** settings are available on all IACBOXes **version 17.2** and **patchlevel 12221** or newer. If the **Data Privacy** setting is not available on your IACBOX, perform the **Online Update**.

6.1.1 WebAdmin Menu

The **Privacy Settings** can be found in the WebAdmin menu **Settings / Data Privacy**.



Deletion Options

The **Deletion Options** can be configured to either directly **delete** data after the configured amount of days - or to **anonymize** it and delete it sometime later.

Default Options:

- Anonymize after 182 days
- Delete after 365 days

Privacy Policy

Configure if the privacy policy should be shown on the *IACBOX Logon Page* and which contact email address will be listed in it.

Hint:

- If the **Login API** is used, then it is required to activate according privacy policy settings in the Login API configuration file instead.

Access Logging

If activated, the activity of WebAdmin users will be logged in **Reporting / Application** whenever a WebAdmin page with sensible data is being accessed.

6.1.2 Additional Resources

- EU GDRP: https://www.eugdpr.org/
- IACBOX Privacy Toolkit: https://www.iacbox.com/en/products/privacy-toolkit/

6.2 Privacy Toolkit

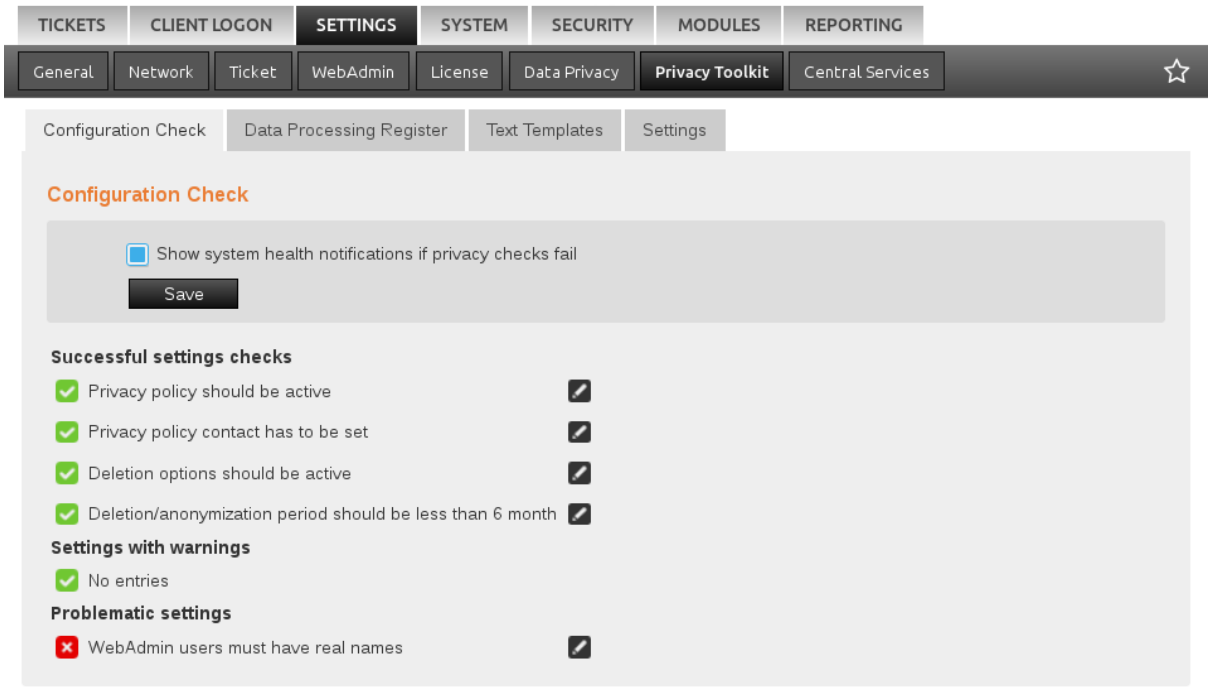
The **Privacy Toolkit** enables you to create automatic reports of accesses, permissions and data usage on your IACBOX, depending on what modules are currently activated. The according WebAdmin menu can be found at **Settings / Privacy Toolkit**.

Hint:

- The **Privacy Toolkit** module can be bought for all IACBOXes **version 17.2** and **patchlevel p12221** or newer. If the **Privacy Toolkit** setting is not available on your IACBOX, perform the **Online Update**.

6.2.1 Configuration Check

In this tab you can enable/disable privacy related system health notifications. Below this setting, the current configuration status of privacy related settings can be examined.

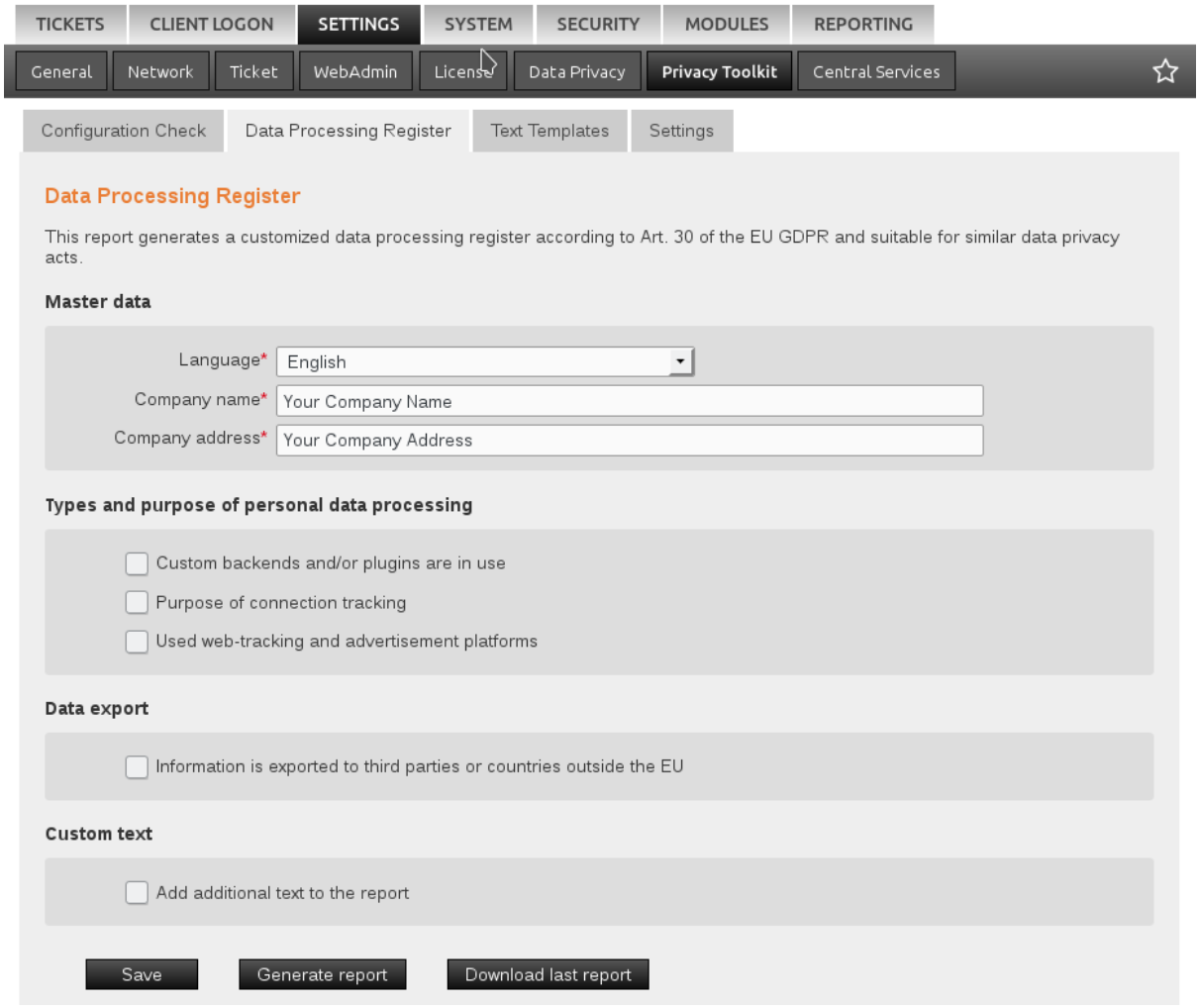


6.2.2 Data Processing Register

The **Data Processing Register** allows you to add text segments to the automatically created report. Besides the **Language** and **Company Address** these segments can be added for a multiple purposes:

- **Custom backends and/or plugins are in use:** The primarily use for this option is to name external backends which receive data via a third party **Login API** plugin - or any other custom implementation of the available IACBOX interfaces.
- **Purpose of Connection Tracking:** E.g. hospitals or government institutions do require connections to be tracked. If you do track connections in other environments, you must add a sufficient reason to do so.

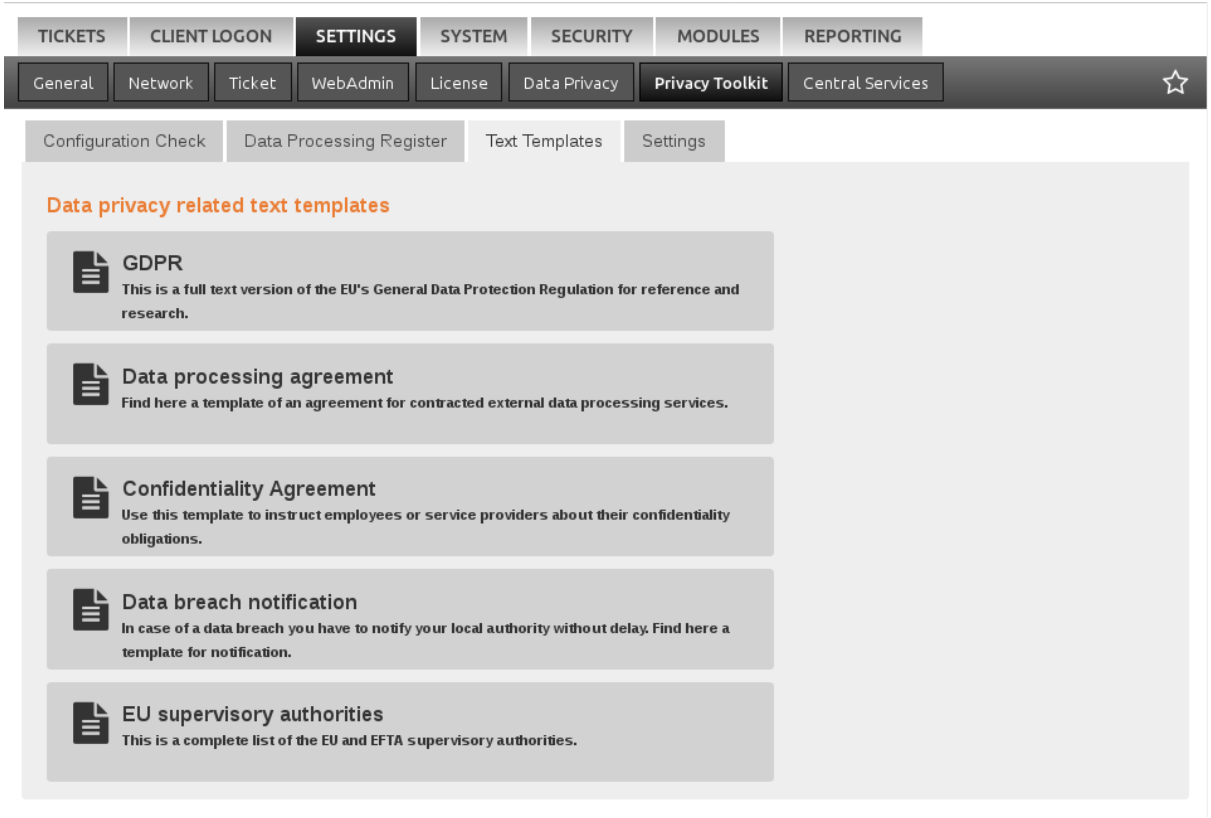
- **Used web-tracking and advertisement platforms:** For example if you use any external data-processing services in combination with the *IACBOX Logon Page* or *Login API*, you can list them here (e.g. Google Analytics, etc.).
- **Information is exported to third parties or countries outside the EU:** E.g. if data is being sent to Facebook Inc or Google LLC, depending on what modules or third party services are used on your IACBOX.
- **Add additional text to the report:** Just for additional text, also allows you to add custom headlines to the final report.



At the bottom of this page you can **Generate and/or Download Reports**!

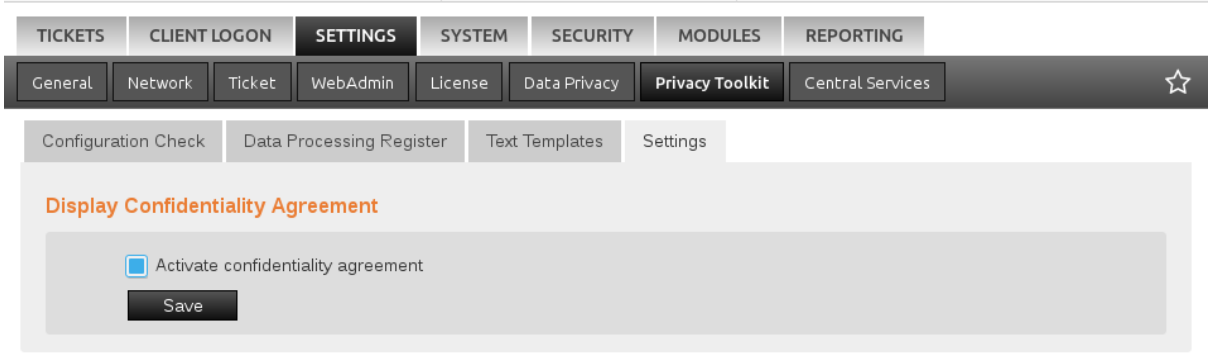
6.2.3 Text Templates

This WebAdmin page holds all available documents you might require.

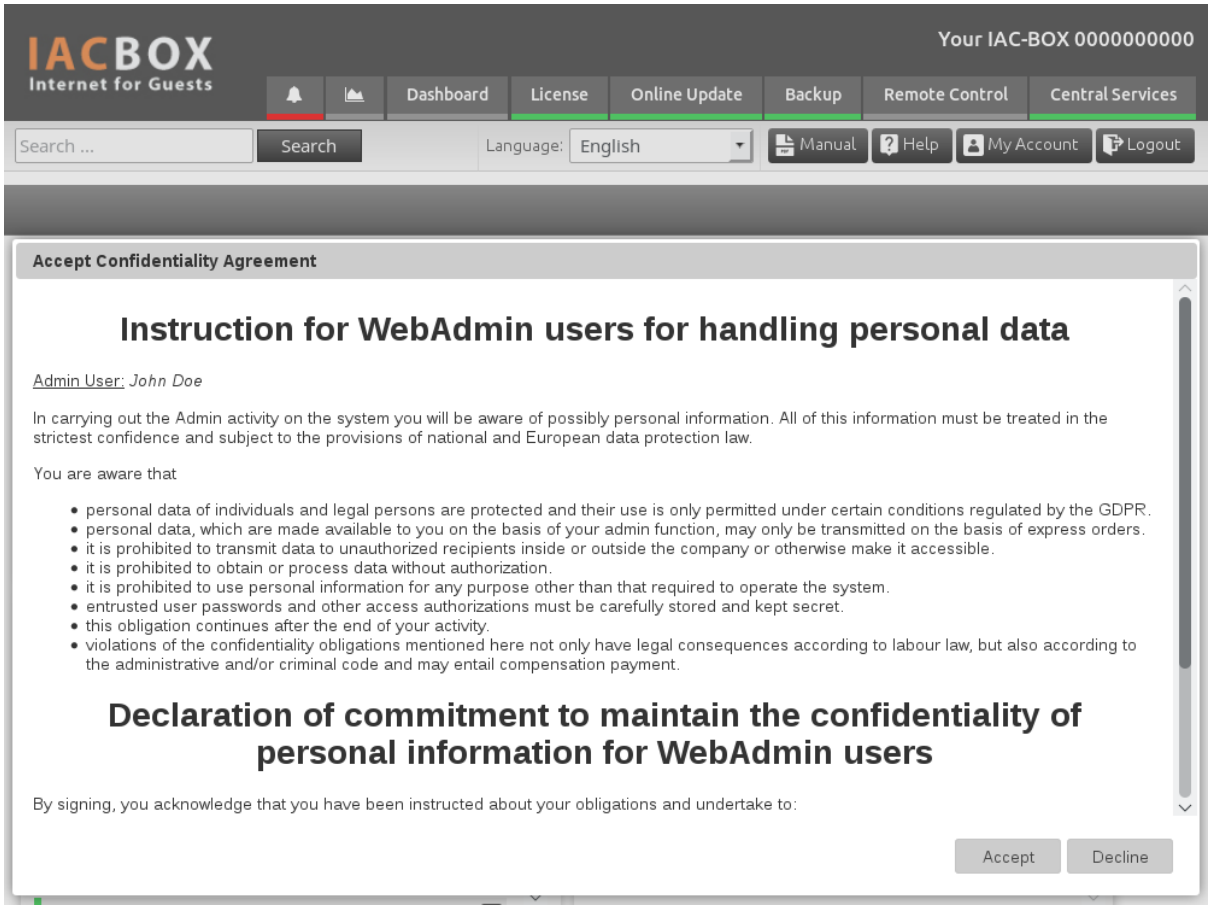


6.2.4 Settings

It is likely that this tab will grow over time. As for now, you can enable the **Confidental Agreement** popup for WebAdmin users. If this is enabled, WebAdmin users will need to accept a confidental agreement with the next WebAdmin page they open.



The popup for the confidental agreement:


IACBOX
Internet for Guests
Your IAC-BOX 0000000000
Dashboard
License
Online Update
Backup
Remote Control
Central Services
Search ...
Search
Language:
English
Manual
Help
My Account
Logout
Accept Confidentiality Agreement
Instruction for WebAdmin users for handling personal data
Admin User: John Doe
In carrying out the Admin activity on the system you will be aware of possibly personal information. All of this information must be treated in the strictest confidence and subject to the provisions of national and European data protection law.
You are aware that
personal data of individuals and legal persons are protected and their use is only permitted under certain conditions regulated by the GDPR.
personal data, which are made available to you on the basis of your admin function, may only be transmitted on the basis of express orders.
it is prohibited to transmit data to unauthorized recipients inside or outside the company or otherwise make it accessible.
it is prohibited to obtain or process data without authorization.
it is prohibited to use personal information for any purpose other than that required to operate the system.
entrusted user passwords and other access authorizations must be carefully stored and kept secret.
this obligation continues after the end of your activity.
violations of the confidentiality obligations mentioned here not only have legal consequences according to labour law, but also according to the administrative and/or criminal code and may entail compensation payment.
Declaration of commitment to maintain the confidentiality of personal information for WebAdmin users
By signing, you acknowledge that you have been instructed about your obligations and undertake to:
Accept
Decline

CHAPTER

SEVEN

REMOTE ADMINISTRATION / INTERFACING

7.1 Batch Access API

This manual describes the **Batch Access API** of the IACBOX and what is possible with it.

Hint:

- The **Batch Access API** is available on IACBOX version **5.0.7615 (p7742)** or newer.
- Some functions were added later on. Ensure to update to the most recent version of the IACBOX in order to access all functions listed in this manual.
- The requesting client must have access to the IACBOX *WebAdmin* interface.

7.1.1 General

With the **Batch Access API** it is possible to **export** and **import** data, as well as to trigger some actions on the IACBOX.

Export Data

- Statistics
- Connection Tracking
- Application Log
- System Log
- Messaging Data
- Current User Info
- License Info
- System Info
- Version
- Ticket Templates
- GDPR Data Processing Register Report
- Ticket Data

Ticket Commands:

- Create
- Log off
- Revoke

Autologon Devices

- Create an Autologon Device
- Delete an Autologon Device
- Import a List of Autologon Devices

System Commands:

- System Reboot/Poweroff (UPS operation support)
- System Backup
- Start Online Update
- Update Log List
- Export Update Log

7.1.2 Usage

In order to obtain or send data, a **HTTP POST** must be sent to the **Batch Access API** interface. This can be done using scripted tools as well as with PHP or other programming environment. In this manual we use **cURL** for demonstration purposes. *cURL* is a simple command line tool for alot of different Operating System and also available as PHP extension.

The software (source and binaries (e.g. Windows binaries, x64 SSL version) can be obtained here: https://curl.haxx.se/download.html

After downloading & extracting the binaries into the **system32** directory, so that **cURL** will become available within the Windows command line.

7.1.3 Authentication

The access is restricted by a **username** and a **password**. For easy test the user **sysop** could be used, but this is not recommended as the password has to be saved in the script. Instead create a new user on the system and allow only the needed rights to this user. Some general actions that don't map to a certain menu need the right for **System -> Services**.

Use the command **insecure** if you don't want libcurl to verify the SSL/TLS certificate of the peer.

With the **lang** data you specify the WebAdmin language that should be used for the *cURL* call. This has influence on some of the exportable data and means that the returend data will be in the specified language. The following languages are available:

- en_US = english
- de_DE = german
- it_IT = italian
- fr_FR = francais

In all examples below please replace **<USERNAME>** and **<PASSWORD>** with your created user.

7.1.4 Export Data

This section will explain how to extract data from the IACBOX according to the list on top of this manual.

Statistics

Statistics includes **tickets** or **revenue**, which both are available in **csv** or **xls** format. Exports can be filtered by **from_date** and **to_date**. Optional fields are:

- **export_id**: Returns the ticket **ids** as first column value. With the ticket id it is possible to send further commands.
- **search_text**: Search for specific ticket names or ticket descriptions, also MAC addresses (e.g. AA:BB:CC:11:22:33 and aabbcc112233) are recognized.
- **issuer**: Search for tickets which were created by a specific user/service (e.g. **sysop**).
- **revoked**: Either show all tickets that have been revoked (1) or all tickets which have not been revoked (0).

cURL command to **retrieve tickets** from device **AA:BB:CC:11:22:33** between **01.01.2015** and **31.01.2015** which were **revoked**:

```
curl --insecure -o tickets.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=statistics&download=tickets&dataformat=csv&from_date=2015.01.01 00:00:00
&to_date=2015.01.31 23:59:59&export_id=1&search_text=AA:BB:CC:11:22:33
&issuer=sysop&revoked=1" https://192.168.1.1/batch.php
```

Connection Tracking

Includes **proxy** and **conntrack** data which both are available in **csv** or **raw** format. Exports can be filtered by **from_date** and **to_date**. Optional fields are:

- **search_text**: Search for websites, connections, IP addresses or MAC addresses.

cURL command to **retrieve** connection tracking proxy data which contains the string **amazon** between **01.01.2015** and **31.02.2015**:

```
curl --insecure -o proxy_log.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=connection_tracking&download=proxy&dataformat=csv&from_date=2015.01.0100:00:00
&to_date=2015.01.31 23:59:59&search_text=amazon" https://192.168.1.1/batch.php
```

Application Log

The **Application Log** of the *WebAdmin* menu **Reports / Application**. Can be exported in **csv** or **xls** format and filtered by **from_date** and **to_date**. Optional fields are:

- **search_text**: Filter the logs with a specific search text.

cURL command to export all **Application Logs** between **01.01.2015** and **31.01.2015** which include the **search_text** "logon".

```
curl --insecure -o application_log.csv --data "username=<USER>&password=<PASSWORD>
&action=application_log&download=logdata&dataformat=csv&from_date=2015.01.0100:00:00
&to_date=2015.01.31 23:59:59&search_text=logon" https://192.168.1.1/batch.php
```

Application Log Tickets only

Exports only **Application Logs** which are caused by **tickets**, e.g. a ticket *login* or *logoff*. Can be exported in **csv** or **xls** format and filtered by **from_date** and **to_date**. Optional fields are:

- **hide_msg**: Extended log messages will not be exportet.
- **search_text**: Filter the logs with a specific search text.

cURL command to export all **Application Logs** related to ticket actions between **01.01.2015** and **31.01.2015** which also include the **search_text** "logon" but **without extended log messages**.

```
curl --insecure -o application_log.csv --data "username=<USER>&password=<PASSWORD>
&action=application_log&download=logdata&dataformat=csv&from_date=2015.01.01 00:00:00
&to_date=2015.01.31 23:59:59&hide_msg=1&search_text=logon" https://192.168.1.1/batch.php
```

System Logs

Exports **system** or **mail** logs, selectable by **generations**, e.g. **generation 0** means today and **generation 1** means yesterday. Only the last **7 days** are available.

Hint:

- The output is a compressed **gzip file**.

---

cURL command to export the compressed **system logs** from today.

```
curl --insecure -o system_log.zip --data "username=<USER>&password=<PASSWORD>&action=system_log&
download=system&generation=0" https://192.168.1.1/batch.php
```

Messaging Data

Exports **Messaging Data** which was previously obtained by messaging modules. Can be exported as **csv** and **xls** and filtered by **from_date** and **to_date**. The following **data-types** are possible:

- **email**: Email addresses which were used to create tickets with the **Email module**.
- **sms**: Phone numbers which were used to create ticket with the **SMS module**.
- **social**: Email addresses which were used to create tickets by logging in with the **Social module**, e.g. **Facebook** or **Google+**.
- **tkrq**: Email addresses which were used to send a request via the **Email ticket request module**.
- **dtc**: Data which was obtained by using the **Data Collector**.

cURL command to export data which was gathered by the **Email ticket request module** between **01.01.2015** and **31.01.2015**.

```
curl --insecure -o messaging_data.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=messaging&download=tkrq&dataformat=csv&from_date=2015.01.0100:00:00
&to_date=2015.01.31 23:59:59" https://192.168.1.1/batch.php
```

User Info

Returns **online** and **maximum concurrent** users in **json** format.

```
curl --insecure -o userinfo.jsn --data "username=<USER>&password=<PASSWORD>&action=json
&want=userinfo "https://192.168.1.1/batch.php
```

License Info

Returns **license data** and **licensed modules** in **json** format.

```
curl --insecure -o licenseinfo.jsn --data "username=<USER>&password=<PASSWORD>&action=json
&want=licenseinfo" https://192.168.1.1/batch.php
```

System Info

Returns **CPU load**, **memory usage** and **hdd usage** in **json** format.

```
curl --insecure -o systeminfo.jsn --data "username=<USER>&password=<PASSWORD>&action=json
&want=systeminfo" https://192.168.1.1/batch.php
```

Version

Returns **software version**, **patchlevel** and **release date** of the IACBOX in **json** format.

```
curl --insecure -o version.jsn --data "username=<USER>&password=<PASSWORD>&action=json
&want=version" https://192.168.1.1/batch.php
```

List Ticket Templates

Returns all available **ticket templates** in json format.

```
curl --insecure -o templates.jsn --data "username=<USER>&password=<PASSWORD>
&action=templates&subaction=get_templates" https://192.168.1.1/batch.php
```

Get last GDPR Data Processing Register Report

Returns the last **GDPR Data Processing Register Report** as .pdf file. If there was no report created yet, create and return a new one. If a new report needs to be created, the settings from the WebAdmin menu **Settings/Privacy Toolkit** will be used.

Hint:

- In order to use this Batch Access API call, the **Privacy Toolkit Module** must be licensed.

```
curl --insecure -o report.pdf --data "username=<USER>&password=<PASSWORD>
&action=privacy_report" https://192.168.1.1/batch.php
```

Ticket Data

Returns **Ticket Data** from the *WebAdmin* menu **Tickets/Manage**. The returned data format is .CSV. The format for the time-span can be either **YYYY.MM.DD hh:mm:ss** or **YYYY-MM-DD hh:mm:ss**.

1.) cURL command to return **all** tickets created within a given time-span (**from_date** - **to_date**). If no **from_date** and **to_date** is provided, all tickets from now until 24 hours back are returned.

```
curl --insecure -o all_tickets.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=get_ticket_data&ticket_type=all&from_date=2018.07.01 00:00:00
&to_date=2018.07.10 23:59:59" https://192.168.1.1/batch.php
```

2.) cURL command to return **used** tickets that are still valid and were created within a given time-span (**from_date** - **to_date**). If no **from_date** and **to_date** is provided, all tickets from now until 24 hours back are returned.

```
curl --insecure -o used_tickets.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=get_ticket_data&ticket_type=used&from_date=2018.07.01 00:00:00
&to_date=2018.07.10 23:59:59" https://192.168.1.1/batch.php
```

3.) cURL command to return **unused** tickets that are still valid and were created within a given time-span (**from_date** - **to_date**). If no **from_date** and **to_date** is provided, all tickets from now until 24 hours back are returned.

```
curl --insecure -o unused_tickets.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=get_ticket_data&ticket_type=unused&from_date=2018.07.01 00:00:00
&to_date=2018.07.10 23:59:59" https://192.168.1.1/batch.php
```

4.) cURL command to return all tickets that are currently **online**.

```
curl --insecure -o online_tickets.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=get_ticket_data&ticket_type=online" https://192.168.1.1/batch.php
```

5.) cURL command to return all **invalid** tickets.

```
curl --insecure -o invalid_tickets.csv --data "lang=en_US&username=<USER>&password=<PASSWORD>
&action=get_ticket_data&ticket_type=invalid" https://192.168.1.1/batch.php
```

7.1.5 Ticket Commands

Ticket Create

Create tickets based on a specified template (by id).

Hint: Ticket Create via Batch Access API is available on IACBOX version 17 or newer.

Required parameter is **use_template**.

- use_template - template_id to use for ticket create (see **List Ticket Templates**)

Optional parameters are:

- return_userdata (1) - returns username + password of created ticket
- tk_username - specify a custom username for the new ticket
- tk_password - specify a custom password for the new ticket (min. length = 4 characters; 7 characters for password only tickets)
- expiration (Days / > 0) - overwrite template expiration period
- time_credit (Min. / > 0) - overwrite template time_credit
- ticket_limit (MB / > 0) - overwrite template ticket_limit
- session_limit (MB / > 0) - overwrite template session_limit
- idle_timeout (Min. / > 0) - overwrite template idle_timeout
- bw_in (Kbit/s / > 64) - overwrite template download_bandwidth
- bw_out (Kbit/s / > 64) - overwrite template upload_bandwidth

```
curl --insecure --data "username=<USER>&password=<PASSWORD>&action=create
&subaction=create_ticket&use_template=248&return_userdata=1
&expiration=15&time_credit=77&ticket_limit=1234
&session_limit=123&idle_timeout=99&bw_in=25000
&tk_username=myticket&tk_password=mysecret" https://192.168.1.1/batch.php
```

Ticket Logout

Logout certain tickets based on the **ticket id**. This also works with multiple **ticket ids**.

```
curl --insecure --data "username=<USER>&password=<PASSWORD>&action=manage_ticket
&subaction=logout&ids=12,22,540,299" https://192.168.1.1/batch.php
```

Ticket Revoke

Revoke certain tickets based on the **ticket id**. This also works with multiple **ticket ids**.

```
curl --insecure --data "username=<USER>&password=<PASSWORD>&action=manage_ticket
&subaction=revoke&ids=12,22,540,299" https://192.168.1.1/batch.php
```

7.1.6 Autologon Devices

Create a new Autologon Device

Add an autologon device as either **static** or **dynamic** device. Optional autologon types are:

- **type=static**: Adds the autologon device as a static device. Requires parameters **mac** and **ip**.
- **type=dyn_mac**: Adds the autologon device as a dynamic device identified via MAC address. Requires parameter **mac**.
- **type=dyn_ip**: Adds the autologon device as a dynamic device identified via IP address. Requires parameter **ip**
- **type=wildcard_mac**: Adds a wildcard entry to affect multiple devices at once via MAC address. Requires parameter **mac**. Valid formats are **AA:BB:CC:11:22:33**, **AA:*:CC:*:22:*** and **AA:??:CC:??:22:??**.
- **type=wildcard_ip**: Adds a wildcard entry to affect multiple devices at once via IP address. Requires parameter **ip**. Valid formats are **172.30.3.54**, **172.30.3.*** and **172.30.3.??**.

Following parameters are available:

- **return_userdata**: Can be used when adding a new autologon device. If value is 1, returns the autologon device id.
- **device_id**: When removing an autologon device, the device_id must be specified.
- **mac**: Used when adding a static device or a wildcard via MAC address device.
- **ip**: Used when adding a static device or a wildcard via IP address device.
- **desc**: For static autologon devices, this will become the name. For all other types, this is the description.
- **ticket_limit**: Ticket limit in **MB**. Can be used while adding a new autologon device. If the ticket limit is used up, the ticket will be revoked and recreated.
- **session_limit**: Session limit in **MB**. Can be used while adding a new autologon device. If the session limit is used up, the ticket will be logged off. If the device is still in the network, it will automatically be logged in again.
- **idle_timout**: Defines the time a ticket will be logged off when the device is completely inactive (turned off or not in the network anymore).
- **bw_in**: The download bandwidth in **Kbit/s** for new autologon entries.
- **bw_out**: The upload bandwidth in **Kbit/s** for new autologon entries.

cURL command to **add a new static autologon device** with the MAC address **AB:12:34:34:56:FF** and IP address **172.30.3.54**. The added device is called **Batch_Test**. The ticket limits for this device should be **1234 MB** as session limit, **5678 MB** as ticket limit and **20.000 Kbit/s** as download and upload bandwidth:

```
curl --insecure -o created_autologon_device_id.tmp --data "username=<USER>&password=<PASSWORD>
&action=autologon_devices&subaction=add_device&type=static&mac=AB:12:34:34:56:FF
&ip=172.30.3.54&desc=Batch_Test&ticket_limit=5678&session_limit=1234&bw_in=20000
&bw_out=20000" https://192.168.1.1/batch.php
```

cURL command to **add a single dynamic MAC device** with the MAC address **AB:12:34:34:56:FF**. The device description should be **Batch_Test** and the idle timout **30 minutes**. The ticket limits for this device should be **1234 MB** as session limit, **5678 MB** as ticket limit and **20.000 Kbit/s** as download and upload bandwidth:

```
curl --insecure -o created_autologon_device_id.tmp --data "username=<USER>&password=<PASSWORD>
&action=autologon_devices&subaction=add_device&type=dyn_mac&mac=AB:12:34:34:56:FF
&desc=Batch_Test&ticket_limit=5678&session_limit=1234&idle_timeout=30&bw_in=20000
&bw_out=20000" https://192.168.1.1/batch.php
```

cURL command to **add a single dynamic IP device** with the IP address **172.30.3.122**. The device description should be **Batch_Test** and the idle timout **30 minutes**. The ticket limits for this device should be **1234 MB** as session limit, **5678 MB** as ticket limit and **20.000 Kbit/s** as download and upload bandwidth:

```
curl --insecure -o created_autologon_device_id.tmp --data "username=<USER>&password=<PASSWORD>
&action=autologon_devices&subaction=add_device&type=dyn_ip&ip=172.30.3.122
&desc=Batch_Test&ticket_limit=5678&session_limit=1234&idle_timeout=30&bw_in=20000
&bw_out=20000" https://192.168.1.1/batch.php
```

cURL command to **add a wildcard MAC entry**, so that all devices which apply to the MAC address wildcard **AB:12:CD:34:*:F5** will be recognized as an autologon device and get an according ticket. The device description should be **Batch_Test** and the idle timout **30 minutes**. The ticket limits for this entry should be **1234 MB** as session limit, **5678 MB** as ticket limit and **20.000 Kbit/s** as download and upload bandwidth:

```
curl --insecure -o created_autologon_device_id.tmp --data "username=<USER>&password=<PASSWORD>
&action=autologon_devices&subaction=add_device&type=wildcard_mac&mac=AB:12:CD:34:*:F5
&desc=Batch_Test&ticket_limit=5678&session_limit=1234&idle_timeout=30&bw_in=20000
&bw_out=20000" https://192.168.1.1/batch.php
```

cURL command to **add a wildcard IP entry**, so that all devices which apply to the IP address wildcard **172.30.3.*** will be recognized as an autologon device and get an according ticket. The device description should be **Batch_Test** and the idle timout **30 minutes**. The ticket limits for this entry should be **1234 MB** as session limit, **5678 MB** as ticket limit and **20.000 Kbit/s** as download and upload bandwidth:

```
curl --insecure -o created_autologon_device_id.tmp --data "username=<USER>&password=<PASSWORD>
&action=autologon_devices&subaction=add_device&type=wildcard_ip&ip=172.30.3.*
&desc=Batch_Test&ticket_limit=5678&session_limit=1234&idle_timeout=30&bw_in=20000
&bw_out=20000" https://192.168.1.1/batch.php
```

cURL command to remove a **static autologon device**, in this example the id **111**.

```
curl --insecure -o log.tmp --data "username=<USER>&password=<PASSWORD>&action=autologon_devices
&subaction=remove_device&type=static&device_id=111" https://192.168.1.1/batch.php
```

cURL command to remove a **dynamic autologon device**, in this example the id **222**. The type **dynamic** must be used for all according types which includes **dyn_mac**, **dyn_ip**, **wildcard_mac** and **wildcard_ip**.

```
curl --insecure -o log.tmp --data "username=<USER>&password=<PASSWORD>&action=autologon_devices
&subaction=remove_device&type=dynamic&device_id=222" https://192.168.1.1/batch.php
```

To automatically revoke tickets together with an autologon entry, add the POST parameter **del_ticket=1**. Note that this does only work for single dynamic autologon entries but not for wildcard entries.

```
curl --insecure -o log.tmp --data "username=<USER>&password=<PASSWORD>&action=autologon_devices
&subaction=remove_device&type=dynamic&device_id=222&del_ticket=1" https://192.168.1.1/batch.php
```

Hint:

- A possibility to upload lists with multiple autologon devices is scheduled for one of the next updates.

7.1.7 System Commands

Reboot System

Supported since **17.0.11166-p11393**. Rebooting an IACBOX via Batch Access API by:

```
curl --insecure --data "username=<USER>&password=<PASSWORD>
&action=services&subaction=reboot" https://192.168.1.1/batch.php
```

Poweroff System

Supported since **17.0.11166-p11393**. In order to support UPS operation (automatic shutdown of the IACBOX in UPS operation), shutting down IACBOX can be done with this call:

```
curl --insecure --data "username=<USER>&password=<PASSWORD>
&action=services&subaction=halt" https://192.168.1.1/batch.php
```

Export Backup

Exports and returns a **system backup**.

```
curl --insecure -o backup.bkp --data "username=<USER>&password=<PASSWORD>
&backup=1" https://192.168.1.1/download_backup.php
```

Online Update

Start the **online update** on the selected IACBOX from outside.

```
curl --insecure -o update.tmp --data "username=<USER>&password=<PASSWORD>&action=online_update
&subaction=doupdate" https://192.168.1.1/batch.php
```

Download Update-Log Lists

Returns a list with all available logfiles from the **online update** (not the actual logs).

```
curl --insecure -o logfiles.tmp --data "username=<USER>&password=<PASSWORD>&action=online_update
&subaction=list_log_files" https://192.168.1.1/batch.php
```

Export Update Log

Exports a specific update log from all available log files on the IACBOX.

```
curl --nsecure -o logfiles.tmp --data "username=<USER>&password=<PASSWORD>&action=online_update
&subaction=fetch_log_files&logfiles=onlupdate_20160621132007.log;
onlupdate_20160621043642.log" https://192.168.1.1/batch.php
```

7.2 Central Services

This manual describes how to configure and use the module **Central Services**.

Hint:

- In order to use the module Central Services, it must be licensed separately.
- The **Central Services** and **Remote Access** will still work after the maintenance of a license is expired.
- Port 1194 TCP/UDP must be opened for outgoing connections on the firewall.

The module **Central Services** includes a set of functions for centralized management of distributed IACBOX systems. This way administrators can remotely access the **IACBOX WebAdmin** interface via the **my.iacbox Partner Portal** and manage the system. The included Central Services functions are extended continuously. The module *Central Services* is available with IACBOX version 3.10.4200 or newer.

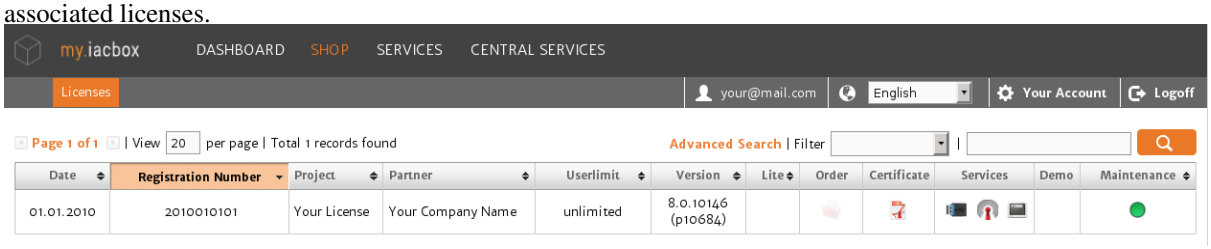
7.2.1 WebAdmin Configuration

Switch to the menu **Settings / Central Services** at the WebAdmin site and activate the service to connect to the **Central Server**.

As soon as the service has been started, a VPN connection to the *Central Server* will be established. The remote access via **my.iacbox Partner Portal** is only allowed once the option **Remote Access** is activated.

7.2.2 Management via my.iacbox

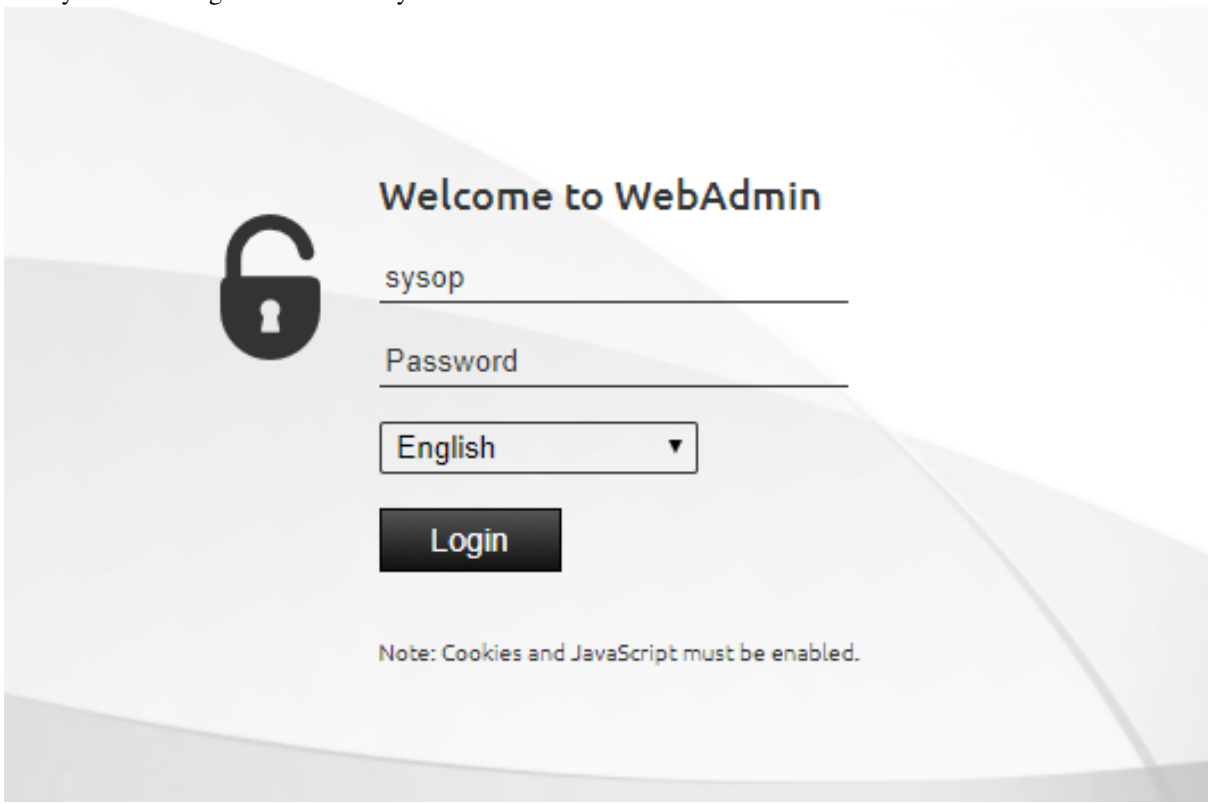
Log in to the **my.iacbox Partner Portal** and switch to the menu **Shop / Licenses** to obtain a listing of all of your associated licenses.



For licenses with activated *Central Services* the appropriate **icon** can be found in the *services column*. A grey icon means that there is no connection to the *Central Server*.

Right click on the license and then click on **Remote Access**. A new browser window opens and you will be connected directly to the WebAdmin interface of the corresponding IACBOX. Or you can also access the *Central Services* menu and establish a connection from there.

Now you can manage the IACBOX system via WebAdmin site as usual.



New in version v17.0: **Single-Sign-On (SSO)** is now supported - you don't have to log in on the IACBOX again.

This feature is enabled for all my.iacbox users in the group *Technical staff* or for any user you want to (just email us).

7.3 VPN Tunnel Configuration

This manual describes how to configure a **VPN tunnel** to access an external **OpenVPN server** from the IACBOX.

Hint:

- In order to use the module **VPN Tunnel**, it must be licensed separately.
- You require at least *OpenVPN version 2.3.4*, older versions are not supported.
- If you encounter problems, then consider to contact OpenVPN consulting services. The IACBOX support can not cover any configuration-related questions of your VPN server.

7.3.1 Creating the OpenVPN Server

You can obtain **OpenVPN** from the official project website: https://openvpn.net/index.php/open-source/documentation/howto.html

This website also covers detailed information on how to install OpenVPN on different operating systems like Linux, Windows, Mac OSX etc.

7.3.2 Generate Certificates & Keys

In order to use **OpenVPN** you need to generate *certificates* and *keys* for both, the **OpenVPN server** and the according **client (IACBOX)**. That for this manual describes how to generate self-signed TLS/SSL certificates.

There are many different tools to generate the certificates and keys. We recommend to use **Easy-RSA** which is a simple **OpenSSL** front-end to generate certificates and keys for both, **Windows** and **Linux**.

Easy-RSA can be obtained here: https://github.com/OpenVPN/easy-rsa

After extracting **Easy-RSA** switch to the directory **easy-rsa/2.0/** where you can find the different build scripts and edit the **vars** file. Based on the **parameters** in the **vars** file, the certificates and keys will be generated. Due to his, **edit/enter** the following important *parameters* in the **vars** file:

- **export KEY_SIZE=2048** The KEY_SIZE should be at least 2048. For enhanced security you can also increase the KEY_SIZE to 4096.
- **export CA_EXPIRE=3650** The CA_EXPIRE defines in how many days the root CA key will expire. For some Eays-RSA installations this is set to 1 year as default so make sure to check this vlaue.
- **export KEY_EXPIRE=3650** The KEY_EXPIRE defines in how many days the created certificates will expire. For some Easy-RSA installations this is set to 1 year as default so make sure to check this vlaue.
- **export KEY_COUNTRY** The **two letter ISO code** for the country where your organization is located. For example **us** or **gb**.
- **export KEY_PROVINCE** The state/region where your organization is located. This should not be abbreviated.
- **export KEY_CITY** The city where your organization is located.
- **export KEY_ORG** The legal name of your organization. This should not be abbreviated and should include suffixes such as *Inc*, *Corp* or *LLC*.
- **export KEY_EMAIL** An email address used to contact your organization.
- **export KEY_OU** The division of your organization handling the certificate.

- **export KEY_NAME** The name of the generated key. For example **iacbox**.

Save the changes you made for the **vars** file. Note that the commands below refer to a Linux system. First run the following commands to initialize the public key infrastructure (**PKI**):

```
. ./vars
./build-ca
```

In order to generate the certificate and key for the OpenVPN server, run the following command. As **server-name** you can select an own name, for example **vpnserver**.

```
./build-key-server server-name
```

Continue with **Enter** and confirm following questions with **y**:

```
Sign the certificate? [y/n]: y
1 out of 1 certificate request certified, commit? [y/n]: y
```

The next step is to generate the certificate and key for the client. Therefore run the following command. as **client-name** you can select an own name, for example **iacbox01**.

```
./build-key client-name
```

Again continue with **Enter** and confirm following questions with **y**:

```
Sign the certificate? [y/n]: y
1 out of 1 certificate requests certified, commit? [y/n]: y
```

The last step is to generate a **Diffie Hellman prime**.

```
./build-dh
```

By now the following files should have been created:

- **ca.crt** - Needed by the *server* and *all clients*, serves as *Root CA certificate*.
- **ca.key** - Needed by the *key signing machine* only, serves as *Root CA key*.
- **dh{n}.pem** - Needed by the *server* only, serves as *Diffie Hellman* prime.
- **vpnserver.crt** - Needed by the *server* only, serves as *server certificate*.
- **vpnserver.key** - Needed by the *server* only, serves as *server key*.
- **iacbox1.crt** - Needed by the *IACBOX client 1*, serves as *client certificate* for just this client.
- **iacbox1.key** - Needed by the *IACBOX client 1*, serves as *client key* for just this client.

For a more secure version you can **optionally** use an TLS-auth key. Generate it with:

```
openvpn --genkey --secret ta.key
```

All the generated certificates and keys are stored in the **keys** directory. In order to use them with *OpenVPN*, copy the **keys** directory to the *OpenVPN* directory where the OpenVPN server daemon runs. On linux this tends to be **/etc/openvpn** and on windows it is usually **C:Program FilesOpenVPNconfig**.

Hint:

- Note that usually **Easy-RSA** sets the file permissions automatically, keep the **.key** files secure and protected.

7.3.3 OpenVPN Server Configuration

Open the OpenVPN configuration file and edit/check the following parameters:

- **port 1194** The OpenVPN default port is set to 1194. If there is a firewall in between the OpenVPN server and the clients, be sure to allow the configured port for input, forward and output.

- **mode server** If the mode is not set to server per default, change it.
- **ca keys/ca.crt** Enter the directory where the CA-file can be found.
- **key keys/vpnserver.key** Enter the directory where the server key file can be found.
- **cert keys/vpnserver.crt** Enter the directory where the server certificate file can be found.
- **dh keys/dh2048.pem** Enter the directory where the Diffie Helmann file can be found.
- **ifconfig 172.17.130.254 172.17.130.253** In this example, the 172.17.130.254 is the IP-address for the tun1 interface of the OpenVPN server and the 172.17.130.253 IP-address is used for point-to-point connections.
- **ifconfig-pool 172.17.130.1 172.17.130.250** This parameter defines the DHCP pool within OpenVPN clients will receive an IP-address. Please note that the IP-address range 172.17.0.0/17 should not be used for the ifconfig-pool. This IP-address range is already used for other functions of IACBOX.
- **route 172.17.130.0 255.255.255.0** This parameter sets a route to the tunnel network 172.17.130.0/24. This route is necessary and needs to be set.
- **push route 10.5.5.0 255.255.255.0** This parameter pushes the defined route to the client (IACBOX). Due to this, the client (IACBOX) knows that the network 10.5.5.0/24 can be reached via tunnel default gateway 172.17.130.254.
- **client-config-dir ccd** This directory should have been pre-created in the default directory where the OpenVPN server daemon runs. When a new client connects to the OpenVPN server, the daemon will check this directory for a file which matches the common name of the connecting client. If a matching file is found, it will be read and processed for additional configuration file directives to be applied to the named client.
- **tls-auth keys/ta.key 0** This **optional** parameter can be set if an TLS-auth key is used. **Attention:** the key-direction field [0/1] is a three-state field! If it is set, it has to be set on the server and the IACBOX, or has to be left out on both sides. It's more secure to use the key direction. On the server-side it has to be **0**, on the client **1**.

This means that if there is a client (IACBOX) with the common name **iacbox1** (or any other common name like for example **iacbox1.vpn**) you need to create a new file named "iacbox1âĂİ ("iacbox1.vpnâĂİ). In this file you can define specific parameters which will only be applied to the corresponding client (IACBOX).

For example:

```
ifconfig-push 172.17.130.98 172.17.130.254
```

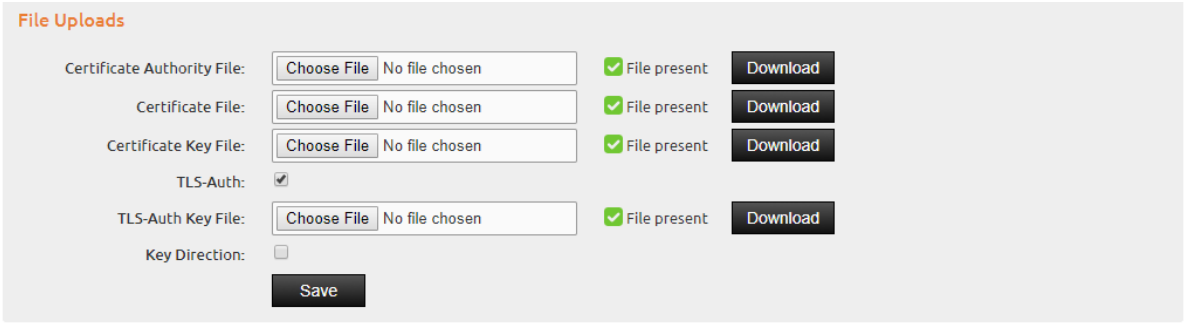
This parameter assigns the fixed IP-address **172.17.130.98** to the client (IACBOX) and sets the clients default gateway to **172.17.130.254**.

```
iroute 172.29.0.0 255.255.0.0
```

This parameter sets a client specific route on the OpenVPN server. In this example, a route to the Surf-LAN network of the corresponding client (IACBOX) is set. It is highly recommend to create a seperate file in the **ccd** directory for each client (IACBOX) connected to the OpenVPN server.

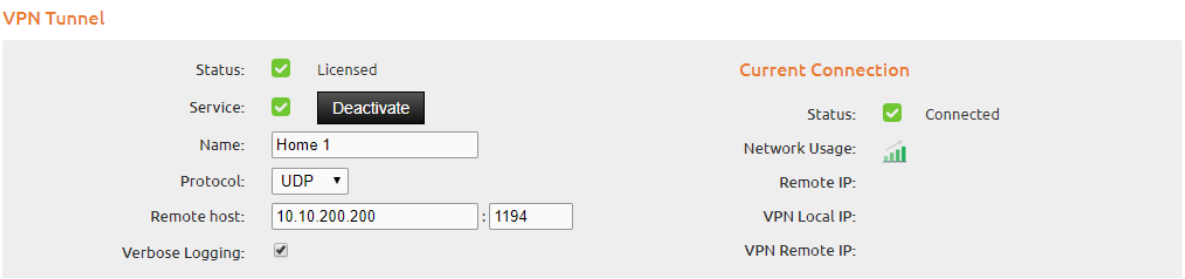
7.3.4 IACBOX Configuration

Activate the VPN tunnel in the WebAdmin menu **Modules / VPN Tunnel**. First of all, you need to upload the certificate and key files on the IACBOX. You need to upload the **ca.crt**, **client1(iacbox1)**.crt and **client1(iacbox1).key** to the system.



If the **optional** TLS-Auth function is active on the server side (have a look at the description of the server config above) the **ta.key** file has to be uploaded here too. If the *Key Direction* is used on the server side then the checkbox has to be active here too.

Enter a name for the VPN tunnel, the remote host or IP-address and the protocol + port according to your OpenVPN configuration (default = 1194/udp). If the connection was successful, the VPN local IP and VPN remote IP will be displayed on the right.



7.3.5 Routing Protection

- **Protect from Surf-LAN** If this is activated, all connections to the VPN tunnel from the Surf-LAN will be blocked.
- **Protect from Management-LAN** If this is activated, all connections to the VPN tunnel from the Management-LAN will be blocked.
- **Protect routing from tunnel** If this is activated, all connections from the VPN tunnel to the IACBOX Surf-LAN, Management-LAN and/or Office-LAN will be blocked.

However there are certain configurations where you need to disable the protection. For example: You want to allow connections from the VPN tunnel to the Surf-LAN. Therefore you need to define a route to the Surf-LAN on the OpenVPN server. You can do this by **editing** the **according file** for the client (IACBOX) in the **ccd** directory of the OpenVPN server and **adding** the route with the parameter **iroute 172.29.0.0 255.255.0.0**. In addition, you need to disable **Protect routing from tunnel** at the VPN tunnel configuration on the IACBOX.

7.3.6 Access to Services

If the client (IACBOX) is connected, you can access the different IACBOX services from the tunnel. If **WebAdmin Access** is enabled, it is possible to connect from the VPN tunnel to the WebAdmin of the IACBOX by using it's tunnel IP-address (e.g. **https://172.17.130.98**).

In addition to the default access services, it is also possible to grant access to custom ports. For example:

- **udp:53** → to see if the DNS works
- **tcp:8080** → check if the proxy server is running

If you want to add multiple ports use blanks as delimiter (e.g. udp:53 tcp:8080).

CHAPTER

EIGHT

LOGON METHODS

8.1 Autologon Devices

This manual describes the function and configuration of **Autologon Devices**.

Hint:

- Important devices should be added as **Static Single Device**.
- Verify that there are *no duplicated entries* in the autologon list.
- The IACBOX does re-assign unused IP-addresses only if the DHCP-range is full. Verify that the Surf-LAN configuration does cover the network requirements.

With the function **Autologon** you can add devices by their **IP** or **MAC address** so that they automatically get logged in and require no further user interaction.

Every configuration, except for the **Static Single Devices**, will automatically log in a device, no matter which kind of traffic was send. There are 5 different modes for **Autologon** devices.

8.1.1 Configuration

The configuration of Autologon devices is quite easy and straightforward. All the management is done under **Modules/Autologon Devices**. You can enter the corresponding data manually or you assign devices directly from the list **Clients Online**. Subsequent editing is possible at all times.

Hint:

- Note that after every change you will need to perform a **Service Restart**.
- If you are about to add or modify an autologon entry, make sure that all tickets of the device are being revoked beforehand at **Tickets / Manage**.

Static Single Device

This autologon mode does work like the common autologon mode for every IACBOX since version 4.0.6562.

You need the **IP-address** and the **MAC-address** of the device you want to add. By adding the device, a **static DHCP lease** will be generated automatically.

This mode is the most secure one, but the device will permanently occupy a license slot.

Dynamic Single MAC

This Autologon mode is based on the **MAC-address** of a device. If the device is active in the Surf-LAN, it will automatically get online without the need to access the logon-page of the IACBOX.

Dynamic Single IP

Just like **Dynamic Single MAC** you can add an **IP-address** for a device, which then will be automatically set online as soon as any traffic is recognized by the IACBOX.

Dynamic Wildcard MAC

With Wildcards you can specify a *MAC address* by using **wildcards**. As soon as a device matches the wildcard and is being recognized by the IACBOX, it will get online. This mode is very helpful if you want to add alot of devices with similar **MAC addresses** (e.g. Access Points).

```
Example:
A:11:22:*:*:*
```

Dynamic Wildcard IP

Similar to **Dynamic Wildcard MAC** you can assign a **range of IP-addresses** by using a wildcard.

```
Example:
172.30.110.* - 172.30.110.1 - 172.30.110.254
172.30.110.?? - 172.30.110.10 - 172.30.110.99
```

Using Lists

Uploading and Downloading IP or MAC address lists is also possible on the **Autologon Devices** configuration page for mass import or export operations.

8.2 Email Login

This manual describes how to provide an email login for your guests.

Hint:

- This module became **free** for all IACBOXes with valid **Software Maintenance** as of 1st december 2016.
- The login by using an *Email* is meant to be **free**, guests can not be charged.
- In order to configure the *Email* login a valid **SMTP configuration** (**Settings / Network**) is **required**.
- The **Email** login requires at least one valid ticket template, configured as 0€ (free).

8.2.1 How it works

On the *Client Login Page* of the IACBOX, guests will notice an **Email** icon. This icon can be used to input an *Email address*. After confirming the email address, the IACBOX will send an email to it. The email will contain the credentials in order to log in, as well as a hyperlink which enables guests to log in immediately. Note that after clicking the email icon, guests will be set online for a configurable amount of time. This can be used to also check *online email services* via HTTP.

Example content of an email:

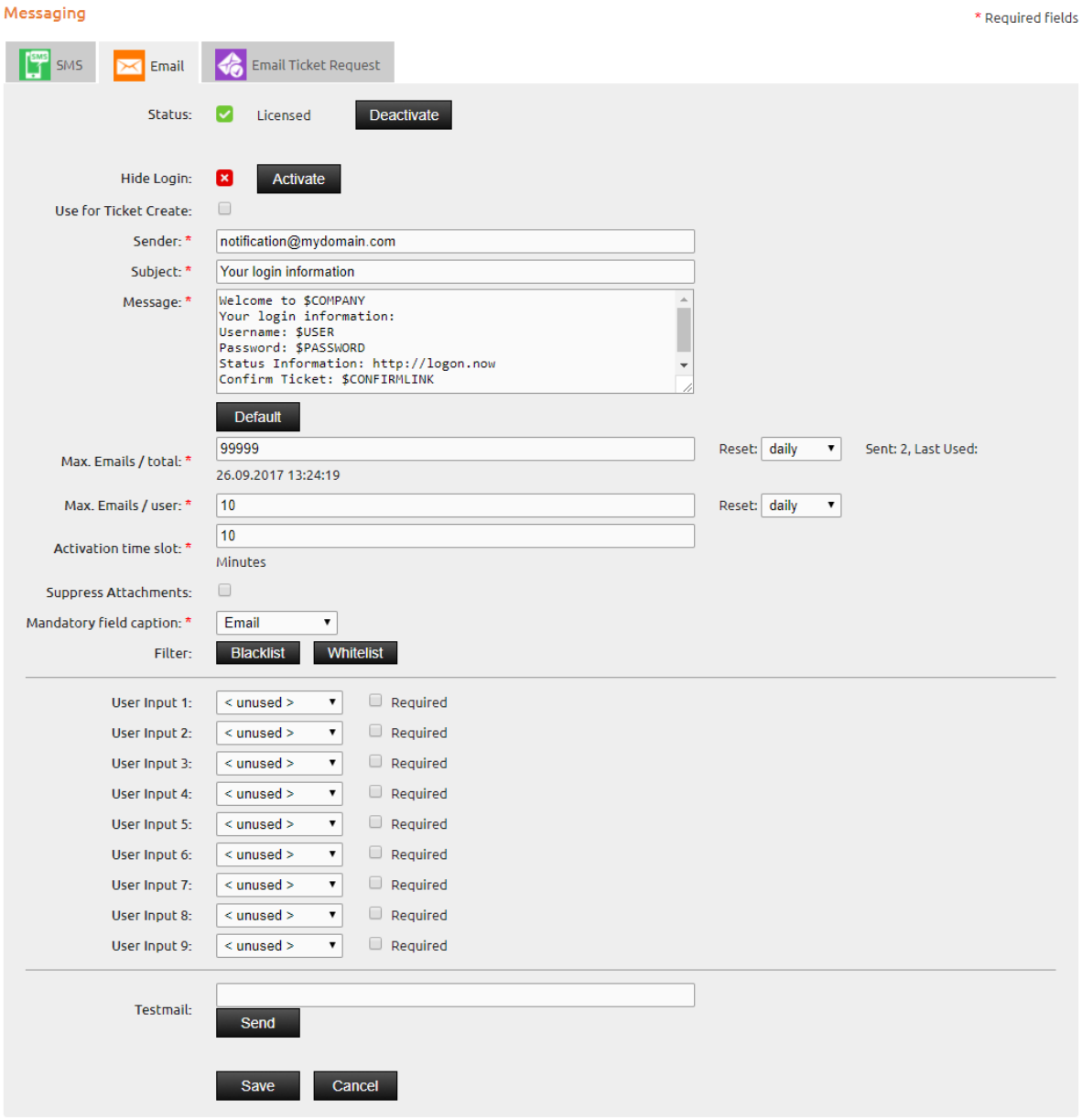
```
Welcome to $COMPANY

Your login information:
Username: $USER
Password: $PASSWORD
```

```
Status Information: http://logon.now
Confirm Ticket: $CONFIRMLINK
```

8.2.2 Configuration

The configuration of the **Email module** can be found in the *WebAdmin* menu **Modules / Interfaces**.



The explicit description of the input fields can be found on the **Help Page** of the *WebAdmin* menu in the right upper corner.

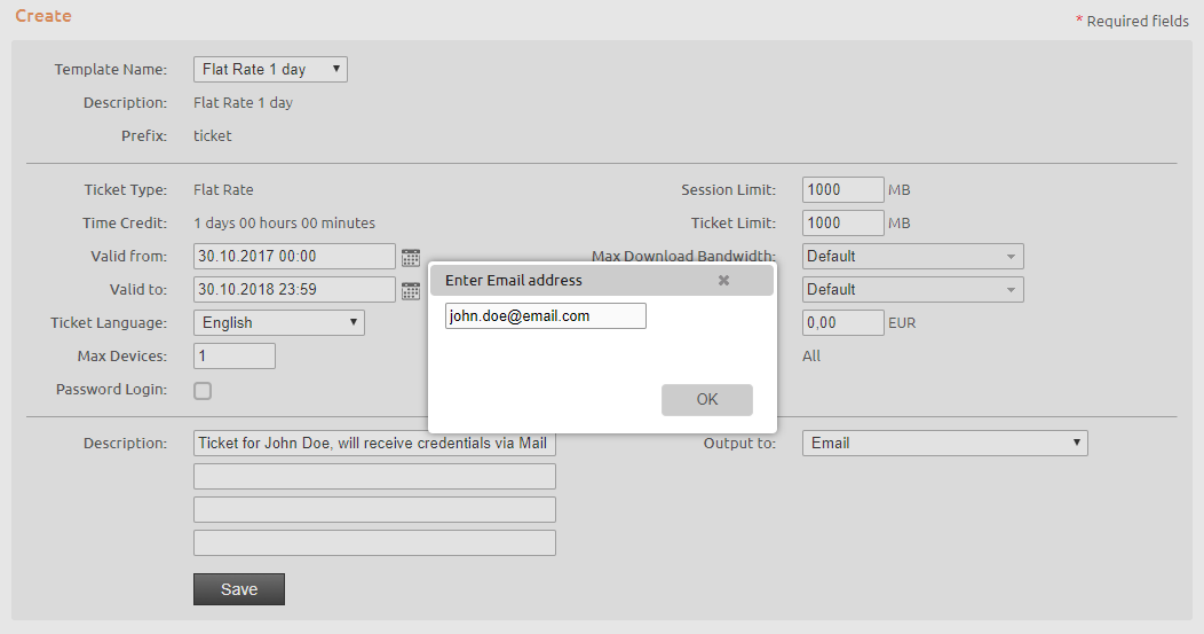
The *Email configuration* allows you to add some restrictions to the usage of this module. Also you can modify the message which is being sent to guests via email.

Hint:

- The **Email** login requires at least one valid ticket template, configured as 0€ (free).

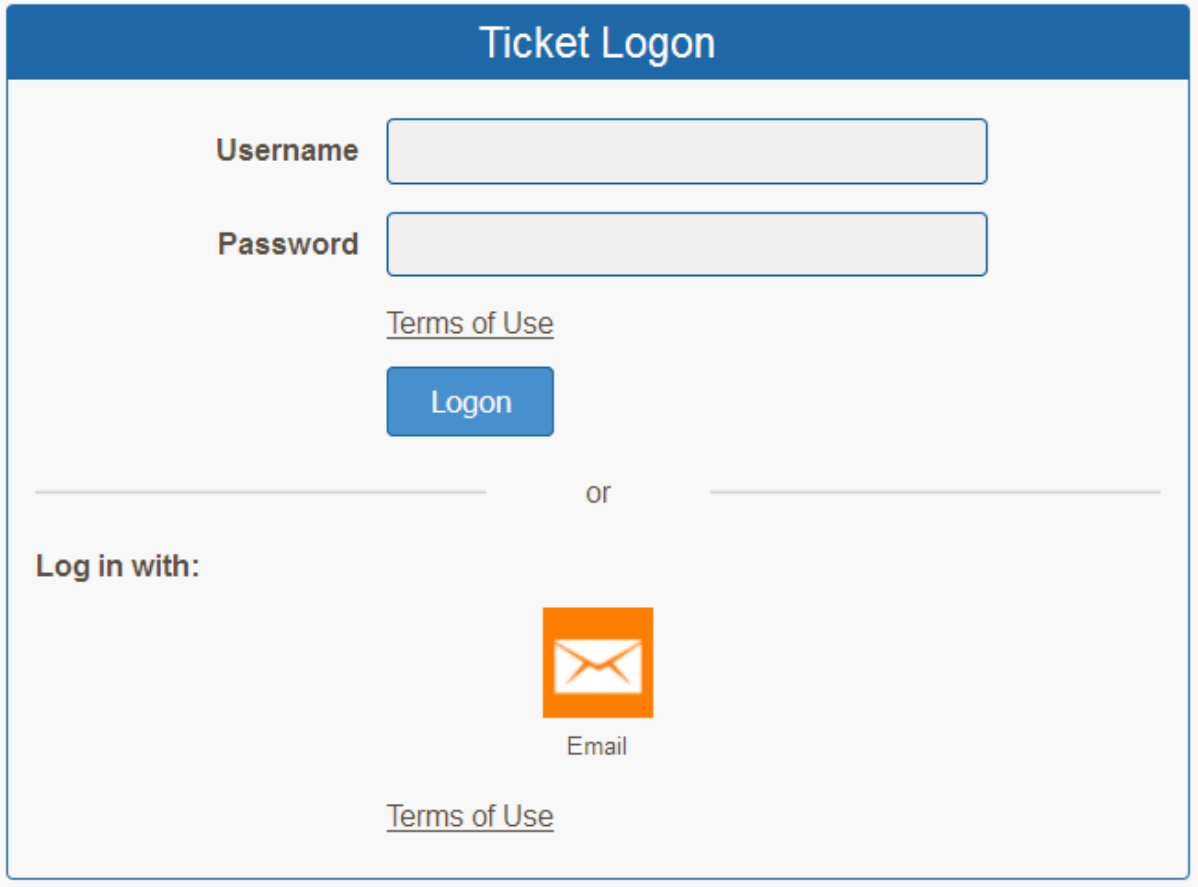
8.2.3 WebAdmin Tickets with Email

If the option **Use for Ticket Create** is active in the *Email configuration*, ticket data (**username**, **password**) of newly created tickets from *WebAdmin* **Tickets / Create** can be directly sent via Email.

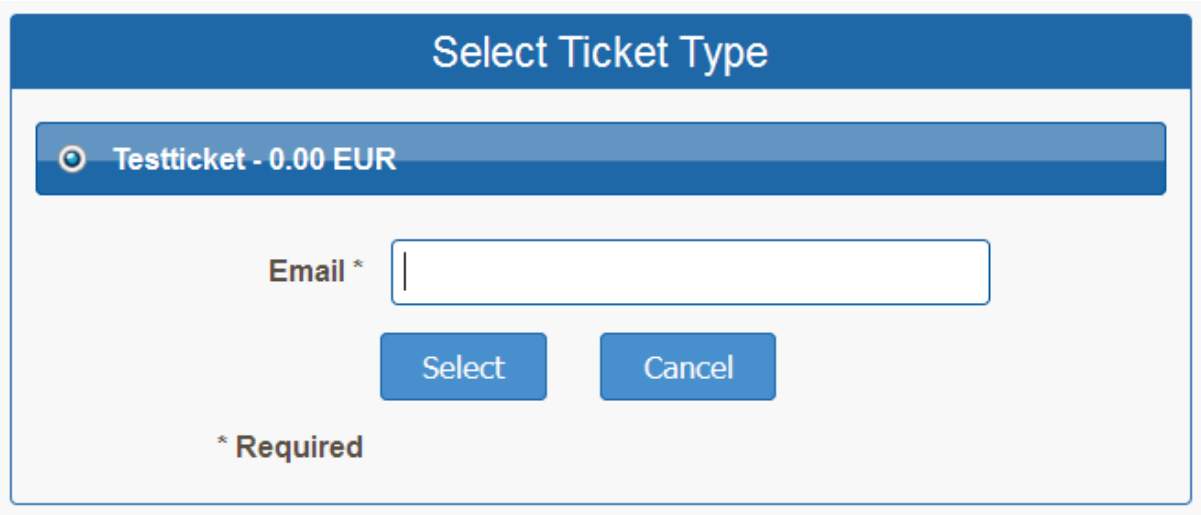


8.2.4 Client Logon Page

After all settings have been made, the result can be seen on the customer logon page.



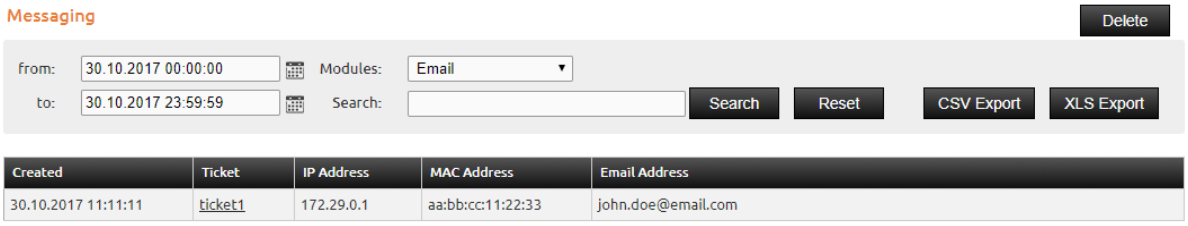
By clicking on the **Email icon**, guests can now enter their *Email address* to continue with the login.



Soon the guest will receive an Email which contains the ticket credentials as well as a hyperlink which enables guests to log in at once.

8.2.5 Stored Data

In the *WebAdmin* menu **Reporting / Messaging** you can always check and also download archived user data.



8.3 External Authentication

This manual describes how to configure the IACBOX in order to use various backends for guest authentication and also for the *WebAdmin* interface.

Hint:

- The **External Authentication** module must be licensed separately.
- Configured backends can also be used to authenticate users for the *WebAdmin*.
- In order to create *Surf-Tickets* with configured backends, a **ticket template** must be assigned to be used with the module **Authentication**. This will also be explained in this manual.

8.3.1 General

The module **External Authentication** allows *Surf-LAN* users to use the default **Ticket Login** box on the *Customer Logon Page* to authenticate with credentials, which are available on external sources. The supported authentication methods are:

- Active Directory
- LDAP
- MSSQL

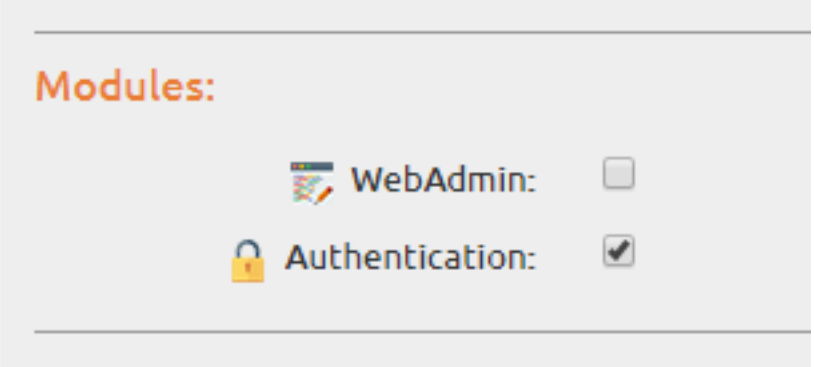
- MySQL
- PostgreSQL
- Radius
- iPass
- Local Database

Hint:

- The **Local Database** is always available, even if the module **External Authentication** is not licensed. The usage of the *Local Database* relates to the *Local Users* which can be created in the *WebAdmin* menu **Tickets / Users**.

8.3.2 Define a Ticket Template

If the **External Authentication** is used for the *Client Logon Page*, then a **Ticket Template** must be configured for this module. After **activating** the **External Authentication** in the *WebAdmin* menu **Modules / Authentication**, navigate to **Tickets / Templates**. Select a desired template to edit or create a new one for this case and configure the restrictions according to your requirements. Before saving the *Ticket Template*, activate the checkbox for **Authentication**, which can be found in the section **Modules**.



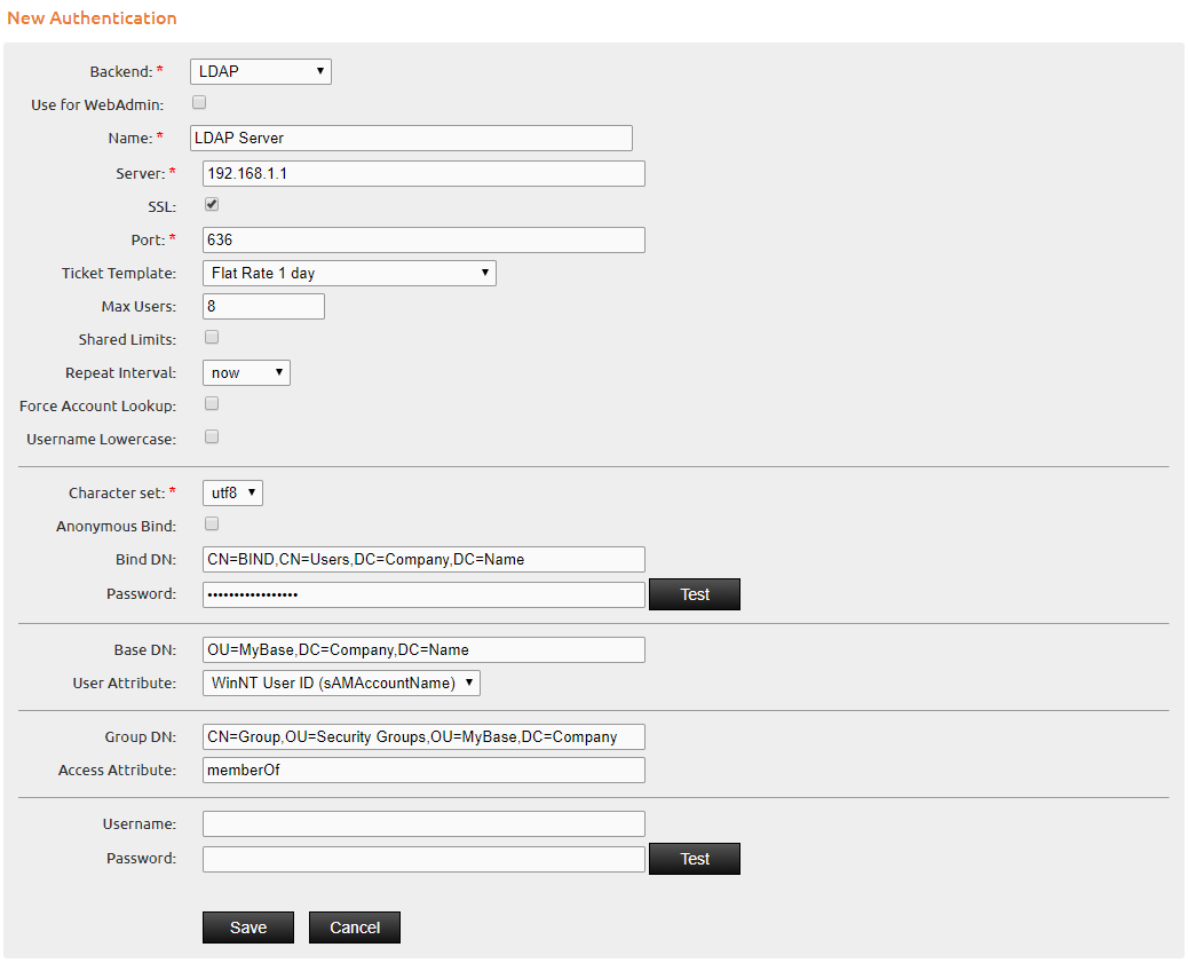
8.3.3 Activate User Template

If the **External Authentication** is being used to authenticate *WebAdmin* users, a *User Template* must be activated for this module. Therefore switch to the *WebAdmin* menu **System / Manage User** and create a new **User Group** which can be used for *WebAdmin* users which do authenticate via the **External Authentication**.

It's now possible to configure any **External Authentication** as **Use for WebAdmin**, then the **User Group** called **ExtAuth** can be assigned to it.

8.3.4 Active Directory / LDAP

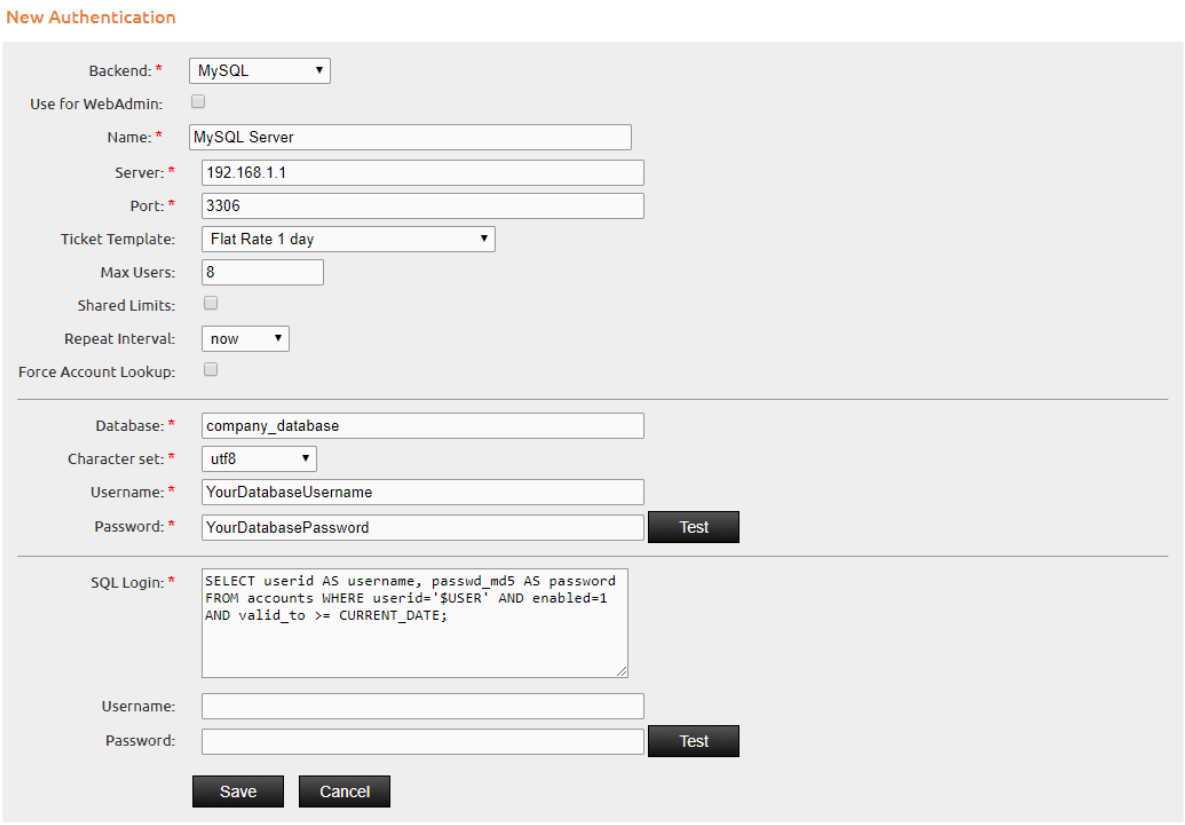
As explained further up in this manual, a ticket template must be configured. The remaining configuration of this module should be pretty much self-explaining.



An explicit explanation of the input fields can also be found in the **help menu of the WebAdmin**:

8.3.5 MySQL / MSSQL / PostgreSQL

The **SQL backends** of the **External Authentication** can use custom SQL statements to authenticate users on either the *Customer Logon Page* or the *WebAdmin* of the IACBOX.



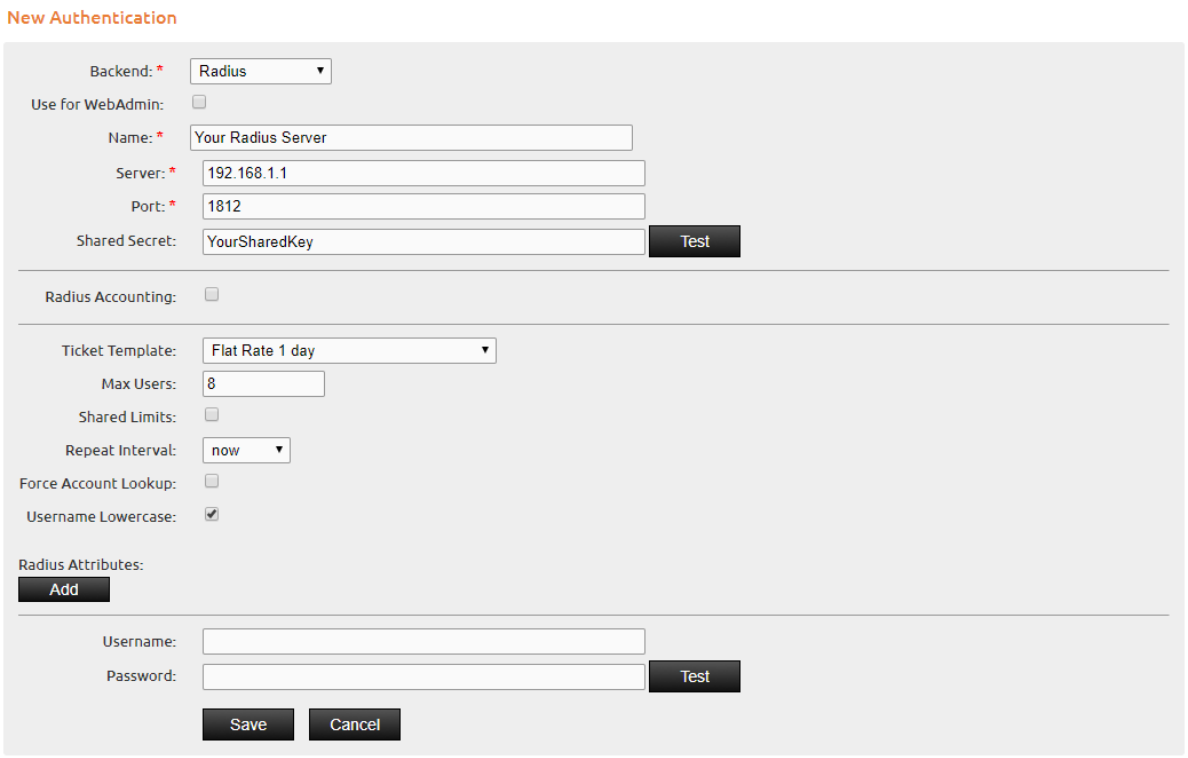
In this screenshot the SQL query is not only interpreting **user_id** as **username** and **passwd_md5** as **password**, but also checking the table columns for the boolean return value of **enabled=1** and **valid_to >= CURRENT_DATE**.

Hint:

- Note that the external SQL server must be able to understand variables like *CURRENT_DATE*. If in doubt, check the according SQL documentation of your server or provider.

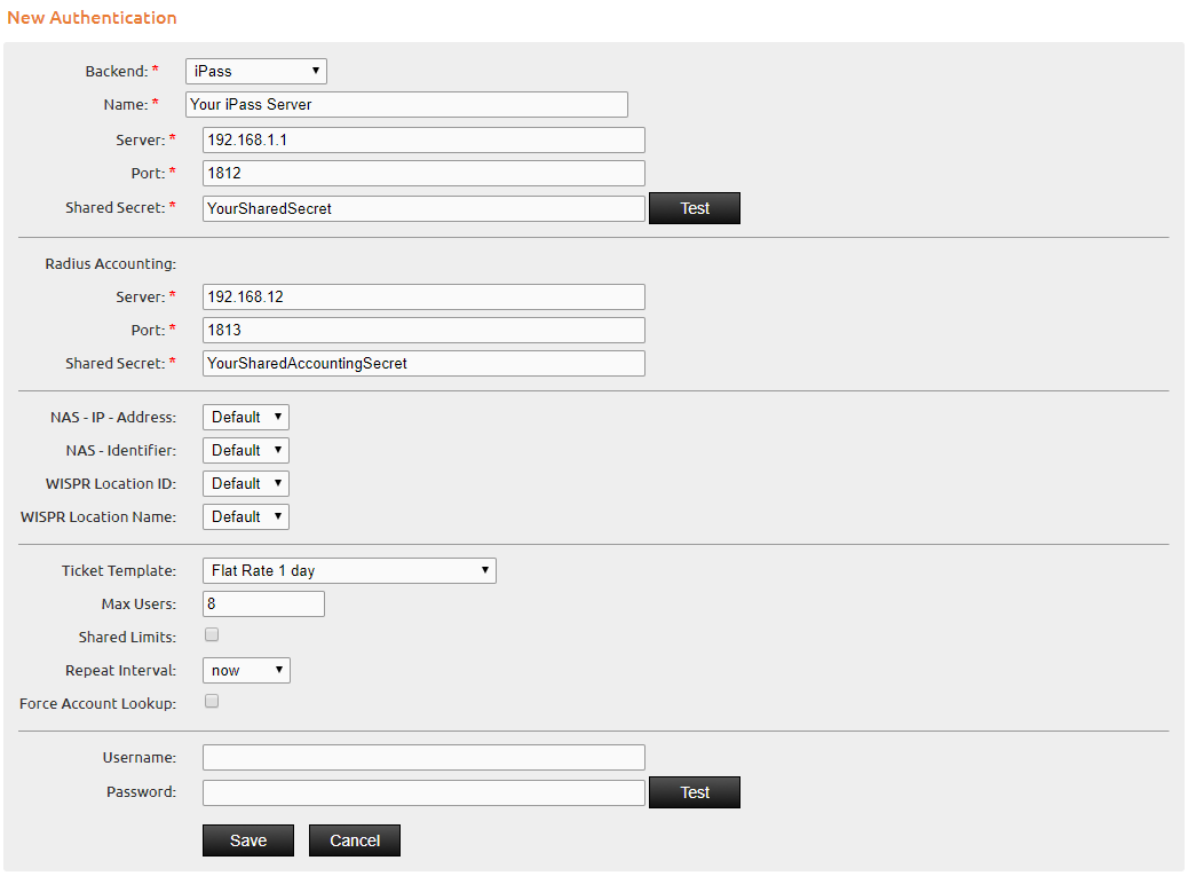
8.3.6 Radius

The **External Authentication** with **Radius** can be used for authentication on the client logon page and on the *WebAdmin* login page. Depending if used for *Client Logon Page* or for the *WebAdmin* login page, the configuration may slightly look different.



8.3.7 iPass

The **External Authentication** with **iPass** can be used for authentication on the client logon page and on the *WebAdmin* login page. Depending if used for *Client Logon Page* or for the *WebAdmin* login page, the configuration may slightly look different.



8.4 Facebook Login

This manual describes how to set up a **Facebook Developer Account** and to configure a new **Facebook App** with it, so that guests can authenticate and log in on the IACBOX by using their Facebook account.

Hint:

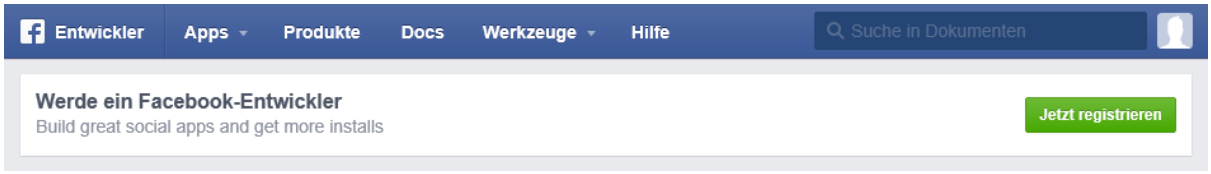
- This module became **free** for all IACBOXes with valid **Software Maintenance** as of 1st december 2016.
- The login with *Facebook Credentials* is meant to be **free**, guests can not be charged.
- To setup a *Facebook Login* for your guests, you will need to create a *Facebook Developer Account* and configure a *Facebook App* with it.
- The *Facebook login* requires at least one valid ticket template, configured as 0€ (free).
- In case you use a *custom certificate* on your IACBOX, pay close attention to change all *redirect URLs* according to your custom hostname.

8.4.1 Developer Account

In order to create a *Facebook app*, you will require a *Facebook Developer Account*. Currently you can transform your regular *Facebook Account* into an Facebook Developer Account on this URL:

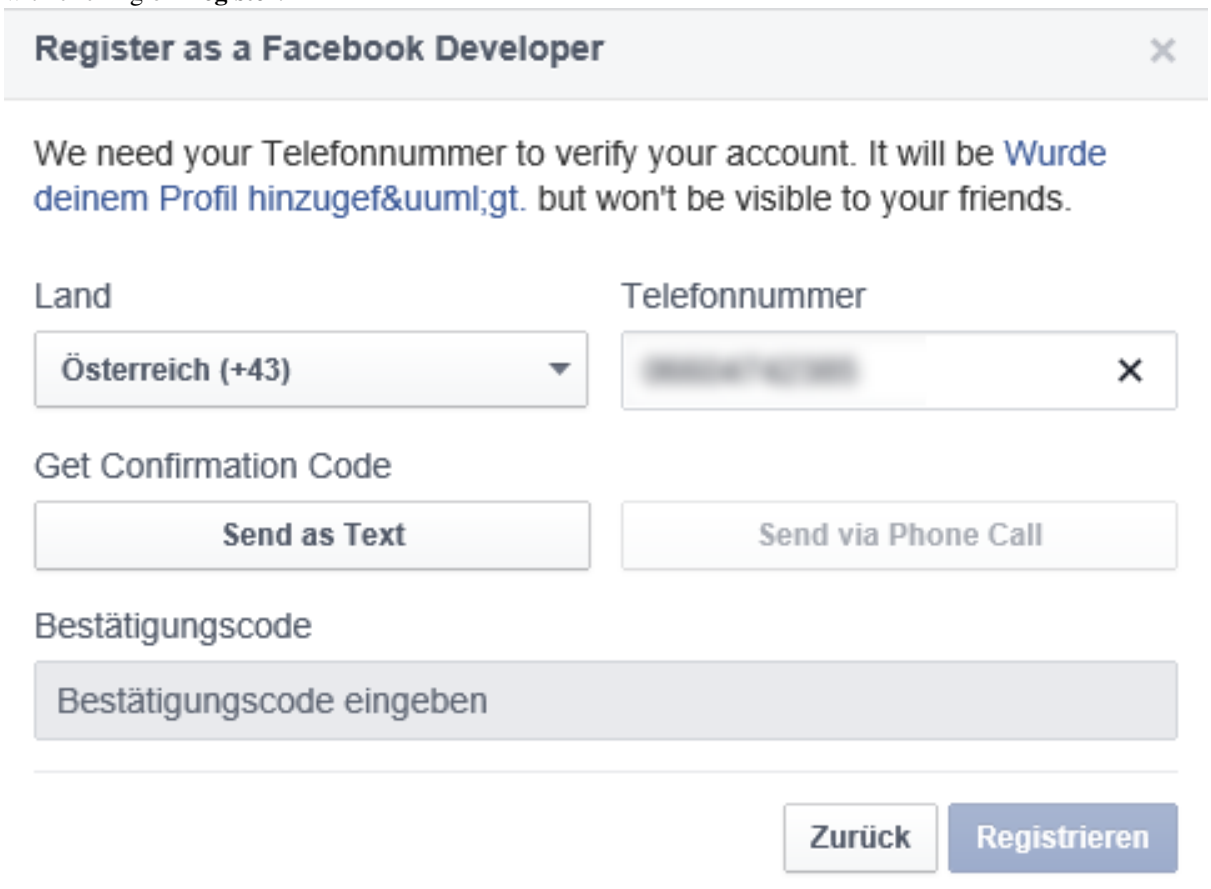
`https://developers.facebook.com/apps`

You will be asked to become a Facebook Developer:

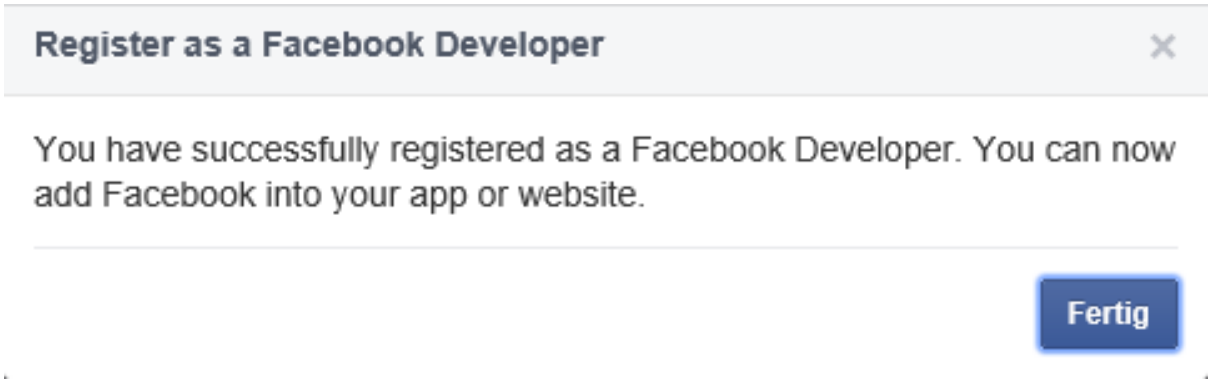


Click on **Register now** to proceed with the activation of your *Facebook Developer* account. Read the **privacy policy**, then use the switch to verify your agreement. Then continue with **Next**.

Now you will need to verify either by **SMS** or by a **phone call**. After you've got your confirmation code, continue with clicking on **Register**.



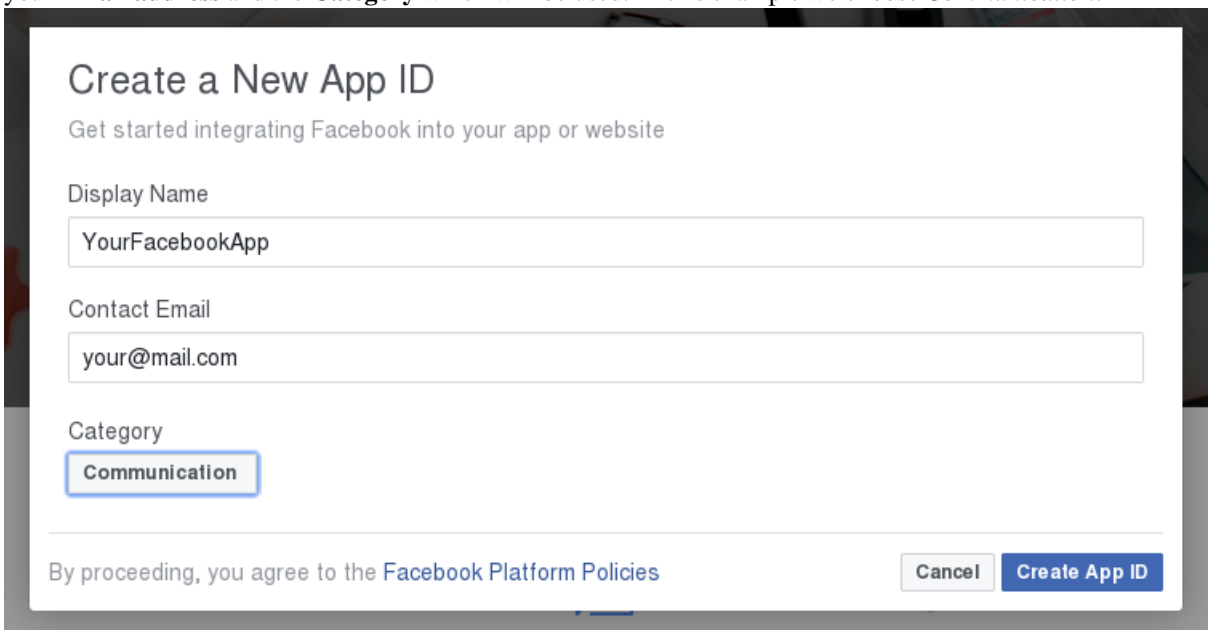
If everything went fine you should see the following message.



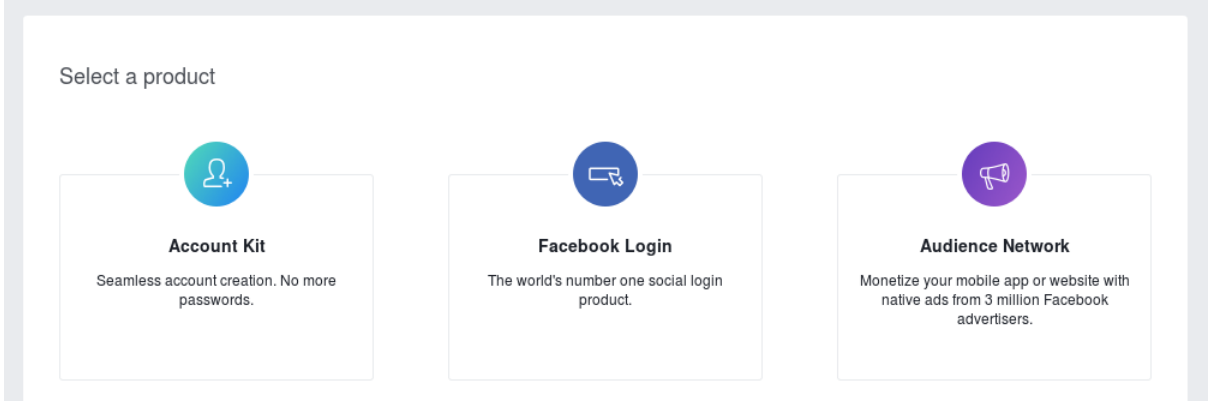
Your developer account is now ready to use.

8.4.2 Creating the Facebook App

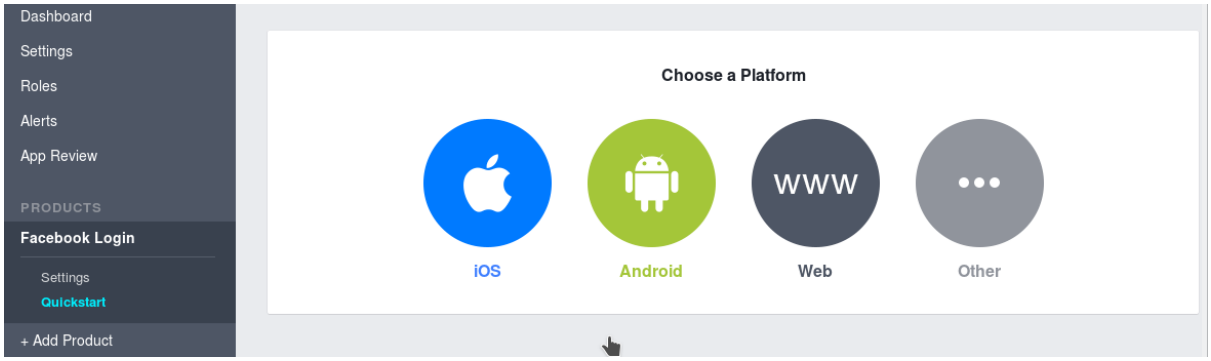
Now create a new **application** by clicking on **My apps**, **Add a new App**. Enter a **name** for your app, as well as your **Email address** and the **Category** which will be used. In this example we choose *Communication*.



Now your new *app* is being created. After this was done, you are now in the configuration menu of your new *app*. Start the configuration by clicking on **+ Add Product** in the navigation on the left side and select **Facebook Login**.



After **Facebook Login** was added, the following screen will appear:

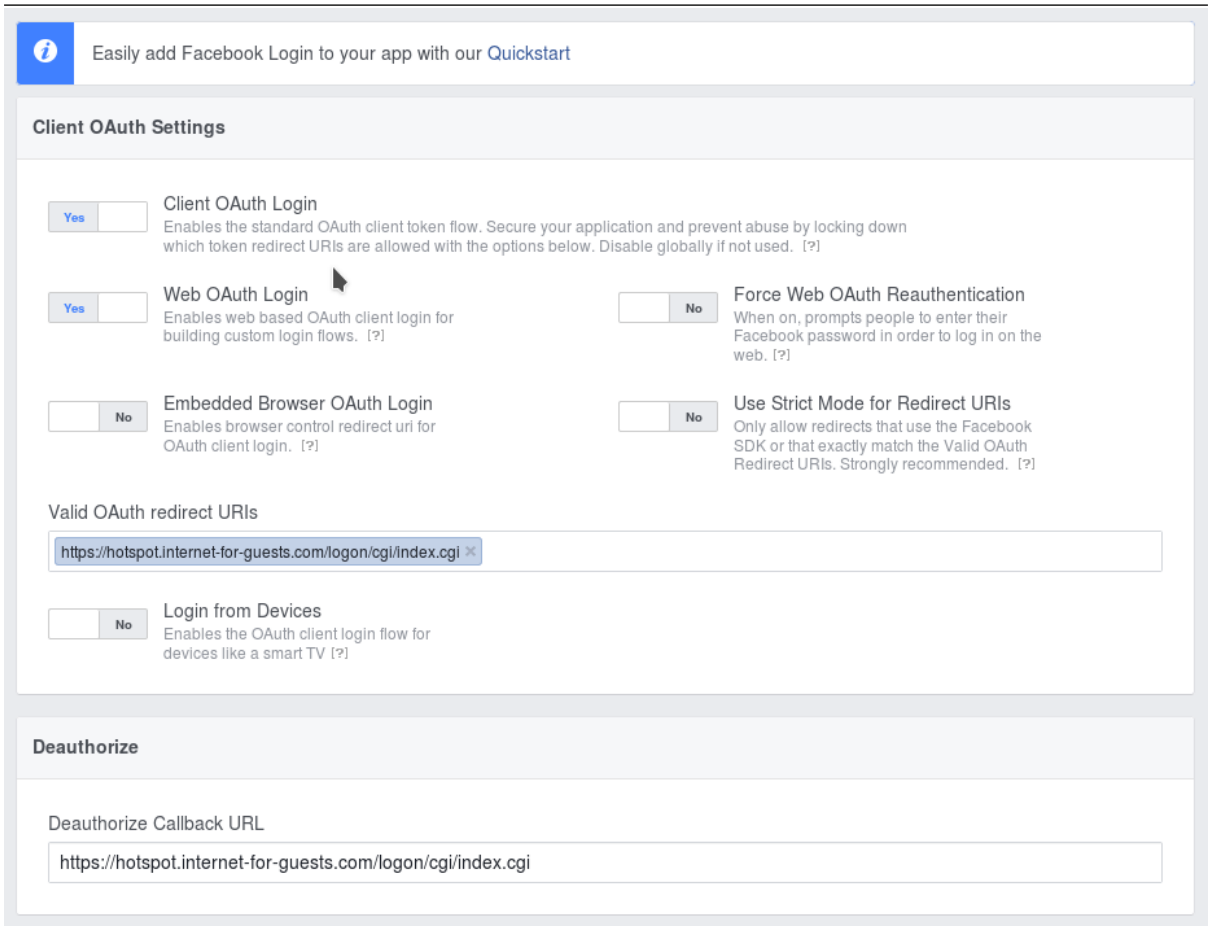


It is not neccessary to *Choose a Platform* here, instead proceed by swtiching directly into the new **Settings** menu, which can be found directly over the **Quickstart** navigation menu entry. This will open the **Client OAuth Settings**. Proceed by copying the redirect URI and the callback URL:

```
https://hotspot.internet-for-guests.com/logon/cgi/index.cgi
```

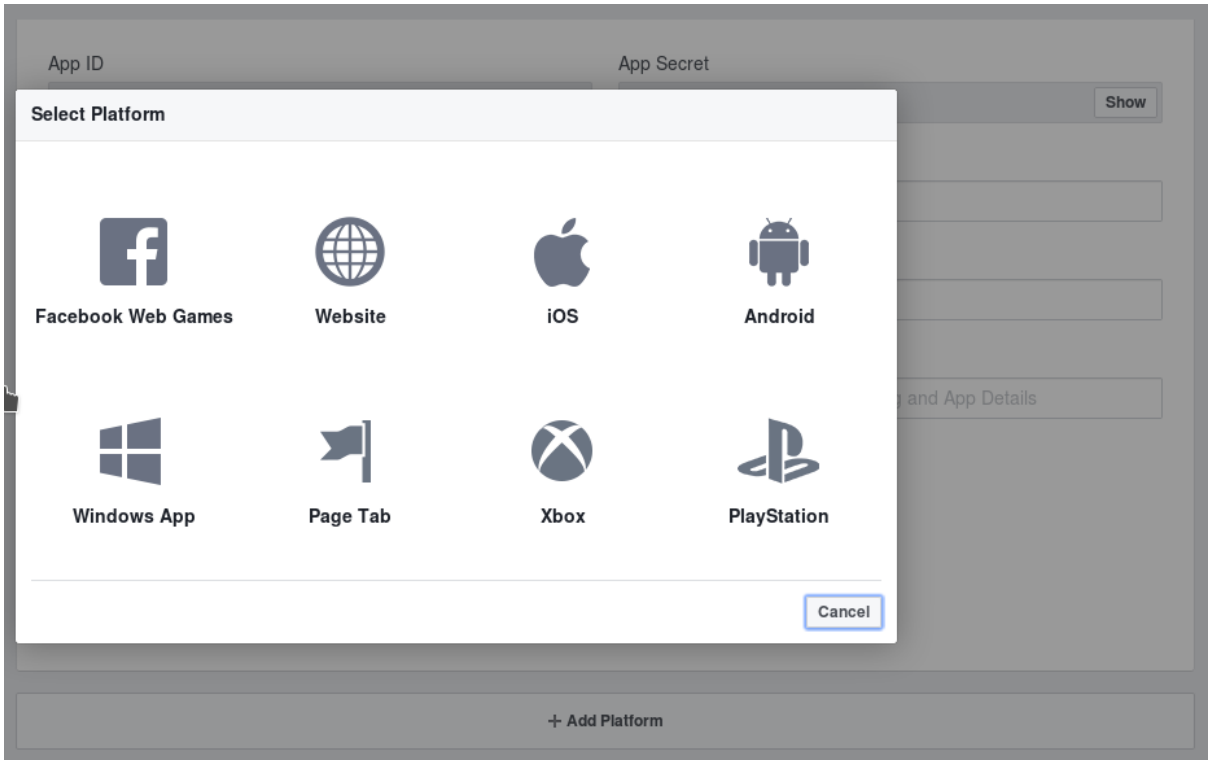
Hint:

- In case you use a *custom certificate* on your IACBOX, pay close attention to change all URLs according to your custom hostname.



After this is done, proceed by clicking on **Save Changes** which can be found on the right bottom of the site.

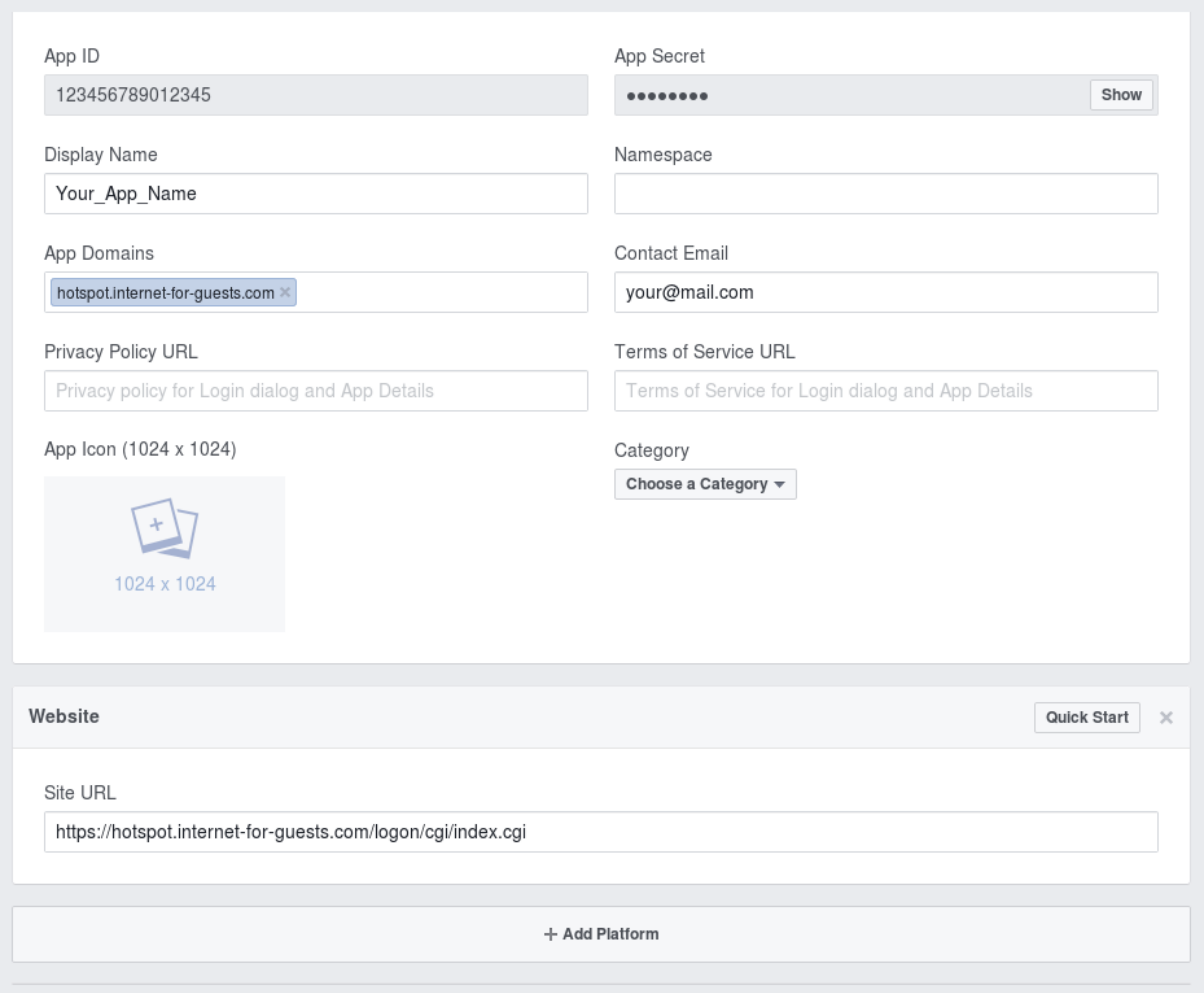
Now click on the top **Settings** entry in the navigation menu. This will bring you to the basic settings of your **app**. In here, click on **+ Add Platform** and then select **Website**.



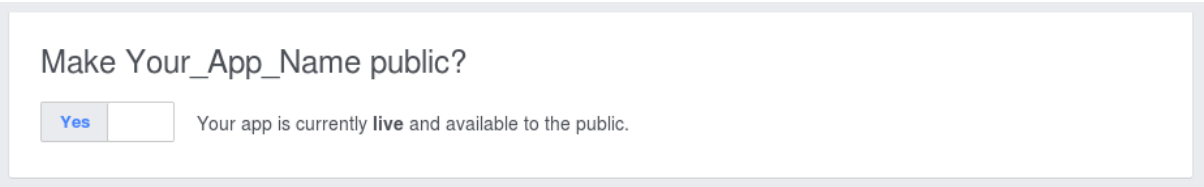
In the next step fill out your **Contact Email**, the **Display Name** of your app and the **Category** selection. Then fill out the fields **App Domains** and **Site URL** according to the following screenshot.

Hint:

- In case you use a *custom certificate* on your IACBOX, pay close attention to change all URLs according to your custom hostname.



Before proceeding to the next step ensure that **Save Changes** was clicked. After this was done, navigate to the menu entry **App Review**. As a final step the Facebook app needs to be moved from the **development** mode into the **live** mode. That for, toggle the switch on top of this site to **Yes** like seen in following screenshot.



After confirming the question, the description text will change to: `Your app is currently live and available to the public.`

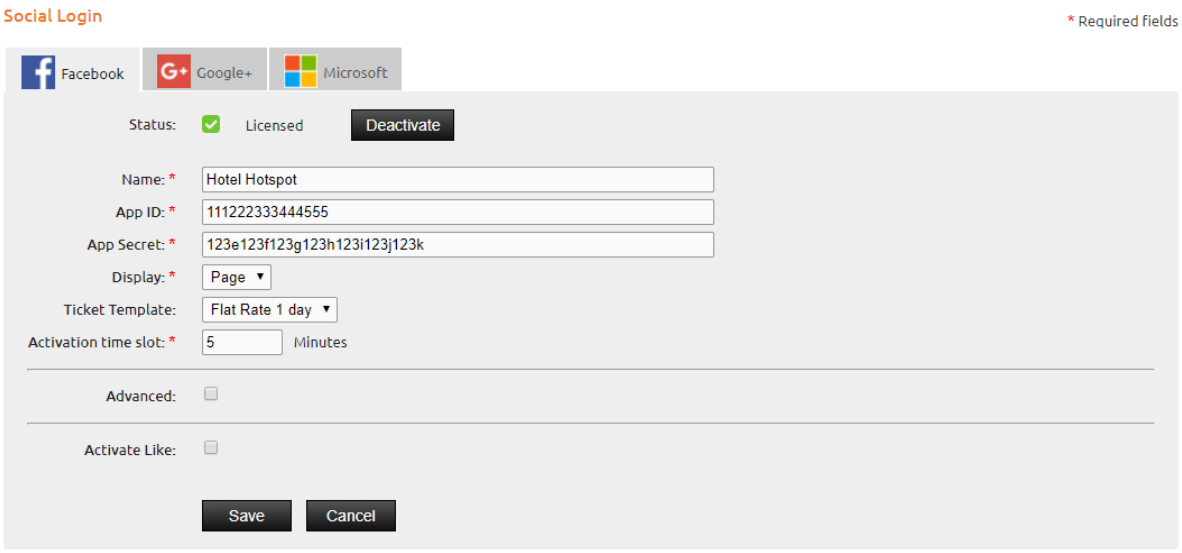
8.4.3 IACBOX Configuration

In the *WebAdmin* of the IACBOX navigate to **Modules / Interfaces** and scroll down to the **Social Login** section. Here you can activate **Facebook**.

Enter your **App Name**, **App ID** and **App Secret**

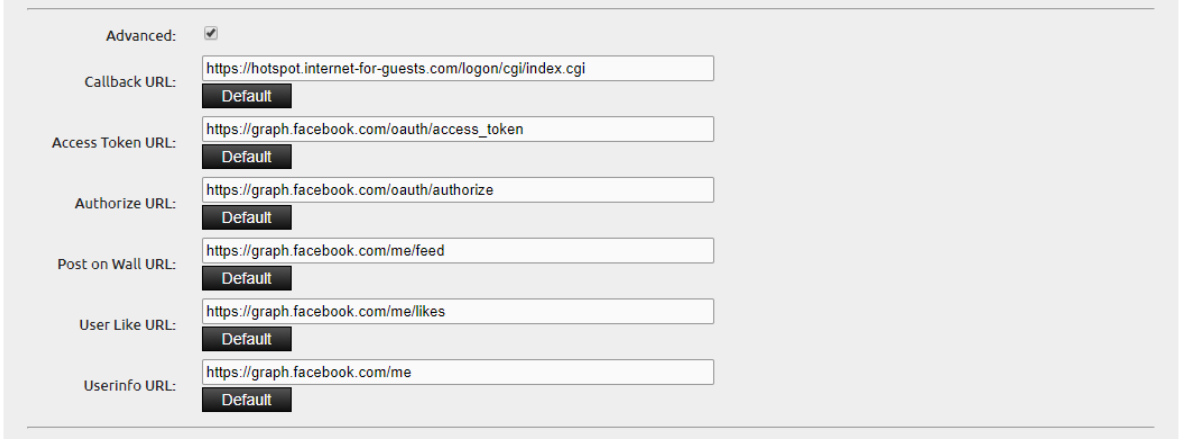
Hint:

- You need to configure a **Ticket Template** to use with **Social Login**.



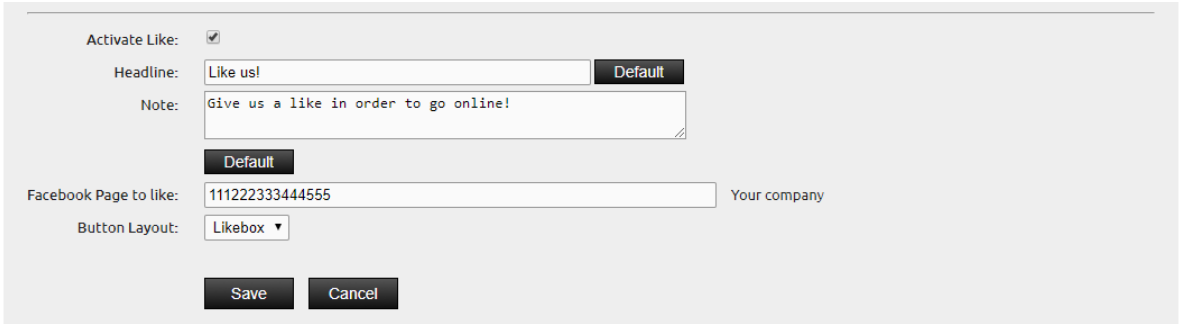
Hint:

- The **Advanced** configuration must be altered if you use a **custom certificate** on the IACBOX.



8.4.4 Activate Like

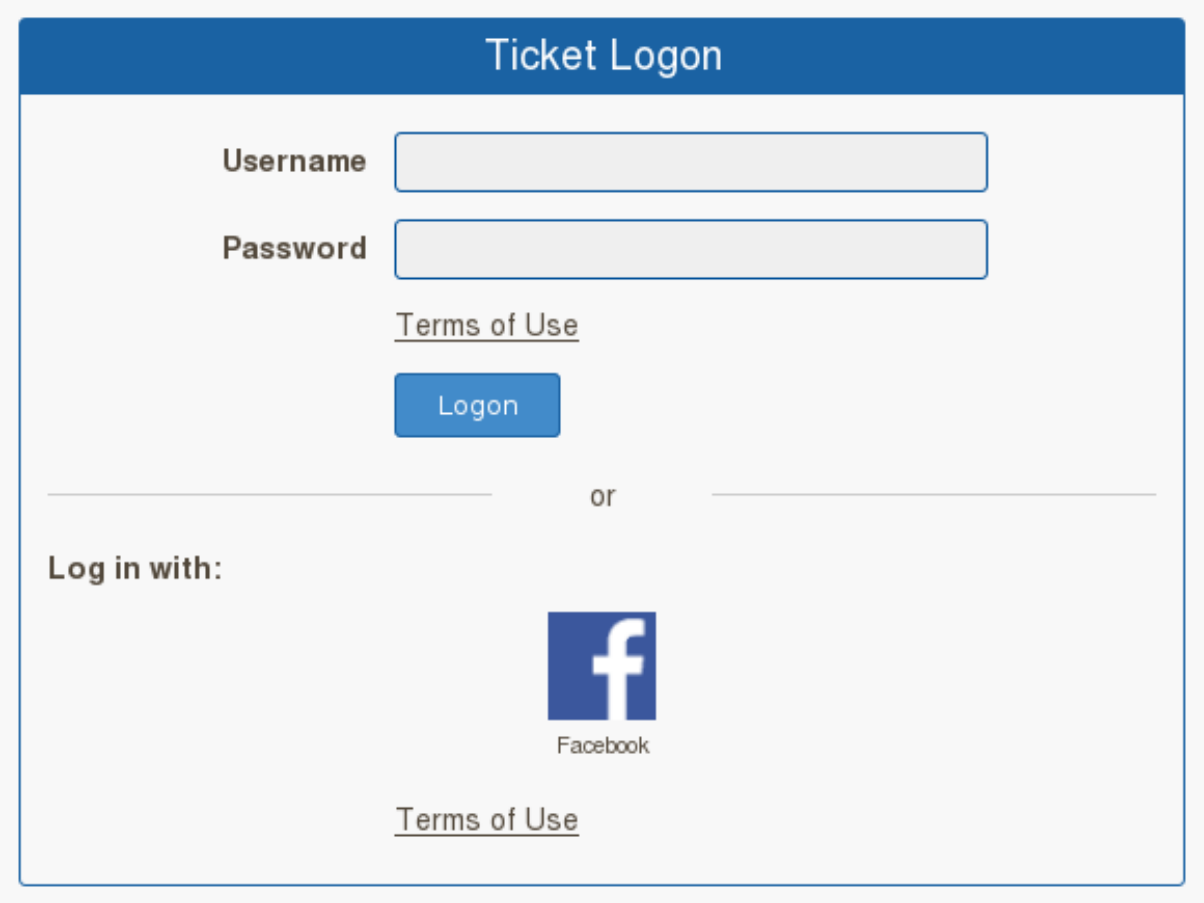
With this setting you can ask guests for a **Like** while they perform login with their facebook account. In case a guest revokes a *Like*, in the process of any re-login the guest will be asked again to like your page.



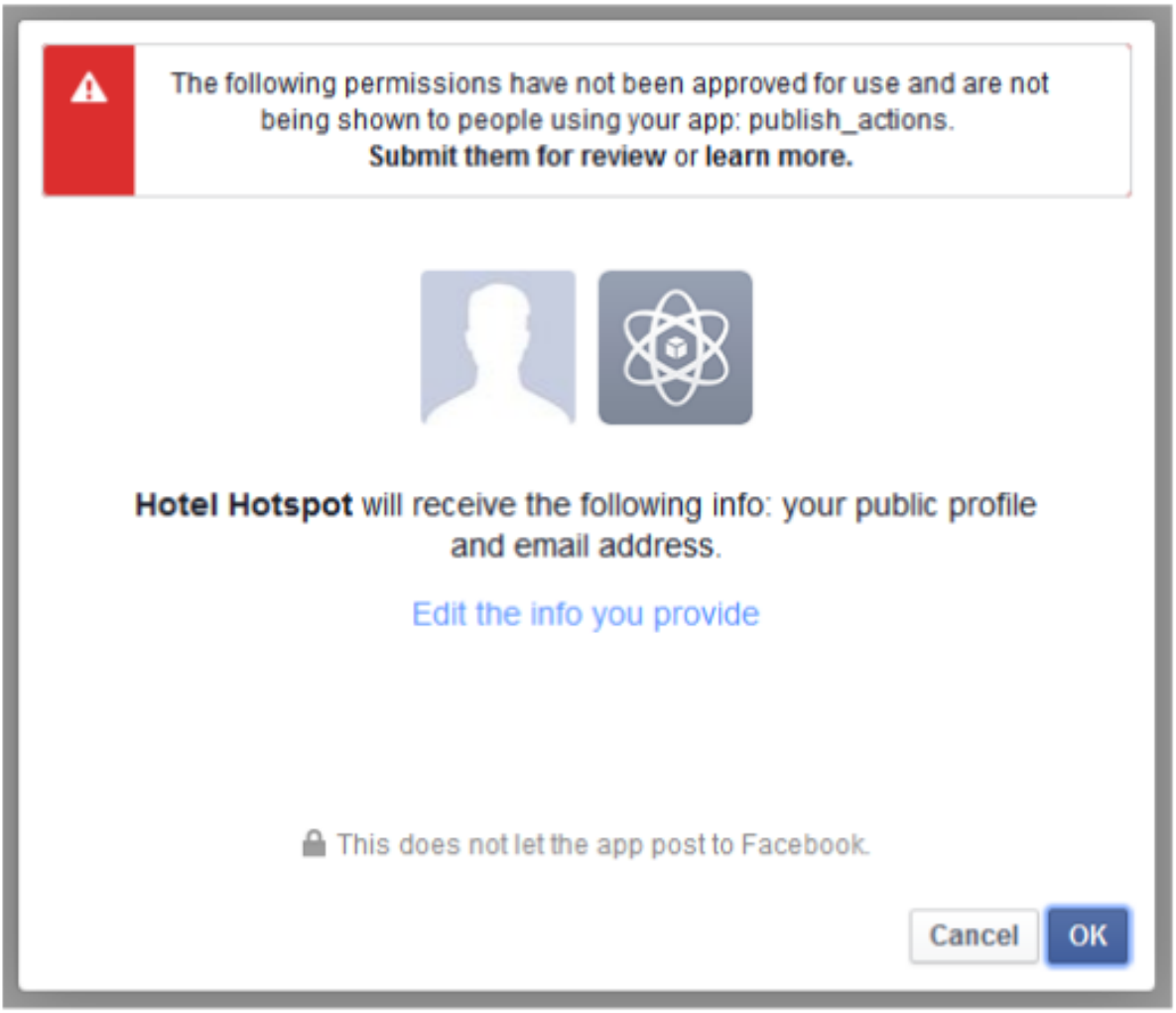
8.4.5 Customer Logon Page

On the *Customer Logon Page* you now are able to choose *Facebook* under *Ticket Logon*. Click on the *Icon* to get to the *Facebook Logon Page*. Note that after clicking on the *Icon* users can access the internet for a predefined amount of time. In this time users should:

- Log in on the *Facebook Login Page*



- Accept permissions which will be asked for



The amount of time in which guests do have access to the internet without logging in with their facebook account can be altered in the IACBOX facebook configuration with the input field **Activation time slot**.

Then optionally you can like a configured *Facebook* page.

8.5 Free Logon

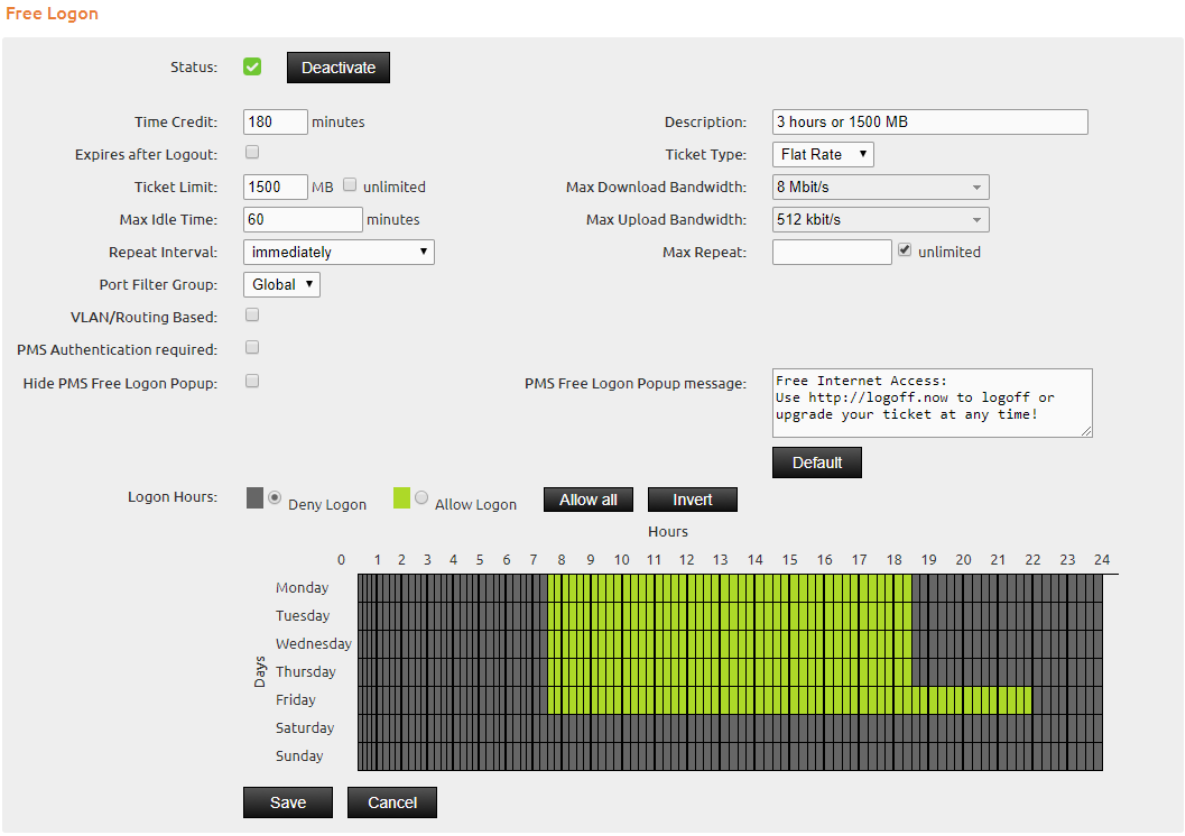
The *Free Logon* essentially is a simple and quick way to offer internet access for guests. This manual describes how to configure the **Free Logon** on the IACBOX.

Hint:

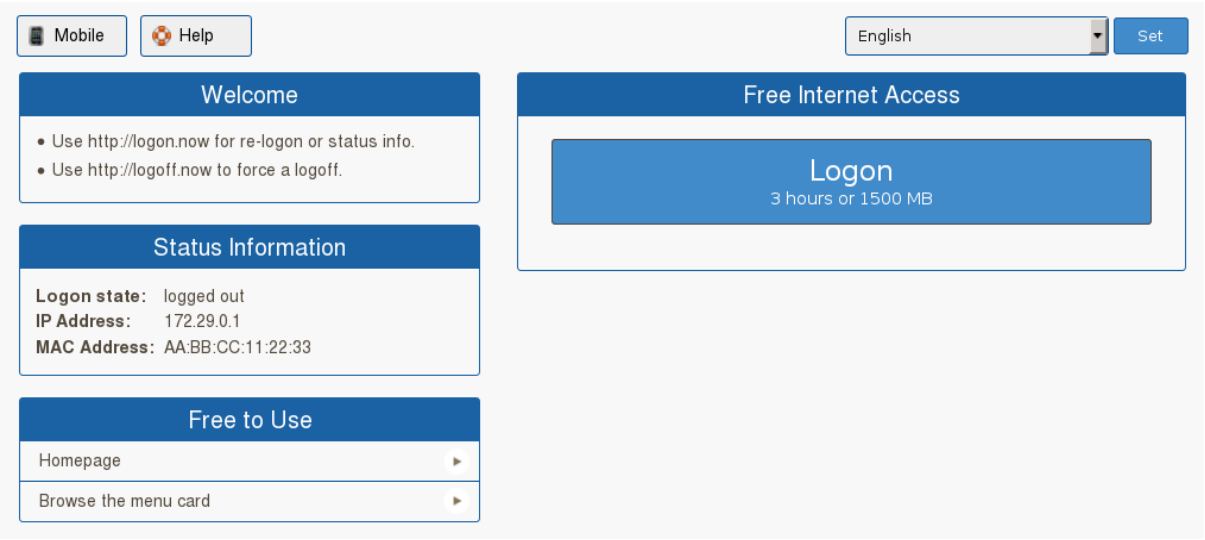
- By using the **Free Logon**, note the **Repeat Interval**. After a *Free Logon* ticket expires, the **guest device** is unable to login again until the *Repeat Interval* expires.
- If you use the function **Expires after Logout**, a **Free Logon** ticket will **Expire** as soon as it does get logged off.

8.5.1 Configuration

The **Free Logon** configuration can be found and activated in the WebAdmin menu **Tickets / Templates**.

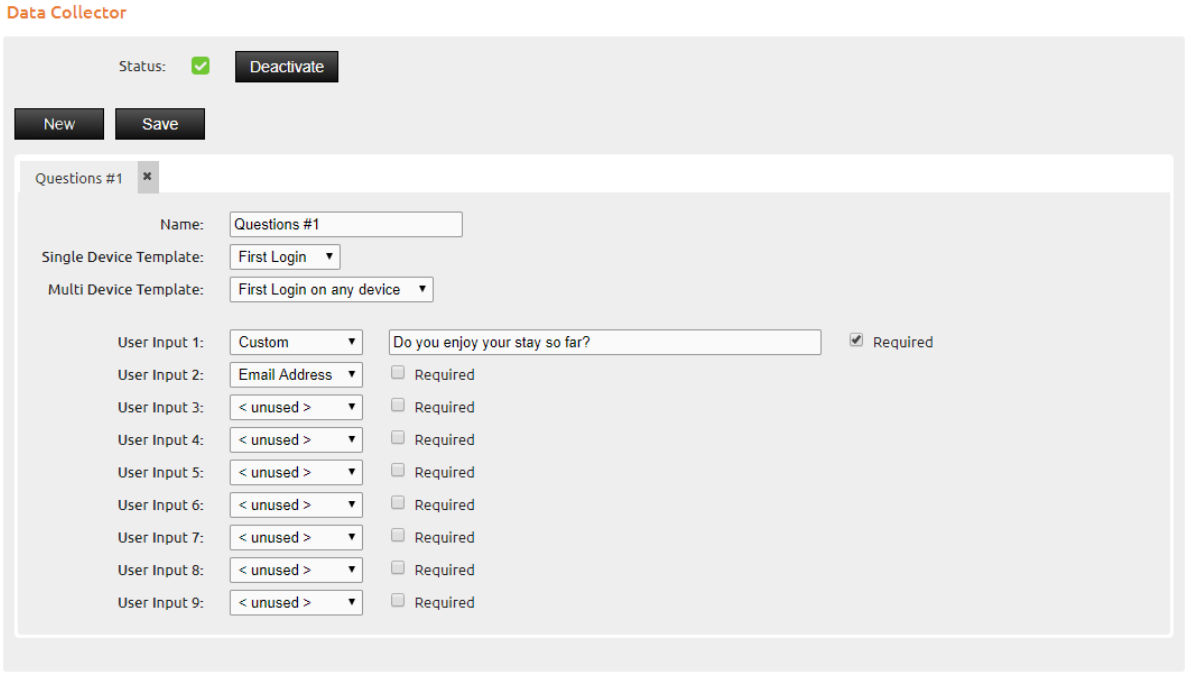


Upon activation, a **Free Logon** button will be visible on the *Client Logon Page*.



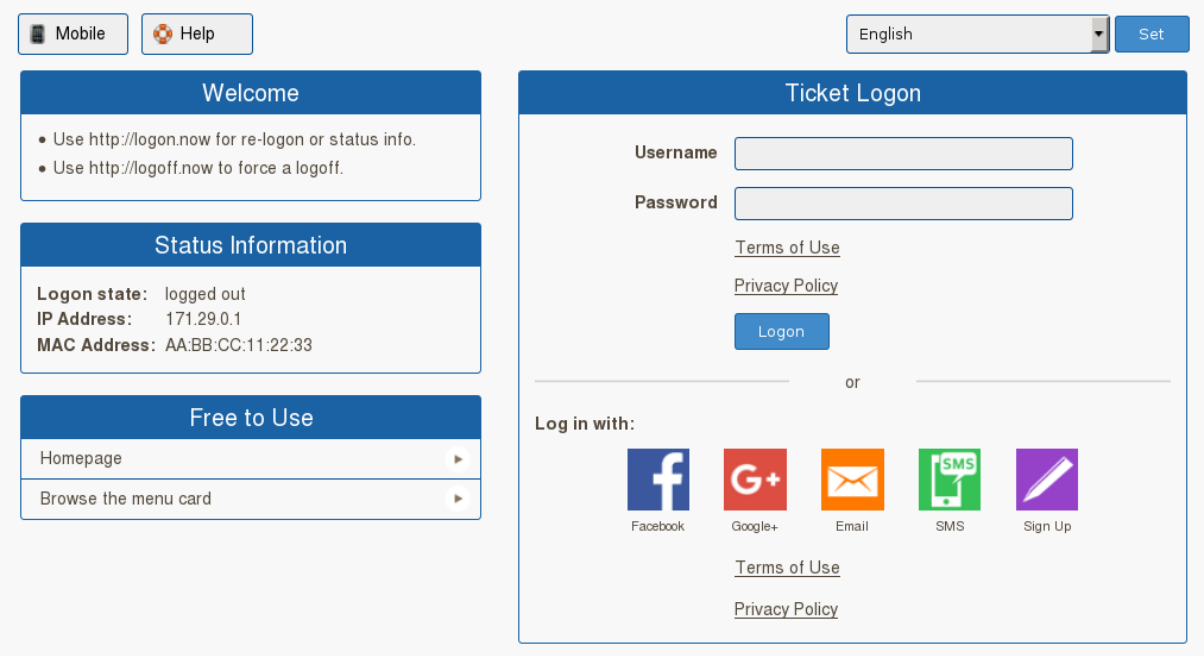
8.5.2 Data Collector

The **Free Logon** can be used together with the **Data Collector** module. This module can be activated and configured in the *WebAdmin* menu **Settings / Ticket**. Here you can ask for specific data which will then be saved with the Data Collector module.



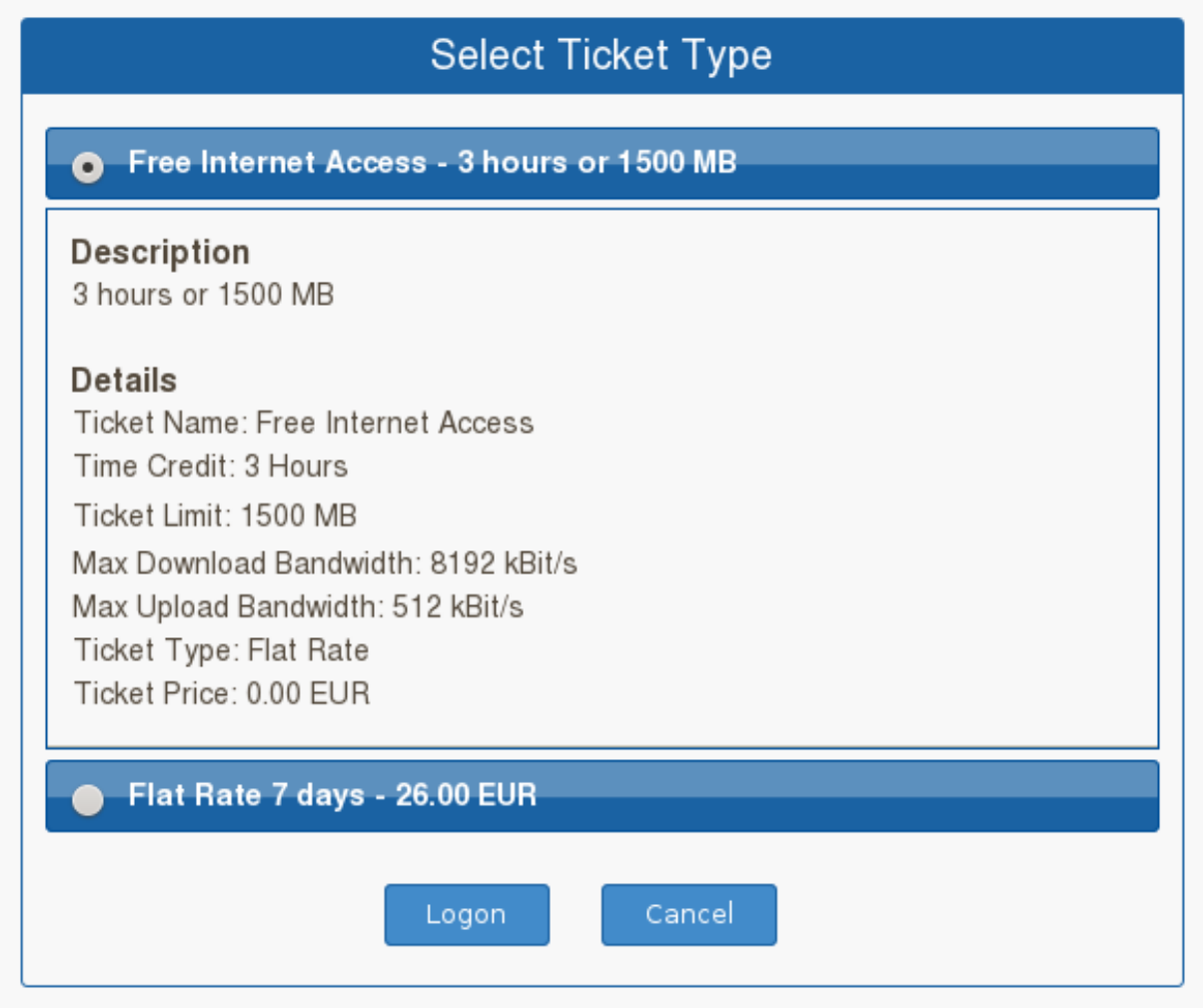
After clicking on **Save** you can select the *Data Collector* profile in the **Free Logon** configuration in **Tickets / Templates**.

Additionally **after** selecting a *Data Collector* profile for the *Free Logon* you can also enable the option **Show as Sign Up**. This will replace the big *Free Logon* button with a much more small button, which will be lined up with icons from the modules *Messaging* and *Social Login*.



8.5.3 Use with PMS System

If you do have a **PMS System** configured on the IACBOX, then it is also possible to enable the **Free Logon** to be shown in the PMS ticket selection. In order to do so, activate the function **PMS Authentication required** in the **Free Logon** configuration. The *Free Logon* will now be listed after guests authenticated with their PMS data.



8.5.4 Use-Cases

The options of the **Free Logon** allow you to meet certain requirements with ease. Attached are some examples:

Example 1:

- **Time Credit**: 30 minutes
- **Ticket Limit**: 500 MB
- **Max. Idle Time**: 60 minutes
- **Repeat Interval**: 600 minutes

Result: Guests can create and use one ticket every 600 minutes. The ticket will be valid for 30 minutes or 500 MB download volume.

Example 2:

- **Time Credit**: 50 minutes
- **Ticket Limit**: unlimited
- **Max. Idle Time**: 60 minutes
- **Repeat Interval**: 60 minutes
- **Max. Repeat**: 5

Result: Guests can create one free ticket every 60 minutes. This ticket can be used for 50 minutes until it expires. After 50 minutes, guests need to wait 10 minutes until they can create and use a new ticket. This process can be repeated 5 times in total.

8.6 Google Login

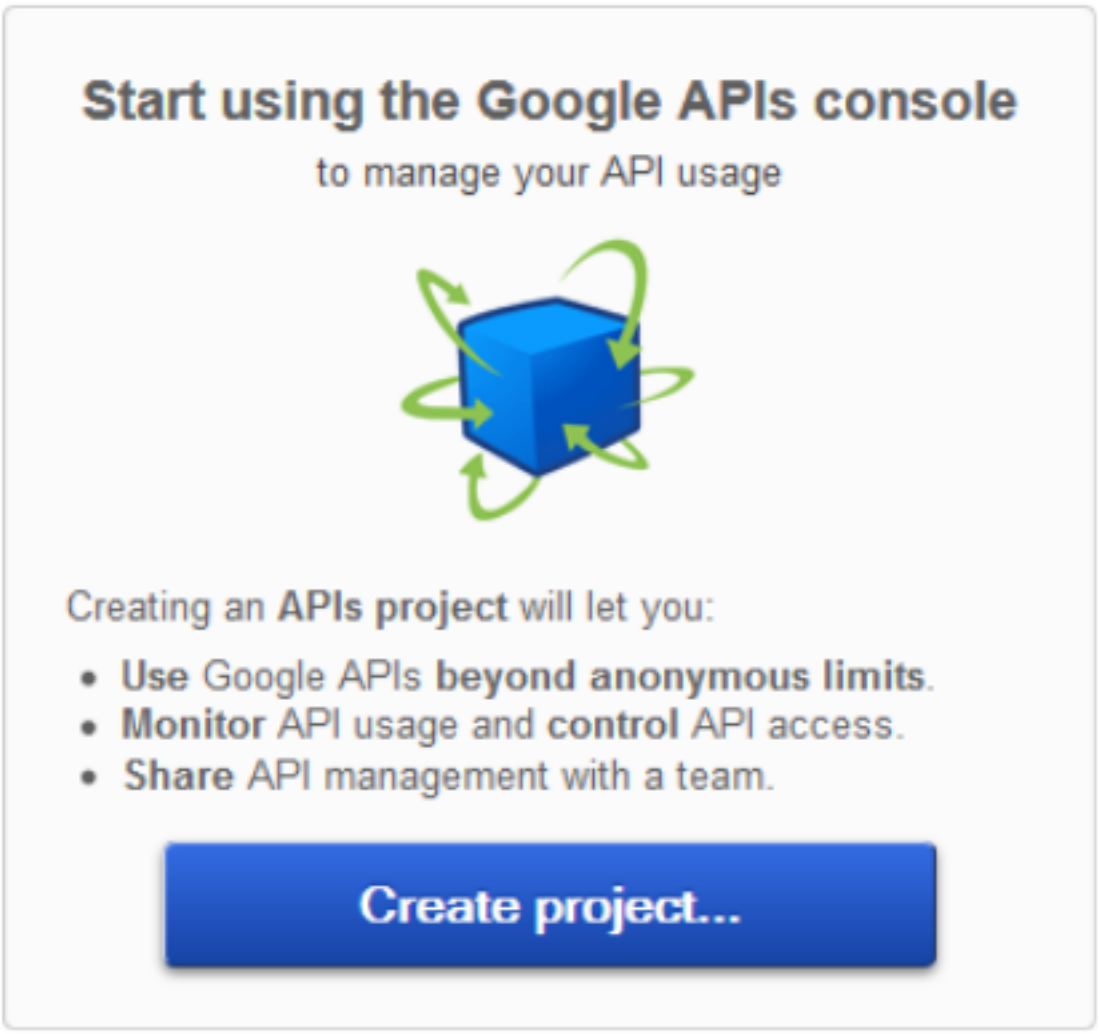
This manual describes how to create a new **Google Project**, so that guests can authenticate and log in on the IACBOX by using their Google account.

Hint:

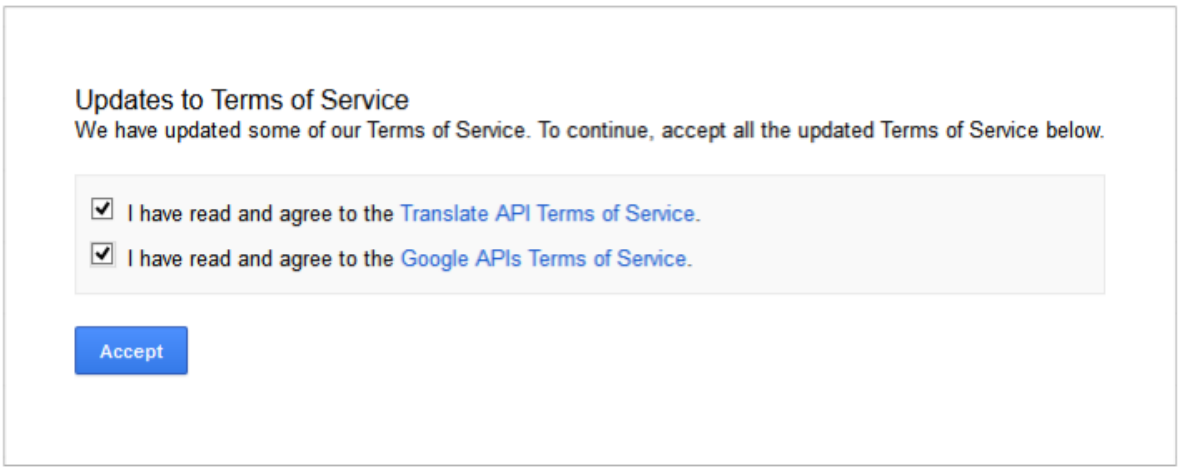
- This module became **free** for all IACBOXes with valid **Software Maintenance** as of 1st december 2016.
- The login with *Google Credentials* is meant to be **free**, guests can not be charged.
- To setup a *Google Login* for your guests, you will need to create a *Google Project* and configure it according to this manual.
- The *Google login* requires at least one valid ticket template, configured as 0€ (free).
- In case you use a *custom certificate* on your IACBOX, pay close attention to change all *redirect URLs* according to your custom hostname.

8.6.1 Creating a Google Project

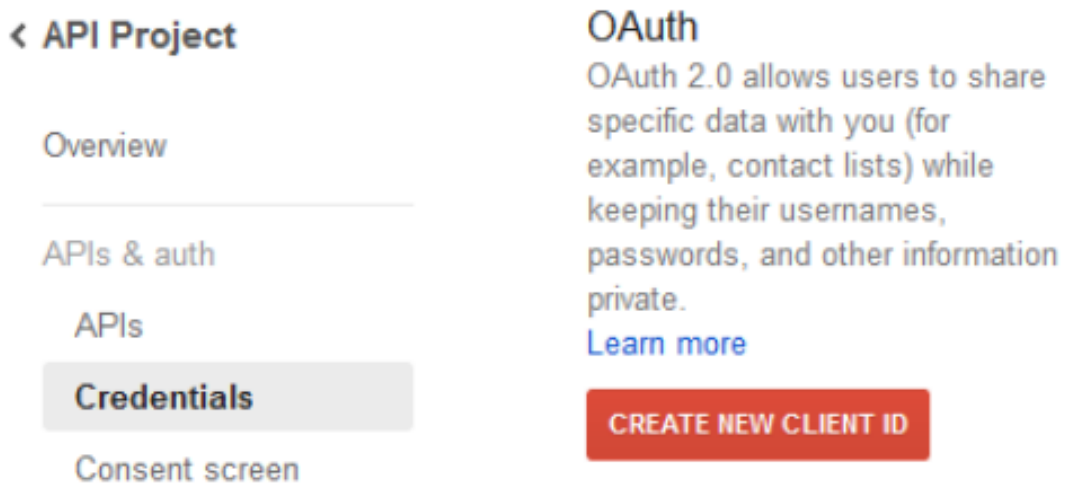
In order to use the *Google Services* you have to create a project beforehand. This can be done on the *Google Developers Page*, public under this URL: `https://code.google.com/apis/console` Log in with your regular Google account and then select **Create project**.



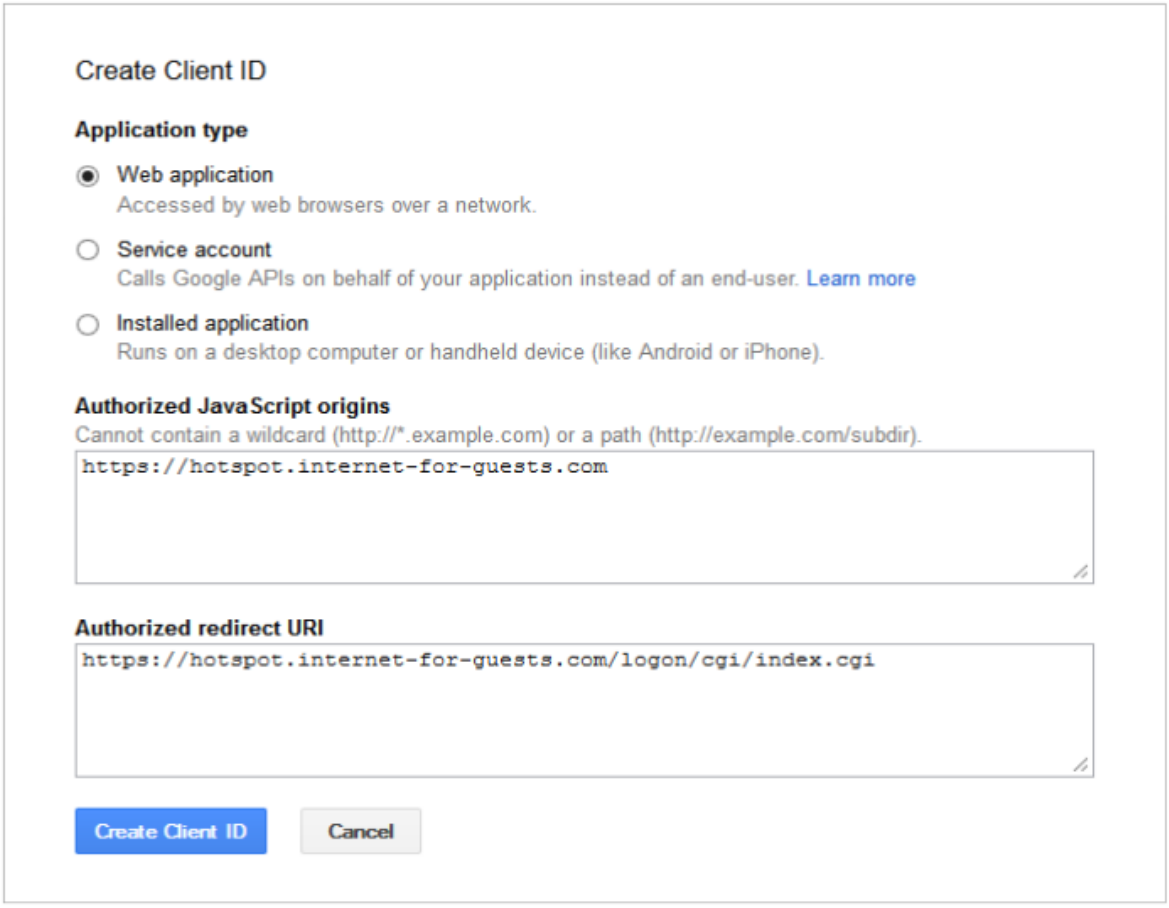
Read and eventually agree to the *Terms of Service*, then continue with **Accept**.



Now on the *left side* below **APIs & auth** click on **Credentials** and then on **CREATE NEW CLIENT ID**.



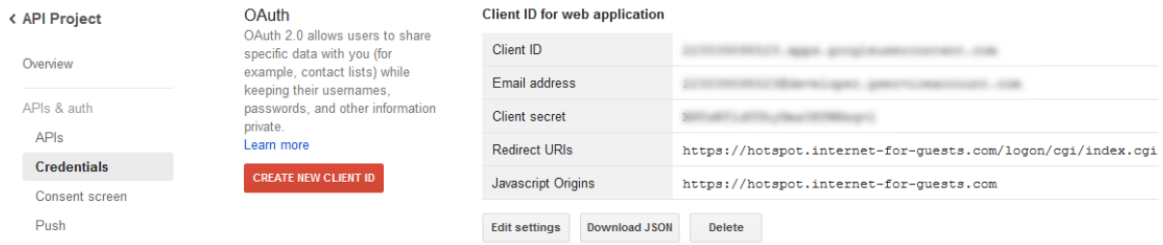
In the new window choose **Web application** as **Application type**. Add the **Authorized JavaScript origins** and **Authorized redirect URI** like shown in the screenshot below.



Hint:

- In case you use a *custom certificate* on your IACBOX, pay close attention to change all *redirect URLs* according to your custom hostname.

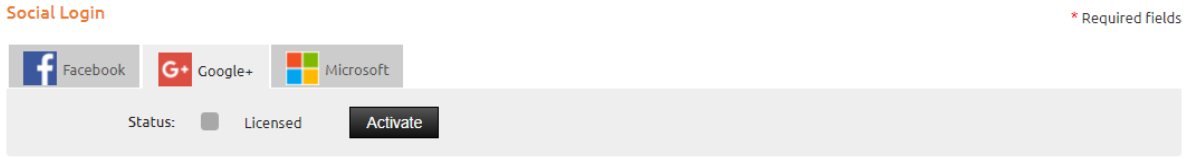
Now write down the **Client ID** and the **Client secret** for the IACBOX configuration.



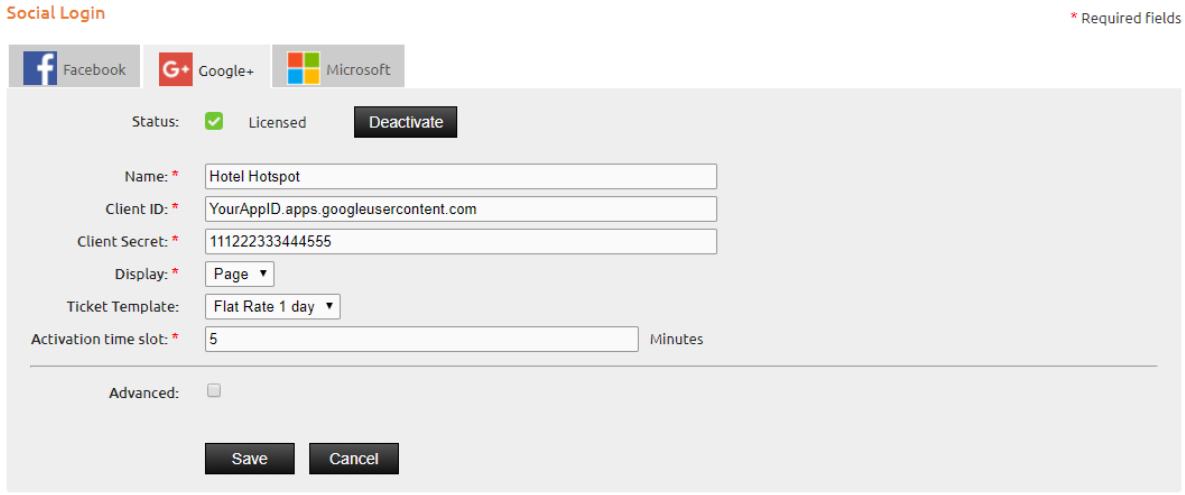
Now navigate to **Consent screen** which can be found in the left menu. Here you only need to type in your **Email address** and the **Product name**. Optionally you can configure any additional inputs and upload a logo. Confirm the changes by clicking on **Save**.

8.6.2 IACBOX Configuration

In the *WebAdmin* of the IACBOX navigate to **Modules / Interfaces** and scroll down to the **Social Login** section. Here you can activate **Google**.

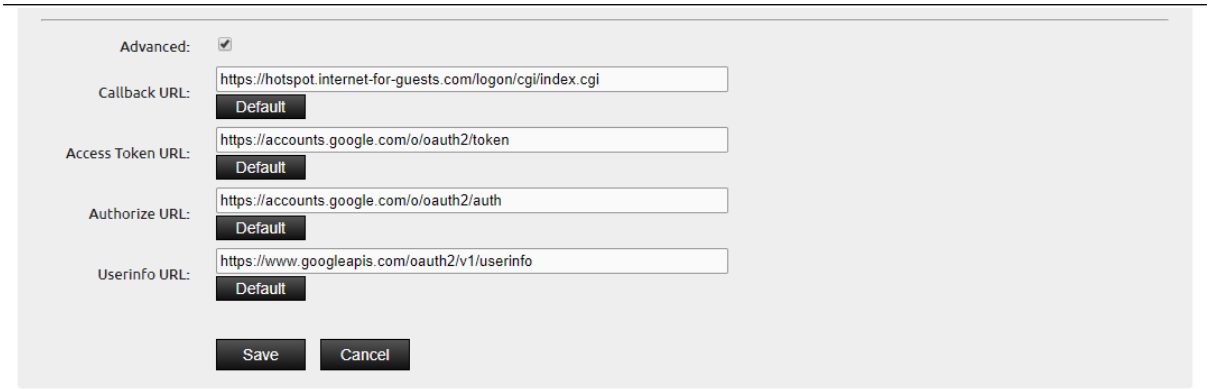


Now fill in the required inputs, including the **Client ID** and **Client secret** from before.



Hint:

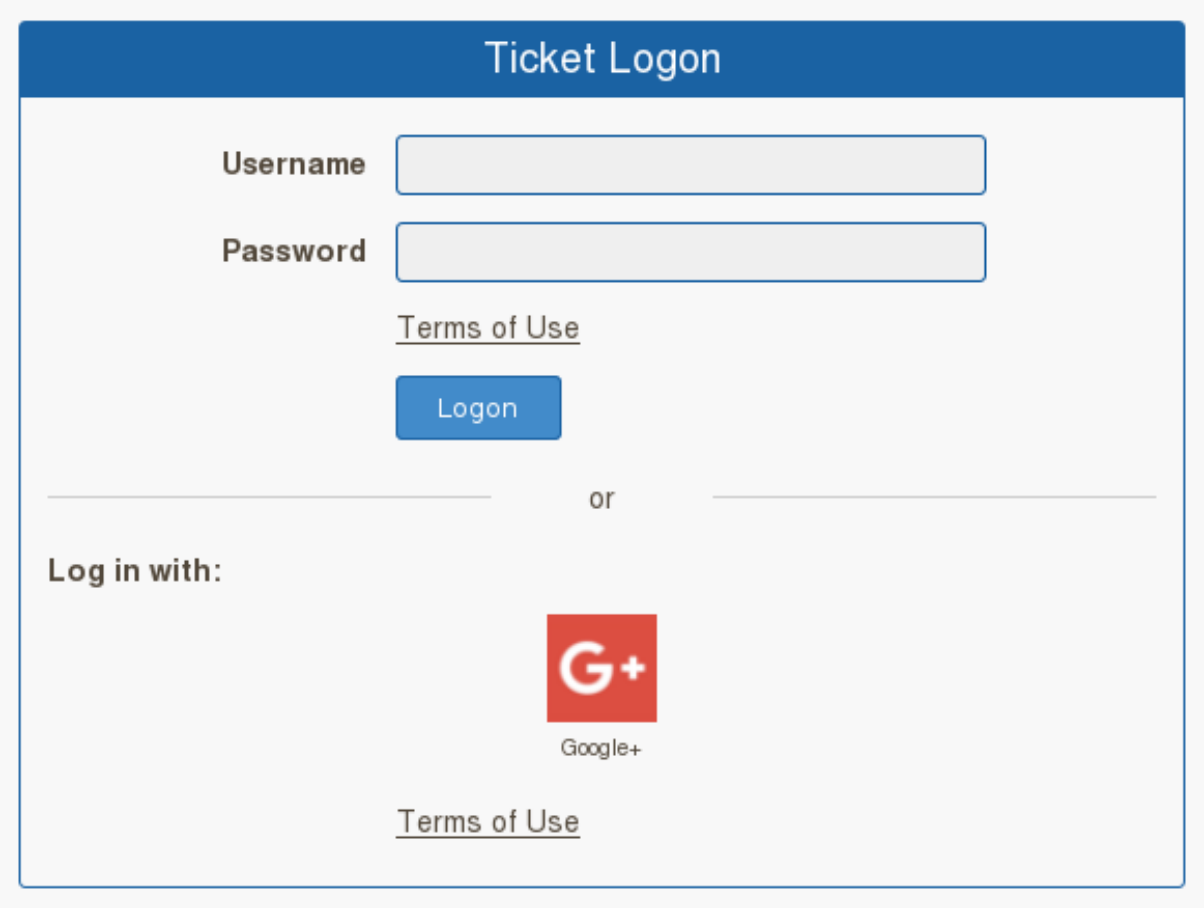
- The **Advanced** configuration must be altered if you use a **custom certificate** on the IACBOX.



8.6.3 Customer Logon Page

On the *Customer Logon Page* you now can see a *Google Icon* under the *Ticket Logon*. Click on the *Icon* to get to the *Google Logon Page*. Note that after clicking on the *Icon* users can access the internet for a predefined amount of time. In this time users should:

- Log in on the *Google Login Page*
- Accept permissions which will be asked for



The amount of time in which guests do have access to the internet without logging in with their Google account can be altered in the IACBOX Google configuration with the input field **Activation time slot**.

8.7 SMS Login

This manual describes how to provide a SMS login for guests on the IACBOX.

Hint:

- This module became **free** for all IACBOXes with valid **Software Maintenance** as of 1st december 2016.
- The login by using a *SMS* is meant to be **free**, guests can not be charged.
- In order to configure the *SMS* login a **SMS Gateway** is **required**. You can also create your own **SMS Gateway** on an external server and then use the *generic HTTP GET/POST Interface* in the IACBOX SMS configuration.
- The **SMS** login requires at least one valid **ticket template**, configured as 0€ (free).

8.7.1 How it works

On the *Client Login Page* of the IACBOX, guests will notice a **SMS** icon. This icon can be used to input a *Mobile Phone Number*. After sending the number, the IACBOX will send a *request* to the configured **SMS Gateway**, then this *SMS Gateway* can send a SMS to the phone number of the guest. The SMS will contain login credentials for the guest.

Example SMS:

```
Welcome to ExampleCompany!
Username: ticket1
Password: dbh3z
Status: http://logon.now
```

Attention:

- Please note that in order to keep the SMS limit *under 160 characters*, the **Status** line will only contain a short link which will be redirected to the **IACBOX Client Login Page**. If you want to add a link which automatically does log in guests with the added credentials, then replace the URL **http://logon.now** with **$LINK**:

```
Welcome to $COMPANY
Username: $USER
Password: $PASSWORD
Status: $LINK
```

Attention:

- With version **17.2.11676** it is now possible to use the variable **$UNSECURELINK**, which essentially is a shorter version of **$LINK**.

8.7.2 Selecting your SMS Gateway

The IACBOX does support a few *SMS Gateway providers and interfaces* out of the box. These providers can be selected directly in the SMS configuration of the IACBOX.

Austria:

- sms.at Business, A-8010 Graz - http://business.sms.at
- A1 Telekom Austria AG, A-1020 Wien - http://www.a1.net

Germany:

- mes.mo - Any-SMS, D-73262 Reichenbach - http://www.mesmo.org
- GOYYA Marketing KG, D-01099 Dresden - http://www.goyya.com

Switzerland:

- eCallâĎć via Email (requires private or business Account), CH-8832 Wollerau - http://www.ecall.ch
- eCallâĎć via HTTPS (requires business Account), CH-8832 Wollerau - http://www.ecall.ch

Turkey:

- Maradit Web Services

Alternatively it is possible to use a custom SMS Gateway. If you do have a *Webserver* or *Email Server* in your environment, you can also use the generic interfaces **Generic via HTTP** and **Generic via Email**.

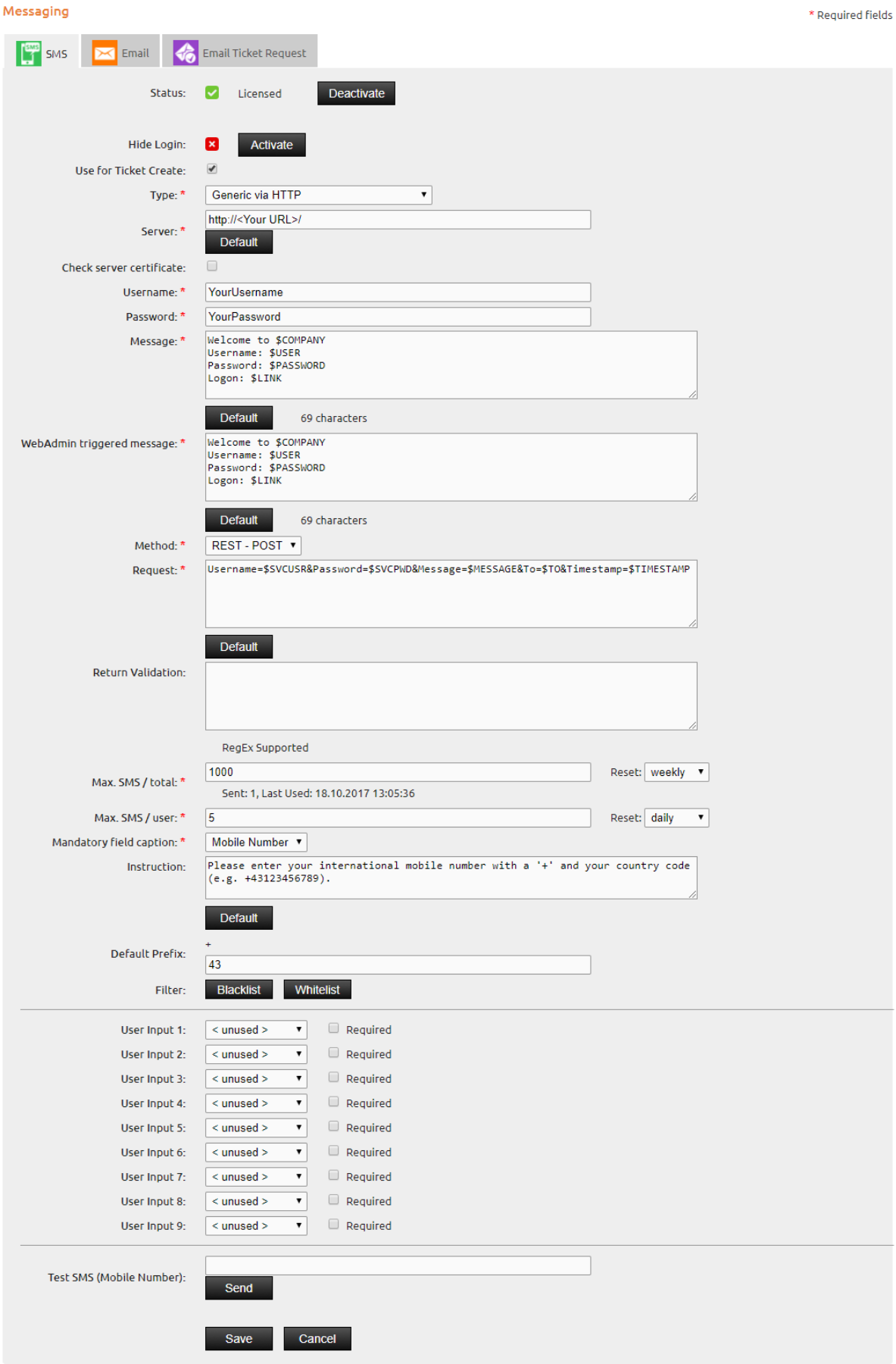
Hint:

- By using one of the 2 generic interfaces, *you must configure your servers to send a SMS* which does contain the information provided by the IACBOX. This can be done by using a *mobile phone network interface* in the sending host or by using custom software which can interact with connected mobile phone devices.

- Any solution developed upon this requirement must be evaluated and maintained by your team.

8.7.3 Configuration

The configuration of the **SMS module** can be found in the *WebAdmin* menu **Modules / Interfaces**.


Messaging
* Required fields
SMS
Email
Email Ticket Request
Status:
Licensed
Deactivate
Hide Login:
Activate
Use for Ticket Create:
Type: *
Generic via HTTP
Server: *
http://<Your URL>/
Default
Check server certificate:
Username: *
YourUsername
Password: *
YourPassword
Message: *
Welcome to $COMPANY
Username: $USER
Password: $PASSWORD
Logon: $LINK
Default
69 characters
WebAdmin triggered message: *
Welcome to $COMPANY
Username: $USER
Password: $PASSWORD
Logon: $LINK
Default
69 characters
Method: *
REST - POST
Request: *
Username=$SVCUSR&Password=$SVCPWD&Message=$MESSAGE&To=$TO&Timestamp=$TIMESTAMP
Default
Return Validation:
RegEx Supported
Max. SMS / total: *
1000
Reset:
weekly
Sent: 1, Last Used: 18.10.2017 13:05:36
Max. SMS / user: *
5
Reset:
daily
Mandatory field caption: *
Mobile Number
Instruction:
Please enter your international mobile number with a '+' and your country code (e.g. +43123456789).
Default
Default Prefix:
+
43
Filter:
Blacklist
Whitelist
User Input 1:
< unused >
Required
User Input 2:
< unused >
Required
User Input 3:
< unused >
Required
User Input 4:
< unused >
Required
User Input 5:
< unused >
Required
User Input 6:
< unused >
Required
User Input 7:
< unused >
Required
User Input 8:
< unused >
Required
User Input 9:
< unused >
Required
Test SMS (Mobile Number):
Send
Save
Cancel

For the fields **Server**, **Username** and **Password** refer to the documentation of your *SMS gateway provider*. The explicit description of the input fields can be found on the **Help Page** of the *WebAdmin* menu in the right upper corner.

Hint:

- The **SMS** login requires at least one valid ticket template, configured as 0€ (free).

8.7.4 Generic Interfaces

In addition to the supported *SMS gateway providers* there is also the possibility to use a *generic interface*. You can select between **generic via HTTP** and **generic via Email**. This enables you to create your own interface.

Generic via HTTP

Enter the *target server* with *username* and *password* for authentication. The input field **user data** from the WebAdmin configuration will be sent to the *target server* with the selected data transmission method (**REST-POST**, **REST-GET**, **JSON**).

Since the *HTTP request* can be customized in the input field *Request*, you can determine how the *target server* receives the data and further processes it.

The following data is transmitted regardless of the selected data transmission method:

- **Username** The username of the SMS gateway
- **Password** The password of the SMS gateway
- **Message** The message which will be sent to the user
- **To** The mobile number which was entered by the user
- **Timestamp** Current timestamp

Generic via Email

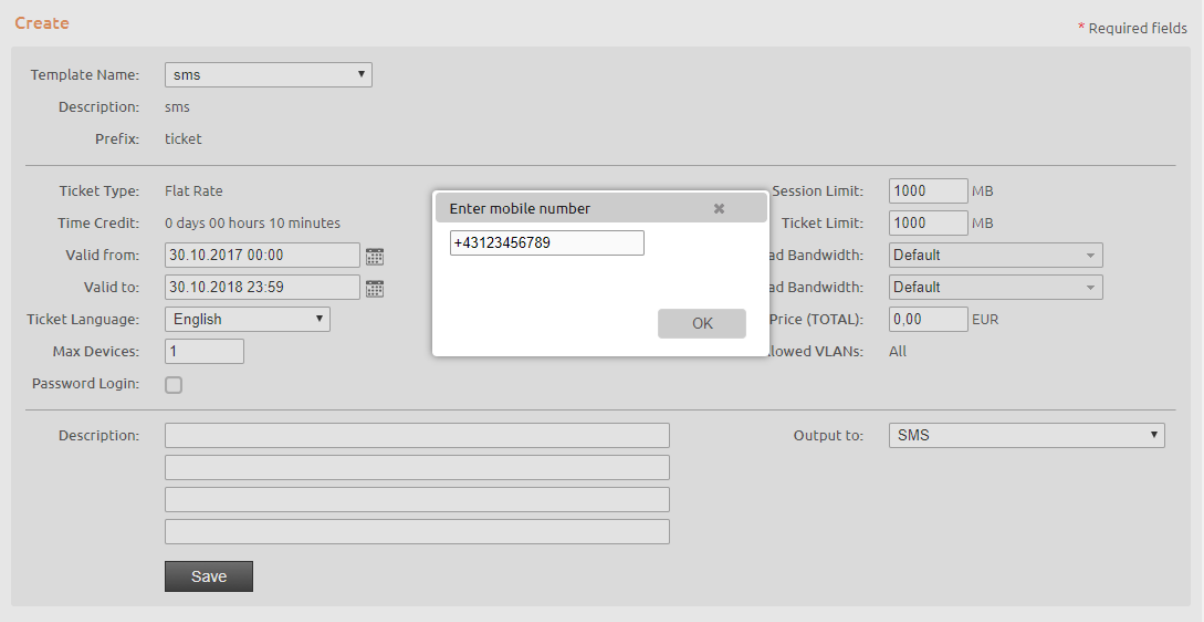
Enter a **sender** (e.g. "iacbox") and the *target email server* which should receive the data from the IACBOX as email. The mobile number which was given by the guest will be added to the configured *target server* address by which an unique email address is being used for each mobile number.

The user data (*username*, *password*, etc.) will then be sent to the generated email address. The *target server* which receives this email will then send an SMS which contains the **Message** to the users phone number.

Example: The configured target server is **sms.ip-plus.net**. When a guest enters a mobile phone number on the *Client Login Page*, it will be added to the configured *target server*, e.g. `+43797979797@sms-ip-plus.net`

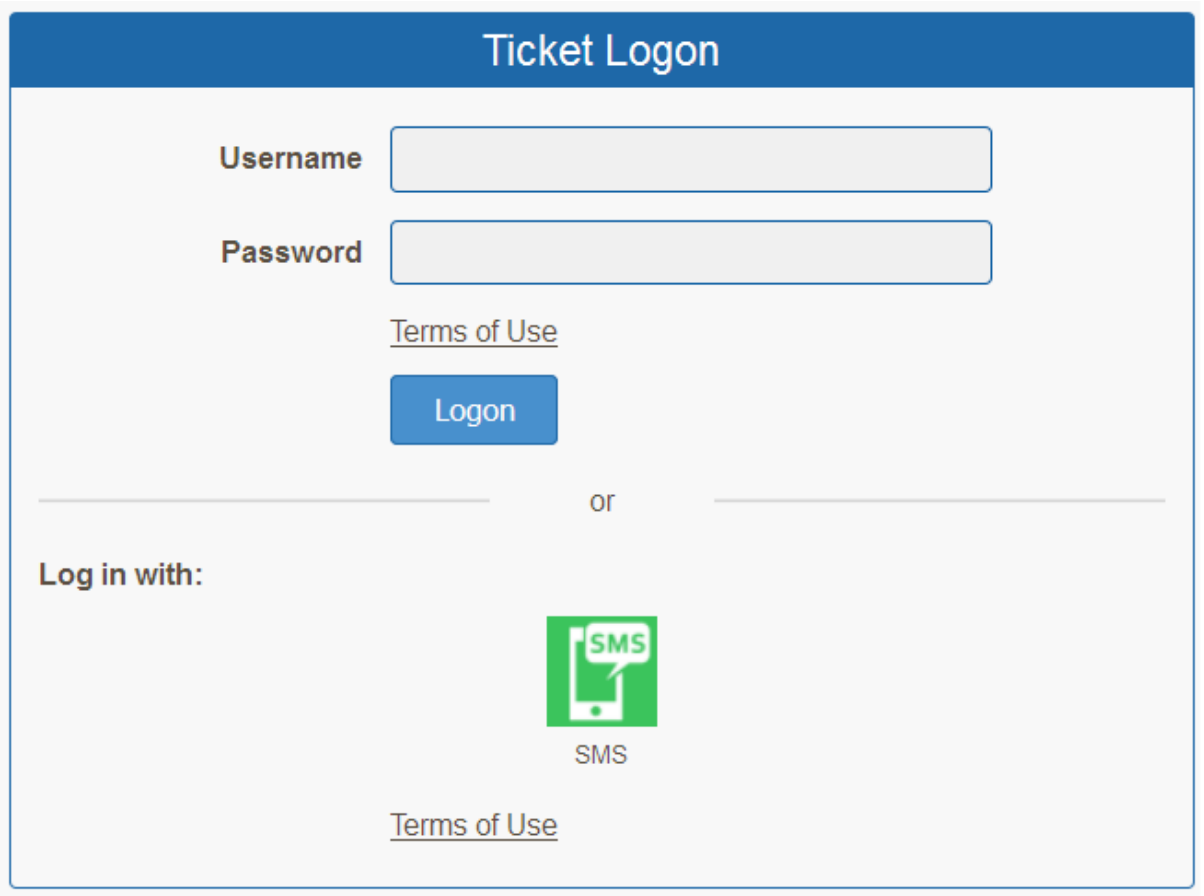
8.7.5 WebAdmin Tickets with SMS

If the option **Use for Ticket Create** is active, ticket data (**username**, **password**) of newly created tickets from *WebAdmin* **Tickets / Create** can be directly sent via SMS.

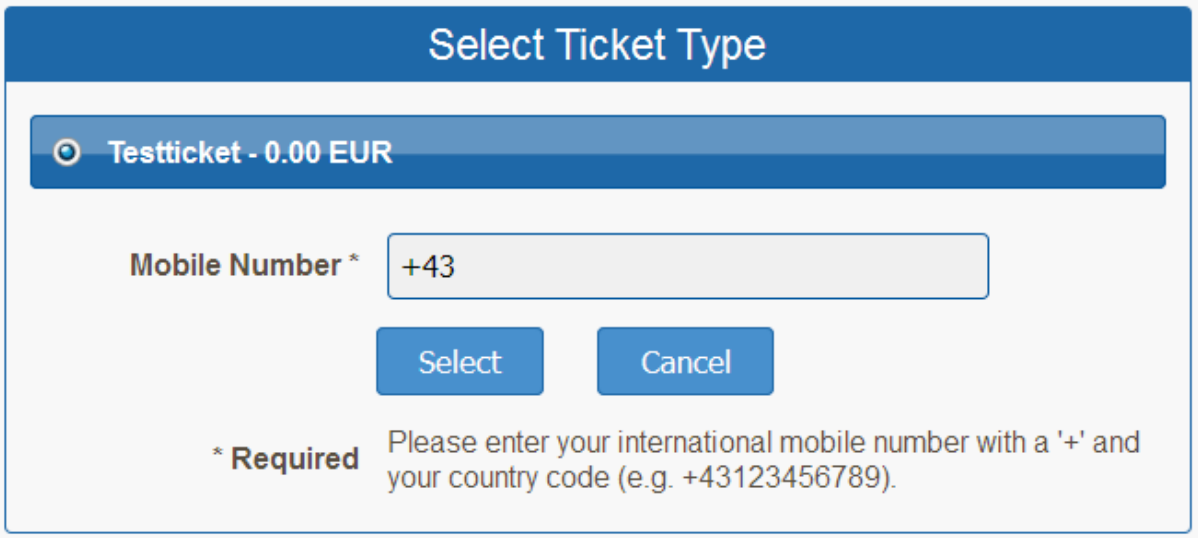


8.7.6 Client Logon Page

After all settings have been made, the result can be seen on the customer logon page.



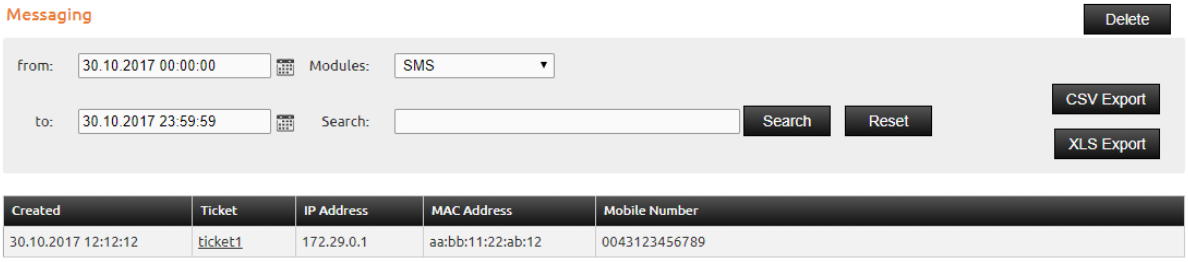
By clicking on the SMS icon, guests can now enter their mobile phone number to continue with the login.



Soon the guest will receive an SMS which contains the ticket credentials. Optionally it is possible to send a link which does log in the mobile device at once. This is explained in the upper section of this manual.

8.7.7 Stored Data

In the *WebAdmin* menu **Reporting / Messaging** you can always check and also download archived user data.



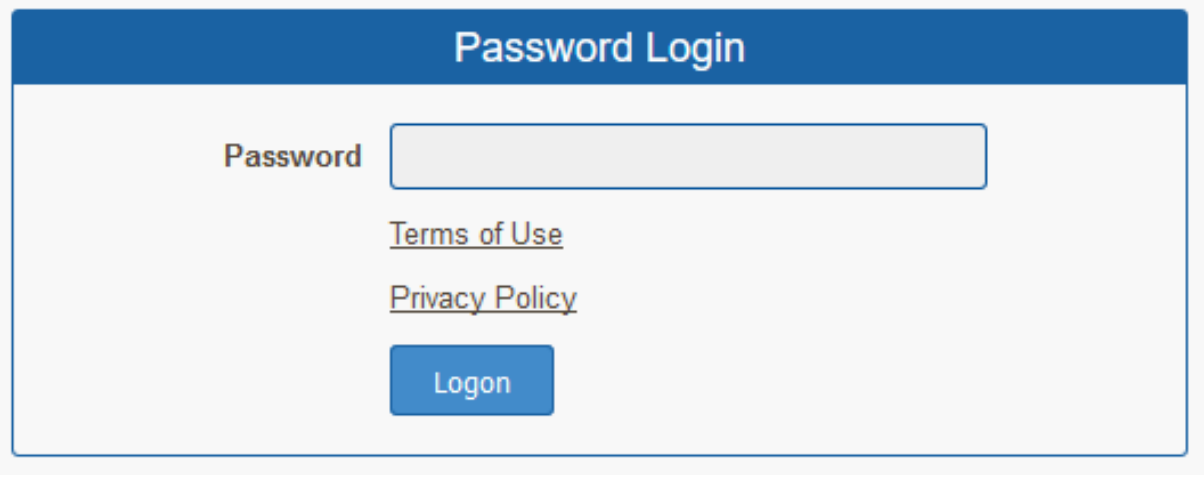
8.8 Password Login

Starting with version 7.0.8982 of the IACBOX, guests can now also log in by using a simple password instead of the known username and password combination. This functionality is called **Password Login** and can be used with several modules like:

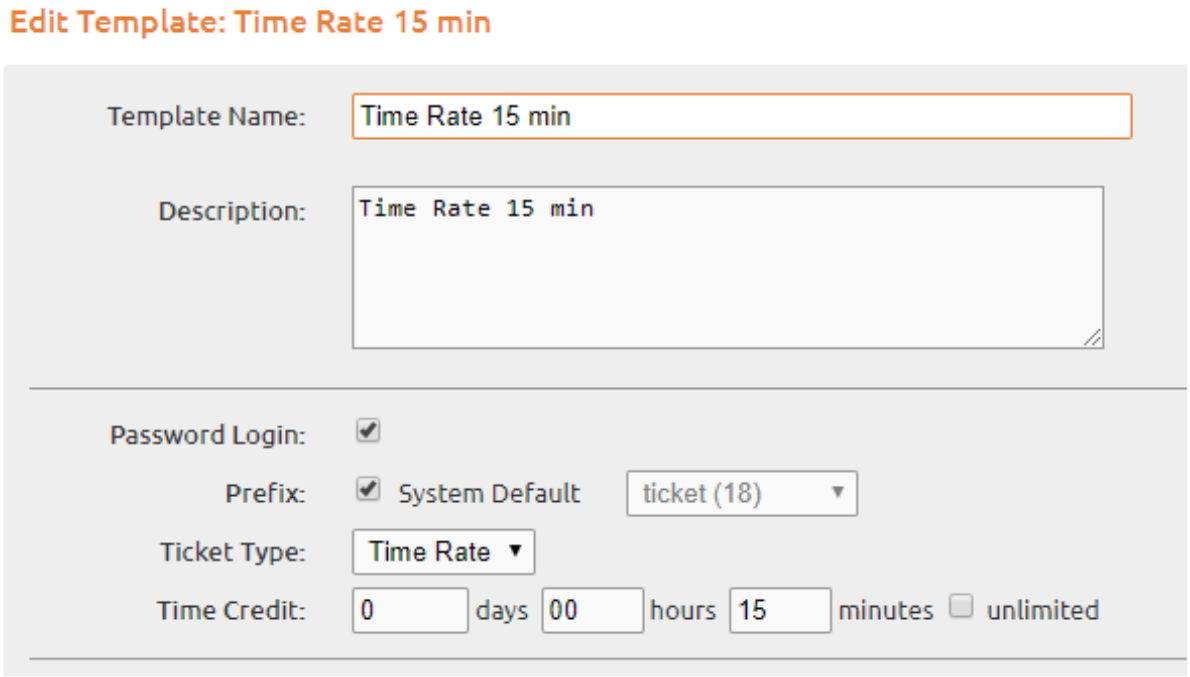
- Regular Ticket Login
- Email Login
- SMS Login
- Online Payment
- External Authentication
- Local Users

In order to activate the basic functionality, the **Password Login** must first be activated in the WebAdmin menu **Client Logon / Design**.

This will also activate the **Password Login** box on the *Client Logon Page*.



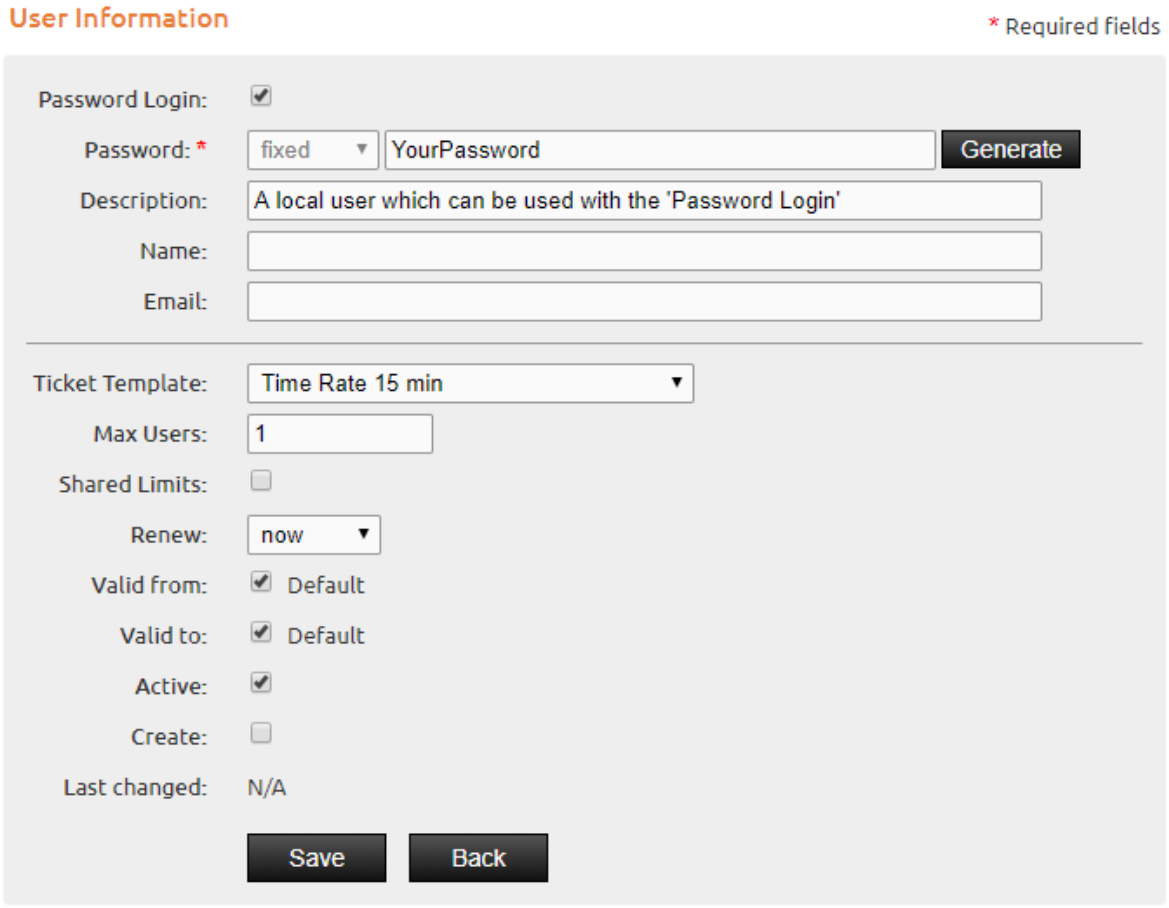
Activation of the **Passwort Login** must be done in **Ticket Templates**. Either create a new or edit an existing *Ticket Template* for usage of the *Passwort Login*.



8.8.1 Usage with Local Users

In order to use the *Passwort Login* for *Local Users*, you must first activate the **Local Database** in the WebAdmin menu **Modules / Authentication**. This functionality is always available, also if the module *External Authentication* was not licensed.

After this is done, navigate to **Tickets / Users**. Here you can create **Local Users**. For this local User you can either choose a static password or generate a random one.



8.8.2 Other Modules

For the modules **Email Login**, **SMS Login**, **Online Payment** and **External Authentication** it is sufficient to select a **Ticket Template** with the *Passwort Login* option enabled.

8.9 PayPal Integration

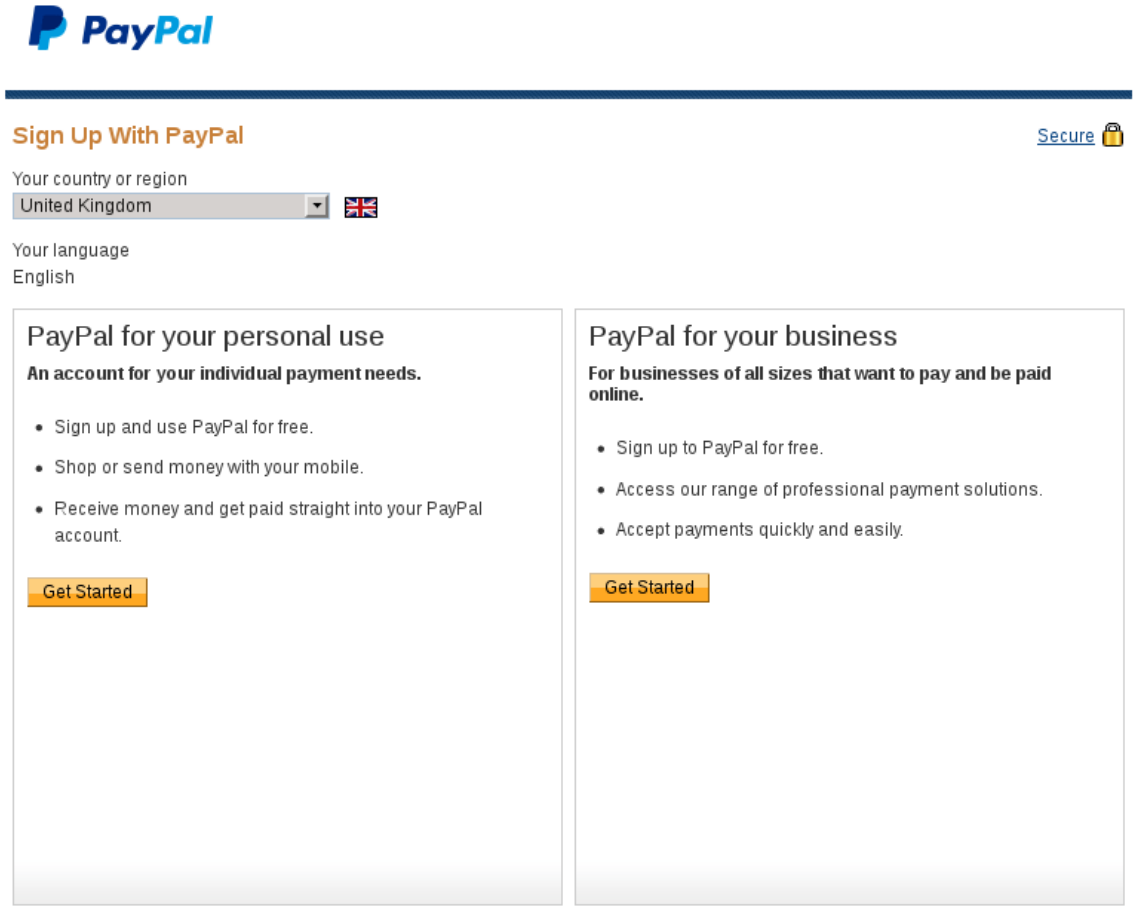
This manual describes how to configure the PayPal API on the IACBOX so guests can pay tickets via their PayPal account.

Hint:

- To use this API a **PayPal business account** is required. This manual covers how to create a PayPal business account.
- Failed or frozen transactions are not within the scope of the IACBOX and can not be supported.

8.9.1 PayPal Account

If you dont already own one, you must create a **PayPal business account**. To register a new account, visit https://www.paypal.com/ and hit **new account**. In the new window, choose **PayPal for your business** as account type.



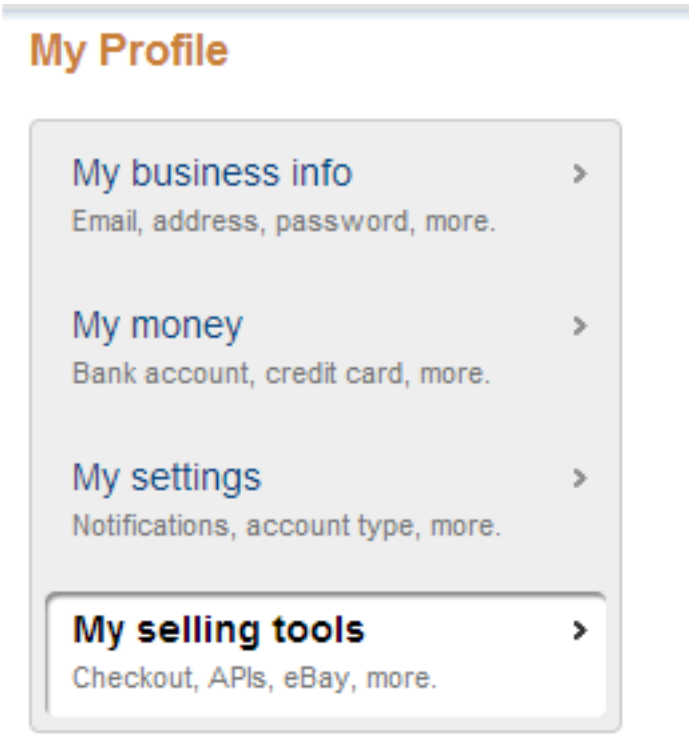
If you do have a **personal account**, then it is also possible to convert it into a **business account** with the following steps:

- Click on **Profile** and then **My Personal Info**
- At the field **Business Information** click on **Update**
- Confirm all your business information and then proceed by clicking **Upgrade**

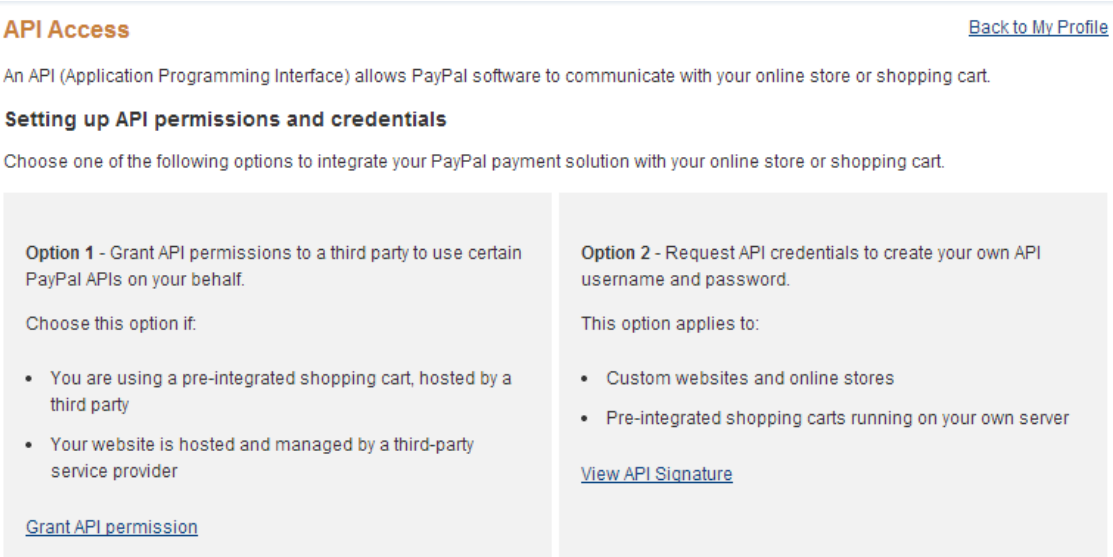
Hint:

- If you have problems creating or upgrading to a **business account**, please consult the PayPal support.

After you've created or upgraded your business account, you will find the section **My selling tools** in your **Profile**.



Here you can set up the **API Access** on the PayPal side. Note that the appearance of this webpage often changes, but the process of creating and setting up the API should be quite similar to this description.



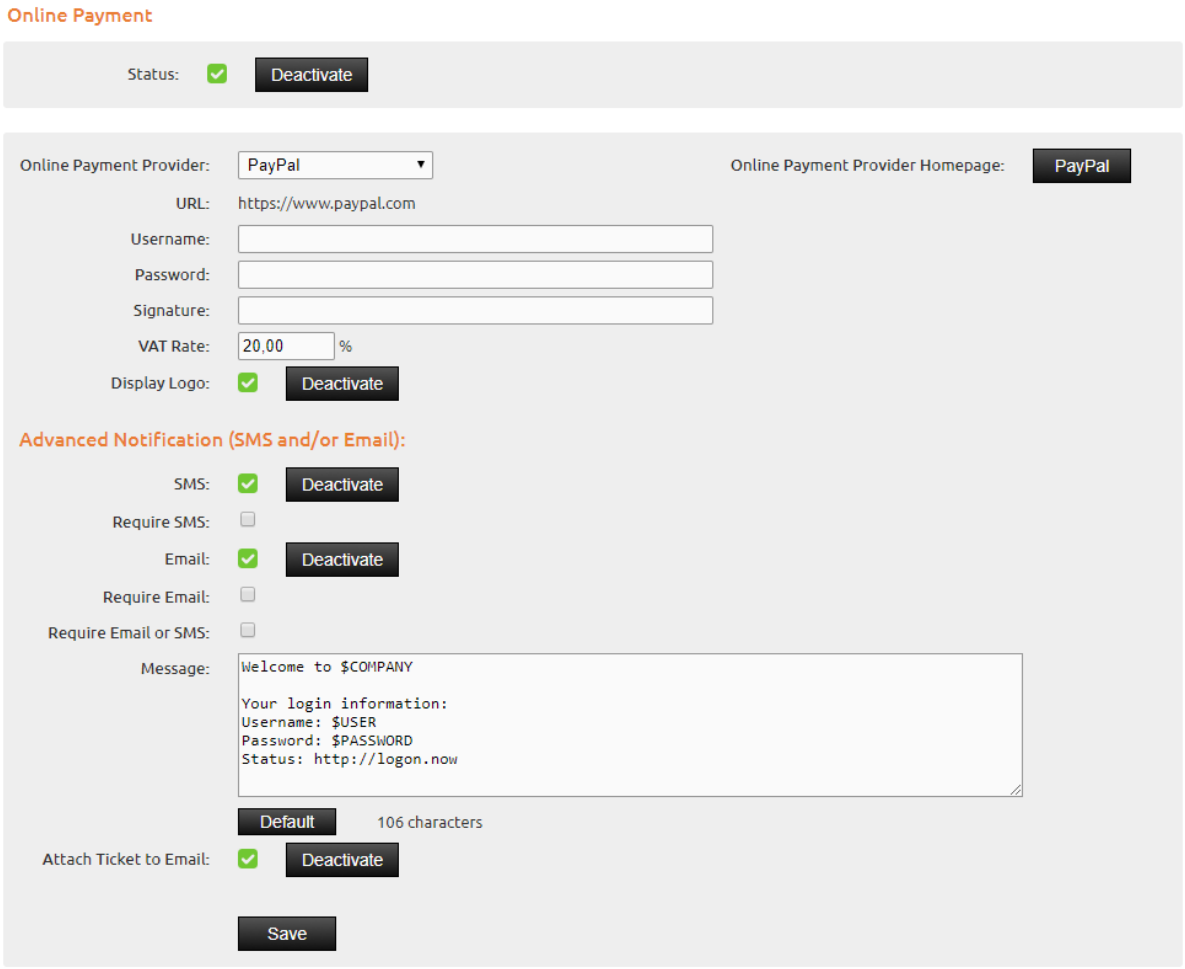
Click on **View API Signature** and write down the **API username**, **API password** and the **signature**. In the next step uncheck the *Sales tax* in the menu **local control**, because the IACBOX will automatically include taxes.

Now navigate to the menu **Website Payment Options** and verify that the **Auto Return** function is disabled. If not, disable it now.

Optionally you can make further adjustments in the menu **Custom Payment Pages**. The process of buying a surf-ticket to use the internet includes, that guests will be redirected to PayPal in order to perform a proper checkout. The *Custom Payment Pages* option allows you to customize this website (e.g. colours, corporate identity).

8.9.2 IACBOX Configuration

In the *WebAdmin* of the IACBOX navigate to **Modules / Online Payment** and activate it. Now use the dropdown menu to add an entry for **PayPal**:



Fill out the form according to the data you got from PayPal earlier. The optional **SMS** and **Email** configuration can only be used, if the according modules are configured proper in **Modules / Interfaces**.

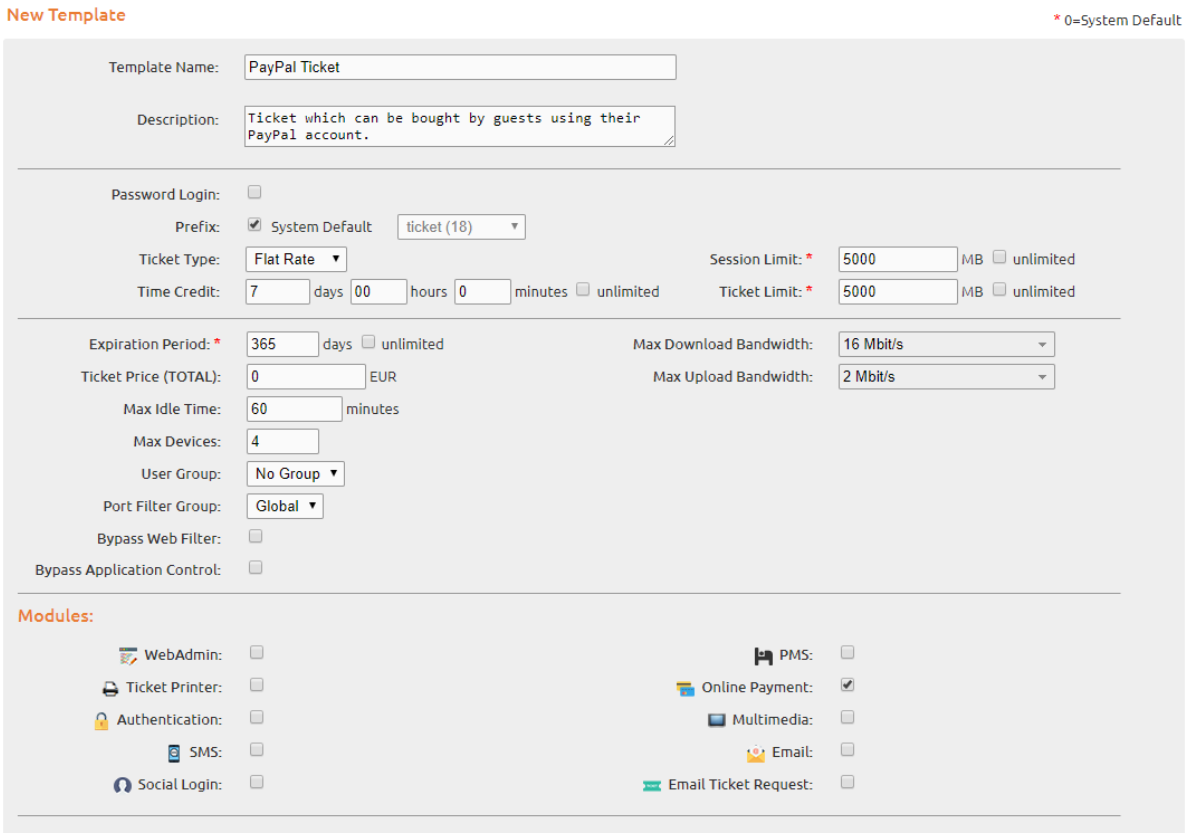
Hint:

- After you've saved the new PayPal configuration, the IACBOX must be restarted.

In the next step navigate to **Client Logon / Design** and enable the **Online Payment**.

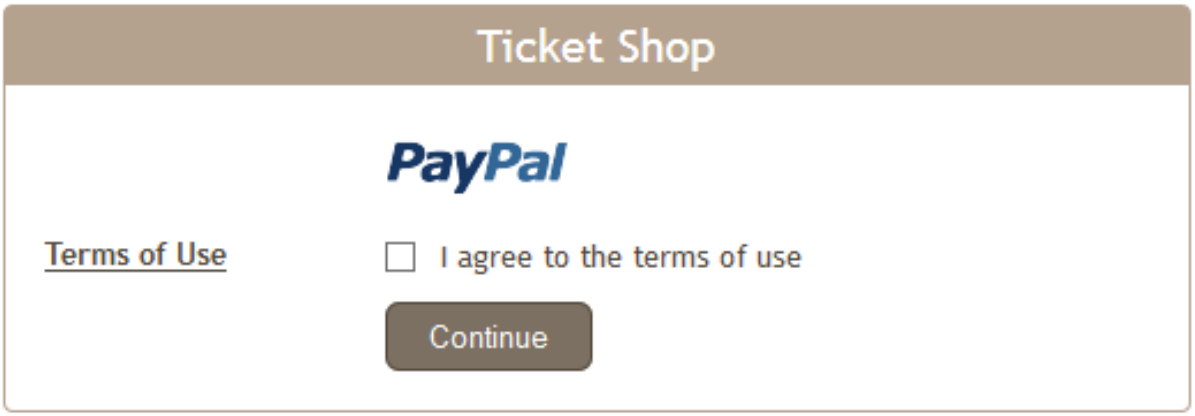
After this is done, navigate to **Tickets / Templates** and create a new *Ticket Template* which will be used for the *Online Payment*. In the screenshot below you can see an example configuration. Note that an *Online Payment* ticket must fullfil following requirements:

- Price must not be 0
- **Online Payment** must be checked unter *Modules*

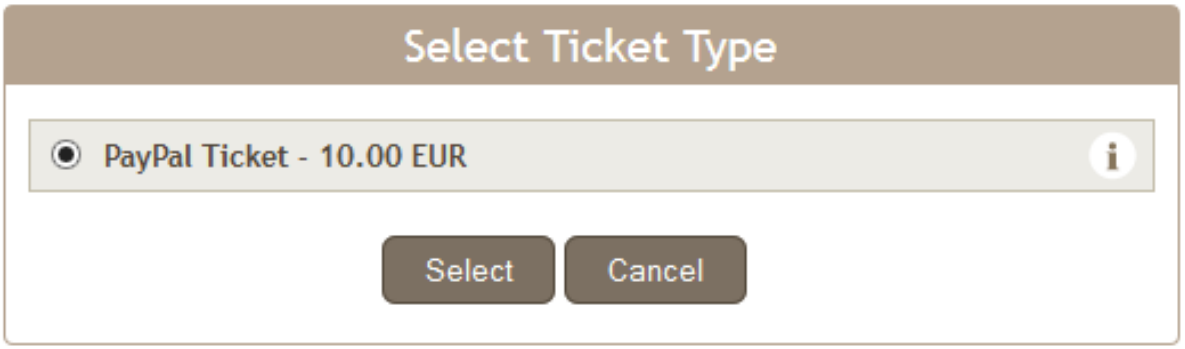


8.9.3 Client Logon Page

The customer login page now lists the **Ticket Shop** and the payment option **PayPal**.



The **Ticket Templates** you assigned for the **PayPal** module can now be purchased.



8.10 PMS Configuration

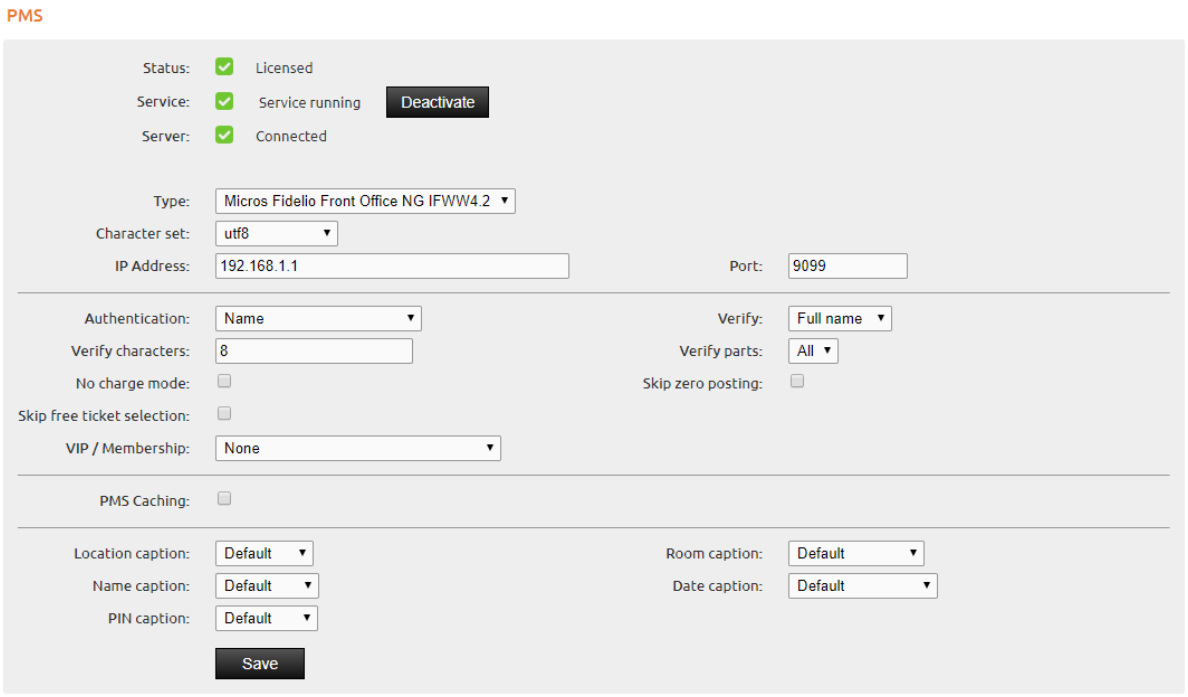
This manual describes how to configure a **PMS System** after it has been connected to the IACBOX. PMS (**Property Management Systems**) are frontoffice systems and mostly used for hotels and facilities. When connected to the PMS system, the IACBOX asks for existing customer data and uses it to verify and authenticate credentials for the login process. Expenses can be directly booked on the hotel rooms bill. The setup will be provided by your PMS IT partner. This module can be enabled in the WebAdmin menu **Modules / Interfaces**.

Hint:

- In order to use the PMS interface on the IACBOX, it must be licensed.
- By using the PMS interface, ensure that the option **Room Logon** is activated in the *WebAdmin* menu **Client Logon / Design**.
- To use this module you need to select and configure at least one **Ticket Template** for it.

8.10.1 Configuration

For the explicit configuration of your PMS system, ask your IT partner. The following screenshot merely demonstrates an example configuration.

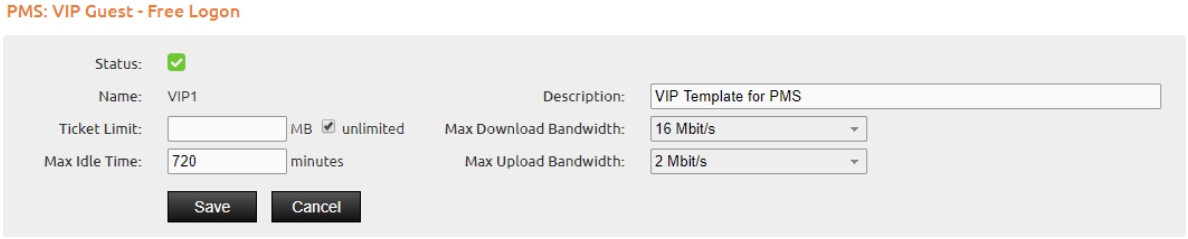


Hint:

- Pay attention to the 3 green lights on top of the configuration, refer to the **Troubleshooting** section of this manual.
- Ensure that the same **character set** is configured on both, the IACBOX and the PMS system.
- A detailed description of the field description can be found in the **help menu** of this *WebAdmin* page.

8.10.2 VIP Guest - Free Logon

This option allows you to differentiate between *default* guests and *VIP guests*. If you configure **VIP Guest - Free Logon**, then a new configuration called **PMS: VIP Guest - Free Logon** will become accessible in the *WebAdmin* menu **Tickets / Templates**. Here you can configure the amount of time, the idle timout and bandwidth which can be used for these *VIP guests*.



Hint:

- For FIAS based protocols the matching data field in the *PMS system* is the **GV** field and in some cases the **A3** field. If one of these fields is **not empty**, then the associated users qualify as *VIP guests*.

8.10.3 Membership Group Mapping

If you enable **Use VIP / Membership group mapping**, a new option called **PMS: Groups** will be accessible at the *WebAdmin* menu **Tickets / Templates**.

Hint:

- You can also use multiple user groups, a separator can be configured in the IACBOX PMS configuration.

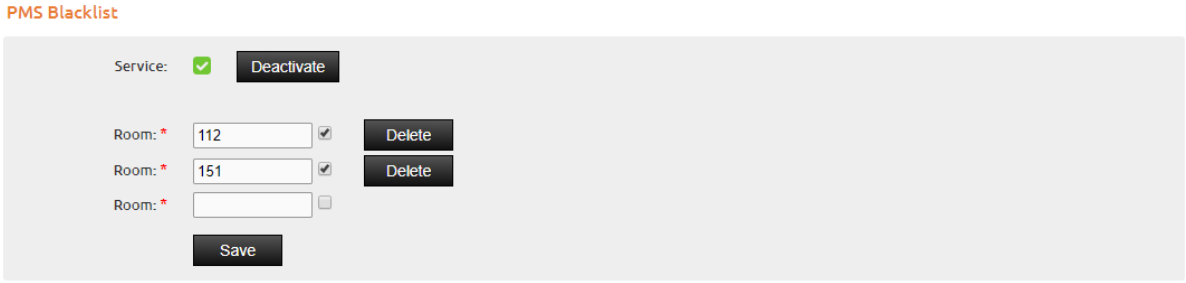
In this according menu you can create **VIP groups** and assign *different ticket templates* to each. For example with different **prices** for gold membership, platin membership (...).

Hint:

- For FIAS based protocols the according data field for **VIP groups** is the **A0** field in the *PMS system*.

8.10.4 PMS Blacklist

With the PMS Blacklist you can exclude rooms from the PMS logon.



8.10.5 Demo PMS

If you configure **Demo PMS** you can test the PMS login with predefined logon data as listed below.

Room Number	Name	Birthday
111	Max Mustermann	01.01.1961
222	John Public	01.01.1971
333	N/A	N/A

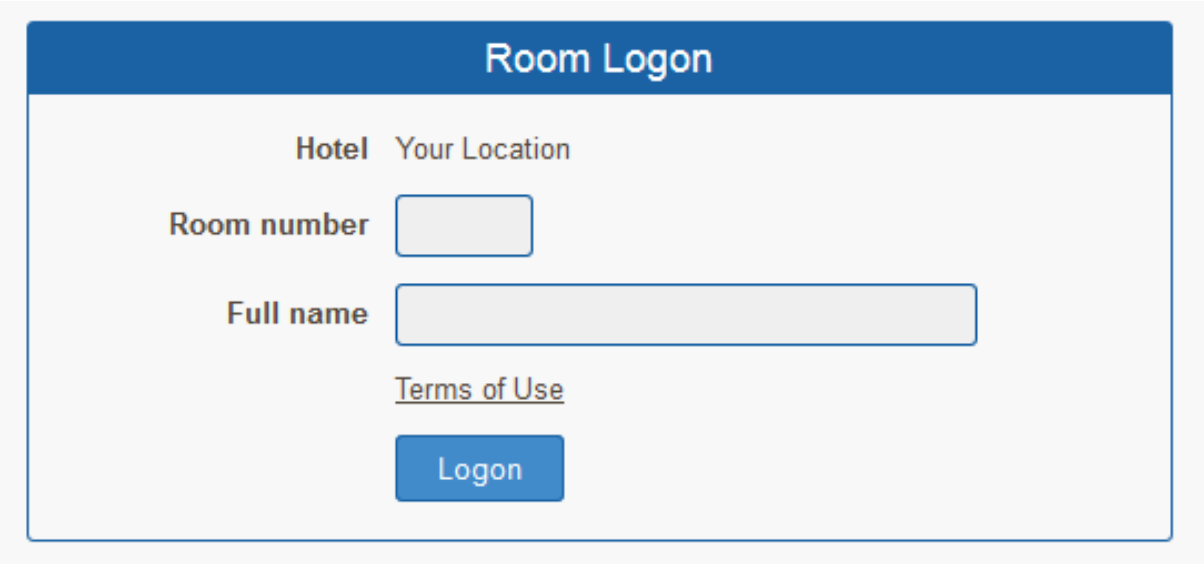
8.10.6 Troubleshooting

The most common problem with PMS systems is that the connection cannot be established. This problem can result from different conditions, for example the PMS system is not reachable in the defined network, or also the connection is not allowed over the configured port (Connection Refused). In order to test the connectivity, you need to try to establish a telnet connection with a random client in the same network (most of the time Office-LAN).

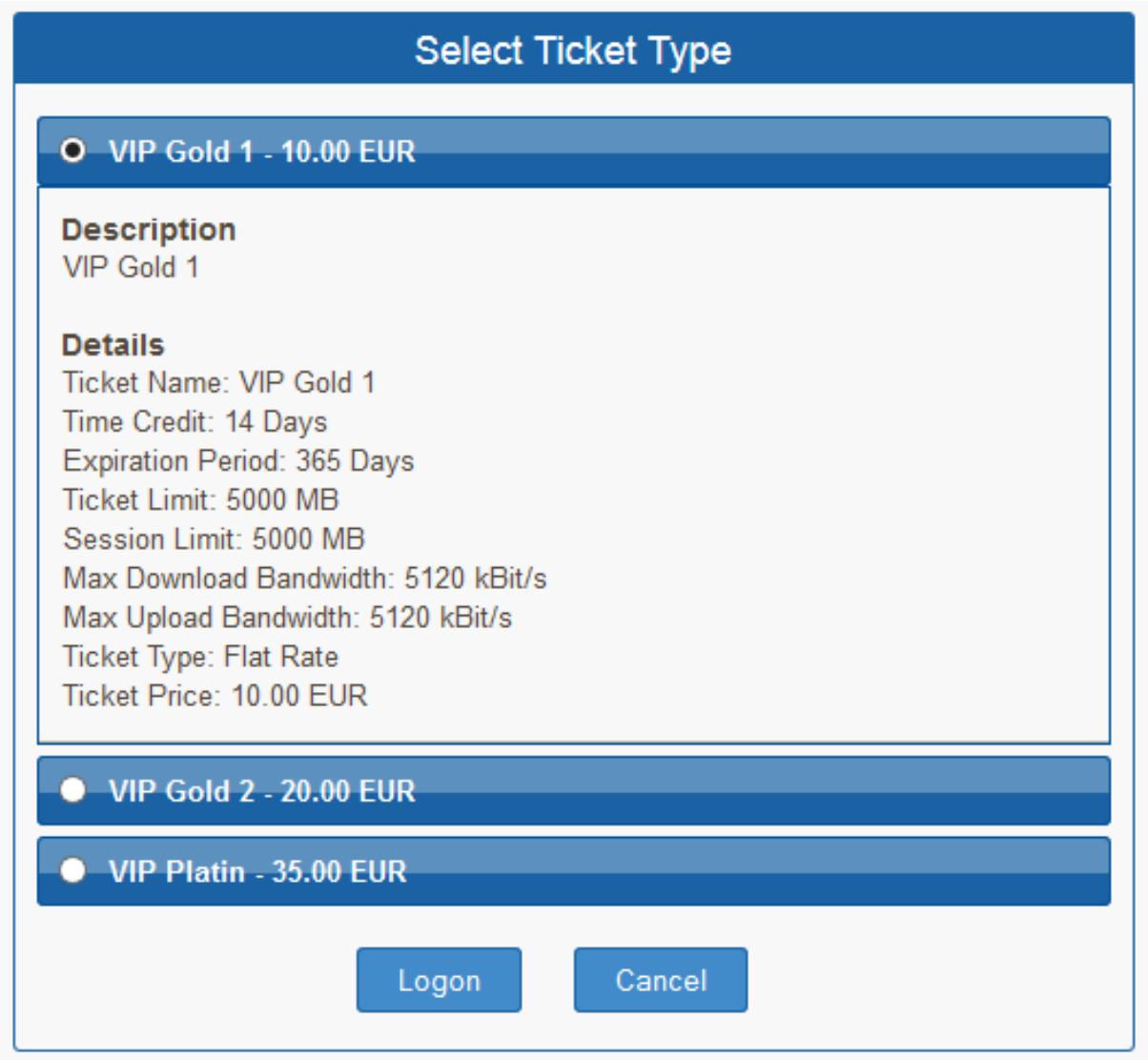
```
telnet 192.168.1.1 9099
```

8.10.7 Customer Logon Page

The logon mask for the PMS authentication will now be displayed on the customer logon page.



After entering the required data, all configured PMS tickets are listed and can be booked by this guest.



8.11 Twitter Login

Attention: The Twitter Login can only be used with the **LoginAPI** If you want to switch read the following manuals (page 180).

This manual describes how to set up a *Twitter Developer Account* and to configure a new *Twitter App* with it, so that guests can authenticate and log in on the IACBOX by using their Twitter account.

Hint:

- The login with *Twitter Credentials* is meant to be **free**, guests can not be charged.
- To setup a *Twitter Login* for your guests, you will need to create a *Twitter Developer Account* and configure a *Twitter App* with it.
- The standard callback-URL if you use the default certificate of your IACBOX is **https://hotspot.internet-for-guests.com:8443/index.php**
- In case you use a *custom certificate* on your IACBOX, pay close attention to change all *redirect URLs* according to your custom hostname.

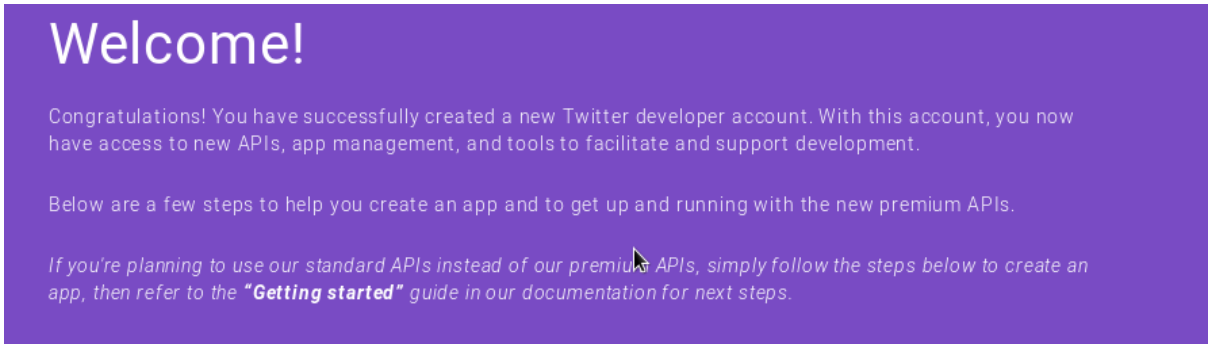
8.11.1 Developer Account

In order to create a *Twitter app*, you will require a *Twitter Developer Account*. Currently you can transform your regular *Twitter Account* into an Twitter Developer Account on this URL:

`https://developer.twitter.com/`

Click on **Apply** to start with the application of your Twitter Developer account. Fill out the forms and read the developer agreement, and verify your agreement by activating the according checkbox and click the button.

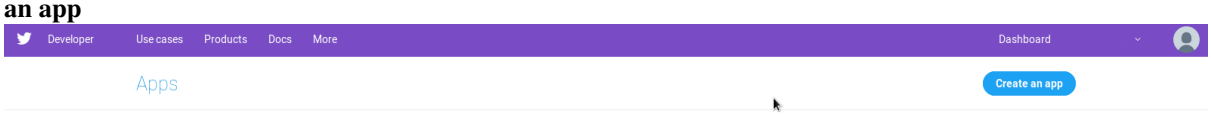
Afterwards you get send an E-Mail with a confirmation link. Click on it and you should see the following message:



Your developer account is now ready to use.

8.11.2 Creating a Twitter App

Now create a new application by clicking on you Username and afterwards Apps. On this new page click **Create an app**

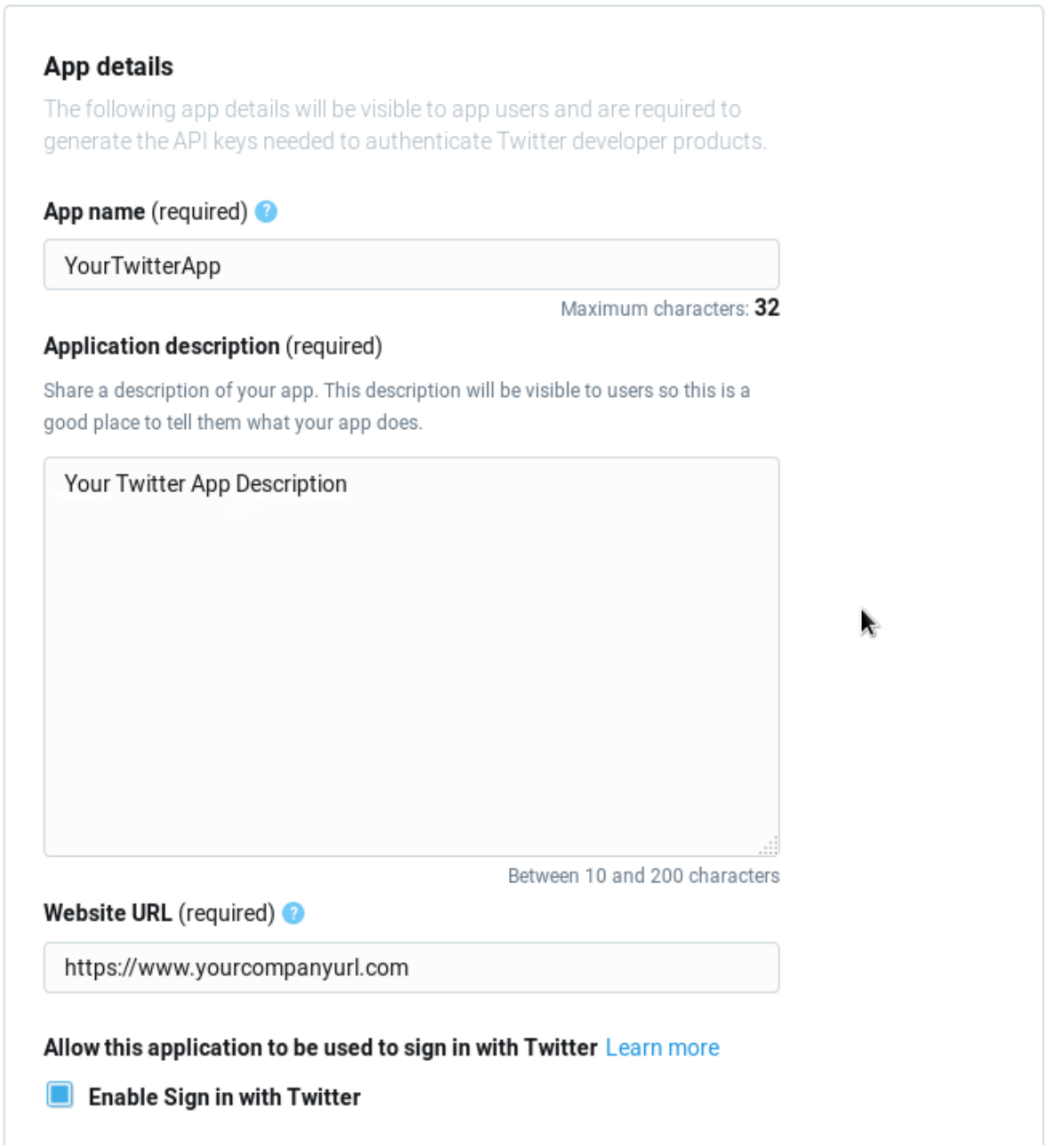


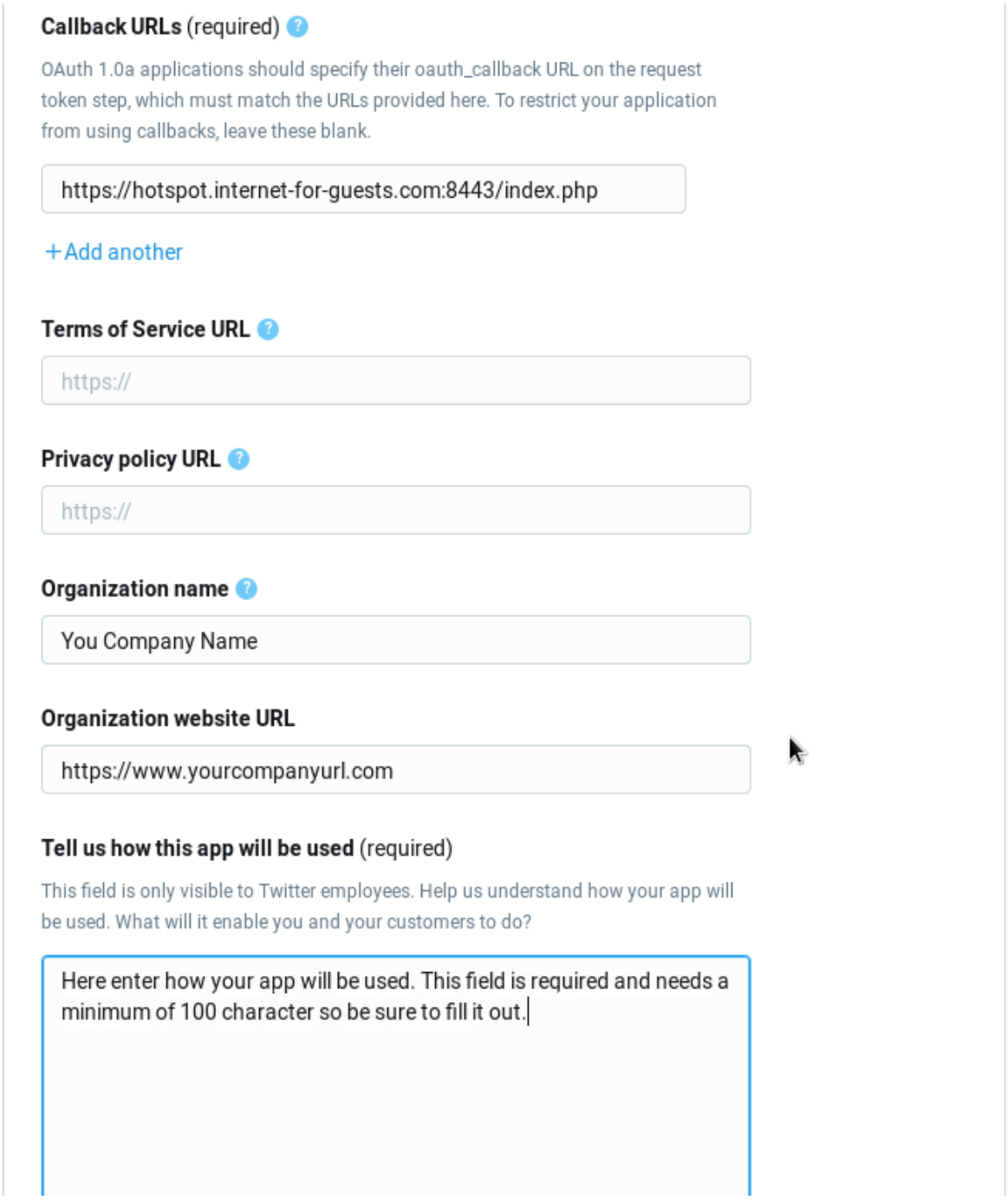
Fill out the form according to your needs. Check the box **Sign in with Twitter** and enter the correct callback-URL:

```
https://hotspot.internet-for-guests.com:8443/index.php
```

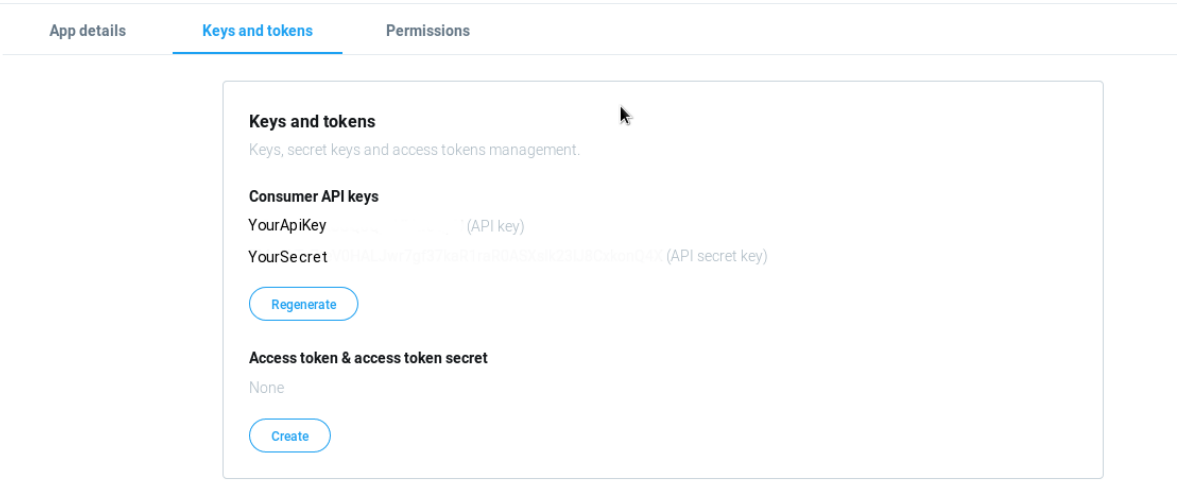
Hint:

- In case you use a *custom certificate* on your IACBOX, pay close attention to change all URLs according to your custom hostname.
- Even with a *custom certificate* you will need the given port **8443**


App details
The following app details will be visible to app users and are required to generate the API keys needed to authenticate Twitter developer products.
App name (required)
YourTwitterApp
Maximum characters: 32
Application description (required)
Share a description of your app. This description will be visible to users so this is a good place to tell them what your app does.
Your Twitter App Description
Between 10 and 200 characters
Website URL (required)
https://www.yourcompanyurl.com
Allow this application to be used to sign in with Twitter Learn more
Enable Sign in with Twitter



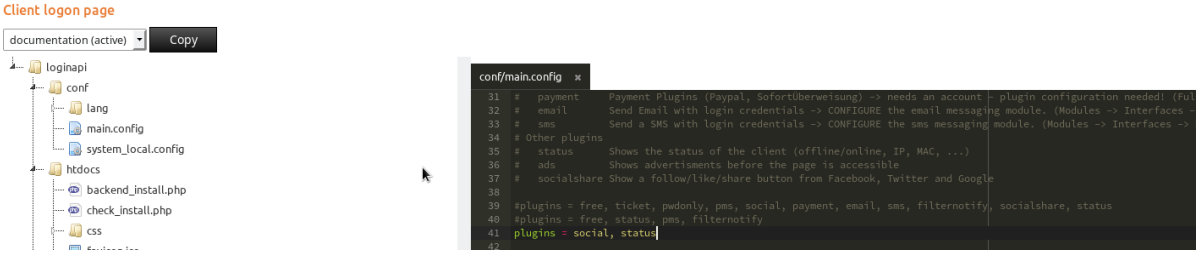
After the creation you can see the information of your created app. In the menu you can find the generated **API-Key** and **Secret**.



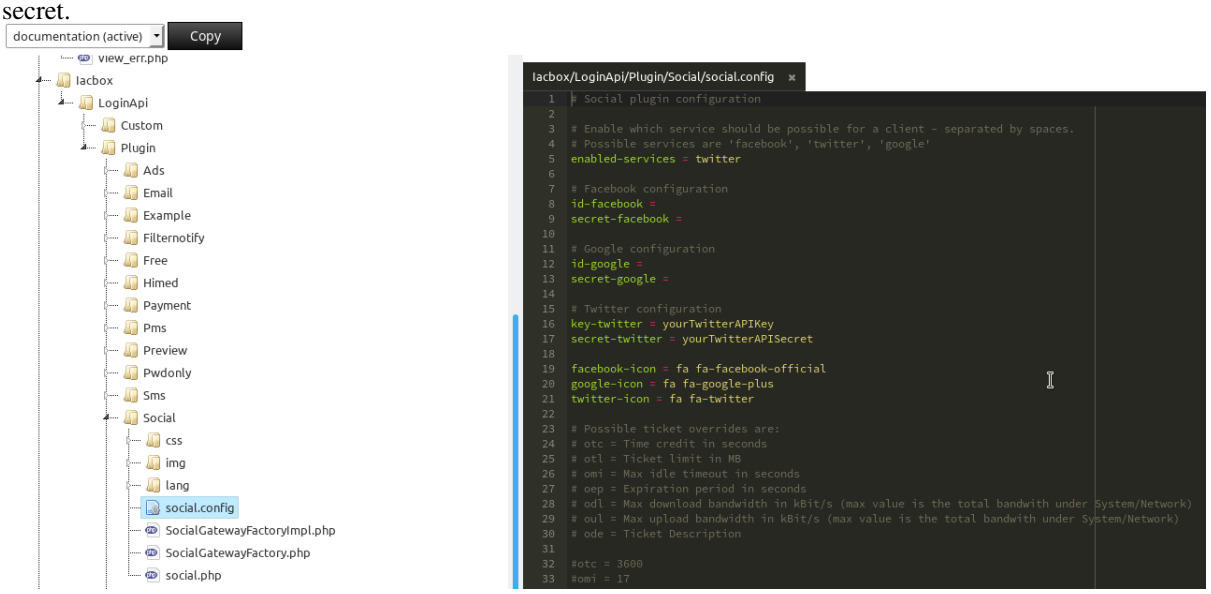
8.11.3 LoginAPI Configuration

Local LoginAPI

Navigate in the WebAdmin of your IACBOX to **Client Logon/ Custom Logon Page**. Open the file */conf/main.config* and add **social** to the configuration value *plugins*.



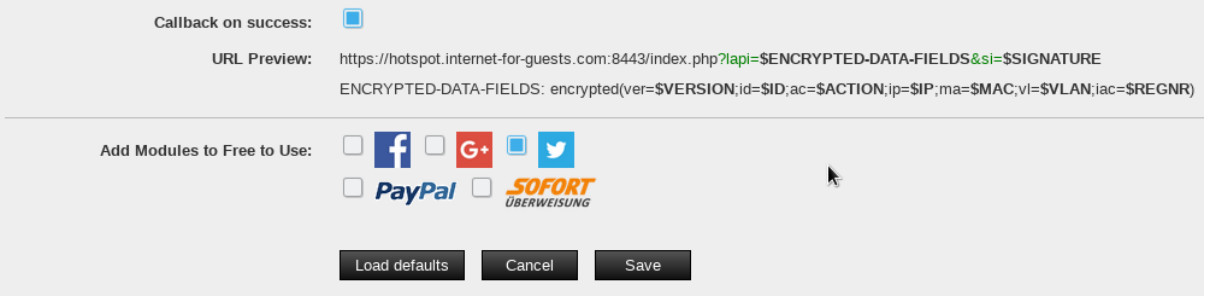
Now open the file **social.config** in the folder *Iacbox/LoginAPI/Plugin/Social* and add your Twitter API key and secret.



Attention: Don't forget to add twitter to the configuration **enabled-services** in order to access Twitter in offline state.

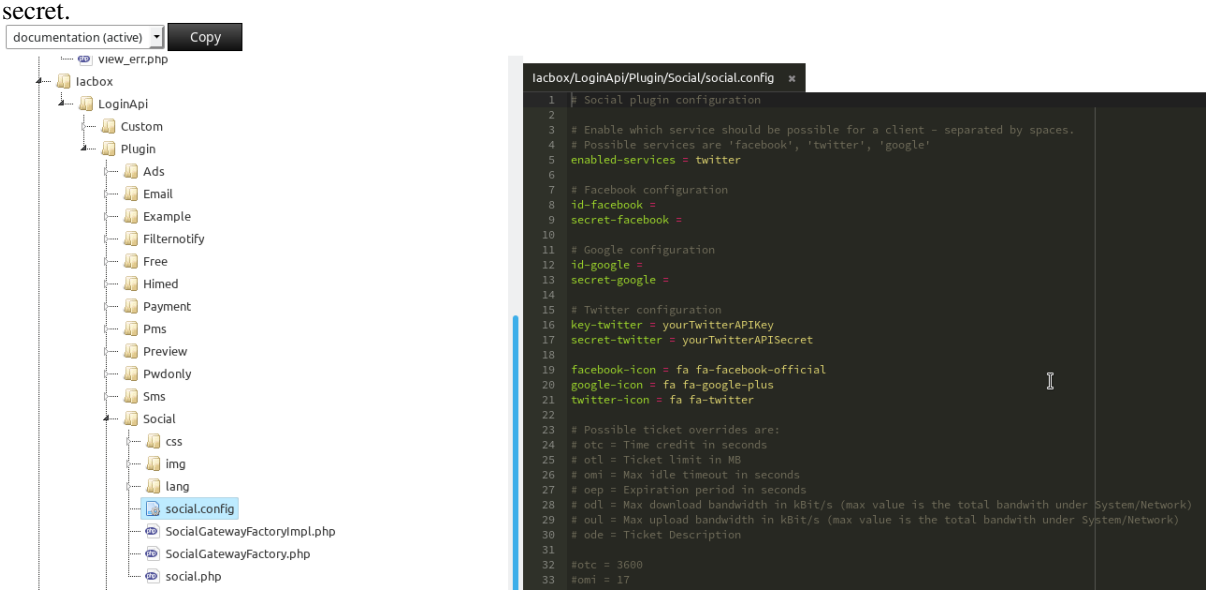
External LoginAPI

If you are using the external LoginAPI activate *Twitter* in the checkboxes below the configuration, else it is not possible to access Twitter in an offline state.



Open the file */conf/main.config* and add **social** to the configuration value *plugins*.

Now open the file **social.config** in the folder *Iacbox/LoginAPI/Plugin/Social* and add your Twitter API key and secret.



CHAPTER

NINE

LOGIN-API

9.1 Installation / Activation

9.1.1 Local mode

Since the local Login-API is pre-installed it just needs to be enabled!

- Navigate to **Modules / Custom Webserver** and activate it.
- The code-editor for easy changes directly in the webadmin interface can be found under **Client Logon / Custom Logon Page**
- Copy the default profile and adapt the file `conf/main.config`
- See the plugin configuration documentation (page 198) for everything else

9.1.2 External mode

Attention:
- The rest of this document covers the installation on an external webserver only.

Hint: The Login-API SDK is written in PHP and runs on **Linux** with **apache, nginx**, or any other webserver that is able to run PHP.

9.1.3 Preconditions

- You need the knowledge to administrate a Linux webserver. Please understand that we can't support basic server administration.
- The SDK is mainly tested with **PHP 5.6**, but should work with PHP 5.4 too.
- Some files are encrypted with the **ionCube** PHP module, so you have to install the ionCube loader in your webserver. We ship a loaders for PHP 5.4 and PHP 5.6 with the SDK.
- Since version 2.0 the way to encrypt the communication between an IACBOX and the webserver has changed to AES-256 with padding (done with PHP module **openssl**). If you are using an older version the PHP module **mcrypt** (AES-128, no padding) is necessary.
- **curl** module (only needed for social and payment plugin).
- A database is only needed if you use a plugin which depends on a specific database.
- Call the `htdocs/check_install.php` script to check your installation and **delete that file afterwards**!

9.1.4 Webserver configuration

- **The public access should be granted to the htdocs folder only!** This setting will also be checked by *check_install.php*. An alias is not enough to secure the access. If somebody gets to know the original path he/she can still access all files. In the example below we refer to the configuration of an apache server.
- **Example**: DocumentRoot /var/www/myloginapi/htdocs
- If you want to use a database as backend it is important to set the correct DB-settings and create the database first (have a look at the instructions at the end of the document howto use DBs (MySQL and PostgreSQL)).

9.1.5 Login-API configuration

Some configurations in the **conf/main.config** need to be in sync with your IACBOX configuration (Modules -> Interface -> Login-API/Custom Logon Page). Find more information in the table below:

Configuration Key	Default
webserver-url	
Necessary when using plugins which are redirecting to an external page (payment, social, ...)	
use-encryption	true
use symmetric encryption	
encryption	AES-256
Per default we use AES-256 (encrypted with the php-module **openssl**) else AES-128	
encryption-shared-secret	
Shared Secret which can be found in the Login API configuration on the IACBOX	
lgnapi-version	2.1
For backwards compatibility we differentiate between the old and new protocol version.	

Attention:
- If the configuration *lgnapi-version* is wrong some plugins will not work anymore (PMS).

9.1.6 RDBMS Backends

This is an **optional** step if you have custom plugins which need a database or you want to log to a database. We have two samples for MySQL and PostgresSQL. These two backends only differ in the create table syntax. The DBs get accessed through the PHP DB abstraction Layer PDO and support many different DBs. It should be easy to adapt these scripts for another DBMS.

Each backend comes with its own installation code. **The only precondition is that you create the database and a user for it first** (do not run with admin or root users in production!).

MySQL

```
# mysql -u root -p
mysql> create database iacbox_loginapi;
mysql> grant usage on *.* to loginapi@localhost identified by 'new_password';
mysql> grant all privileges on iacbox_loginapi.* to loginapi@localhost;
```

Replace *new_password* with a long random password (> 10 chars).

PostgresSQL

```
# createdb -h localhost -U postgres iacbox_loginapi
# psql -U postgres iacbox_loginapi
loginapi=# CREATE ROLE loginapi WITH PASSWORD 'new_password' NOSUPERUSER NOCREATEDB NOCREATEROLE;
loginapi=# ALTER DATABASE iacbox_loginapi OWNER TO loginapi;
```

Now create the tables with install-script: Open this link in your browser: `http://your.domain-or-ip.com/loginapi/backend_install.php` and click on *Install*. If everything went fine you should see an *Ok* for each table.

Attention: REMOVE the *backend_install.php* and *check_install.php* after installation since leaving them in-place is considered a security risk!

9.2 Software development kit

New in version 6.0: SDK for external mode

New in version 8.0: SDK for external and local mode

Current SDK Version: 18.0

Hint:

- The SDK comes preinstalled on every IACBOX with the *custom logon page/custom webserver*.
- If a central installation (for many systems) is needed, the SDK can be installed on an external webserver.

The SDK provides a small PHP framework which abstracts all the tiny details away and lets you easily make customization and/or create new plugins and extensions.

The SDK written in PHP provides you with a sample login page and plugins for many different authentication methods. Starting with version 2 the SDK is also used for the local custom webserver.

9.2.1 Downloads

You will find the SDK for external use in the download section of our homepage or in the my.iacbox partner-portal.

9.2.2 Login-API operation modes

Ahead of any other decisions that have to be made, you have to be aware of two very different operation modes of the Login-API. You have to choose which one serves your needs:

1. External Mode: this is the normal mode of version 1.x. The Login-API SDK is hosted on an external webserver by yourself.

- **Advantages:**
  - Allows to point any number of IACBOXes to this login page.
  - One single place to make changes.
  - Centralized data storage possible.
- **Disadvantages:**
  - You have to host the SDK on an external webserver which has to be maintained and configured by yourself.
  - Slower than local mode and consumes uplink bandwidth.
  - Without any failover architecture this external server is a single point of failure.

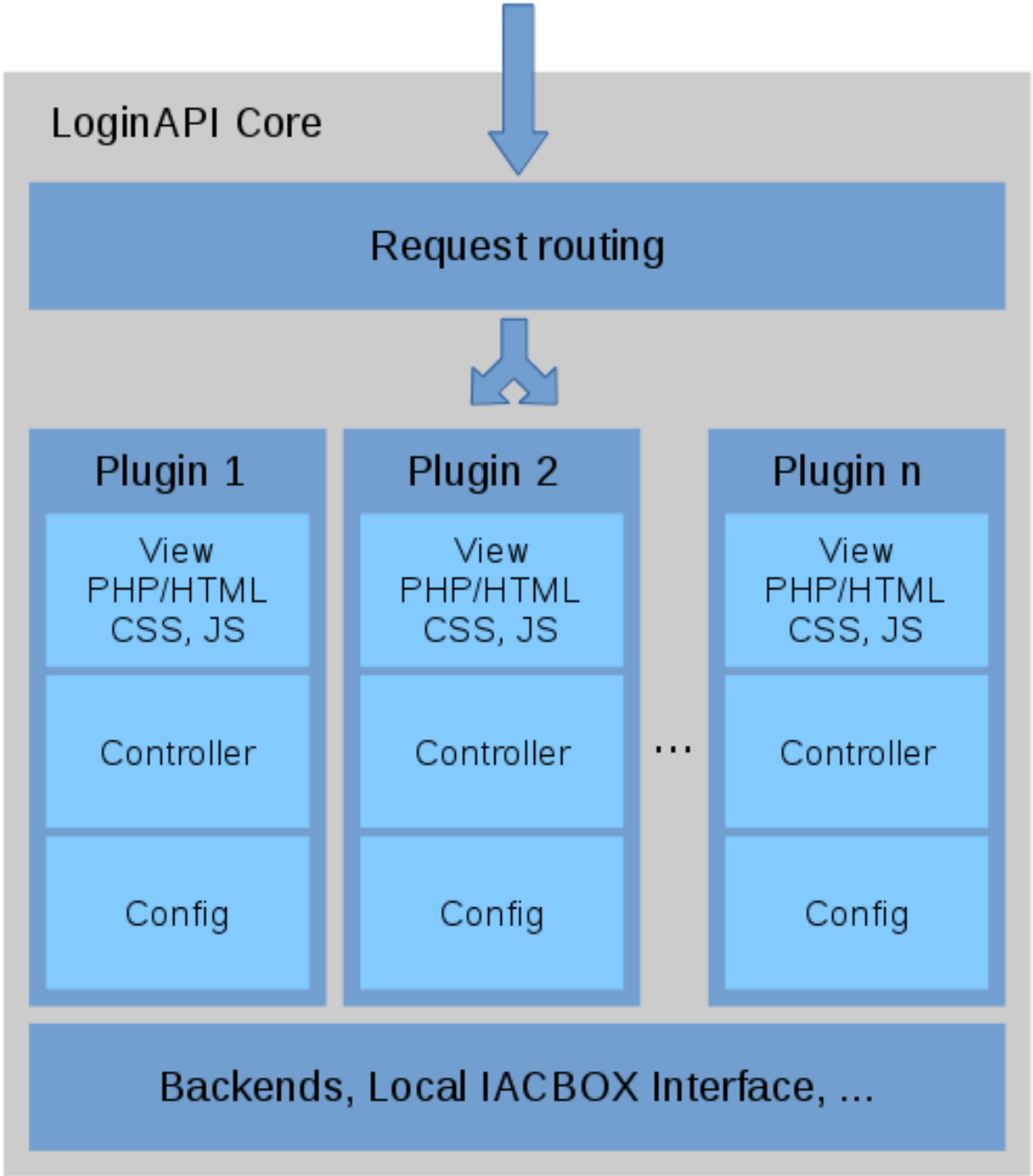
2. Local Mode: Since version 8.0 an IACBOX comes equipped with a **custom logon webserver** with PHP 5.6 and the preinstalled SDK replacing the traditional loginpage. So this can also be used to just change the look of the login page like it wasn't possible before.

- **Advantages:**

  - The login page ist fast no matter how slow or unreliable your uplink is.
  - The traffic to your login page is only local and does not utilize your upstream.
  - Make easy and fast changes with the built-in code editor.
  - Fewer redirects compared to the external mode, less complex plugins.

- **Disadvantages**
  - If you are using multiple systems you have to make your changes on every IACBOX seperately.
  - If you want to manage a central database with authentication data and having logs of logins the local mode still allows that but it is harder to achive
  - Currently no local database available.

Login-API SDK architecture



9.2.3 Configuration

The configuration of the Login-API SDK is handeled in two different places. The main configration can be found in **conf/main.config**. Everything important regarding the Login-API is handeled there.

Configuration key	Since	Default
profile-desc	2.0	[profilename]
Name of the profile		
company-name	2.0	
Will be the title of the logon page		
company-name-legal	2.0	
Shown on the end of the logon page		

Continued on next page

Table 9.1 – continued from previous page

Configuration key	Since	Default
logo	2.0	login_logo.png
Logo which should be used		
background-image	17.0	login_bg_full.jpg
Backgroundimages shown on the logon page		
show-welcome-header	17.0	true
Should the Headermessage be shown		
show-lang-select	17.0	true
Should the language selection be shown		
template	17.0	index_view.php
Determines which template will be the default tempalte		
plugins	2.0	
Plugins which should be used for the logon.		
languages	2.0	en
Available languages for the logon page. Only needed for the external version		
fallback-language	2.0	en
Will be used if a language could not be found		
location-based	2.0	false
Enable location based configuration		
location-id-fields	2.0	iacbox
Fields to determine different locations		
base-url	2.0	https://hotspot.internet-for-guests.com
Should not be changed except you use another base-URL for the Surf-LAN		
webserver-url	2.0	
Url of your Webserver used for plugins with callbacks from externel servers		
sender-name	2.0	
Name of the sender (used in payment plugin only)		
sender-email	2.0	
Email address which will be shown in email (used in payment plugin only)		
log-level	2.0	INFO
Determines the log level for the Login-API		
redirect-url	2.0	
Redirect **after** the login		
show-terms	2.1	true
Show the terms of use (no checkbox)		
show-privacy-policy	2.1	true
Show the privacy policy (no checkbox)		
force-terms	2.0	true
Force to accept the terms of use		
force-privacy-policy	2.1	false
Force to accept the privacy policy		
encryption	2.0	
Has to match the configuration in **Module/Interfaces/Login-API** on the IACBOX		
encryption-shared-secret	2.0	
The shared secret can be found in **Module/Interfaces/Login-API** on the IACBOX		
use-browser-auto-settings	2.0	
The Browser should use autocomplete, autocapitalization, spellcheck and autocorrect features		
custom-services	2.0	
Register one or more custom services (separated by commas)		
test-template	2.0	
Creates a fake session for testing purposes		
refresh-id-mapping	2.0	
Refresh call to a certain page		
lgnapi-version	2.0	

Continued on next page

Table 9.1 – continued from previous page

Configuration key	Since	Default
Version of the API / protocol		
use-rdbms	2.0	
Loads the RdbmsConnector to hold and manage a connection to your database.		

Attention:
- Please note that the configuration **use-rdbms** can only be used with an external DB
- Create a secure connection to your DB! (e.g.: VPN)

The second part of the configurations are the plugin specific configurations. This files can be found in **Iacbox/LoginApi/Plugin/[Plugin]/[plugin].conf**. Which configurations have to be done depends on the plugins you have activated in **conf/main.conf**. For further informations which configurations are possible read the Plugin configuration (page 198) documentation.

Existing plugins

Pluginname	Since	Type	Usage
Ticket	1.0	AuthPlugin	loc, ext
Ticket/ Voucher Login			
Pms	1.1	AuthPlugin	loc, ext
PMS Login			
PwdOnly	1.4	AuthPlugin	loc, ext
Password only Login			
Free	2.0	AuthPlugin	loc, ext
Free login coupled with Free Logon or Take online			
Social	2.0	AuthPlugin	loc, ext
Social login (Facebook, Google+, Twitter)			
Payment	2.0	AuthPlugin	loc, ext
Payment plugin (PayPal, SofortÃijberweisung)			
Email	2.0	AuthPlugin	loc, ext
Email Login coupled with Email Messaging Module			
Sms	17.0	AuthPlugin	loc, ext
SMS Login coupled with SMS Messaging Module			
Status	2.0	Others	loc, ext
Status information of the client			
Ads	2.0	Others	loc, ext
Show advertisement before logon			
Socialshare	2.0	Others | loc, ext	
Show sociale like/share buttons			

loc = available in local mode, ext = available in external mode

9.2.4 Styling

Template changes can be made on two different places. The structure of the main page can be changend in **htdocs/index_view.php**. It is also possible to creae your own page and include it in the *htdocs/index.php*. The main CSS file can be found in **htdocs/css/style.css**. We recommend to use the overlay file **htdocs/css/style_overlay.css** for changes to the CSS because the *style.css* will be updated regularly.

Some plugin specific CSS classes can be found in the corresponding CSS file **Iacbox/LoginApi/Plugin/[Pluginname]/css/**. If there is a need to change the template structure of a plugin this can be done in the file **[pluginname](_local).php**. Please note that if you create your own plugin you are not bound to the naming convention for templates we are using in the already existing plugins.

9.2.5 Translations

General translations are stored in **conf/lang/[language].lang**. Plugin specific translations can be found in **Iacbox/LoginApi/Plugin/[Pluginname]/lang/**. The LoginApi provides translations for all 23 languages of the normal logon page but not alle languages are fully translated.

Hint:

- Since version 17 it is possible to add translations for the selection of gender and building mapping in the PMS configuration.
- The translation-keys need to be lowercase and seperated per "-" (dash)
- Example: building-mapping = 1:house-a; 2:house-b

9.2.6 Logging

System log

Externale Mode

As default the SDK logs to your syslog - depending on your Linux distribution this is **/var/log/syslog** or **/var/log/messages**. On newer systems which use **systemd** you can see your logs with `jounalctl -f`. All messages start with *LGNAPI* which makes it simple to filter (adding `| grep LGNAPI`). If you encounter any errors (especially white pages what means PHP had an fatal error) then you should look into your webserver log - for apache this is very often **/var/log/apache2/error.log** or similar.

Local Mode

The SDK logs per default to the syslog of the IACBOX. All messages start with *LGNAPI* which makes it simple to filter. The System logs can be downloaded in the menu *Reporting/System*.

If you encounter any errors (especially white pages what means PHP had an fatal error) then you should look into the log of the Login-API webserver. This log can be accessed if you connect via FTP with the user **sysop**.

Database

If you want to log the messages in a database, which is available in both modes, there is already a custom service called **DBLogger** part of the SDK. This is made for external Login-API but theoretically possible in local mode too when writing to an **external** database (But this is propably a performance issue). Be sure to have this config lines in your `main.config`

Listing 9.1: conf/main.config

```
use-rdbms = true
db-type = <database type like pgsql or mysql>
db-host = <database host>
db-name = <database name>
db-user = <database user>
db-pwd = <database password>
```

9.3 How to create your own plugin

If you need a login method which is not covered by the shipped plugins you can easily add your own. In the following how to we will cover all needed steps to create your own plugin.

9.3.1 Name conventions

Type	Mandatory	Example
Plugin folder	yes	Free
Capitalize the first letter, pluginname equals Foldername		
Class and filename	yes	FreePlugin FreePlugin.php
Capitalize the first letter, expression "Plugin" has to be added		
Configuration file	yes	free.config
Name of the folder, file ending has to be .config		
Language file	yes	lang
Has to be this name		
Language files	yes	free.en.lang
Lowercase, short country code (e.g. en), file ending has to be .lang, everything separated through a "."		
View-file of a plugin	no	free.php
Lowercase, name of the folder		
Css and javascript	no	css/ js/
Folder names have to be in lowercase		
Css and javascript files	no	free.css free.js
Lowercase		

9.3.2 Needed Steps

Please note that some name conventions are mandatory. You can see this conventions in the table above. The other file and folder names are examples we are using in the SDK code you are free to modify them.

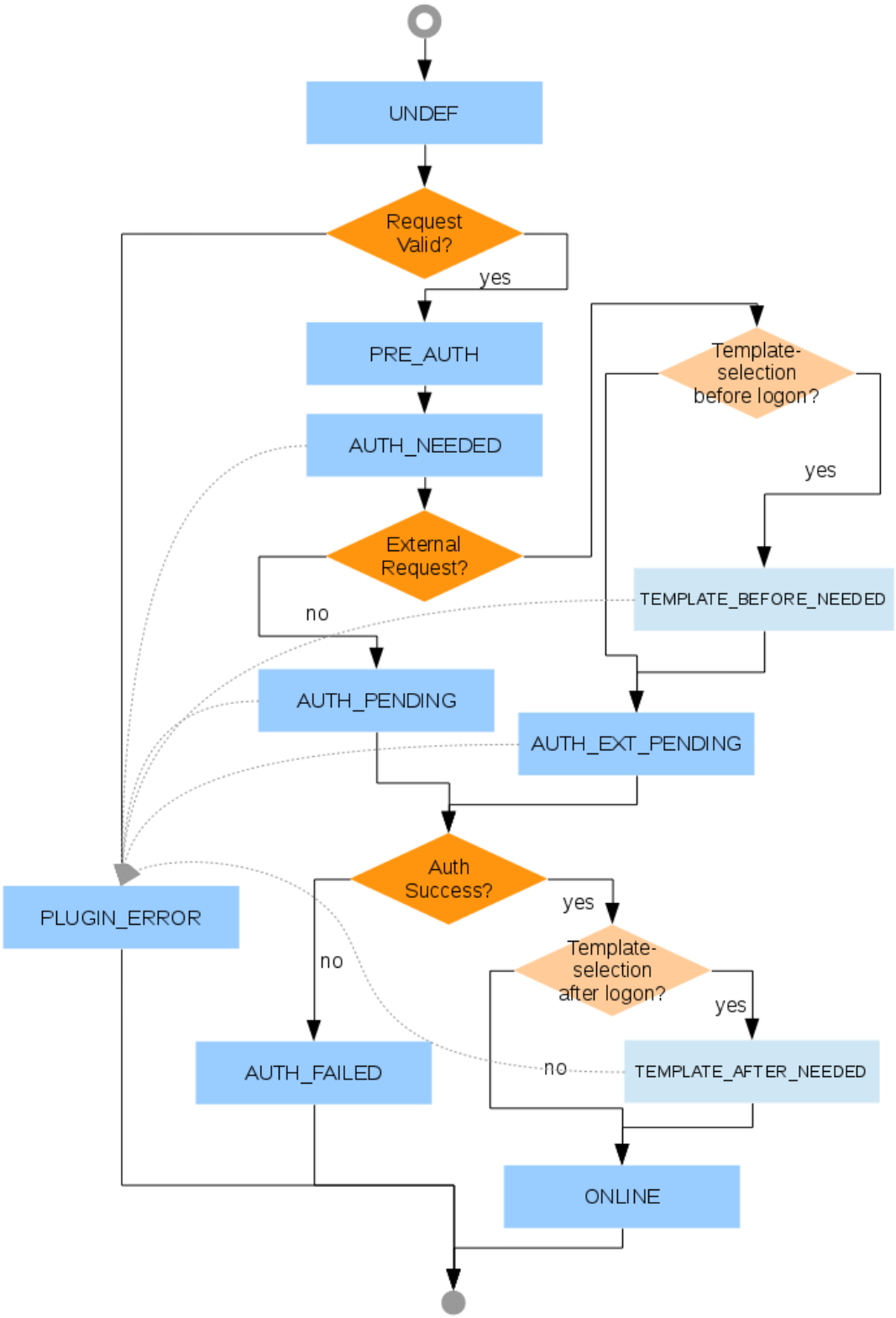
1. Create a folder with the desired plugin name in the folder **Plugin**
2. Add the following files and folders:
    - **[Pluginname]Plugin.php**
        - Main code of the plugin

        - Class name must be identical with this name
    - **[pluginname].php**
        - View file of the plugin
    - **[pluginname].config**
        - Configuration file
    - **Folder "lang"**
    - **[pluginname].[lang].lang**
        - Has to be in the folder **lang**
        - For each used language a language file has to be added
    - **Optional: CSS and JS files**
3. **Implmentation details:**
    - Your plugin class has to implement the interface **Plugin**
    - In a regular case your plugin will use the PHP *traits* **VisualPluginDefaults** and **AuthPluginDefaults** to include common code.

9.3.3 States

Using states correctely is important to have a working plugin as this is the main criteria what has to happen next whatÂt's especially important for multi-step authentication methods like PMS or external method like social or payment login. During authentication a client is in different states. This state is always saved in the **Session** object.

- **UNDEF**: Initial state or unresolved state
- **PRE_AUTH**:We have a valid redirect and a session but no auth plugin was chosen (during first page rendering)
- **AUTH_NEEDED**: Request parsing was ok - waiting for the user to choose a login-type
- **AUTH_PENDING**: We are waiting for an IACBOX callback with an success/error code
- **AUTH_EXT_PENDING**: If the backend needs external redirects (like social or payment logins)
- **TEMPLATE_BEFORE_NEEDED**: Mainly for prepaid tickets (paypal, etc) - to know what to charge a selction has to be made
- **TEMPLATE_AFTER_NEEDED**: Mainly for PMS - authentication was ok, but template selection is needed but template selection is needed
- **TICKET_CREATION_PENDING**: If AUTH_EXT_PENDING was successful, the ticket creation on the IACBOX is pending
- **ONLINE**: Authentication was successful - user is online
- **TEMPORARILY_ONLINE_PENDING**:User is taken online and an action is required
- **TEMPORARILY_ONLINE**: User is taken online for a short amount to make an action
- **ERRORS_START**: Marker
- **ROUTING_ERROR**: General error state for the login API
- **PLUGIN_ERROR**: Something went wrong inside the plugin, or with a needed external backend (like no connection to the PMS, ...)
- **AUTH_FAILED**: Authentication failed



9.3.4 Useful functions for creating your own plugin

Before we start we will describe with some useful methods you can use when developing your own plugin.

Get config values

We provide a number of methods to get the config from your plugin config file.

- **$key**
  - Represents the key (given as String) from the value you want to use
  - Example: *example-icon*
- **$section**
  - This parameter is optional
  - You can hand over a specific section as String
  - Example: *v17*
  - We recommend to use a more generic approach as we provide the method *$this->loginApi->getLocationId()*
- **$default**
  - This parameter is optional
  - Define a default value if the config could not be found
- **$keyValueDelimier**
  - This parameter is optional
  - Only available for the method **getMap**
  - Use only a specific field delimiter instead of default ":" colon
- **$entryDelimiter**
  - This parameter is optional
  - Only available for the methods **getlist** and **getMap**
  - Use only a specific field delimiter instead of default [s,;]+

```php
<?php

// Returns a config value as string
$this->config->get($key, $section, $default);

// Returns a numeric value casted to an integer
$this->config->getInt($key, $this->loginApi->getLocationId(), $default);

// Returns a numeric value parsed as float
$this->config->getFloat($key, $this->loginApi->getLocationId(), $default);

// Returns a boolean value of a config kex in the given section
$this->config->getBoolean($key, $this->loginApi->getLocationId(), $default);

// Returns a list of splitted strings, - used delimiters are whitespace, "," comma or ";" semicol
$this->config->getList($key, $this->loginApi->getLocationId(), $default, $entryDelimiter = null);

// Returns a map - used delimiters are whitespace, "," comma or ";" semicolon.
// To seperate the list in a key => value the delimiter ":" is used
$this->config->getMap($key, $this->loginApi->getLocationId(), $default);
```

Get location Id

The method **getLocationId()** enables you to get the id with which the client hits the Login-API. It is crucial to use this method if you want to have a generic approach in your plugin and use the location-based configuration from the **conf/main.config**.

```php
<?php
$this->loginApi->getLocationId();
```

Get translations

With the method **translate()** you can call translation keys you defined before hand in your lang files.

- **$key**
  - Represents the key (given as String) from the value you want to use
  - Example: *example-icon*

```php
<?php
$this->translate($key);
```

Check if the Login-API is used locally

The method **isLocal()** is an easy check if you use the local Login-API. Returns true if used locally or false.

```php
<?php
$this->loginApi->isLocal()
```

9.3.5 Code Example Main Implementation

Attention:
- In the whole example we use **Example** as plugin name
- We will start with the Plugin implementation itself

First of all you need to add the namespace to the plugin and add the import of the namespaces which we are listet in the code example below.

Listing 9.2: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

namespace Iacbox\LoginApi\Plugin\Example;

use Iacbox\LoginApi\Core\AbstractPlugin;
use Iacbox\LoginApi\Core\AuthPlugin;
use Iacbox\LoginApi\Core\VisualPlugin;
use Iacbox\LoginApi\Core\AuthPluginDefaults;
use Iacbox\LoginApi\Core\VisualPluginDefaults;
use Iacbox\LoginApi\Core\Session;
```

Next you need to define the class. Extend your Plugin from the class **AbstractPlugin** and implement the Interfaces **AuthPlugin**, **VisualPlugin**. To use some default implementation we are providing use the traits **AuthPluginDefaults**, **VisualPluginDefaults**.

Listing 9.3: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php
class ExamplePlugin extends AbstractPlugin implements AuthPlugin, VisualPlugin {
use AuthPluginDefaults, VisualPluginDefaults;
```

In the next step you need to define a name for your plugin how it should be able be called in the **conf/main.config** and a name how the Plugin should be displayed. The name has to be the same as the plugin.

For the next part we are going to define a icon and the name which should be displayed on the landing page. We will use the method to get the config which was described before. We recommend to copy the code provided below and change the keys according to your needs.

- **renderIcon($addCss = null)**

Listing 9.4: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php
        /**
         * @inherit
         */
        public function getName() {
                // This is the internal name of the plugin used in the config - this has to be ex
                return 'example';
        }

        /**
         * @inherit
         */
        public function getDisplayName() {
                return 'Example of a plugin';
        }
```

  - The param $addCss is optional
  - The method used outputGlyphicon() adds the glyphicon with the default class "icon" and the class you defined if you use the param $addCss

- **renderName()**
  - Add the translation of the plugin name you want to be displayed on the landing page
- **render($slot)**
  - Renders the plugin and defines were it should be positioned on the landing page with the param $slot
  - **Possible Slots:**
    - *SLOT_AUTH* Positions the plugin in the main section of the landing page (e.g. Ticket plugin)
    - *SLOT_BEFORE_FOOTER* Positions the plugin before the footer (e.g. Status plugin)
    - *SLOT_BOTTOM* Positions the plugin below the footer

In the next part you can include css and javascript files. We positioned the possibility to add css and js files to the plugin to load them only if they really are needed. You can add as many css and js files as you need for your Plugin. In the method **getJsIncludes()** you can also see how to differentiate between local and external mode.

In the last part of the main implementation of a plugin we are going to the core part. In the method **processRequest()** we are handling the different behaviour of the plugin. To navigate through the different phases of the plugin we are using states. Find all the possible states above under the point States. In our example we are going to create a simple ticket login. We will also provide an implementation for the local and the external mode.

We are starting with the backbone of the plugin. First you need to think of which states are needed to route through your plugin correctly. The authentication always starts with the State AUTH_NEEDED. Because the local Login-API is on the IACBOX it is not needed to use any further states but in the external mode (Login-API lies on an external webserver) we will need to use the AUTH_PENDING state. We also check if the the plugin was entered with the wrong state (every state except AUTH_NEEDED, AUTH_PENDING, ONLINE).

- **$getParams**
  - In production LoginApiImpl passes $_GET.
- **postParams**
  - In production LoginApiImpl passes $_POST.
- **&$session**
  - State with the information about the session. Is called by reference.

Listing 9.5: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

/**
* @inherit
*/
public function renderIcon($addCss = null) {
        // This is called when the icon gets rendered for this plugin.
        // In the mobile version only the icons will be shown
        $this->outputGlyphicon($this->config->get('example-icon', $this->loginApi->getLocationId(
}

/**
* @inherit
*/
public function renderName() {
        // Renders the name of this plugin shown to the user in desktop mode.
        // The name is translated, so a language key is used.
        // Find further informations for translate() in the PHPdoc
        echo $this->translate('example');
}

/**
 * @inherit
 */
public function render($slot) {
        // The template index_view.php calls this method at different places with different slot
        // The usual slot for authentication plugins is SLOT_AUTH
        // Possible slots are SLOT_AUTH, SLOT_BEFORE_FOOTER, SLOT_BOTTOM
        if ($slot == VisualPlugin::SLOT_AUTH) {
                // Include the representation of the plugin for the LoginApi login page
                include(__DIR__.'/example.php');
        }
}
```

Listing 9.6: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

/**
 * @inherit
 */
public function getCssIncludes() {
        // If a own CSS file is needed for this plugin it can be defined here and will be automat
        // Please note that it's necessary to add the folder path ([Plugin]/[name]/[name].css).
        return array('Example/css/example.css');
}

/**
 * @inherit
 */
public function getJsIncludes() {
        // If a own JavaScript file is needed for this plugin it can be defined here and will be
        // Please note it is necessary to add the folder path ([Plugin]/[name]/[name].js).
        // It's also possible to differentiate between local and external mode
        $js = array();
        if ($this->loginApi->isLocal()) {
                array_push($js, 'Example/js/example_local.js');
        } else {
                array_push($js, 'Example/js/example_external.js');
        }
        return $js;
}
```

Listing 9.7: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

/**
 * @inherit
 */
public function processRequest($getParams, $postParams, &$session) {
        if ($session->getState() == Session::AUTH_NEEDED) {
                //Implement your code here
        } else if ($session->getState() == Session::AUTH_PENDING) {
                //Implement your code here
        } else if ($session->getState() != Session::ONLINE) {
                //Implement your code here
        }
}
```

We will now start with the concrete implementation of the Code which is needed for the state AUTH_NEEDED. At first we get the username and password from the post parameters. If you want to use the configuration value *force-terms* from **conf/main.config** it is important to check this value in the implementation. This check should to be independent from the difference between local and external mode. If the configuration is active (true) but the value could not be found in the post parameters we cancel the authentication and return the state AUTH_FAILED. A generic error message will be generated if the plugin returns with this state so you don't need to create your own.

Listing 9.8: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

if ($session->getState() == Session::AUTH_NEEDED) {
        $usr = $postParams['username'];
        $pwd = $postParams['password'];
        // True if config force-terms is true/yes and have been accepted
        if ($this->loginApi->getConfig()->getBoolean('force-terms', $this->loginApi->getLocationI
                        && !(array_key_exists('termsofuse', $postParams) && $postParams['termsofu
                return $session->setState(Session::AUTH_FAILED);
        }

        // Implementation follows below
} else if ()
```

In the next step we are going to implement the logic for the local login. First we check if we are in local mode. You can find a description of the metho **isLocal()** above. We set the state to AUTH_PENDING. In local mode it should not be needed to get in this state but we can catch a possible error there. To take the client online call the method **loginByCredentials()**. If the client could be taken online the method returns a boolean. Now we just check the return value and add a error message. If the client could not be taken online you can get the error message of the IACBOX throught the following array in the session object **$session->lastRequest['err']**. With the method **addErrorMessage()** you can add your own error message and the one from IACBOX.

In the implementation of the external mode we prepared a method **redirectToIacBox()** which sends a post request to the IACBOX from a array you defined before. In our example we create the array $paramMap with the following params:

- **id**
  - The ID which identifies the client. This is a random token
- **ac**
  - Which action should be taken
  - In our example and most of the time it is *logon*
- **type**

Listing 9.9: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

if ($this->loginApi->isLocal()) {
        // Local LoginApi part
        $session->setState(Session::AUTH_PENDING);
        $success = $this->localInterface->loginByCredentials($usr, $pwd, $session);
        if (! $success) {
                $this->loginApi->addErrorMessage('logon-failed');
                if (array_key_exists('err', $session->lastRequest) && $session->lastRequest['err']
                        $this->loginApi->addErrorMessage('', false, $session->lastRequest['err']);
                }
                $this->log->error('ticket', "Auth Failed with the following RC [".$session->lastR
        }
        if ($this->log->debugging()) {
                $this->log->debug('ticket', 'Login by ticket '.($success ? 'success' : 'failed'));
        }
        return $session->setState($success ? Session::ONLINE : Session::AUTH_FAILED);
} else {
        // Implementation of the external mode
```

  - Logon type which needs to be used
  - **The following types are available**
    - cred = credentials (e.g. Ticket login)
    - to = takeonline (e.g. Social login)
    - free = free logon (e.g. Free logon)
    - create = create ticket per id (e.g. Payment login)
    - pms = take onliner per roomnumber or other pms fields (e.g. PMS login)

- **user**
  - Username which was entered in the input field
- **pwd**
  - Password which was entered in the input field

After we created the array to login we set the state to AUTH_PENDING. Afterwards we call the method **redirectToIacBox()** to redirect our params to the IACBOX to take the client online. To ensure the scripts stops its execution we use the php method **exit()**.

In the next step we cover the behaviour when entering the plugin with the state AUTH_PENDING. Like in the state before we need to differentiate between local mode and external mode.

If we enter the ExamplePlugin in this state locally no further steps are needed because the login was finished in AUTH_NEEDED. If the plugin still gets entered in the state AUTH_PENDING it equals an error and we add a error message and return the error state PLUGIN_ERROR.

In external mode we cover the return values of the IACBOX which were saved in the session. We check if the key **rc** is 0. This means no error happend during the login. Any other return code represents a specific error. If the rc was 0 we return with the stete ONLINE. If we an error occured we return a error message with the error of the IACBOX and the state AUTH_FAILED.

In the last part we check if the plugin was entered with the state ONLINE. This should not happen because no further steps are needed. We return a error message and the state PLUGIN_ERROR to show something did not work correctly. At last we return the present state to catch the error in case we forgot to return the state in one of our cases.

Listing 9.10: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

if ($session->getState() == Session::AUTH_NEEDED) {

        if ($this->loginApi->isLocal()) {
                // Implementation of local Login-API part
        } else {
                // External Login-API part
                $paramMap = array(
                        'id'            => $session->id,
                        'ac'            => 'logon',
                        'type'          => 'cred',
                        'user'          => $usr,
                        'pwd'           => $pwd
                );
                $session->setState(Session::AUTH_PENDING);
                $this->log->info('ticket', 'Ticket login pending');
                $this->loginApi->redirectToIacBox($paramMap);
                exit();
        }
} else if ($session->getState() == Session::AUTH_PENDING) {
        //Implementation of behaviour in state AUTH_PENDING
}
```

Listing 9.11: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

if ($session->getState() == Session::AUTH_NEEDED) {
        //Implementation of behaviour in state AUTH_NEEDED
} else if ($session->getState() == Session::AUTH_PENDING) {
        if ($this->loginApi->isLocal()) {
                // Should not happen, as the login is made blocking and locally - see above
                $this->loginApi->addErrorMessage('error-plugin');
                $this->log->error('ticket', 'Unknown Error! Local call in State AUTH_PENDING');
                return $session->setState(Session::PLUGIN_ERROR);
        } else {
                // Awaited callback with an success or error code
                if ($session->lastRequest['rc'] == 0) {
                        $this->log->info('ticket', 'Ticket login was successful');
                        return $session->setState(Session::ONLINE);
                } else {
                        $this->loginApi->addErrorMessage('logon-failed');
                        if (array_key_exists('err', $session->lastRequest) && $session->lastReque
                                $this->loginApi->addErrorMessage('', false, $session->lastRequest
                                $this->log->error('ticket', "Auth Failed with the following RC ["
                        }
                        return $session->setState(Session::AUTH_FAILED);
                }
        }
}
```

Listing 9.12: Iacbox/LoginApi/Plugin/Example/ExamplePlugin.php

```php
<?php

if ($session->getState() == Session::AUTH_NEEDED) {
        //Implementation of behaviour in state AUTH_NEEDED
} else if ($session->getState() == Session::AUTH_PENDING) {
        //Implementation of behaviour in state AUTH_PENDING
} else if ($session->getState() != Session::ONLINE) {
        $this->loginApi->addErrorMessage('error-plugin');
        $this->log->error('ticket', 'Plugin was entered with the wrong State '.$session->getState
        return $session->setState(Session::PLUGIN_ERROR);
}
return $session->getState();
```

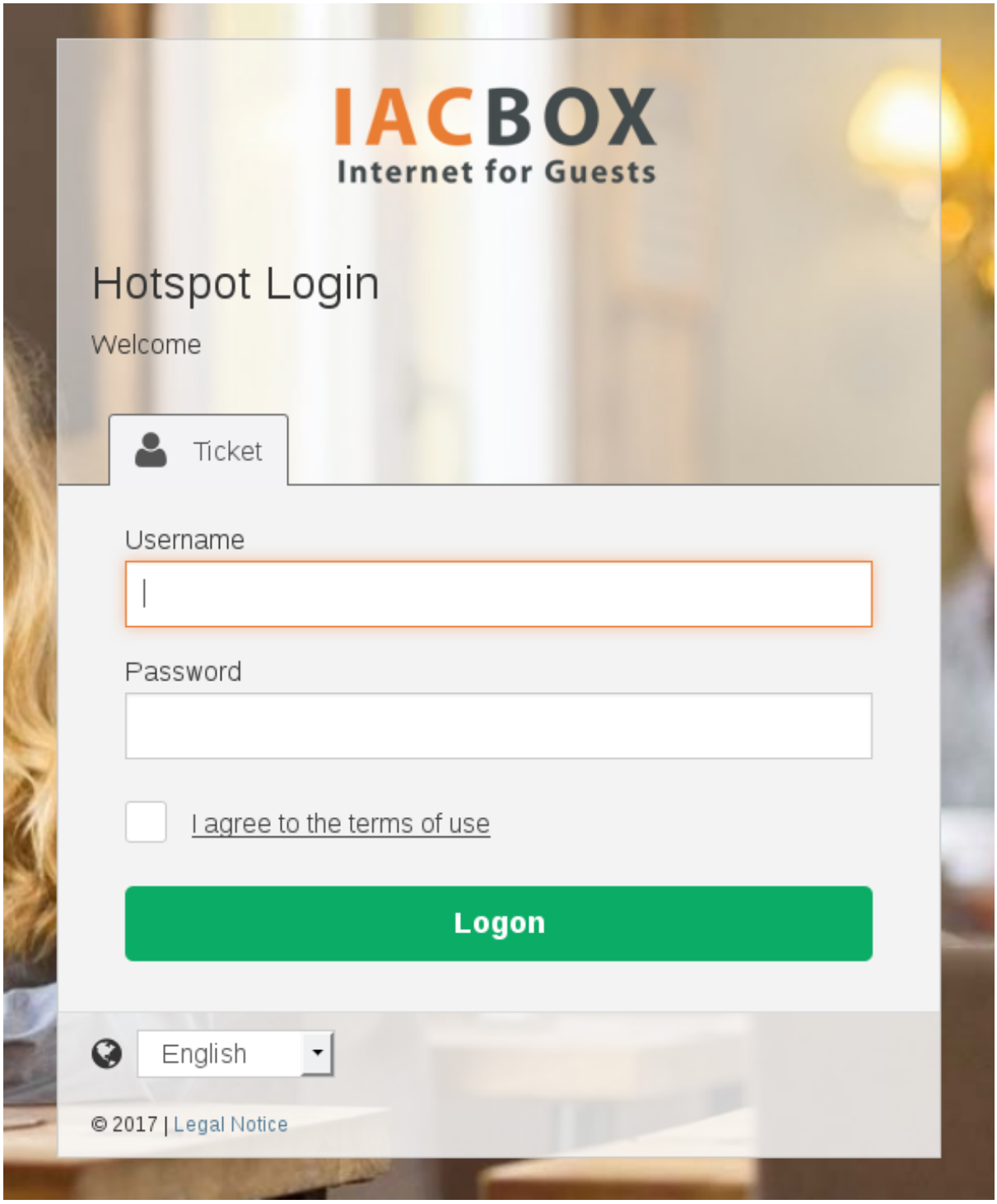
9.4 Plugin configuration

Every plugin is configured differently. Find below more informations regarding the different configuration keys.

9.4.1 Authentication plugins

Ticket

The ticket logon is a standard login with credentials (username and password) or any configured external authentication module (e.g. LDAP, radius, ...).



Configuration Key	Since	Mandatory	Default
ticket-icon	2.0	Yes	fa fa-user
Sets the icon for the plugin			

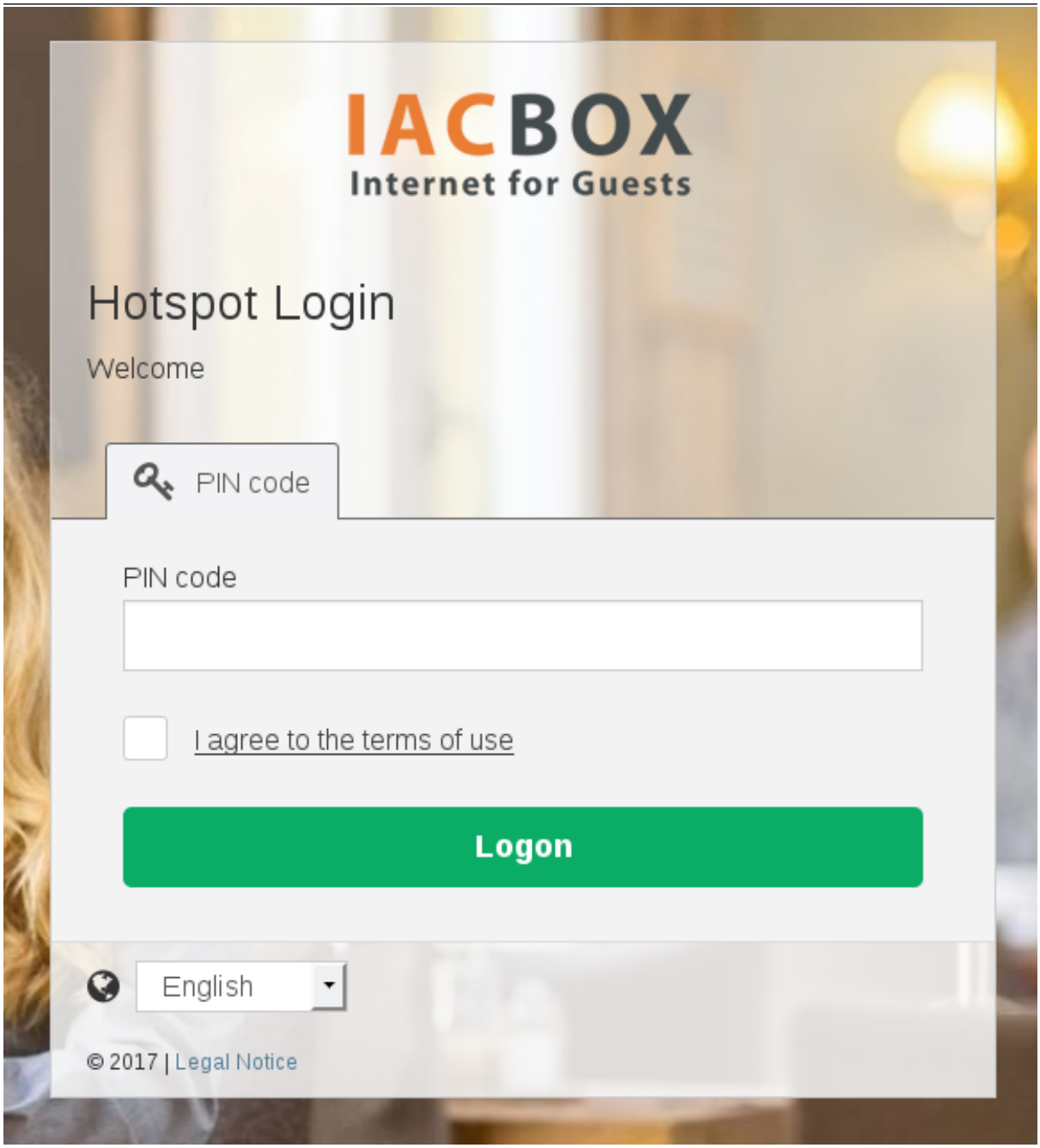
PwdOnly

The pwdonly logon is a standard login with a password. It can be created exactly the same way as when using the default logon page of the IACBOX.

Hint:

- The password login has to be activated on the IACBOX.

- The password login has to be enabled in a ticket template.
- For further information see our documentation regarding the Password Login (page 163)



Configuration Key	Since	Mandatory	Default
pwdonly-icon	2.0	Yes	fa fa-key
Sets the icon for the plugin			

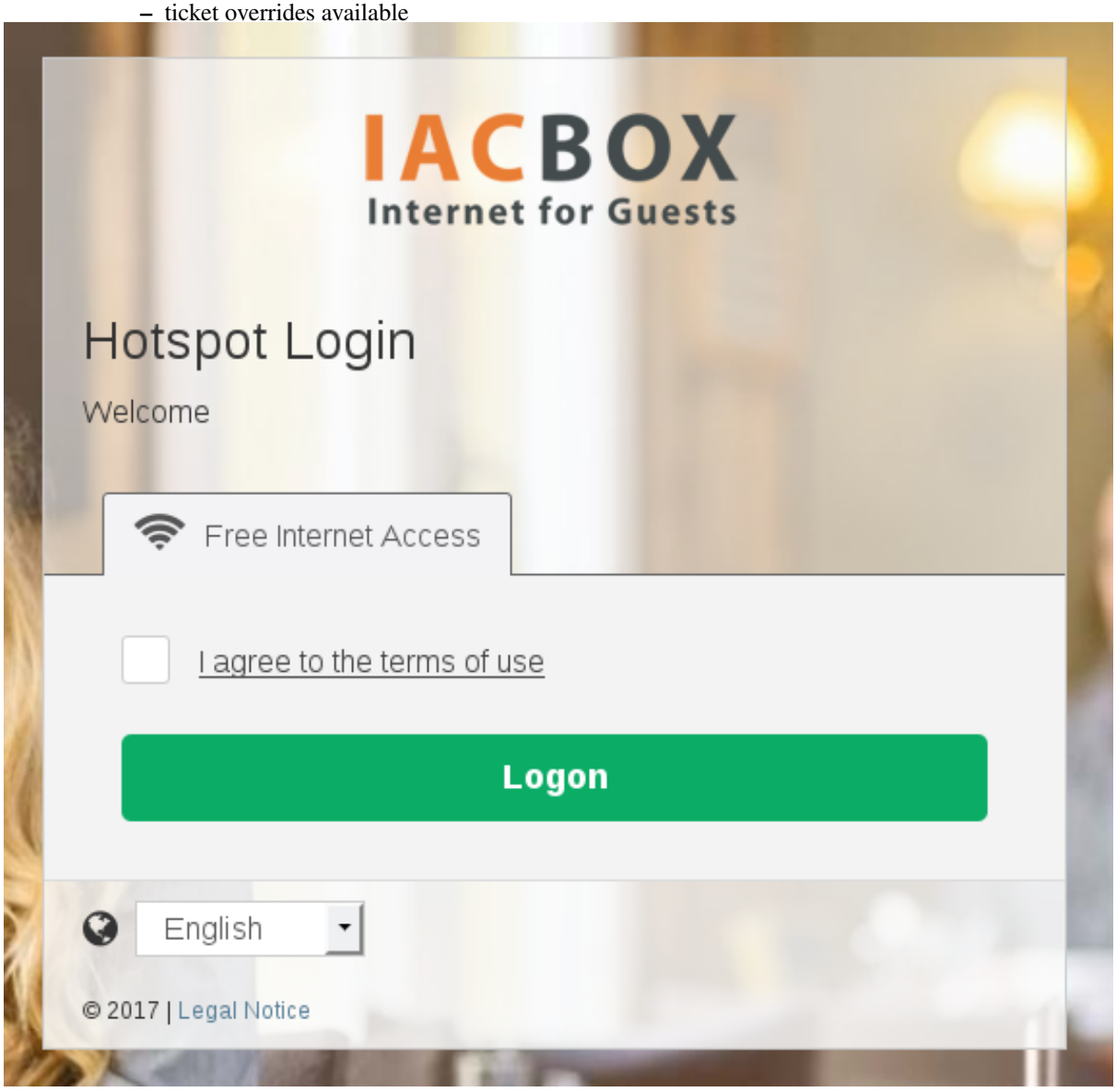
Free

With the free plugin you can take your clients online without any further authentication steps. The plugin supports two different modes:

- **free:**
  - available since API version 2

  - linked with **Ticket/Templates/Free Logon** of the IACBOX
  - **no** ticket overrides available

- **to:**
  - available since version 17.0
  - the template, which you configured in **Modules/Interfaces/Login API/Custom Logon Page**, will be used
  - ticket overrides available



Configuration Key	Since	Mandatory	Default
free-icon	2.0	Yes	fa fa-wifi
Sets the icon for the plugin			
free-logon-type	17.0	Yes	free
Determines if the plugin is using the settings of Free Logon of the IACBOX			

Only available with the logon type to

Configuration Key	Since	Mandatory	Default
otc	17.0	No	3600
Time credit in seconds			
otl	17.0	No	17
Ticket limit in MB			
omi	17.0	No	500
Max idle timeout in seconds			
oep	17.0	No	10800
Expiration period in seconds			
odl	17.0	No	2048
Max download bandwidth in kBit/s (max value is the total bandwith under System/Network)			
oul	17.0	No	1024
Max upload bandwidth in kBit/s (max value is the total bandwith under System/Network)			
ode	17.0	No	LoginApi Free Plugin take online
Ticket description			

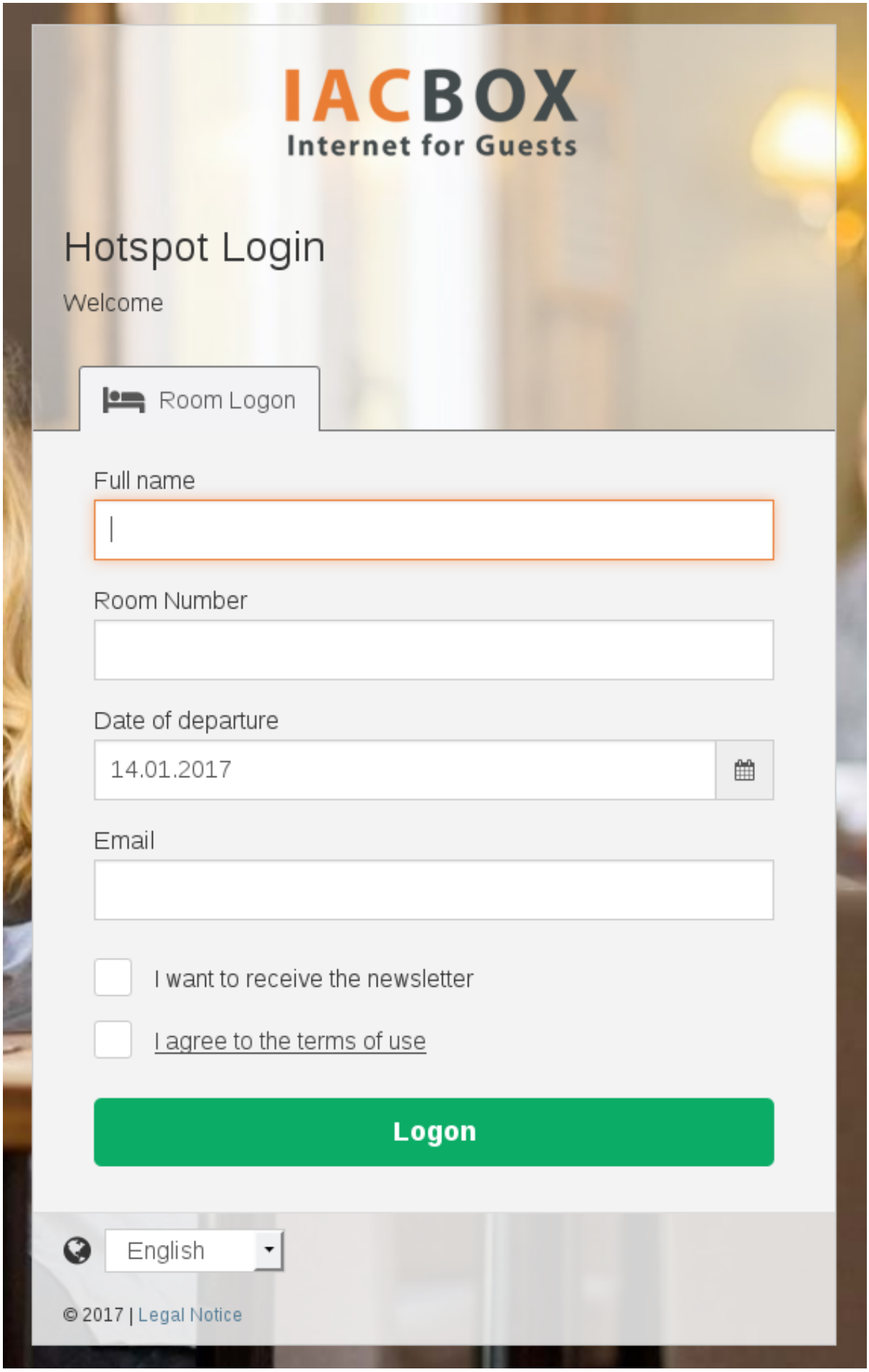
PMS

The PMS plugin works the same way as it does on the IACBOX landing page. After the successful authentication you get to a second page to select a ticket.

New in version 17.0: We also support the PMS **Himed**, **Guestline** and **ASAj**.

Attention:
The following configuration keys have to be in sync with your PMS configuration in Modules/Interfaces/PMS:

- auth-fields
- name-check


IACBOX
Internet for Guests
Hotspot Login
Welcome
Room Logon
Full name
Room Number
Date of departure
14.01.2017
Email
I want to receive the newsletter
I agree to the terms of use
Logon
English
© 2017 | Legal Notice

Configuration Key	Since	Mandatory	Default
pms-icon	2.0	Yes	fa fa-hotel
Sets the icon for the plugin			
auth-fields	2.0	Yes	room, name
The fields which are needed for the authentication			
name-check	17.0	No	full
Changes the label of the name field			
show-email	2.0	No	false
Shows an email field (create a custom service to further work with the information)			
email-syntax-check	2.0	No	true
Set if the email-adress should be checked for a correct syntax			
email-mandatory	2.0	No	false
Set if the email-field is mandatory			
show-building-selection	2.0	No	false
Enables the number padding and building mapping			
room-nr-padding	2.0	No	0
Number of digits which has to met in the roomnumber			
room-nr-type	2.0	No	number
Room number only contains digits or is alphanumeric			
building-mapping	2.0	No	
Specific Mapping for the room number e.g.: A:Aaaaaaa;B:Bbbbbbb;C:Ccccccc			
show-address-person	2.0	No	false
Enables the field how to address a person e.g. Mr. or Mrs.			
address-person-mapping	2.0	No	
Specific mapping for address of a person			

Guestline

The Guestline plugin is **only available** in the LoginAPI and is not supported of the default Logon page.

New in version 18.0.

Attention: Plugin can not be used in **external mode**

Attention: **PMS** module needs to be licensed to use this plugin

Configuration Key	Since	Mandatory	Default
guestline-icon	18.0	Yes	fa fa-hotel
Sets the icon for the plugin			
guestline-operator-code	18.0	Yes	
Sets the operator code of the Guestline Endpoint			
guestline-pwd	18.0	Yes	
Sets the password of the Guestline Endpoint			
guestline-site-id	18.0	Yes	
Sets the site id of the Guestline Endpoint			
guestline-interface-id	18.0	Yes	
Sets the interface id of the Guestline endpoint			
guestline-endpoint	18.0	Yes	
Sets the endpoint url of Guestline			
auth-fields	18.0	Yes	room, name, departure_date
The fields which are needed for the authentication			
exact-match	18.0	No	2
Sets the matching accuracy of the name (0 exact, 5 inaccurate)			

Continued on next page

Table 9.2 – continued from previous page

Configuration Key	Since	Mandatory	Default
show-ticket-selection	18.0	No	true
Sets if a ticket selection should be available			
*allow-multi-guest-login**	18.0	No	true
Changes the label of the name field			
show-email	2.0	No	false
Shows an email field (create a custom service to further work with the information)			
email-syntax-check	2.0	No	true
Set if the email-adress should be checked for a correct syntax			
email-mandatory	2.0	No	false
Set if the email-field is mandatory			
show-building-selection	2.0	No	false
Enables the number padding and building mapping			
room-nr-padding	2.0	No	0
Number of digits which has to met in the roomnumber			
room-nr-type	2.0	No	number
Room number only contains digits or is alphanumeric			
building-mapping	2.0	No	
Specific Mapping for the room number e.g.: A:Aaaaaaa;B:Bbbbbbb;C:Ccccccc			
show-address-person	2.0	No	false
Enables the field how to address a person e.g. Mr. or Mrs.			
address-person-mapping	2.0	No	
Specific mapping for address of a person			

Himed

The Himed plugin works exactly as the PMS Plugin and also has the same configuration keys. Additional it is possible to define a redirect url because Himed allows a flexible redirect url.

Hint:

The following placeholder are available:

- $USERID = UserId of the Himed client

Configuration Key	Since	Mandatory	Default
Same configuration values as the PMS Plugin			
himed-redirect-url	17.0	No	
Redirect URL after Himed authentication			

Email

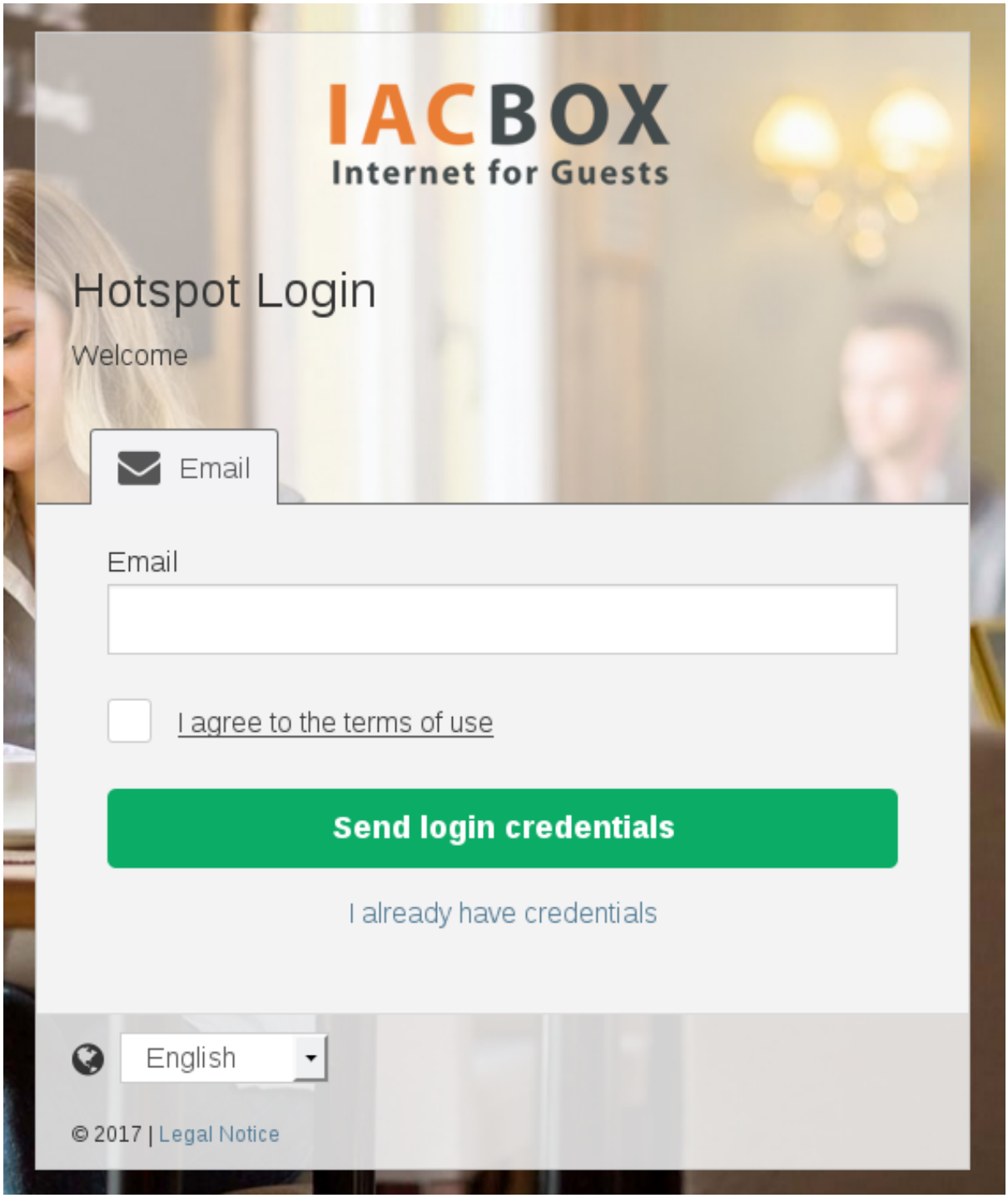
The email plugin is linked with the messaging module **Email**. Therefore the ticket template will be used which is configured in the email settings (Modules -> Interfaces -> Email).

If you want to use the pwdonly login with the email plugin follow the hint.

Hint:

- The password login has to be activated on the IACBOX.
- The password login has to be enabled in a ticket template.
- For further information see our documentation for the Password Login (page 163)

New in version 2.0.



Configuration Key	Since	Mandatory	Default
email-icon	2.0	Yes	fa fa-envelope
Sets the icon for the plugin			
pwd-only	2.0	Yes	false
Set to true if you are using a pwd-only template			

SMS

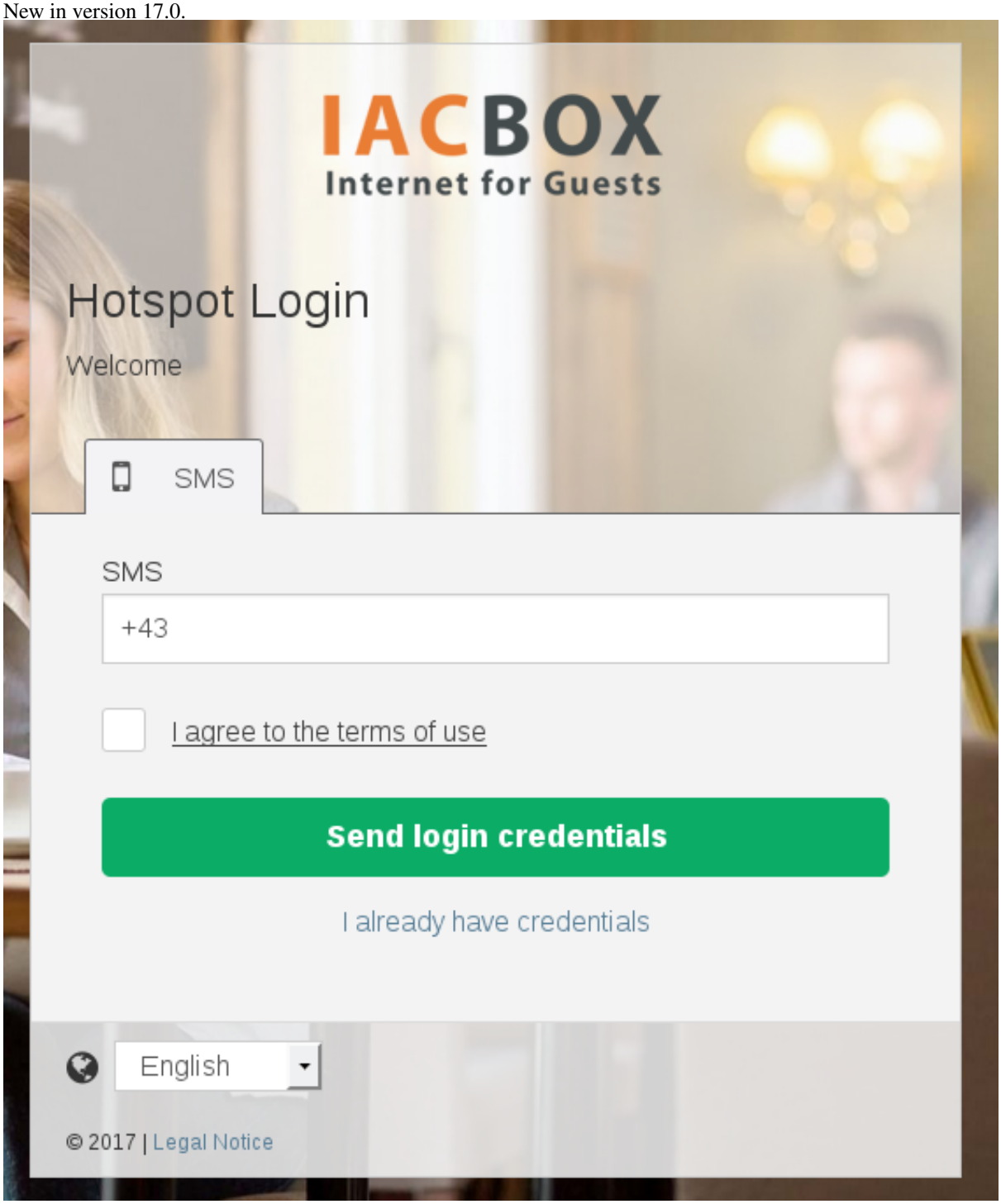
The SMS plugin sends credentials to the given phone number via the configured SMS backend. This has to be configured on Modules -> Interfaces -> SMS.

If you want to use the pwdonly login with the sms plugin follow the hint.

Hint:

- The password login has to be activated on the IACBOX.
- The password login has to be enabled in a ticket template.
- For further information see our documentation for the Password Login (page 163)

New in version 17.0.



Configuration Key	Since	Mandatory	Default
sms-icon	17.0	Yes	fa fa-mobile
Sets the icon for the plugin			
pwd-only	17.0	Yes	false
Set to true if you are using a pwd-only template			

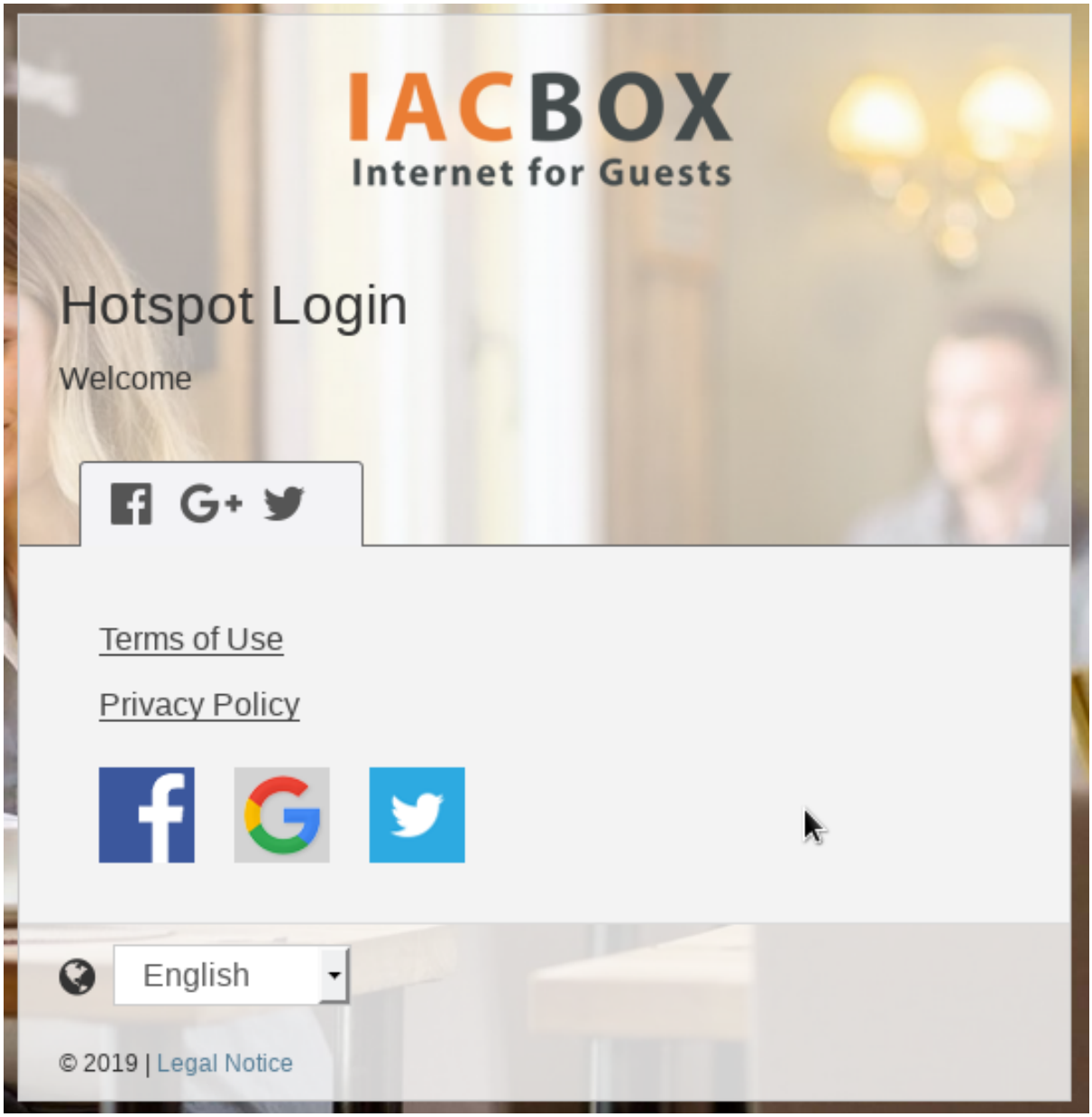
Social

The social plugin grants a user internet access via login through an external social platform. Currently supported platforms are **facebook**, **twitter** and **google**.

The plugin doesn't have a connection to the module *Social Login* of the IACBOX and can therefore be used without further licensing.

Attention: You need to add the following parameters to your callback-url in the configuration of the app for a successful login: **?auth=social&hauth.done=<Provider>** where provider is one of **twitter** or **google** depending on the used platform.

Attention: Since **Facebook API-Version 2.10** dots **.** are not allowed in the redirect urls anymore. Pleasse modify the callback url to **?auth=social&hauth_done=Facebook**.



Configuration Key	Since	Mandatory	Default
enabled-services	2.0	Yes	
Enable which service should be possible for a client			
id-facebook	2.0	Yes	
Set the id of your facebook application			
secret-facebook	2.0	Yes	
Set the secret of you facebook application			
id-google	2.0	Yes	
Set the id of your google application			
secret-google	2.0	Yes	
Set the secret of you google application			
key-twitter	2.0	Yes	
Set the key of your twitter application			
secret-twitter	2.0	Yes	
Set the secret of you twitter application			
facebook-icon	2.0	Yes	fa fa-facebook-official
Sets the icon for the facebook login			

Continued on next page

Table 9.3 – continued from previous page

Configuration Key	Since	Mandatory	Default
google-icon	2.0	Yes	fa fa-google-plus
Sets the icon for the google login			
twitter-icon	2.0	Yes	fa fa-twitter
Sets the Icon for the twitter login			
otc	2.0	No	
Time credit in seconds			
otl	2.0	No	
Ticket limit in MB			
omi	2.0	No	
Max idle timeout in seconds			
oep	2.0	No	
Expiration period in seconds			
odl	2.0	No	
Max download bandwidth in kBit/s (max value is the total bandwith under System/Network)			
oul	2.0	No	
Max upload bandwidth in kBit/s (max value is the total bandwith under System/Network)			
ode	2.0	No	
Ticket description			

Payment

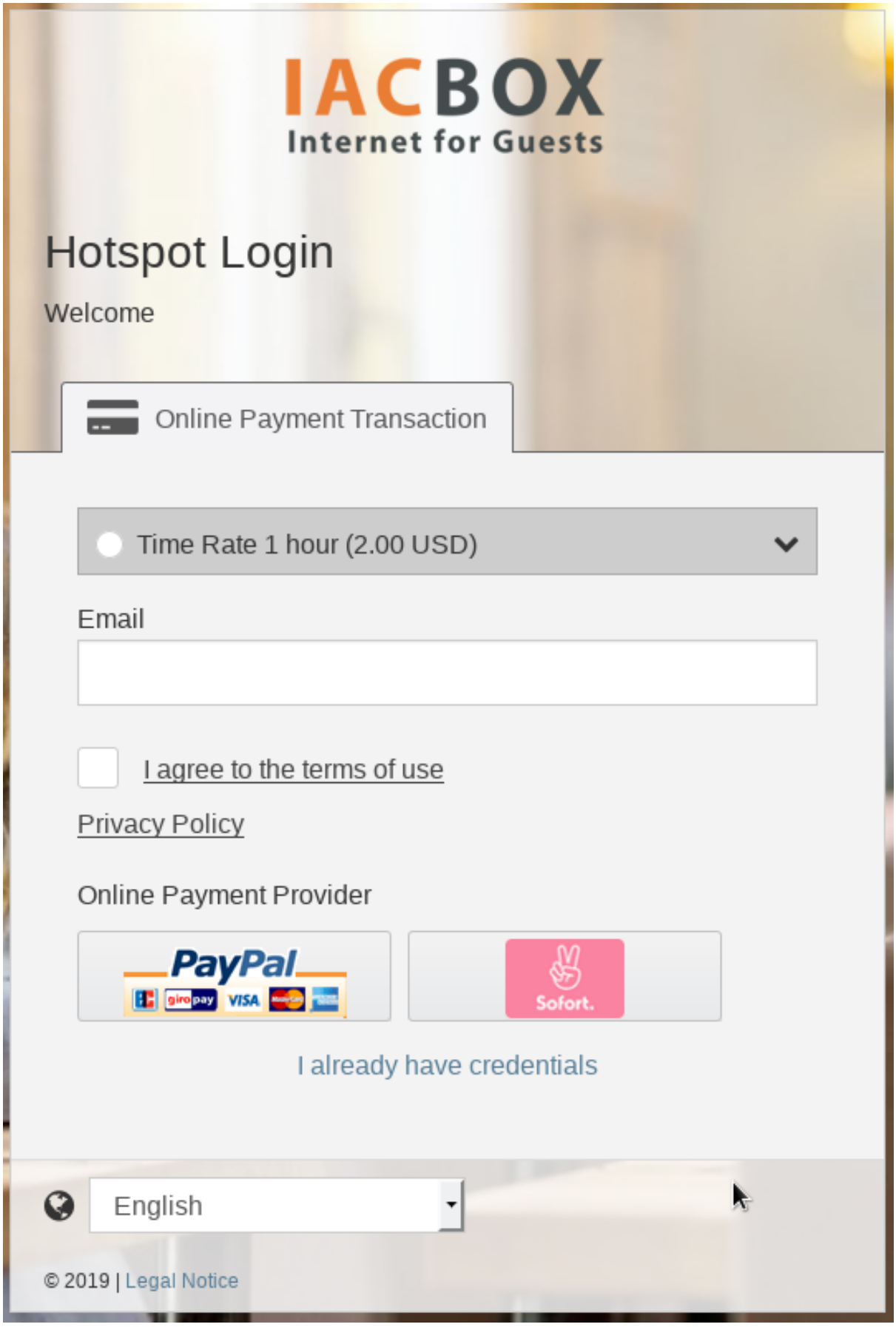
This plugin allows you to login via a payed ticket. Currently we support the following two payment providers **PayPal** and **Sofort ÃIJberweisung**.

If you want to use the configuration value send-email configure the WebAdmin setting in the menu **Settings/Network/SMTP Proxy**.

Attention:
We are providing some experimental payment providers:

- Stripe
- WorldPay
- 2Checkout
- AuthorizeNet

They are implemented in the code but no buttons are provided for their usage. **You have to implement them yourself.**


IACBOX
Internet for Guests
Hotspot Login
Welcome
Online Payment Transaction
Time Rate 1 hour (2.00 USD)
Email
I agree to the terms of use
Privacy Policy
Online Payment Provider
PayPal
Sofort.
I already have credentials
English
© 2019 | Legal Notice

Configuration Key	Since	Mandatory	Default
payment-icon	2.0	Yes	fa fa-credit-card-alt
Sets the icon for the plugin			
enabled-services	2.0	Yes	
Enable which service should be possible for a client			
append-location-id	2.0	Yes	
Location id will be send to the payment provider			
test-mode	2.0	Yes	false
Set the sandbox mode for the payment provider			
send-email	2.0	Yes	
Email with login data will be send appends a email field			
currency	2.0	Yes	
Set the currency (has to be in sync with the IACBOX)			
paypal-username	2.0	Yes	
Set the username for Paypal			
paypal-password	2.0	Yes	
Set the password for Paypal			
paypal-signature	2.0	Yes	
Set the Paypal signature			
sofort-account-id	2.0	Yes	
Set the account id for SofortÃIJberweisung			
sofort-key	2.0	Yes	
Set the key for SofortÃIJberweisung			
sofort-project-id	2.0	Yes	
Set the project id for SofortÃIJberweisung			

Experimental payment provider which are not tested

Configuration Key	Since	Mandatory	Default
stripe-key	2.0	Yes	
Set the key for Stripe			
worldpay-installation-id	2.0	Yes	
Set the installation id for WorldPay			
worldpay-account-id	2.0	Yes	
Set the account id for WorldPay			
worldpay-secret-word	2.0	Yes	
Set the secret word for WorldPay			
twocheckout-number	2.0	Yes	
Set the number for 2Checkout			
twocheckout-secret	2.0	Yes	
Set the secret for 2Checkout			
authorize-net-id	2.0	Yes	
Set the id for Authorize.Net			
ode	2.0	No	
Ticket description			

Planned redirect / ReviewPro

To activate this plugin, add **plannedredirect** to the plugin setting in **main.config** - this is done automatically for the local LoginAPI.

```
plugins = <any other plugin>, plannedredirect, status
```

This plugin takes a client offline in the night and redirects it to an arbitrary page when the device is activated in the morning again.

There are two main types supported:

- **ReviewPro**: This redirects to an in-stay survey of ReviewPro
- **Generic**: This redirects to an arbitrary website

The concept is to take a client offline in the night, as devices don't behave in a user friendly way when disconnected while they are used. If the device gets disconnected in the night (default 04:00) where most users are sleeping a new DHCP request including a connection check is done when the device is activated again. Taking the client offline is needed to be able to do a redirect as most connections are encrypted nowadays and only the connection check is done with an unencrypted HTTP call.

For example: a guest checks in on Monday 3pm - after 48h (Wednesday 3pm) in the following night (Thursday 4am) the client gets disconnected and redirected to the survey when the device gets activated again. Set the timeout to 24h if this should happen 1 day earlier.

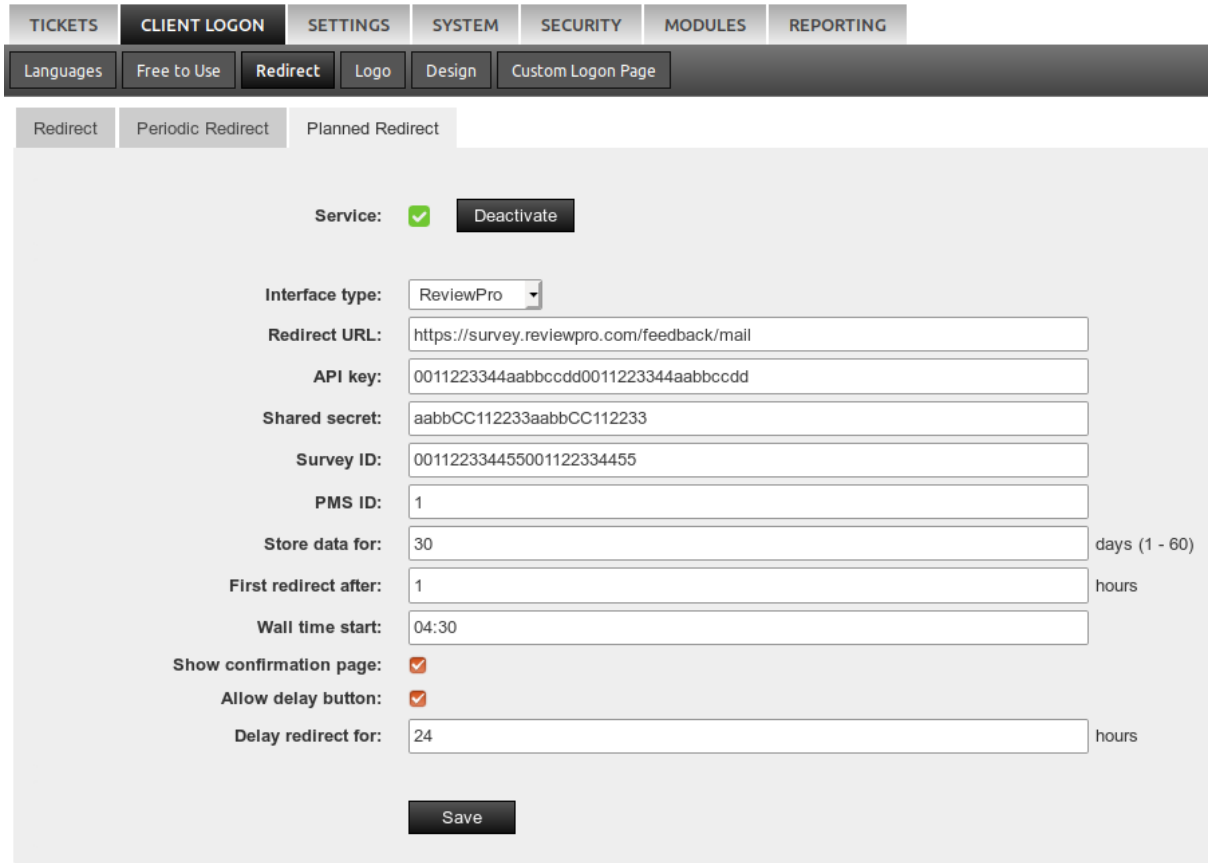
ReviewPro

Hint:

- There is only **one** redirect per stay, so nobody gets annoyed and the hotel gets quality feedback.
- A guest does not have to re-authenticate again after the "Back to internet" button was pressed.

The idea is to get valuable feedback during the stay of a guest. After the set time interval (default 48h) in the following night the client is taken offline to be able to redirect the device.

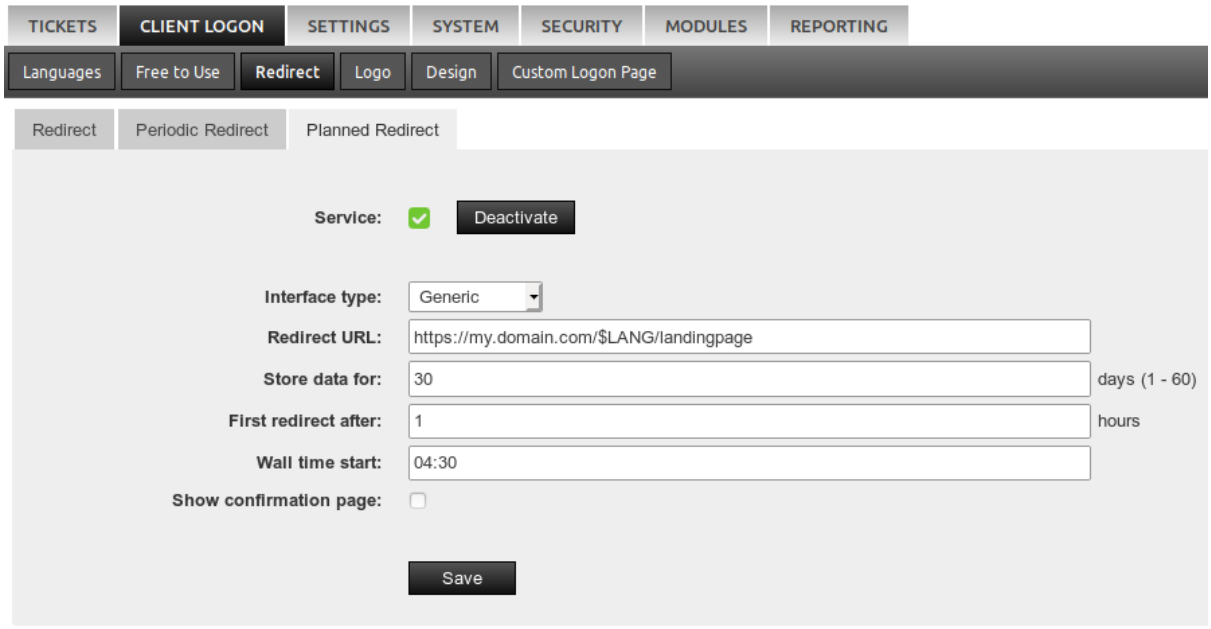
We strongly recommend to activate the **confirmation page** which asks the guest if s/he wants to do the survey. Without the confirmation page the quality of the feedback will suffer as many guests just want to get online again and would give any feedback just to achieve this.



Configuration values

Configuration	Mandatory	Default	Description
Redirect URL	Yes		Use the default URL unless instructed otherwise
API key	Yes		ReviewPro provides this API key
Shared Secret	Yes		ReviewPro provides the shared secret
Survey ID	Yes		ReviewPro provides you with this ID. It can be used for many locations that should show the same survey
PMS ID	Yes		The ID of your PMS used to differentiate all hotels that show the same survey
Store data for	Yes	30	Number of days the IACBOX stores which devices have already done the survey
First redirect after	Yes	48	After how many hours the first redirect should be done in the following night
Wall time start	Yes	04:00	Daytime at which the devices should get disconnected
Show confirmation page	No	False	If the guest should get asked if s/he wants to do the survey (recommended)
Allow delay button	No	False	If an additional delay button should be shown
Delay redirect for	No	24	How many hours until the confirmation page is shown again

Generic



Supported placeholders in the URL are.

- **$LANG** = The detected language of the client as two-letter ISO code like (en, de, it, ...). Any unknown language (to the LoginAPI) will result in the fallback language from `main.conf`.
- **$IP** = IP address of the client
- **$MAC** = MAC address of the client
- **$VLAN** = The VLAN/route ID
- **$LOCID** = The location ID - depending on the setting this is the VLAN/route ID and/or the registration number

---

Hint: It's important to have a prominent **Back to internet** button on your website. This signals the IACBOX

DOKU integration

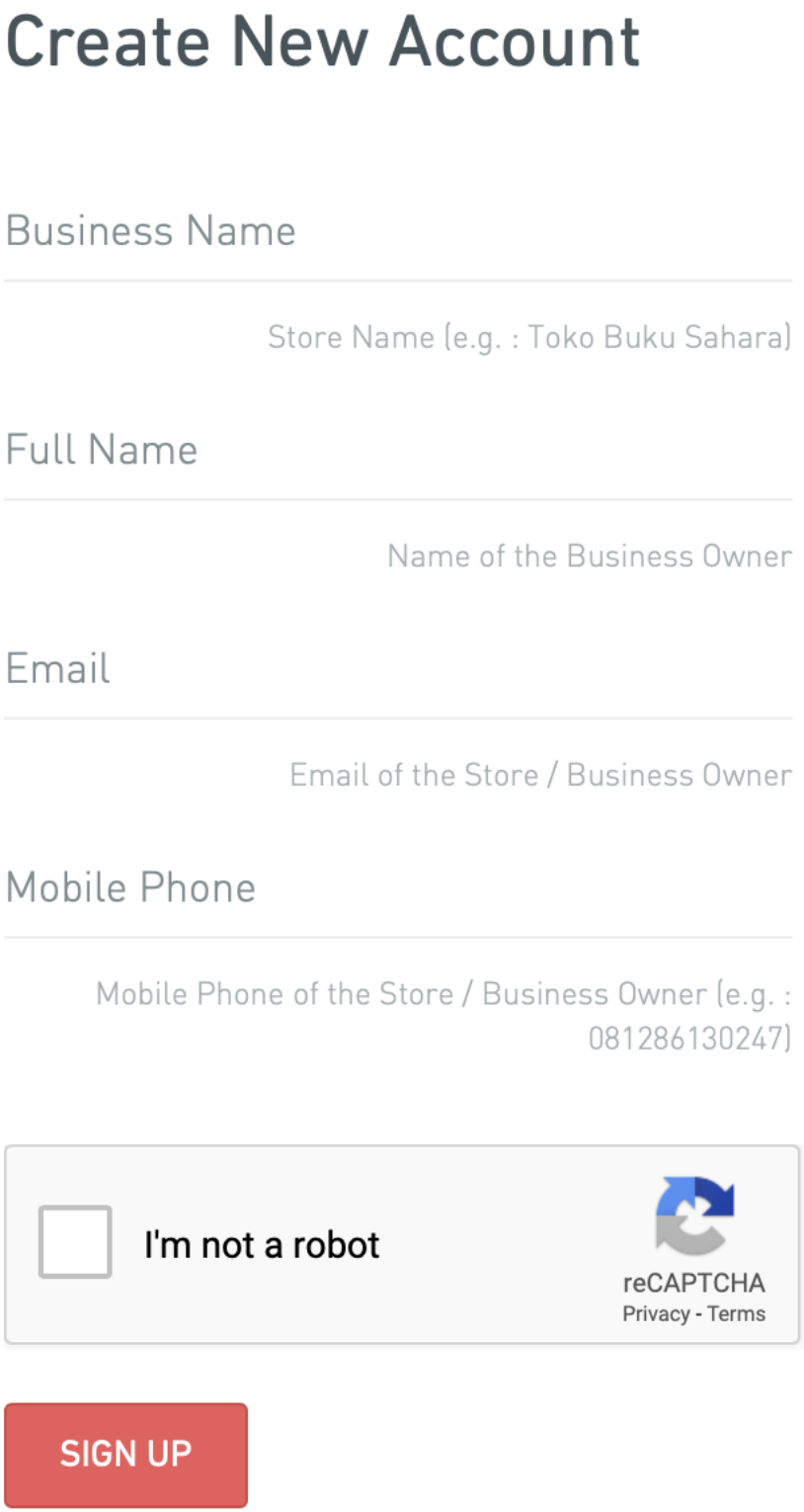
This manual describes how to configure the DOKU API on the IACBOX so guests can pay tickets using their credit card, and DOKU Wallet via DOKU payment gateway.

Hint:

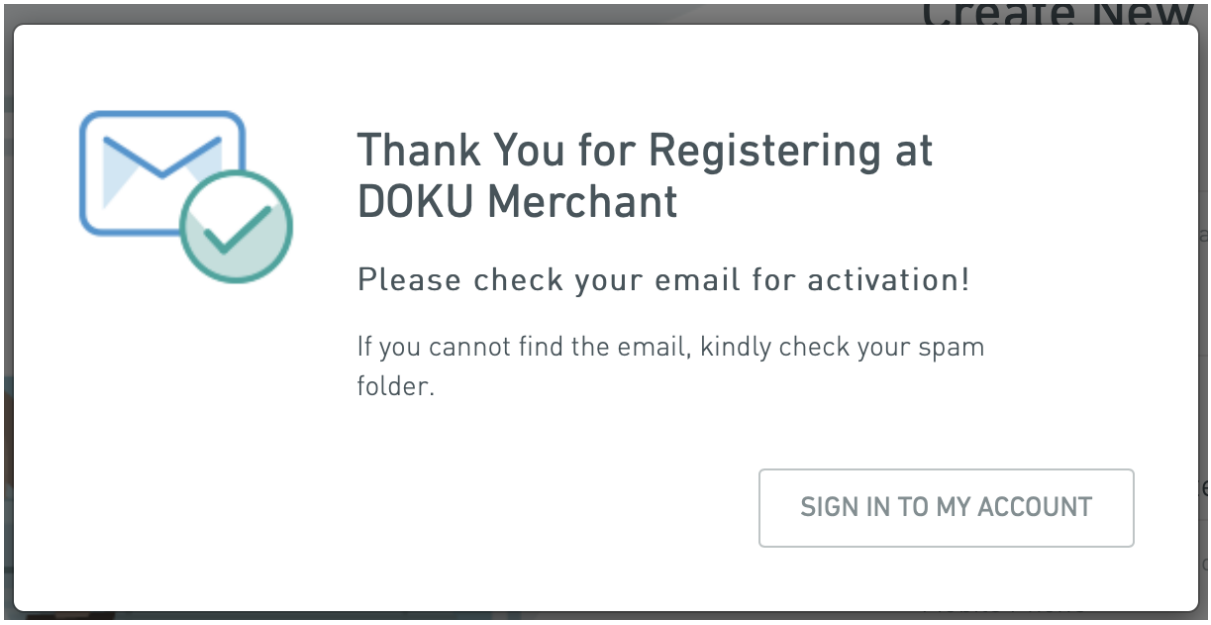
- DOKU is an **indonesian payment gateway**
- To use this API a **DOKU Merchant** account is required.

DOKU merchant account registration

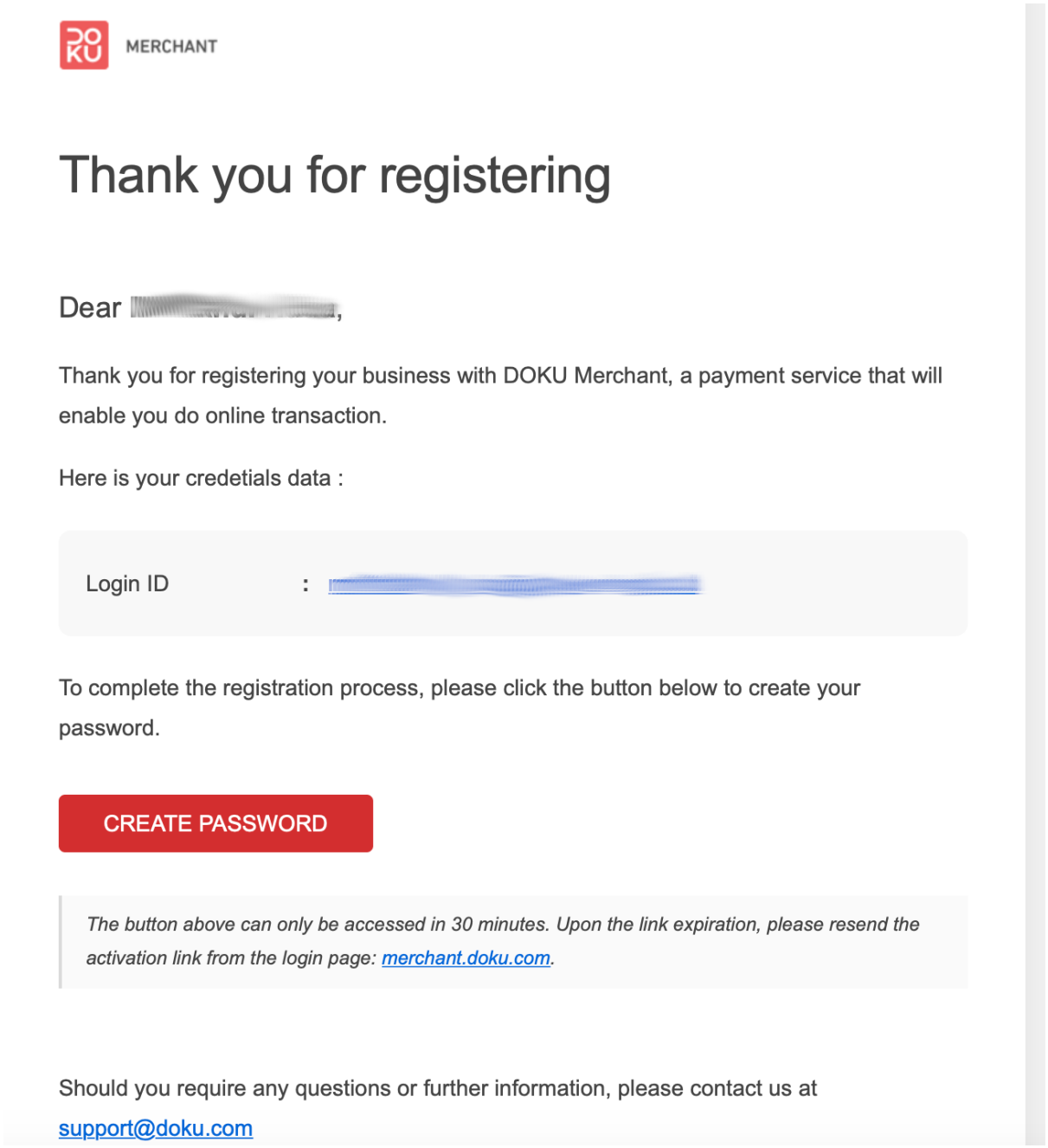
To register new account, visit https://merchant.doku.com/acc/register fill out all fields and then hit the Sign Up button.



Upon successful registration, this screen will show up

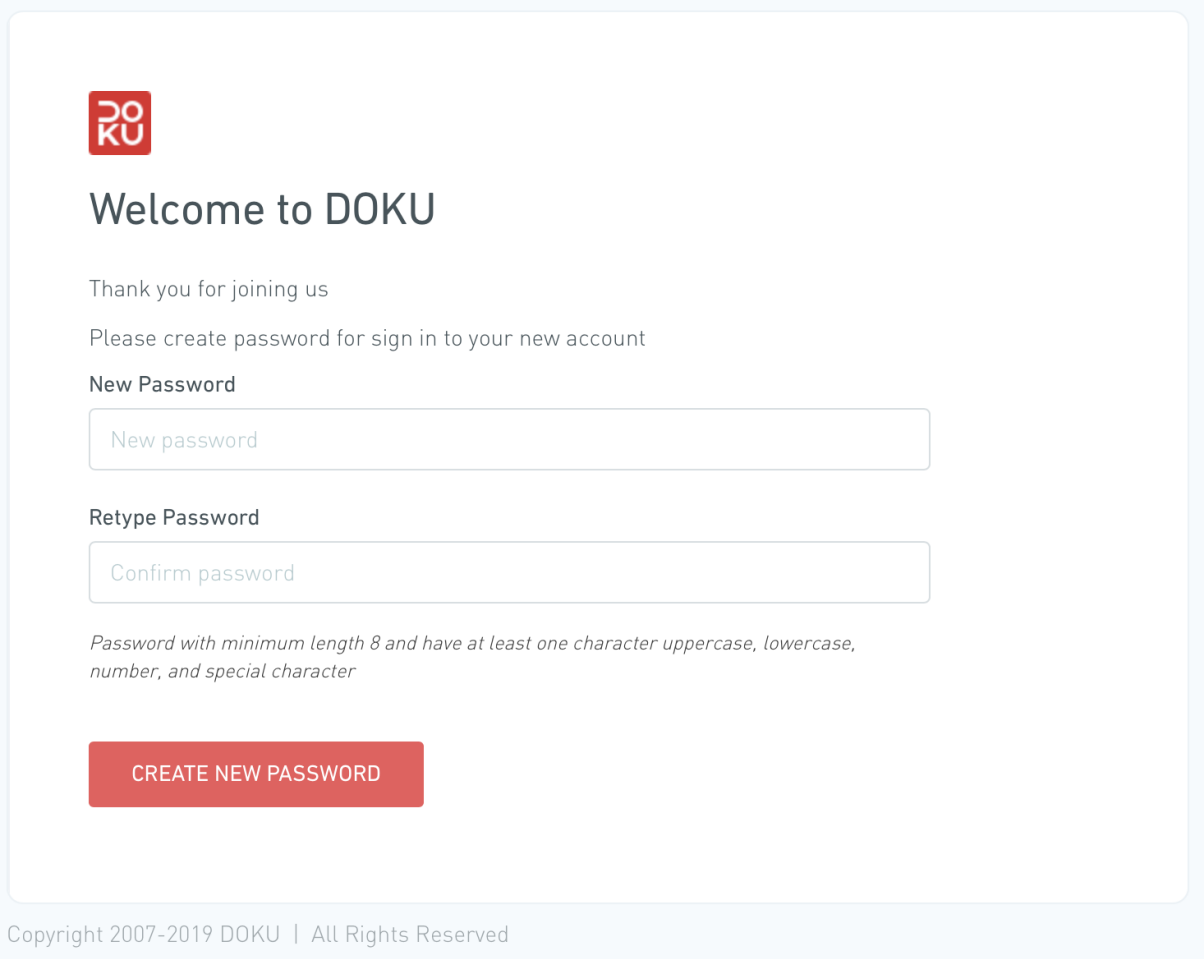


You can click at **SIGN IN TO MY ACCOUNT** but first, check your registered email to create a password for your account.

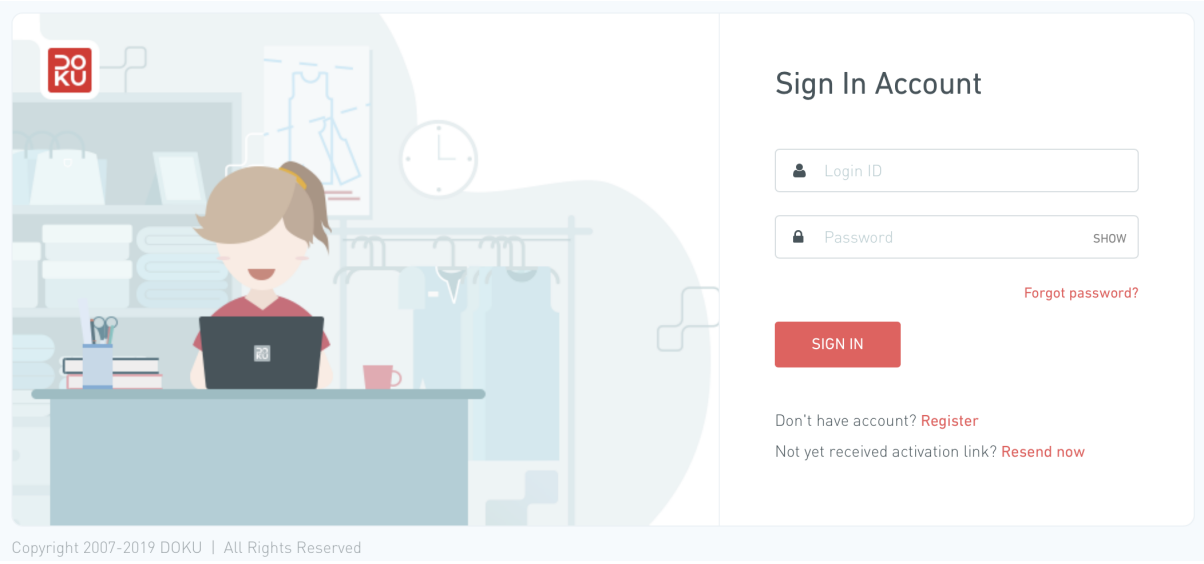


Click at that **CREATE PASSWORD** button, will take you to the next step.

Fill in your desired password.

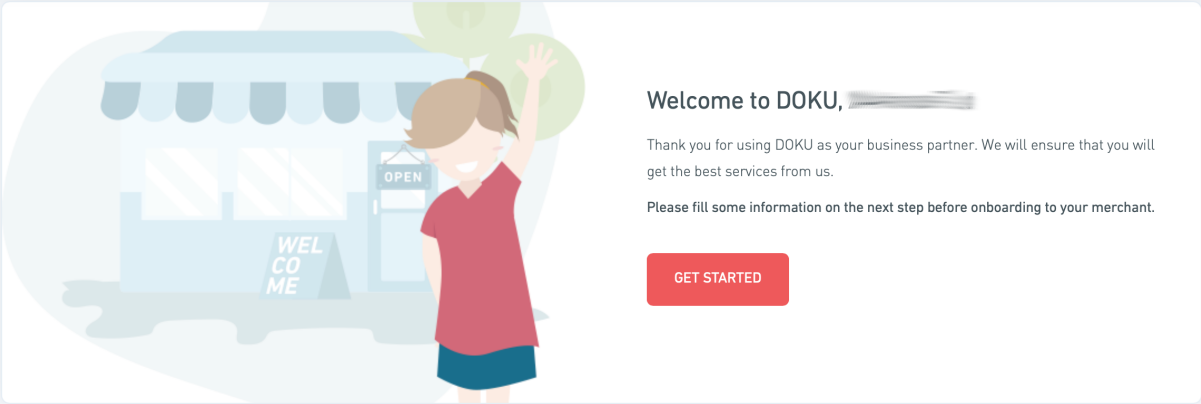


After successful password creation, you can now Sign-In.



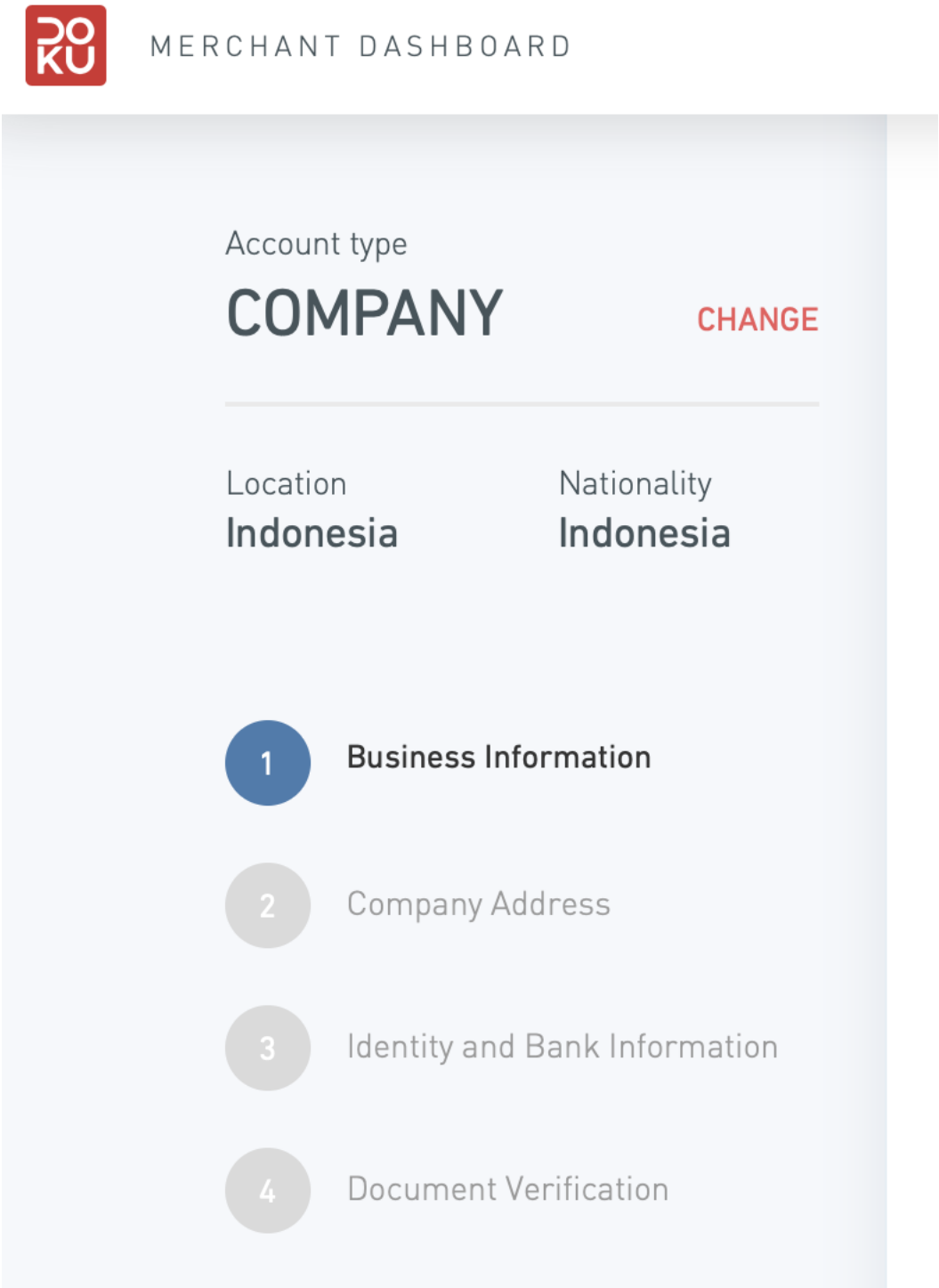
Now login with your registered email and password in this form.

After the first successful login, you will see this welcome message. Click on the **GET STARTED** button.



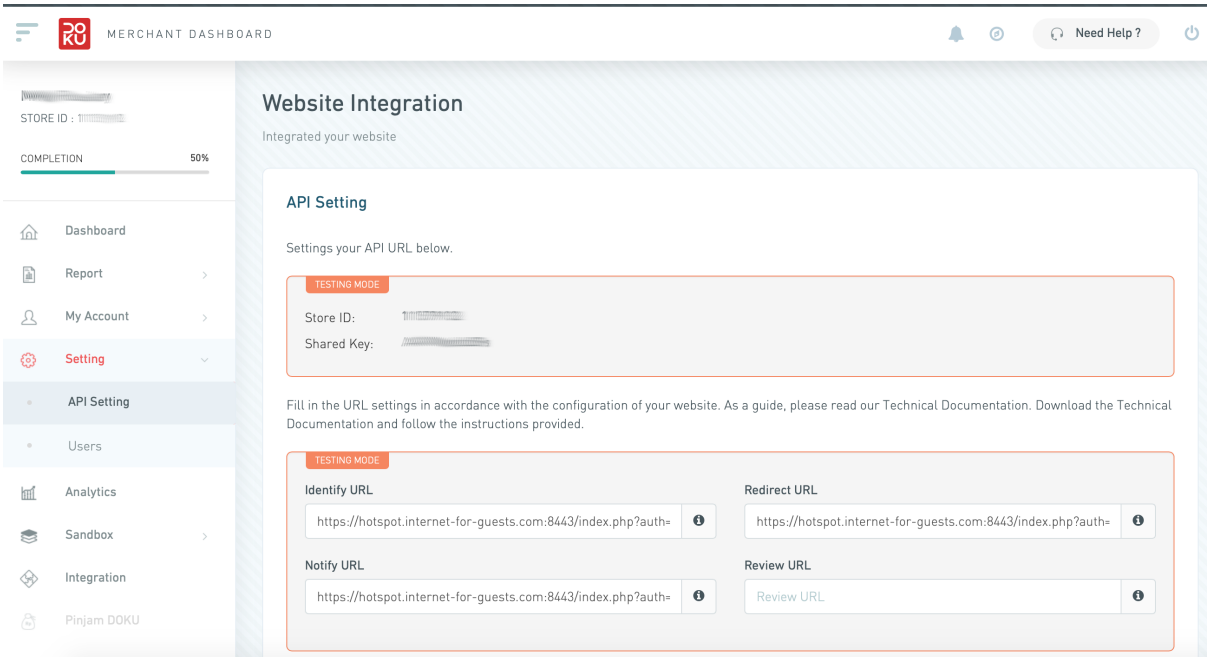
And after that please complete the business type data, and click on the **SAVE AND NEXT** button.

Please provide all necessary data and documents as required in the next steps.



Meeting all requirements above, and after your account has been activated by DOKU, you can now see your dashboard.

Go to Setting and then API Setting, to get information about Store ID and Shared Key, also fill in the necessary URLs.



Here you get your **Store ID** and a **Shared Key** which has to be set in the IACBOX configuration.

Fill **Identify URL**, **Notify URL**, and **Redirect URL** according to your IACBOX domain configuration.

With our default domain this is `https://hotspot.internet-for-guests.com:8443/index.php?auth=payment`

If you have a custom domain use `https://<your-domain-goes-here>:8443/index.php?auth=payment`

Midtrans integration

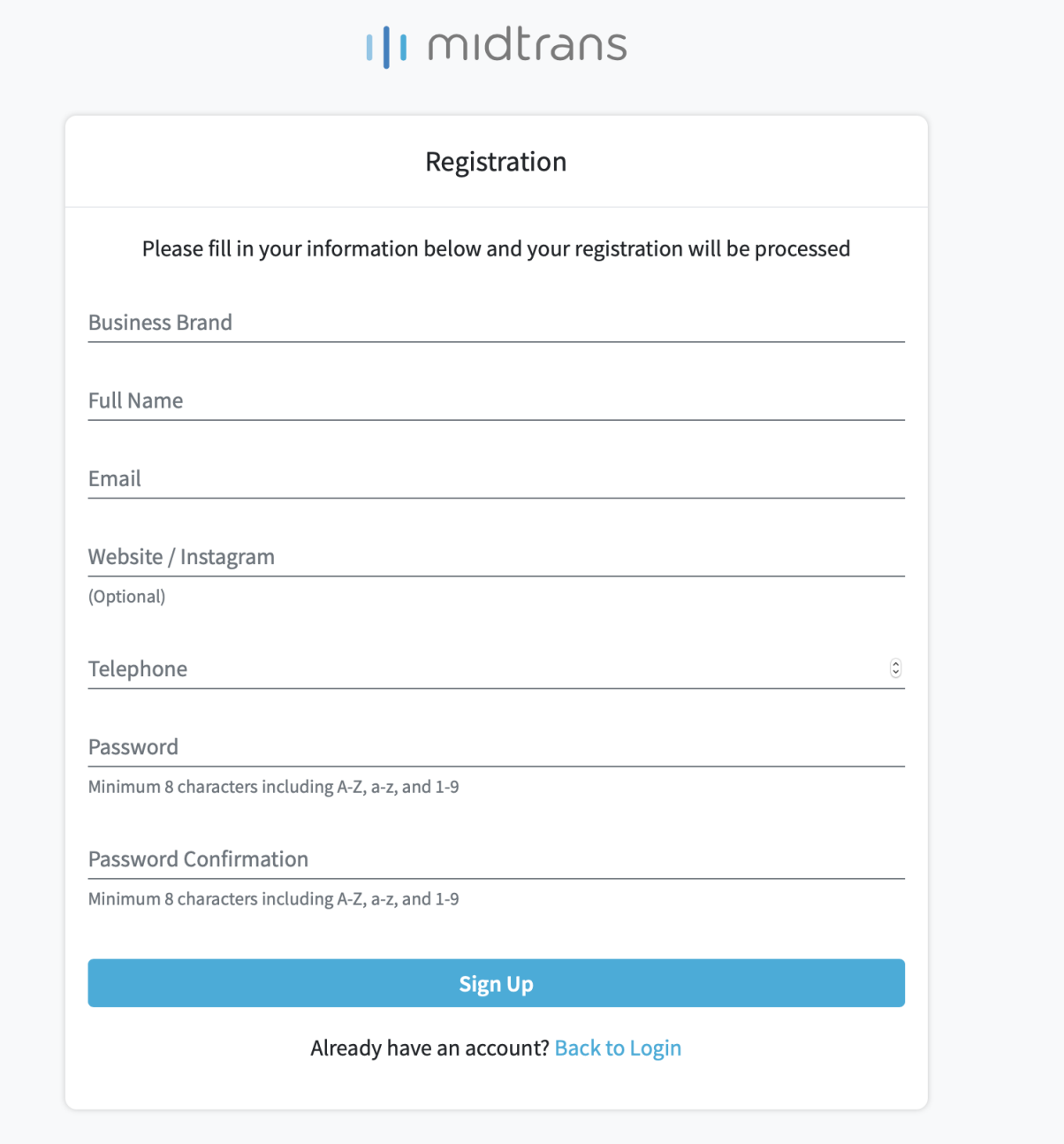
This manual describes how to configure the Midtrans API on the IACBOX so guests can pay tickets via their credit card, and GoPay Balance via Midtrans payment gateway.

Hint:

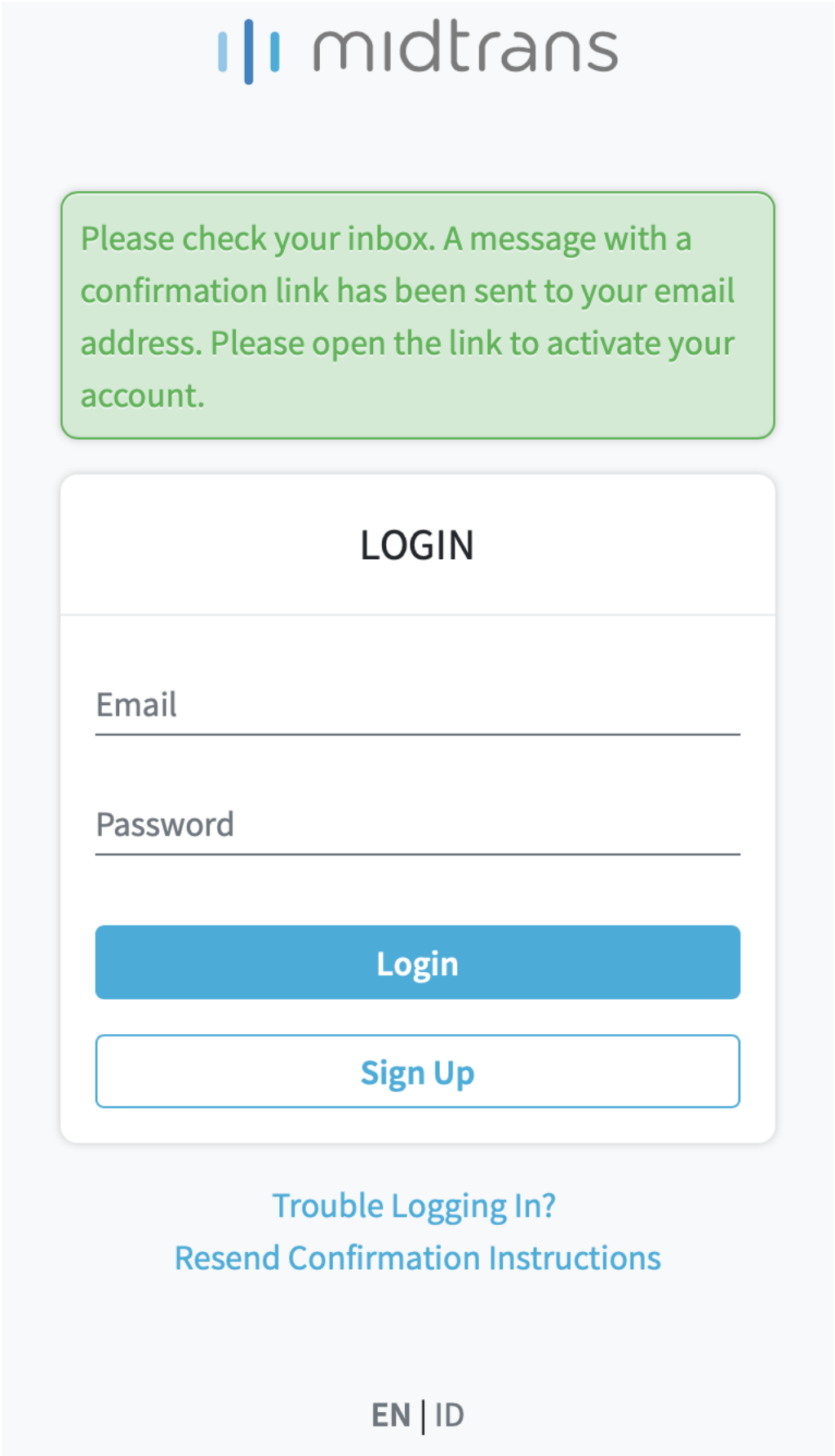
- DOKU is an **indonesian payment gateway**
- To use this API a **Midtrans Passport Business** account is required.

Midtrans passport account registration

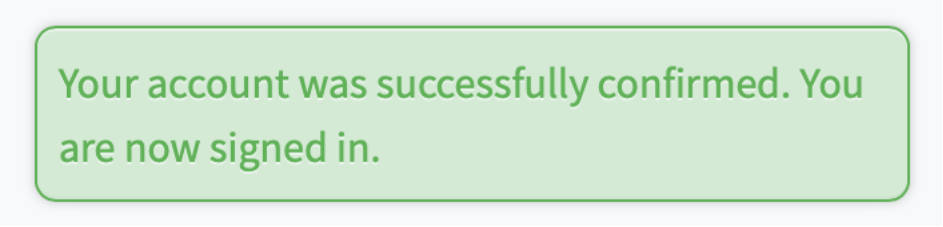
To register new account, visit https://account.midtrans.com/register fill out all fields and then hit the Sign Up button.



Upon successful completion this screen will show up

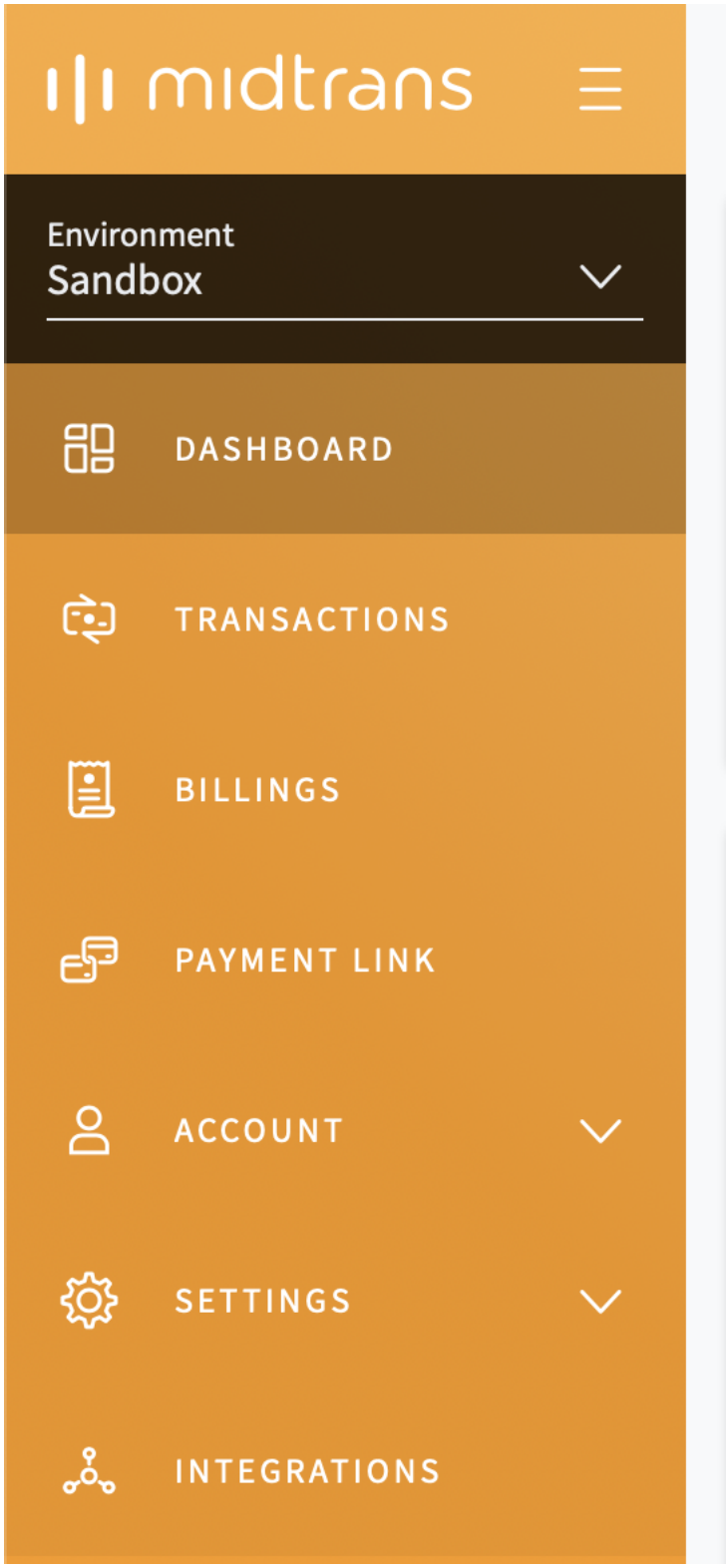


Check your registered email and click on **Activate My Account** button. This will activate your account, and opens a page like above, but with the following message



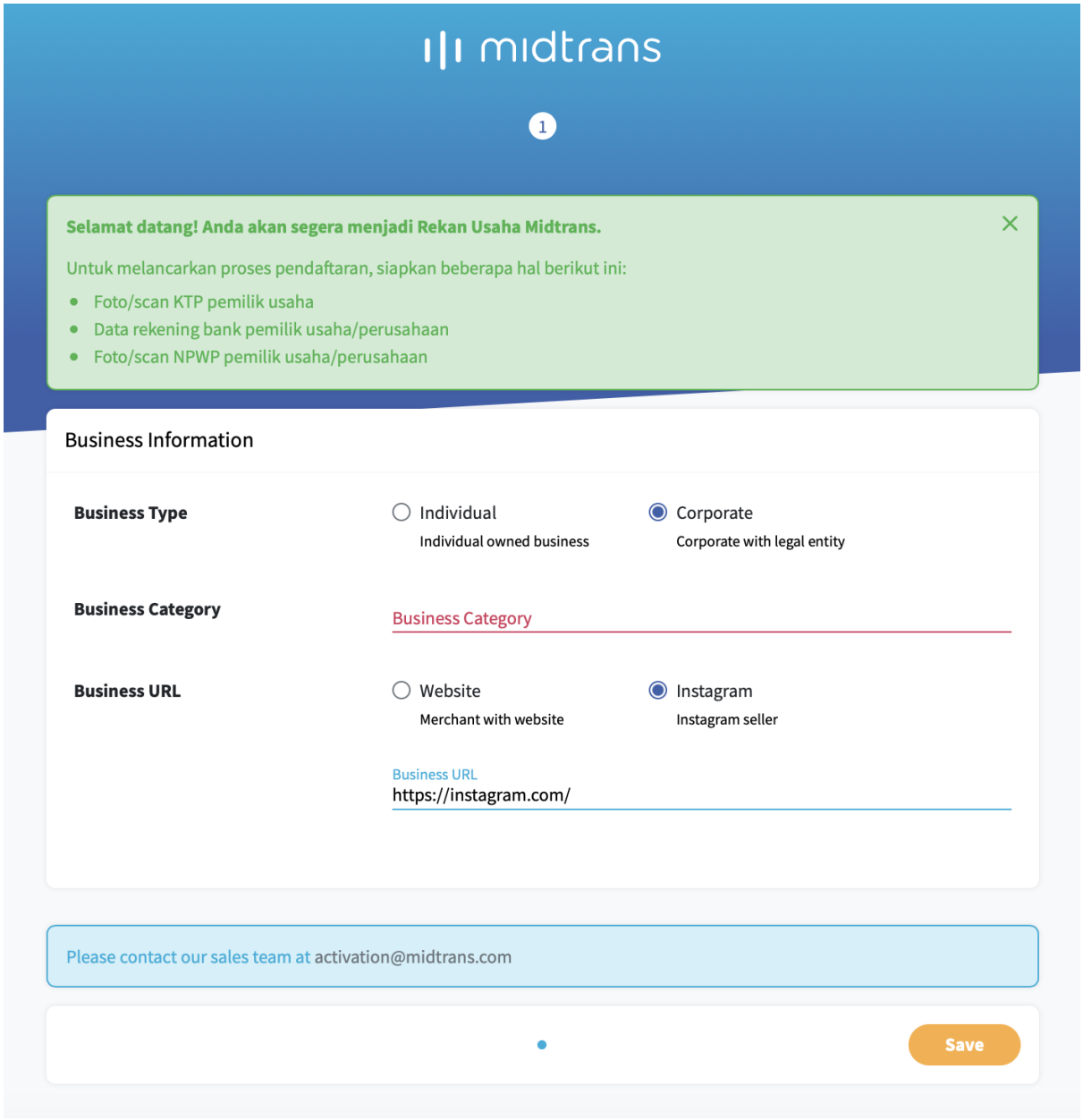
To login use the email and password of your account registration in order to see your dashboard. This process only activate account, and for doing integration you need to submit few documents in the dashboard.

After the login process you will be presented Sandbox Environment, because the documents is still not completed yet. The Sandbox menu will looks like this one on the left side of the page



To complete the registration process and activate the integration, please change **Environment** to **Production**.

Follow the link to complete the registration. Here you can choose your Corporate in Business Type

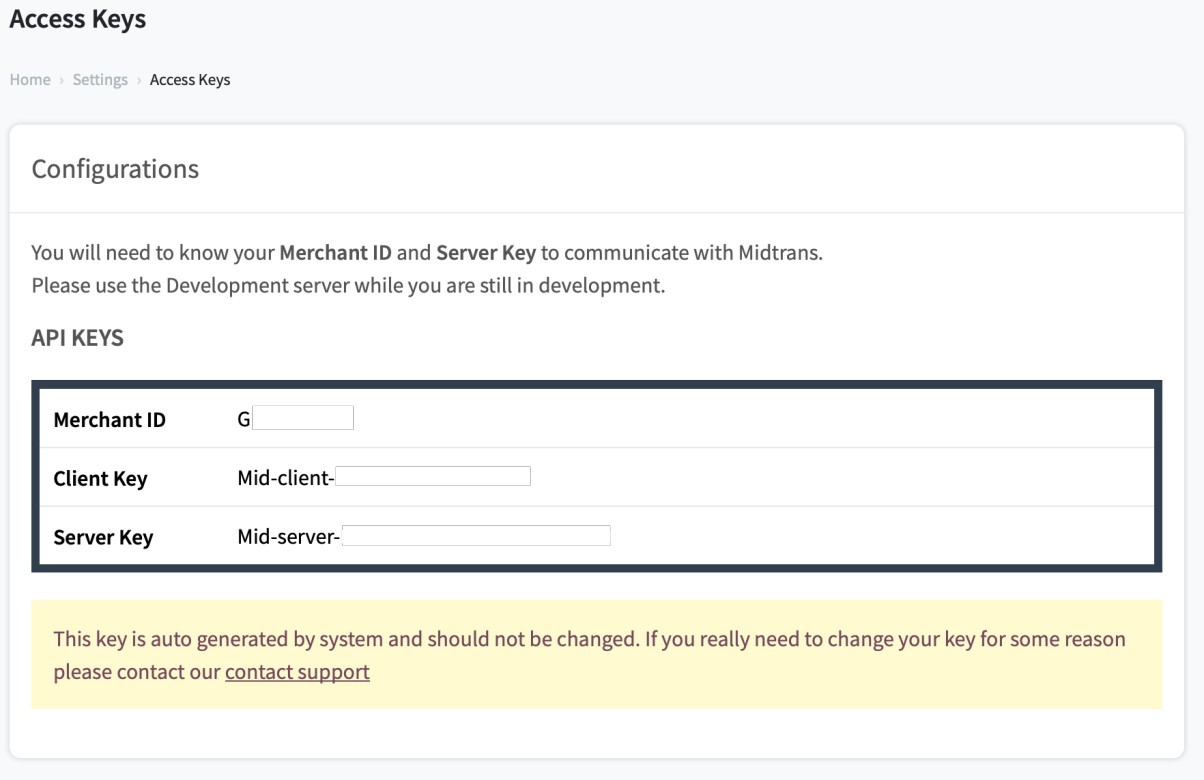


After saving this form, you can contact the sales team (activation@midtrans.com) for further instructions to activate your business account.

After account activation

Please get information about API Keys, that will be used in IACBOX configuration.

Login to your Dashboard, and then click on Settings (down arrow to the right of menu), and then choose Access Keys.



You can copy/paste this values to your IACBOX configuration.

9.4.2 Other plugins

Status

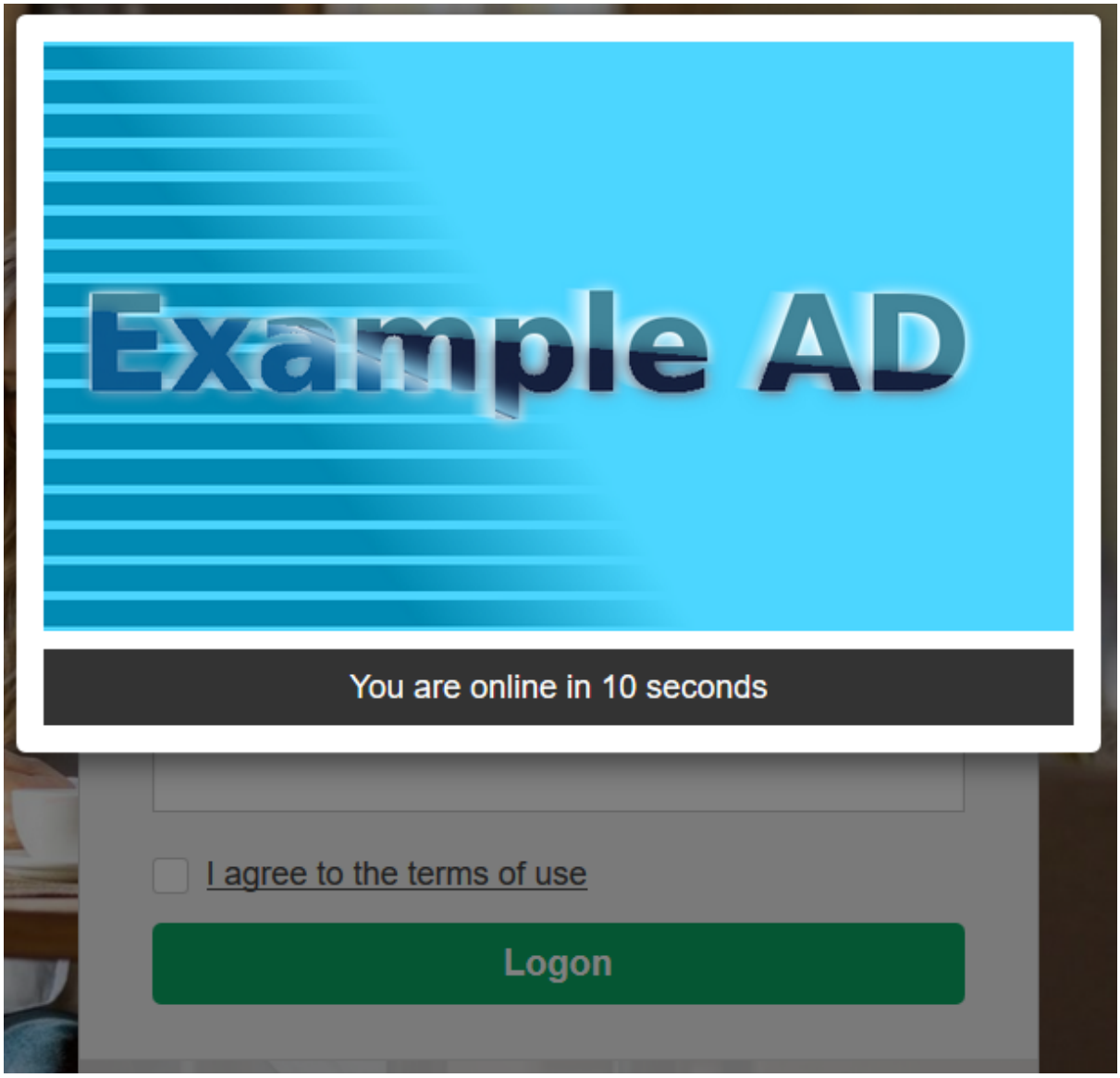
This plugin is the equivalent to the status pop-up on the default logon page and shows your current status.

The status will be displayed above the footer. Please note that **no** pop-up will be opened, so if you redirect the client after a successful login the status will not be visible to the user. A user can manually reach the login page with the following url: http://logon.now

Since version 17.2 the status can be refreshed via a icon

Ads

The Ads plugin allows you to show a modal popup on the landing page for a certain time. The user has to watch it until s/he can access the landing page.

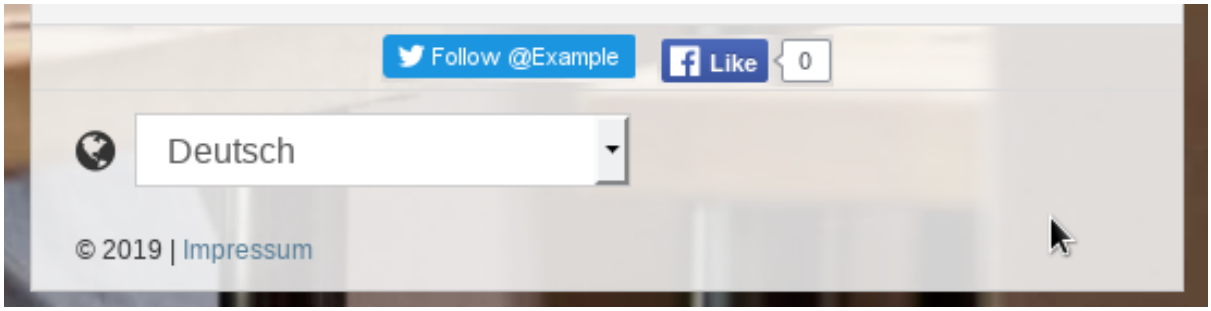


Configuration Key	Since	Mandatory	Default
display-time	2.0	Yes	
Seconds how long the popup should be shown			
ads-type	2.0	Yes	
Type of the ad (image or video)			
image-file	2.0	No	
Image file to be shown			
video-file	2.0	No	
Video file to be shown			
video-poster	2.0	No	
Poster image for the video			

Socialshare

The socialshare plugin offers the possibility to like/share/follow a side. You can use the services of Twitter, Facebook and Google.

The plugin will be shown above the footer.



Attention: Google+ Like Button was removed because Google+ is deprecated with **March 2019**

Configuration Key	Since	Mandatory	Default
enabled-services	17.0	Yes	
Enable which service should be possible for a client			
twitter-name	17.0	Yes	
Name of the Twitter account which should be followed			
twitter-show-screen-name	17.0	No	true
Show the name of the twitter account			
twitter-show-count	17.0	No	false
Show the count of your twitter followers			
fb-like-url	17.0	No	
Url to the page the user should like/ recommend			
fb-action	17.0	Yes	like
You can choose between like or recommend			
fb-layout	17.0	Yes	button
Choose the layout of your button			
fb-data-show-faces	17.0	No	false
User should see the faces of friends who liked the page			
fb-data-share	17.0	No	false
Add a share button to the like or recommend			

9.5 Location based landing page

Since version 2 it is possible to make different configurations and styling for different location or VLANs.

We differntiate between two types of location based configuration. The first is to show different templates and the second is to define different configurations per location. You can mix this two types as you like. The location based functionality has to be enabled in **conf/main.cofig**.

Configuration key	Default
location-based	false
Activates the location based functionality	
location-id-fields	iacbox
Data fields which will be used. Currently available are **iacbox** which is the registration number and **vlan** for the VLAN-ID (or R	

Please note that the IACBOX has to send the needed data for the location. Check your settings in **Modules/Interface/Login API/Custom Logon Page**. The following parameters have to be present in the field **First call data fields with placeholders** when using location based: vl=$VLAN (location per vlan), iac=$REGNR (location per registration number).

Possible location-id-fields configurations

location-id-fields Configuration	Example
vlan	v10
iacbox	2001010101
iacbox, vlan	2001010101-v10

9.5.1 Show different templates per location

After you enabled the location based functionality switch to the file **htdocs/index.php**. Comment in the following code part at the end of the file and comment out the last two lines of the code like the example below. Customize the different cases for your needs. Please note that when using VLANs it is necessary to add the letter **v** before the VLAN-Id.

Listing 9.13: htdocs/index.php

```
<?php

switch ($loginApi->getLocationId()) {
        // The location ID can be the Reg-Nr or the VLAN-ID or a combination - define that in con
        case 'v10':
                // only VLAN matching
                include_once('index_view_XXX.php');
                break;
        case '2001010101': /* intentional fall-through */
        case '2001010102': /* intentional fall-through */
        case '2001010103':
                // only a registration number (of course only useful if more than one system conn
                include_once('index_view_YYY.php');
                break;
        case '2001010101-v12':
                // a combination of reg-nr and VLAN-ID
                include_once('index_view_ZZZ.php');
                break;
        default:
                // There are cases where clients without cookie support (or proper redirect) end
                // needed informations (ip, mac, vlan) so we don't have a location ID - this temp
                // that should provide at least the "restore session" button for humans - most of
                // be hit by apps.
                $template = $loginApi->getTemplateFile();
                include_once($template);
                break;
}

// --- This needs to be out commentet if you want to work location-based
//$template = $loginApi->getTemplateFile();
//include_once($template);
// ---
```

9.5.2 Show different configurations per location

Location based configurations can be made in all *.config* files by grouping config values with so called *sections* which are location IDs in square brackets like **[my-location-id]**.

Attention: **Place sections for location based settings always at the bottom of config files** because all following config values after an location ID belong only to this location. You should always have a default section at the top without any section header.

Example:

```
languages = de, en, fr, it

# ... leave all present configs above and add at the end of the file!

[your-location-id]
languages = es, pt, en

# Example location-id-fields = vlan
```

```
[v10]
languages = es, pt, en

# Example location-id-fields = iacbox
[2001010101]
languages = es, pt, en

# Example location-id-fields = iacbox, vlan
[2001010101-v10]
languages = es, pt, en
```

Available configurations:

The following configuration keys are availabe for the location based functionality in the **conf/main.config**:

- company-name
- comapny-name-legal
- company-website
- logo
- background-image
- show-welcome-header
- show-lang-select
- template
- plugins
- languages
- fallback-language
- redirect-url

If you want to group multiple systems or you have your location information in a database you can provide a custom service by implementing the interface **Iacbox/LoginApi/Core/LocationService** and load that service in the *conf/main.config* with

```
custom-services = MyLocationService
```

We are delivering an example implementation with **Iacbox/LoginApi/Custom/DBLocationService**. It uses a database, but you can develop your own version that connects to a different service.

9.6 Custom services

Custom services allow you to add simple extensions to the LoginAPI core. We already a provide a number of different interfaces as extension points. Simple add a class in the custom directory and implement the interface of interest.

Currently we provide the following custom services:

Name	Description
LogBackend	If you want to provide your own backend for log messages. DbLogger is already an example shipped with the SDK.
StateChangedListener	If you want to react on state changes - most important if a client got online (like our example NewsletterSubscriber))
LocationService	For a location based setup implement this interface to provide your own location IDs based on MAC addresses, VLANs, IACBOX IDs and others
RedirectService	For a custom redirect service implement this interface to provide your own redirects based on your business logic.

To load a custom service add your class name to the config entry **custom-services** in **conf/main.config**. Multiple classes have to be space or comma seperated. Please note that the name of a class and file has to be the same (MyService -> filename Custom/MyService.php).

9.6.1 Example Implementation StateChangedListener

Execute custom code after a successful login with the payment plugin.

Listing 9.14: Iacbox/LoginApi/Custom/ExampleStateChangedService.php

```
<?php

namespace Iacbox\LoginApi\Custom;

use Iacbox\LoginApi\Core\AbstractService;
use Iacbox\LoginApi\Core\Session;
use Iacbox\LoginApi\Core\StateChangedListener;

/**
 * Sample implementation of a custom service
 */
class ExampleCustomService extends AbstractService implements StateChangedListener {
        /**
         * If you want to do something for example send an HTTP request to a CRM system do it in
         */
        public function onStateChanged(Session $session, $oldState, $newState) {
                if ($oldState != Session::ONLINE && $newState == Session::ONLINE && $session->autl
                        //do something here
                }
        }
}
```

9.6.2 Example Implementation LocationService

Execute custom Code for a customized LocationService

The method **resolveLocationId()** is necessary because it will be called automatically. Don't change it!

Listing 9.15: Iacbox/LoginApi/Custom/ExampleLocationService.php

```php
<?php

namespace Iacbox\LoginApi\Custom;

use Iacbox\LoginApi\Core\AbstractService;
use Iacbox\LoginApi\Core\LocationService;
use Iacbox\LoginApi\Core\LoginApiException;

use \PDO;

/**
 * This is a sample implementation of a LocationService that gets the location mapping
 * from a DB. Currently this can only be used on an external webserver with a DB.
 */
class DbLocationService extends AbstractService implements LocationService {

        protected $connector = null;

        public function onServiceLoaded() {
                $this->connector = $this->factory->getRdmsConnector();
                if ($this->connector == null) {
                        throw new LoginApiException('DbLocationService - no RdmsConnector availab
                }
        }
        public function resolveLocationId($requestFields) {
                // Add your custom location code here
        }

}
```

9.6.3 Example Implementation RedirectService

Example Redirect Service

The method **doRedirects()** of the interface **RedirectService** is called at the very end after a successful login for each client. You can redirect each client based on your business logic here. This method **should exit** after sending the redirect via a header.

> **Attention:** Don't forget that captive browsers of Android smartphones **close** after a successful login. You will see redirected websites only on iOS and desktop browsers.

9.7 API definition

> **Attention:** This is the definition of the **raw API**. Chances are good that you can use our software development kit (SDK) (page 182) written in PHP either on an external webserver or the preinstalled version on our custom webserver on an IACBOX. It hides away the low-level details you usually don't need to implement yourself.

Listing 9.16: Iacbox/LoginApi/Custom/ExampleRedirectService.php

```php
<?php

namespace Iacbox\LoginApi\Custom;

use Iacbox\LoginApi\Core\AbstractService;
use Iacbox\LoginApi\Core\RedirectService;
use Iacbox\LoginApi\Core\LoginApiException;

class ExampleRedirectService extends AbstractService implements RedirectService {

        protected $loginApi;

        public function onServiceLoaded() {
                $this->loginApi = $this->factory->getLoginApi();
        }

        public function doRedirect(Session $session) {
                // Add your custom redirect code here

                // For example ask a backend where to redirect the user
                $url = getRedirectUrlFromBackendXy($session->mac);

                header('Location: '.$url);
                exit();
        }
}
```

9.7.1 General

The Login-API uses **indirect communication over HTTP redirects**. All information is passed as URL GET parameters, so there's no need for the external webserver to access the IACBOX directly and no port forwardings and no VPN tunnels are needed.

Protocol version

API versions are always in the format `<major>.<minor>` and should be interpreted like - two different major versions are maybe incompatible with each other - having the same major but different minor version there is a compatible set of data fields and options based on the older version. Minor version updates just add but don't remove data fields and options. Upgrading within the same major release will be safe.

9.7.2 Where to start?

- It's very important to understand the communication flow, so take a look at the flow charts below to understand the redirects.
- Look at our PHP SDK (Software Development Kit) for a production ready implementation which is designed to get you up and running with just a few modifications. Therefore you need only little development skills to have a working implementation or test setup. Of course you are free to develop your own login page in any language with any framework you want.
- Really read this documentation!

9.7.3 Supported logon types

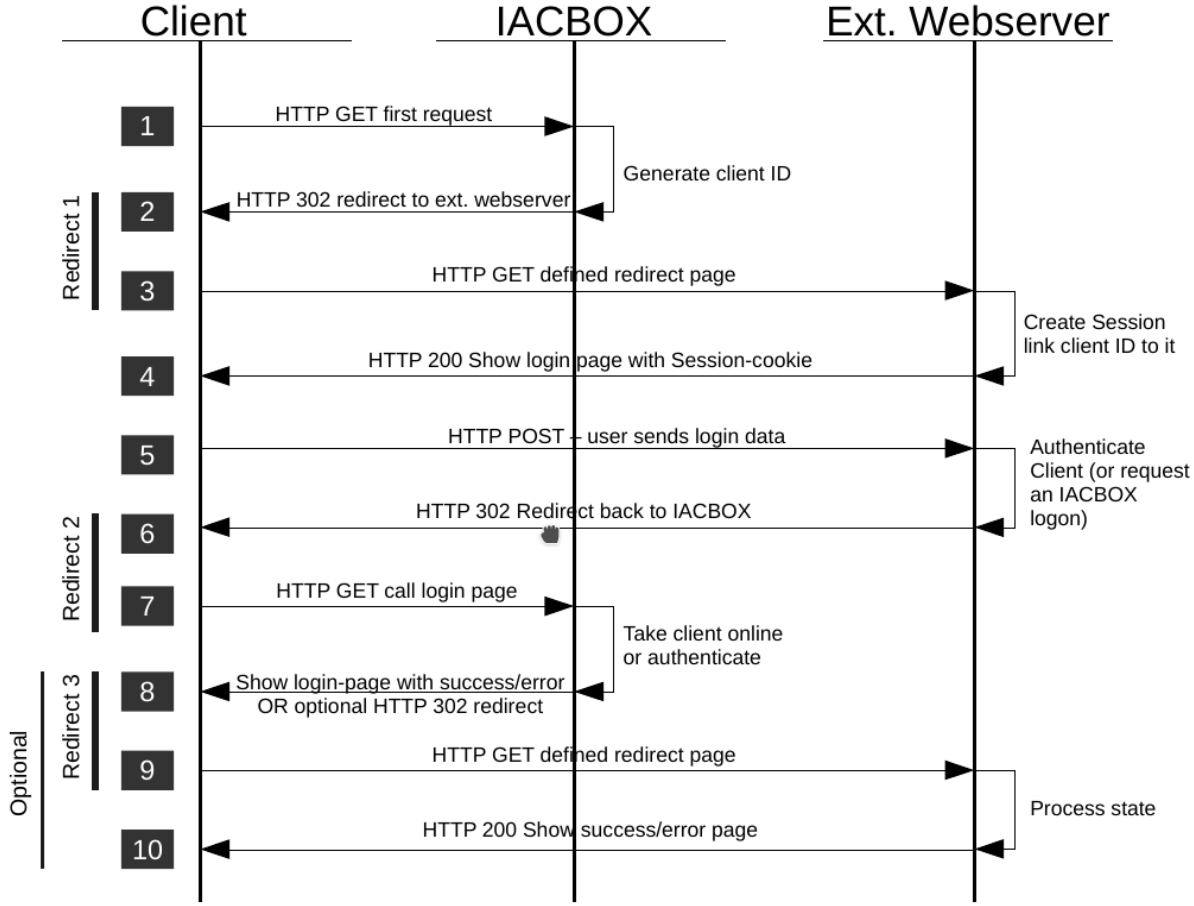
The LoginAPI supports multiple logon methods of the IACBOX but not all. The supported methods can be used all at the same time. Here is the complete list of supported types:

- Type **to**: Take the client online without any authentication (useful if you do the authentication on your external webserver yourself, or the user does not need to authenticate).
- Type **cred**: Normal ticket logon with username and password, local users (Navigate to **Ticket/Users**) and **external authentication** (Radius, LDAP, SQL DB, ...)
- Type **pms**: Login with Roomnumber (mandatory) and other defined PMS data fields like birthday, arrival-day, ...
- Type **create**: Create a ticket by sending a ticket template ID. The new SDK uses this for the local PMS version to create tickets based on the template but it could be used to use any authentication method.
- Type **free**: Create a with the free template. The big advantage of the "free" settings is the interval functionality like (30min per day) which can't be done with just creating a free ticket with type **to**.

9.7.4 Communication flow - general version

In the following section the communication flows is described step by step. This document covers all non-PMS login types (type **to** (take online) and type **cred** (credentials), type **free** (free logon), type **create** (used for payed tickets) which is a more generic approach.

The PMS version is covered later in this document.



1. The client is in state offline and wants to show an arbitrary page. Please note that it works only flawless if this is an unecrypted HTTP call, SSL/TLS works only if the "Redirect offline SSL Traffic" is active (which has other performance/load implications!)
2. The IACBOX catches this connection, creates a new unique ID for this client and sends an HTTP 302 (moved temporary) redirect to the client. The redirect URL consists of the specified external landing page and additional parameters which contain data fields like IP/MACaddress, Vlan ID, ...
3. The client calls the redirect (1) URL on the external webserver.

4. On your webserver a session (cookie based) has to be created to remember the client ID for this browser. The webserver responses with the login page (and the session cookie).
5. The enduser fills out the login form and submits the form. A HTTP POST will be sent to the webserver.
6. **If there was an error on your side then the user gets your login page again and starts again with step 5. On success the**

    (a) type = **to** (take online): your webserver authenticates the client login data against its backend and sends the needed parameters to make the IACBOX generate a ticket and set this client online.

    (b) type = **cred** (credentials): or you want the IACBOX to do the authentication. Currently supported logon types are normal tickets, local users, extauth modules. The redirect has to contain the user credentials with logintype to the IACBOX. The second redirect is made to the IACBOX which contains the needed parameters to either set this client online or try to authenticate it there.

7. The client calls the redirect (2) URL.
8. The IACBOX now takes this client online (case 6.1) or tries to authenticate it (case 6.2). If you wish another callback on success or the login failed, another redirect (3) is made. Otherwise the IACBOX landing page is shown with the success/error message.
9. If a callback was configured then the client calls again the external webserver.
10. Your landing page can now visualize the state success/error and display any other information.

Outgoing redirects (IACBOX/ext or loc. webserver)

The outgoing URL parameters are all setable by yourself. The default values match our SDK implementation and are identical with the parameters for incoming requests. The URL parameter names (default: lapi and si) are changeable to also support REST like URLs instead of the query style. You can change this parameters only for outgoing requests - incoming always need the query style format. Leave the default parameter names unchanged if you want to be compatible with the SDK.

Example with our default parameter names:

https://your.domain.com/path/login?lapi=vLQT8uyK1...gNIm4_Ec0w&si=Ezo5zswu...OxaNCepI

Allowed URL placeholders in outgoing redirects

Hint:

- *Redirect*: You find the redirect numbers in the flow chart above.
- *Required*: Only in the scope of the redirects specified

$VERSION

- Description: The *protocol* version.
- **Required**, Redirect: 1 and 3
- Format: <major>.<minor>
- Default field: ver
- Example: 2.1

$ID

- Description: The ID which identifies the client. This is a random token.
- **Required**, Redirect: 1 and 3
- Format: 16 bytes in base64Url, 22 chars
- Default field: id

- Example: 7yXYL...1LlCw

$ACTION

- Description: The action should be triggered. Redirect 1 is **auth**, Redirect 3 is **cbk**. Maybe gets extented in the future
- **Required**, Redirect: 1 and 3
- Format: Enum **auth** (1), **cbk** (3)
- Default field: ac
- Example: auth

$IP

- Description: IP address of this client
- **Optional**, Redirect: 1
- Format: X.X.X.X
- Default field: ip
- Example: 172.29.0.15

$MAC

- Description: MAC address of this client
- **Optional**, Redirect: 1
- Format: MAC address without colons, lowercase, 12 chars
- Default field: ma
- Example: 00359ac076c4

$VLAN

- Description: VLAN-Id of this client or an empty string if this IACBOX has no VLANs configured. If the IACBOX rus in routing mode and you prefer the routing mode and you prefer the routing match, then subtracting 4096 from this ID > 4096 will give you the ID of the route.
- **Optional**, Redirect: 1
- Format: 0-4096 for VLAN IDs, 4097-8192 for route-IDs
- Default field: vl
- Example: 17

$REGNR

- Description: The registration number of this IACBOX if more than on box calls your server
- **Optional**, Redirect: 1 and 3
- Format: Version Nr with 10 chars
- Default field: iac
- Example: 2014112233

$USERSONL

- Description: Users online
- **Optional**, Redirect: 1
- Format: Number: 0-99999
- Default field: uo
- Example: 527

- Since: Version 1.1

$USERSONLPERC

- Description: Users online as percentage (rounded to integers)
- **Optional**, Redirect: 1
- Format: Number: 0-100
- Default field: uop
- Example: 53
- Since: Version 1.1

$MAXUSERS

- Description: A voluntary max. value. Usually the same value as LICUSERS. 0 < MAXUSERS <= LICUSERS
- **Optional**, Redirect: 1
- Format: Number: 10-99999
- Default field: mu
- Example: 800
- Since: Version 1.1

$LICUSERS

- Description: Max. users defined in the license. Note: unlimited = 99999
- **Optional**, Redirect: 1
- Format: Number 10-99999
- Default field: lu
- Example: 1000
- Since: Version 1.1

$RC

- Description: The return code that represents the state of this request. See all possible states below.
- **Required**, Redirect: 3
- Format: 1-4 digits state code
- Default field: rc
- Example: 0

$ERROR

- Description: The error-message if the $STATE indicates an error
- **Optional**, Redirect: 3
- Format: Error message as text (translated if possible)
- Default field: err
- Example: Wrong username or password.

$USERURL

- Description: The URL the user wanted to get before s/he got redirected to the lofin page. This is no always reliable especially for HTTPS URLs. Try to keep this at the end as it can get long.
- **Optional**, Redirect: 1
- Format: URL

- Default field: userurl
- Example: http://example.com

Incoming redirects (ext or loc. Webserver/IACBOX)

Incoming redirects to logon/authenticate a client need this parameters:

lapi

- Description: The data fields (optionally encrypted)
- **Required**
- Format: Base64Url encoded

si

- Description: Signature of the value sent in **lapi**
- **Required**
- Format: Base64Url encoded

Example with our default domain name - the parameters have to be that way:

https://hotspot.internet-for-guests.com/logon/cgi/index.cgi?lapi=vLQT8uyK1...gNIm4_Ec0w&si=Ezo5zswu...OxaNCepI

Data fields

The data which is send between the IACBOX and your server is packed and encrypted/encoded in the **lapi** URLParameter. The fields are **;** (semicolon) separated key=value pairs in the format: `key1=value1;key2=value2;key3=value3;...`

Ensure that you cut out all possible semicolons from the data fields!

ver

- Description: The *protocol* version
- **Required**
- Format: <major>.<minor>

id

- Description: The ID which identifies the client. Just send it the way you received it
- **Required**
- Format: 16 bytes in base64Url, 22 Chars

ac

- Description: The action that should be triggered. Usually you use **logon** for all login related things. Since version2 there's also a refresh (ref) action if you lost the client session and have to recover it with a refresh cycle.
- **Required**
- Format: Enum: logon, ref

type

- Description: The type of authentication that was/should be used. In case of **to** and **free** you have done the authentication on the server yourself and tell the IACBOX it should simply take this client online without any further checks or are using the free logon service. In every other case the IACBOX does the authentication and you have to pass username and password or a template id.
- **Required**

- **Format: Enum:**
  - **to** = Take online
  - **cred** = Credentials - user/pwd logon
  - **pms** = for PMS logons
  - **free** = free logon
  - **create** = template id

lang

- The ISO language code which is used to translate the login error/success messages. This error messages are translated into many languages. Have a look at **Client Logon/Terms of Use** for a list of available languages.
- **Required**
- Format: 2 char language code like en, de, fr, it

user

- Description: If the IACBOX does the authentication (type is not **to**) this is mandatory.
- **Required**
- Format: String

pwd

- Description: If the IACBOX does the authentication (type is not **to**) this is mandatory.
- **Required**
- Format: String

desc

- A custom text set as description of the generated ticket
- **Required**
- Format: String
- Since: Version 1.3

userurl

- Description: If you use no final callback and the IACBOX should redirect the user to his/her original wanted URL. Only for successful logins.
- **Required**
- Format: URL

ref

- Description: Used only for action **ref**. This can be a custom ID if you need to support a mapping of a returning refresh call to a certain sub page. See chapter **Recover from lost session**
- **Required**
- Format: String

Ticket overrides

You may also append these data fields which override values of the ticket template

otc

- Description: Time credit

- Format: Number of seconds

otl

- Description: Time limit
- Format: Number of MB

omi

- Description: Max idle timeout
- Format: Number of seconds

oep

- Description: Expiration persiod - just use **time()+seconds** to get an absolute timestamp
- Format: Unix timestamp, Number of seconds

odl

- Description: Max download bandwidth (max value is the total download bandwidth under **System/Network**)
- Format: Number of kBit/s

oul

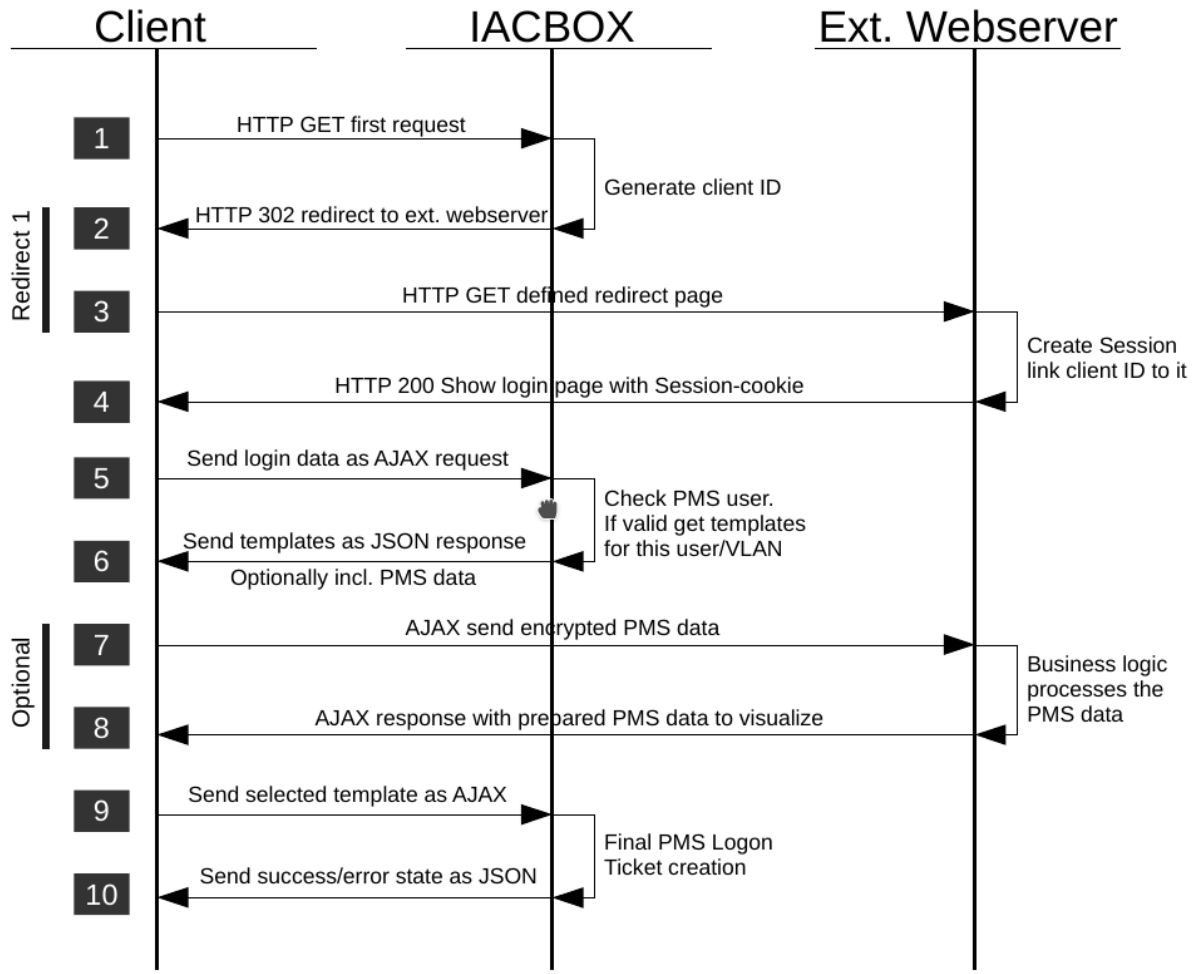
- Description: Max upload bandwidth (max value is the total upload bandwidth under **System/Network**)
- Format: Number of kBit/s

ode

- Description: Custom ticket description - useful if you want to add an external ID which can be used to search later for this ticket
- Format: String

9.7.5 PMS Logon

Communication flow - PMS version



1. The client is in state offline and wants to show an arbitrary page.
2. The IACBOX catches this connection, creates a new unique ID for this client and sends an HTTP 302 redirect to the client. The redirect URL consists of the specified external landing page and additional fields like IP/MACaddress, Vlan ID, ...
3. The client calls the redirect (1) URL on the external webserver.
4. On your webserver a session has to be created to remember the client ID for this browser. The webserver responses with the login page (and the session cookie).
5. The enduser fills out the login form and submits the form. An AJAX call via HTTP POST is send to the webserver with the login data.
6. The IACBOX verifys the user data (roomnumber, name, PIN, ...) by checking the PMS. If the user already has a valid ticket the success state **online** is send as response. In the normal offline state all ticket templates visible to this user (depending on optional configured VLANs) are returned as JSON structure. If there is only one free template and skipfreetemplates is active in the PMS configuration then the client is online rigth now. In any case starting with API version 1.3 there now can be additional PMS fields passed encrypted.
7. If additional PMS data was sent the client browser now does an AJAX call to your webserver with the encrypted PMS data. Edit the Javascript if you don't want that.
8. Depending on the business logic on the server side there is now a chance to individualize the logon page by sending arbitrary data back which has to be interpreted by the Javascript (eg. Show the name or the roomnumber of this guest).

9. If a template selection has to be made, the user now selects a ticket template which is send again as AJAX request to the IACBOX which tries to set this client online.
10. The success/error state is returned as JSON response. The page can now shown the success/error message.

Please note the PMS system in local mode is handled completely different, because it does not need any javascript code.

Incoming AJAX requests (Browser -> IACBOX external only)

The external PMS version uses no incoming redirects like the normal version. It uses AJAX requests from the client browser which need the parameters below. Please note that because of a crossdomain situation here we have to use a JSONP call, which has to be an HTTP GET call.

Since version 2.1 we also support the PMS systems Himed and ASAj.

lapi

- Description: JSON data
- **Required**
- Format: JSON string as POST data

callback

- Description: Has to be **dataCBK** for the SDK to work
- **Required**
- Format: String

userid

- Description: The Login-Api Id
- **Required**
- Format: 16 bytes in base64URL, 22 chars

tt_id

- Description: The selected ticket template Id
- **Required**
- Format: Number

pms_room

- Description: Room number
- **Required**
- Format: String

pms_himed_room

- Description: Room number for the PMS Himed
- **Required for the PMS Himed**
- Format: String

pms_name

- Description: Room number
- **Required**
- Format: String

pms_fname

- Description: First name needed for PMS ASAj
- **Required for the PMS ASAj**
- Format: String

pms_lname

- Description: Last name needed for PMS ASAj
- **Required for the PMS ASAj**
- Format: String

pms_pin

- Description: Pin
- **Optional**
- Format: String

pms_arrivalday

- Description: Arrival day
- **Optional**
- Format: String DD.MM.YYYY

pms_departureday

- Description: Departure day
- **Optional**
- Format: String DD.MM.YYYY

pms_birthday

- Description: Birthday
- **Optional**
- Format: String DD.MM.YYYY

Example with our default domain name - the parameters have to be that way:

https://hotspot.internet-for-guests.com/logon/cgi/index.cgi?lapi=tt&callback=dataCbk&userid=d3D3C...RUA&tt_id=35&pms_roo

JSON answer with the ticket templates

```
{
        "rc" : 0,
        "state" : "tsel",
        "tt" : [
                {
                        "id" : 435,
                        "name" : "Free Access",
                        "price" : "0.00 EUR",
                        "txttyp" : "Ticket Type: Flat Rate",
                        "lifetime" : "Time credit: 30 Minutes",
                        "expires" : "Expires: 1 Days",
                        "bw_out" : "Max. upload bandwidth: 512 kBit/s",
                        "bw_in" : "Max. download bandwidth: 1024 kBit/s",
                        "session_limit" : "Session Limit: 2000 MB"
                },
                {
                        "id" : 174,
                        "name" : "Premium Speed",
                        ...
                }
```

```
        ],
        "pms" : "0Wdhuvpa...ufuPpn",
        "pms_si" : "bGcpGgRt...diKAtY"
}
```

The strings in the example above are available in over 20 languages and get translated into the browser language.

Returning PMS data

Please note that this feature is only supported with certain PMS types (All FIAS based protocols like Fidelio, Protel). Some fields maybe also have different values with different PMS types (like VIP).

The PMS fields **pms** and **pms_si** (= HMAC signature of the encrypted pms field) are optional if configured in the login API settings. The javascript sends these fields to your webserver where the data can be decrypted and processed. The response can trigger custom visual changes on the logon page to react on the current user.

Possible data fields

id

- Description: Id ffrom PMS

names

- Description: First- and lastname
- **Response field names:**
  - fname (first name)
  - lname (last name)

room

- Description: Room number

adate

- Description: Arrival day

ddate

- Description: Departure Day

bdate

- Description: Birthday

vip

- Description: VIP group name(s)

Below there's a JSON answer if the client is online. This can be already the answer after the user sent the PMS data if there is still a valid ticket for this user or there is only one free ticket template active with enabled skipfreetemplates option in the PMS configuration. The PMS fields are only added if no ticket template selection was needed.

```
{
        "rc" : 0,
        "state" : "online",
        "pms" : "0Wdhuvpa...ufuPpn",
        "pms_si" : "bGcpGgRt...diKAtY"
}
```

9.7.6 Recover from lost state/session

If you have pages/services a customer can reach when s/he is already online you will have the problem that the session is most propably lost - you can't identify the client any more. This is the case because nowadays nearly all OS use a captive browser to do the authentication and so the session cookie is only available in this browser. If the user now comes back to your landing page with the normal browser the cookie is not there and so you don't have the device info like MAC, IP, VLAN aso. To recover this information we provide a so called *refresh call* which are technically two redirects one to the IACBOX and the second back to the LoginAPI code including the missing device information. Just redirect the user to this URL (adapt the domain if you have a custom domain):

```
https://hotspot.internet-for-guests.com/logon/cgi/index.cgi?lapi=ref&ref=<refid>
```

Replace <refid> with *1* if you don't need it or use a refresh ID that is used in a mapping.

Refresh ID mappings

On an incoming refresh call from the IACBOX the LoginAPI request router does not know anything what to do with this call and will show the normal login page. So if you want to show another page you will need a mapping in main.config like that (without the brackets < and >):

refreshidmapping = <refid>:<targetpage.php>

9.7.7 General information

URL length

Please remember that the URLs should be as short as possible. The IACBOX proxy can handle URLs with a total length up to 8000 chars. This is also the reason why we use Base64 encoding for the data fields to have the smallest encoding overhead possible.

Possible States

All non 0 states are errors.

State	Description
0	OK request successfully processed
1-9998	Error codes - please have a look at our SDK for all possible values.
9999	General/unknown error

Base64 URL safe encoding

We use **base64Url** (See [1]) as encoding at different places including URLs to save as much space as possible. Base64Url is the same as Base64 except that character 62 (plus **+**) and 63 (slash **/**) are replaced with the URL safe characters **-** (minus/dash) and _ (underscore). Decoding works the inverse way. See [2] for the whole alphabet. In addition to that the padding is truncated after encoding (all **=** chars at the end are removed) and gets appended to the string before the Base64 decoding is done (pad **=** until the string length is a multiple of 4). Please have a look at the SDK for a sample implementation in PHP.

User URL

User URL is the URL the user wanted to see before s/he was redirected to your login page (Step 1). If you want the user to be redirected to this page after a successful logon add the needed placeholder **$USERURL** to your login URL. Depending on if you have configsample_logonpage_viewured a final callback after the logon on side of the IACBOX (Step 8) the redirect has to be done at different places.

- If you want a final callback you have to do the redirect yourself (the PHPSDK includes this functionality already)
- If no callback is used, the IACBOX does the redirect if the data field userurl is provided.

9.7.8 Security

- It's recommeded to use encryption. The URL parameter (lapi) which holds the data fields is encrypted and so is neither read nor changeable for the user or an attacker in between.
- But also if encryption is not used, all URLs have to be signed with a HMAC so both sides can trust the calls.
- It's also recommended that your logonpage uses SSL/TLS to protect the communication, especially in wireless networks.

HMAC signature

Because the communication between the IACBOX and the external webserver is achieved by URL redirects we have to be sure that the data fields can't be manipulated. To do so the data fields (content of the lapi parameter) of every call in each direction are signed with a HMAC signature. Compared to just hashing a key and the message a HMAC hash is safe against a lengthextension attack which easily allows you to create a valid MAC with different content.

In the current version this is a **HMAC SHA256** (See [3]) encoded as Base64Url. The shared secret used for this HMAC is set in the configuration and has to be identical on the IACBOX and your webserver. For further informations see [4] and [5].

Salted keys for unencrypted communication

Attention: Attention - this is an incompatible change of version 2.x! In version 1.x no salt was added to the HMACs.

- If you use encryption then the key of the HMAC is not salted, because the encryption adds a salt anyway - so with encryption the HMAC is the same as in version 1.x.
- If you don't use encryption starting with version 2.x you have to salt the key used for HMAC generation to get a different signature also if the clear text is the same. The salt gets prepended to the resulting HMAC and separated with a dollar **$** like `<salt-b64>$<hmac-b64>`. Use 8 random bytes as salt which are base64Url encoded. `HMAC = base64url(hmac_sha256(cleartxt, salt+secret))`

Used shared secret:

```
v09q5JFPZCv_nwMRyKsRWtDS9JtFghzR
```

Clear text:

```
ver=2.1;id=dZDzvCrCdz2MxsN2GqlMtw;ac=auth;ip=172.29.0.1;ma=8fa72685eb68;vl=0;iac=201601
```

Salt Base64Url encoded:

```
V1fhYVxaj5w
```

HMAC with prepended salt Base64Url encoded:

```
V1fhYVxaj5w$boR-6lCDj1QXkIweZzoaGoA2PyCe8kQjyCipnTSyj0Q
```

Encryption

It's recommeded to encrypt the URL parameters, especially if you transmit user credentials or PMS data to the IACBOX. In our SDK for PHP the encryption code is fully implemented and you simply have to turn it on (default).

If you develop the client code yourself use this settings:

- Algorithm is **AES-256** in CBC mode (See [6])
- As key use `SHA256(sharedsecret)` what ensures the needed keylength of 32 bytes
- Use 16 random bytes as IV (initialisation vector/salt) which have to be prepended to the encrypted text.
- PKCS7 Padding is used
- Encode the result with Base64Url

```
encrypted = base64url( IV + AES256(datafields) )
```

Implementation tests

Clear text used for the examples below:

```
ver=2.1;id=dZDzvCrCdz2MxsN2GqlMtw;ac=auth;ip=172.29.0.1;ma=8fa72685eb68;vl=0;iac=2016010
```

Used shared secret:

```
v09q5JFPZCv_nwMRyKsRWtDS9JtFghzR
```

The result of every encryption (of the same text) is different because of the used IV/salt. Here are the 16 bytes Base64Url encoded used for the encryption and HMAC below:

```
hELE1zweeT2yT1JVLQ8auQ
```

Expected AES256 encryption incl. prepended IV (Base64Url encoded):

```
hELE1zweeT2yT1JVLQ8auQkn_CXQVEBj4SPEes0a8PDa0F2bU6-JFtH_SNAYJQb-Zd-RqGzvMIk
UbhhrU5Ll78h_UbDv4PfRVD5N5I37anPXvAi7__fO3yJ_ISFc3qf6baYjVx-cqZdlP36o6ODAGw
```

HMAC of the Base64Url encoded string:

```
kbihE5UaIIiT2q4P65qPfNUpw5cVtyZDxZKIiLFGb8E
```

9.7.9 Things to consider, common pitfalls

- You will get multiple requests per client to your webserver and some of them origin from all kinds of software or mobile apps operating on port 80 or 443, so don't underestimate the number of requests to your server. Because of this it's wise to avoid a heavy loaded login page with lots of images and other elements. You can show more advanced content on the final callback page if the client was successfully set online.
- Because all images, javascripts and possible iFrames included in your login page are loaded by the client browser (which is still in offline state) have to either come from your webserver with the same hostname or this URLs have to be whitelisted in the "Hidden Free to Use" section as well. Note, that whitelisting works only for assets with static URLs and will not work properly for web services of big companies with CDN (content delivery networks) like google (googleanalytics), facebook, twitter, aso. where the same domain name resolves to different IP addresses very quickly. This is especially a problem with javascript widgets (twitter, ...) and user tracking tools like google analytics. You can have rich content on the final callback page where the client is in online state.
- Avoid URL rewriting for the login page at your webserver since this results again in HTTP 302 redirects which multiplies the problem of getting too many unwanted connections.
- Check your session timeout at your server - it should have a resonable value > 20min and < 3h.
- Consider some kind of load balancing or failover setup at your webserver to avoid downtime. When your webserver is not reachable, nobody can login.

9.7.10 Testing

Keep in mind that while your're testing your implementation and you login/logoff frequently you have to revoke your ticket each time otherwise you will not see the login page again and the IACBOX will not apply new values like ticket overrides, etc..

Use non HTTPS URLs for testing because TLS traffic is handled completely different and will not even work if HTTPS redirects in offline state are disabled.

9.7.11 Monitoring

The IACBOX comes with builtin monitoring which you should use to monitor your external webserver. This tells you if your landing page is reachable or not.

Navigate to **System/Monitoring**, activate it if needed and add a **New Device**. As host define the domain of your webserver. Then click on the edit icon on the right and click on the tab **Checks** and add one with **New**. As **Test Type** select HTTP(S) and save. You can append the path if you want to check the landing page instead of the host only by adding the parameter `-u /path/to/index.php?ping=1` . The parameter `?ping=1` is only useful if you use our SDK because the script exits very early, causes least possible load and does not pollute the log with errors. If you want to get an email notification or a rsyslog message to your remote syslog server you can configure that too. Navigate to **System/Notifications** and add the following line to the large text field:

```
ACCEPT " MONITORING STATE " my.name@example.com
```

With a rsylog server:

```
ACCEPT " MONITORING STATE " my.name@example.com rsyslog.example.com:514
```

9.7.12 References

1. http://en.wikipedia.org/wiki/Base64
2. http://tools.ietf.org/html/rfc4648#page8
3. http://en.wikipedia.org/wiki/SHA2
4. https://tools.ietf.org/html/rfc2104
5. http://en.wikipedia.org/wiki/Hashbased_message_authentication_code
6. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

CHAPTER

TEN

LOGON PAGE

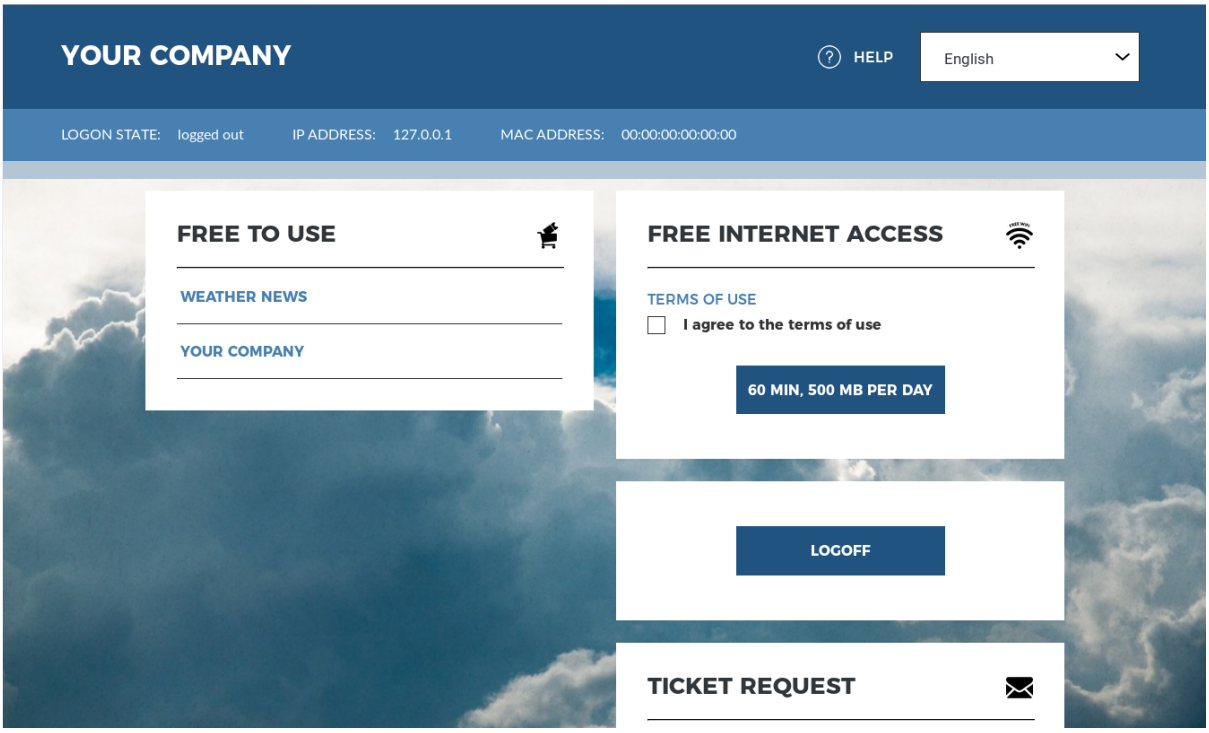
10.1 Background image

New in version 17.0: Background images of the logon page can now be easily exchanged for our new *Metro* styles.

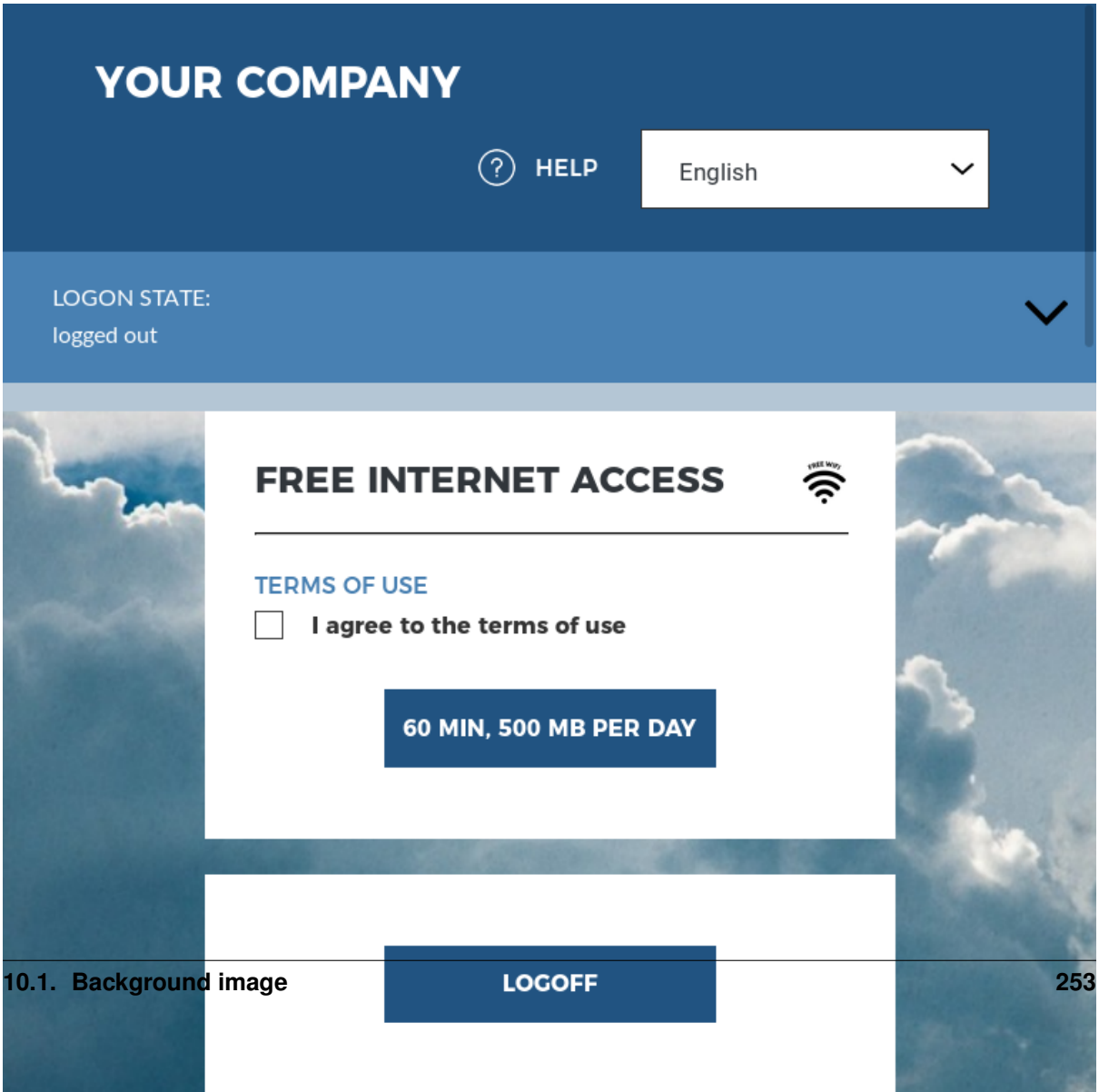
10.1.1 Understanding the responsive design

The new *Metro* styles are responsive, so the page adapts to the screen size of the client-device. Currently 3 different sizes are used which map roughly to screen sizes of **desktops, tablets and mobile phones**. The background image can be seen in the desktop and table size, but not in the version for mobile phones as there is not enough space left what would justify to load the image.

Desktop (large displays)

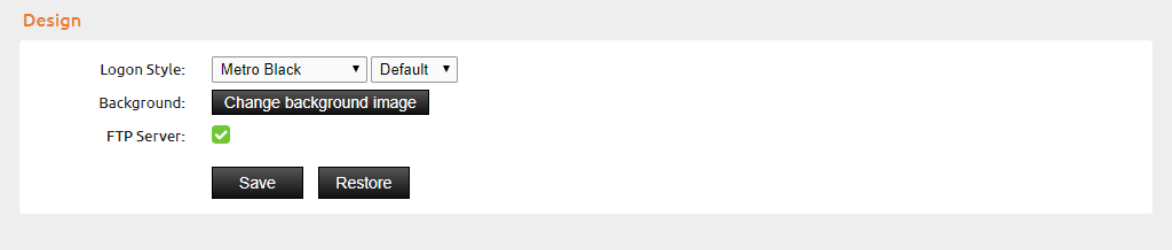


Tablet (medium displays)

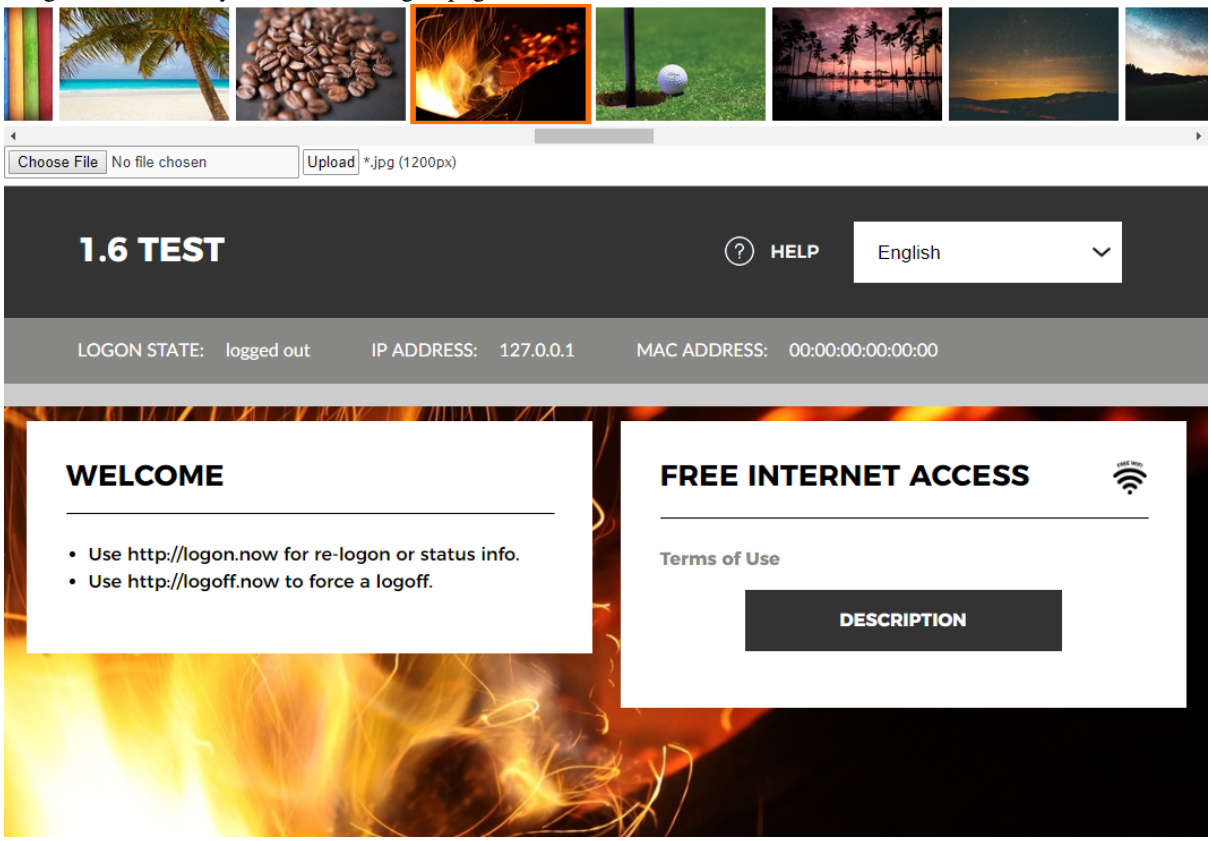


10.1.2 How to change the backgound image

To change the background image navigate to **Client Logon / Design** and click on the tab **Themes** and then on the button **Change backgound image**.



In the popup window you can now choose one of the preinstalled images just by clicking on it. The new backgound image is immediately active on the logon page!



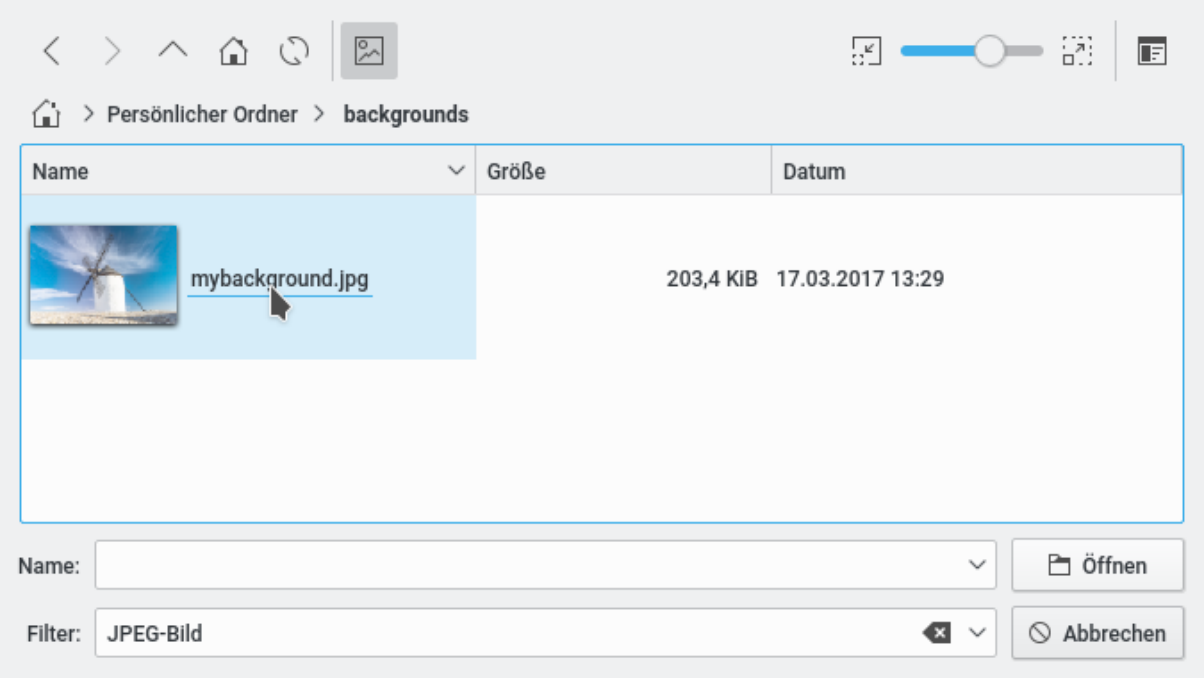
10.1.3 Upload your own image

If you want to upload your own image, click on **Choose File**.

Choose an JPEG image form your PC.

Attention:

- Format: Only **JPEG** images are suppoted. Ensure your image has a *.jpg or *.jpeg file extension.
- Size: The image should be scaled to a width of about **1200px**. Remember that a too big image can slow down your logon page and leads to a bad user experience!



The uploaded background gets active immediately. All custom backgrounds are listed at the beginning of the scroll list.

10.1.4 Deleting a custom image

Click on the **X** symbol in the upper-right corner of the image to delete a custom image. Preinstalled images can not be deleted.

Hint: If you want to restore the original image you can click on **Restore** to fully reset the current style. **Note that this also deletes custom changes made via FTP**.

10.2 Customize Logon Page

Hint:

- **HTML and CSS knowledge is required** to modify the design of the IACBOX *Client Logon Page*.
- Note that some templates may be changed by future IACBOX versions. If this is the case, any changes performed need to be migrated again.
- Always use the same template variables as in the original when creating your own template design.
- As an alternative you can also use the built-in feature **Redirect before Logon**. This allows you to redirect users to a defined website (e.g. corporate website) before the actual logon process. The website defined requires a button/link which then redirects guests back to the IACBOX *Client Logon Page* for logging in.

10.2.1 Redirect before Logon

With **Redirect before Logon** it is possible to redirect guest devices to an external website.

This can be used to place hints, additional information, terms of use, restaurant menu cards, etc. on this website.

Since guest devices still have to log in on the regular IACBOX *Client Logon Page*, they need to be redirected onto it again.

Therefore you may place a button or link on your external website which then redirects the users to the IACBOX *Client Logon Page*. Use the following link: https://hotspot.internet-for-guests.com/logon/cgi/index.cgi

If you use the feature **Free Logon** on the IACBOX, you can call a link which would directly log in users via the **Free Logon**. This is done by sending 2 additional **GET parameters** which you can find in the link below. https://hotspot.internet-for-guests.com/logon/cgi/index.cgi?freeperperiod=1&accept_termsofuse=1

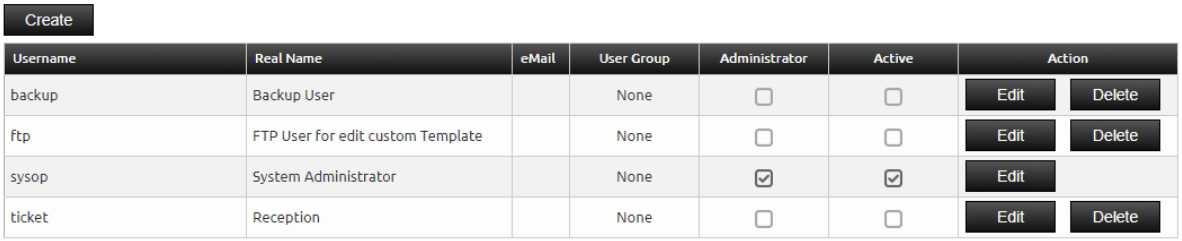
Note again that for this, the Free Logon (page 149) must be activated and configured in the WebAdmin menu **Tickets / Templates**.

10.2.2 Edit HTML templates

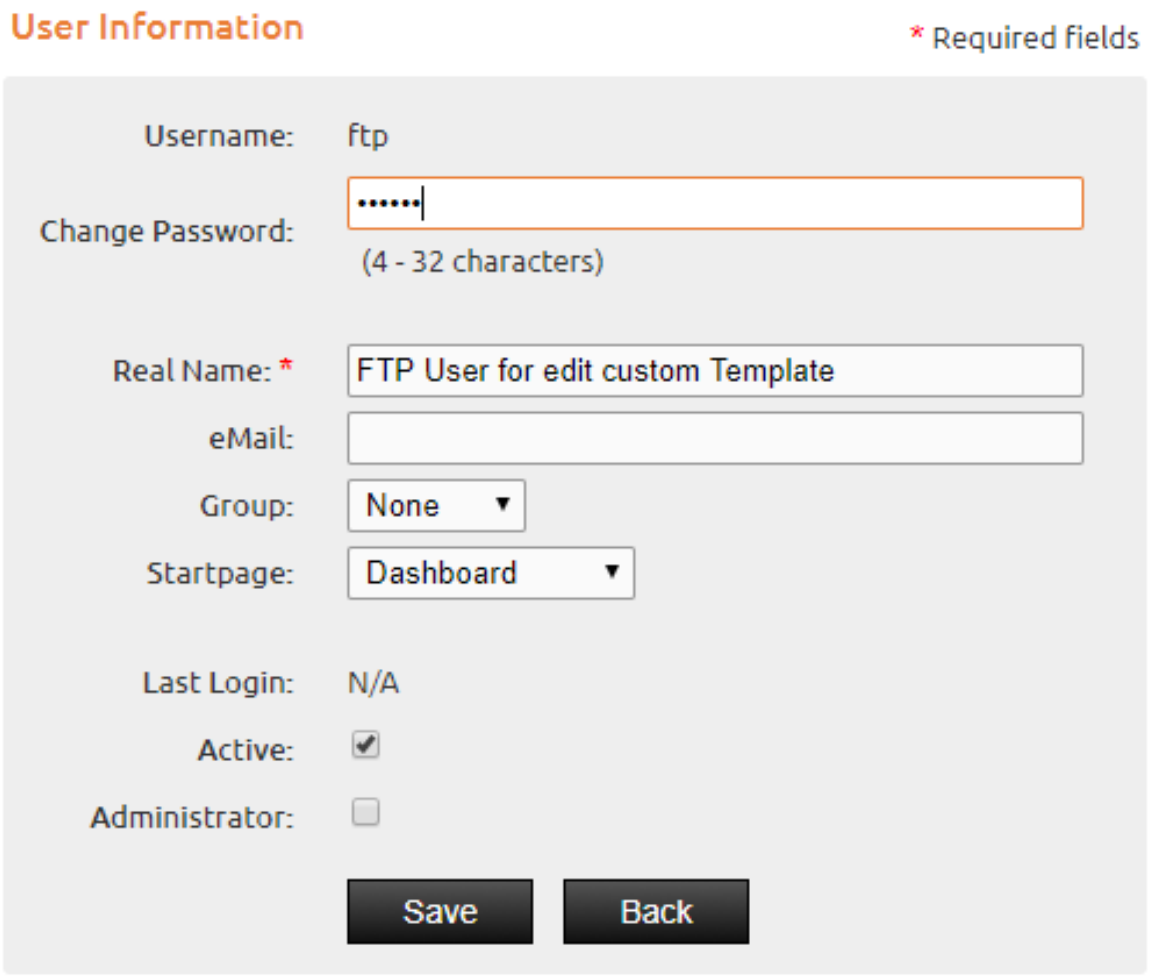
In case you want to edit the *Client Logon Page* directly, you can edit the **HTML template files**. **FTP access** is required for transferring the HTML template files to/from the IACBOX system.

Activate the FTP User

The IACBOX provides an ftp user which is disabled by default. Therefore you have to activate the user and assign a password. Switch to the WebAdmin menu **System / Manage User** and edit the user **ftp**.

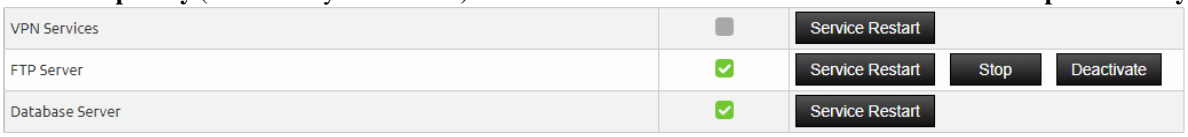


At the window shown below, set a **password** for the ftp user, **activate** it and then **save** the changes.



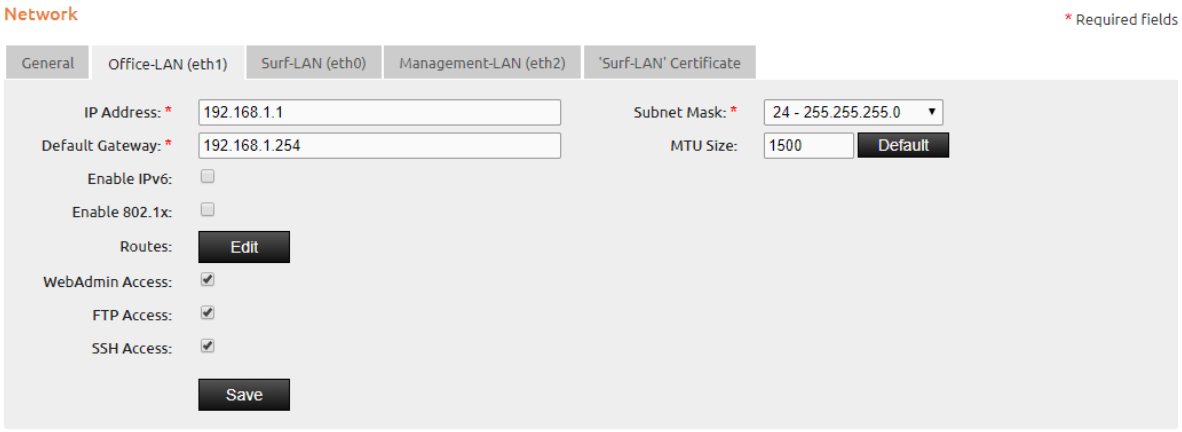
FTP Service

Next, switch to the menu **System / Services** and activate the **FTP service**. The **Start** button only activates the FTP service **temporary** (until next system reboot) whereas the **Activate** button activates the FTP service **permanently**.



Open FTP port

Now you need to open the **FTP port** in order to access the templates. Therefore switch to the menu **Settings / Network** and activate **FTP Access** for the **Office-LAN** interface. To make the changes take effect, save the settings and then reboot the system.



Activate Custom Design

To modify the *Client Logon Page* html templates, you have to **activate** the **Custom** design setting. Switch to menu **Client Logon / Design**. In the tab **Themes** change the **Logon Style** from **Classic Flat - Default** to **Classic Flat - Custom** at the bottom of this page. You can switch back to the default design at any time if something goes wrong.

Hint:

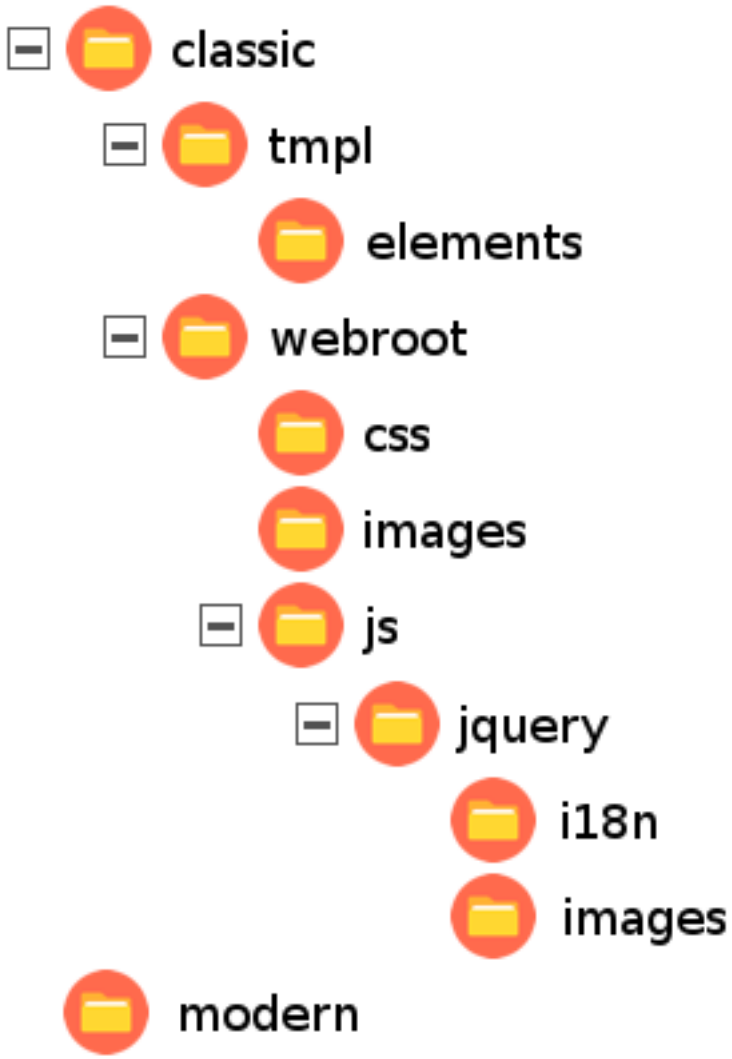
- The template files in the FTP directory are only visible after switching to **Custom** design.

FTP Server Connection

The FTP server supports FTP and SFTP (secure FTP) connections. Use a FTP client of your choice to connect to the server. As host use the IP address of the IACBOX and log in with the credentials of the previously activated FTP user and password.

Directory Structure

The directory **modern** is currently not in use. Switch to the directory **classic** to modify the template files.



Directory Information:

- **tmpl** Includes HTML template files
  - index.tmpl Template file for default logon page
  - mobile.tmpl Template file for mobile logon page
  - help.tmpl Template file for help
  - error-404.tmpl Template file for error 404 page
  - dgerror.tmpl Template file for HTTP content filter
- **tmpl/elements** Contains various HTML design elements like checkboxes, input fields, combo boxes etc.
- **webroot/css** Contains CSS files for HTML element styles
  - box.css CSS styles for box design (Ticket Logon, Status Information, etc.)
  - nobox.css CSS styles for box design of old browser and mobile logon page (e.g. IE6)
  - design.css CSS styles for default logon page
  - mobile.css CSS styles for mobile logon page
  - fix-ie6.css CSS styles for IE6 browser
  - fix-ie7.css CSS styles for IE7 browser
- **webroot/images** Contains images for customer logon page

- **webroot/js** Contains Java-Script files for customer logon page

10.2.3 Template Customization

The following part will explain how to edit the now accessible template files based on examples.

Hint:

- Changes of the *Client Logon Page* can not be handled by the IACBOX Support.
- All changes must be performed for the **desktop** and the **mobile** template.

Example: Text Replacement

Customization is tricky because almost all texts on the *Client Logon Page* do depend on many factors and are available in different languages. To replace a text with your own variant a **JavaScript** must be added at the end of the template you are editing. To replace the title text of the module PMS **Room Logon** with your own variant, for example **Hotel Logon**, open the file **index.tmpl**. Now scroll down to **the very bottom** of the file **index.tmpl** and add the following code **above the html and body closing tags**.

```
<html>
<body>
..
</body>
</html>
```

This customization relates to the CSS class **headline**. Basically this method can be adapted on all contents.

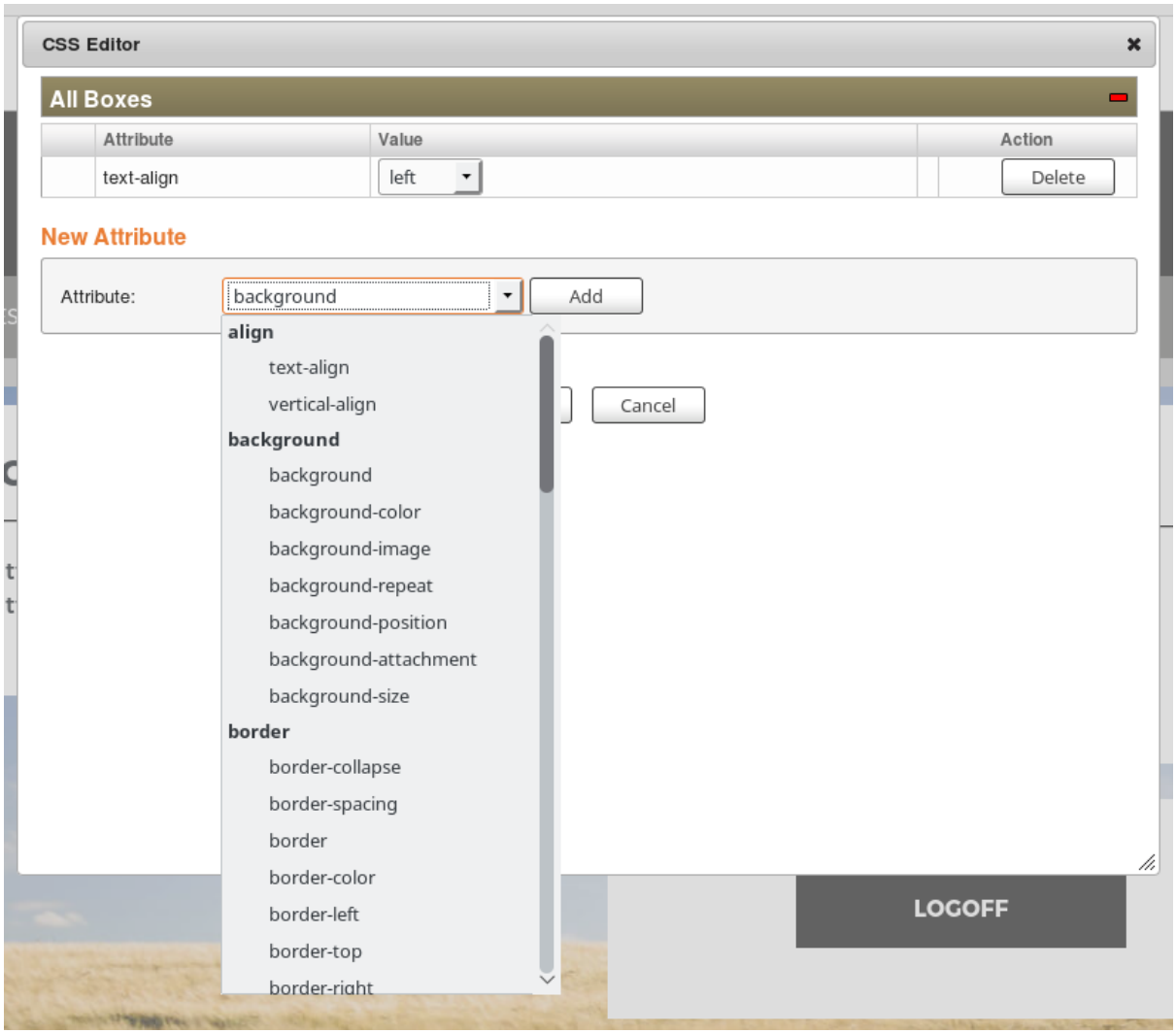
10.3 EasyWeb CSS Editor

The EasyWeb CSS Editor allows you to change the customer logon site by yourself and adapt it to your needs. You can change the color, position, font size etc. of all existing elements. In order to open the EasyWeb CSS Editor, click on the element you want to change.

10.3.1 General

The EasyWeb CSS can be accessed in the WebAdmin menu **Client Logon/Design-Themes**, where new themes can be created and edited with the CSS editor. Once a theme has been opened for editing, the various elements of the login page can be selected.

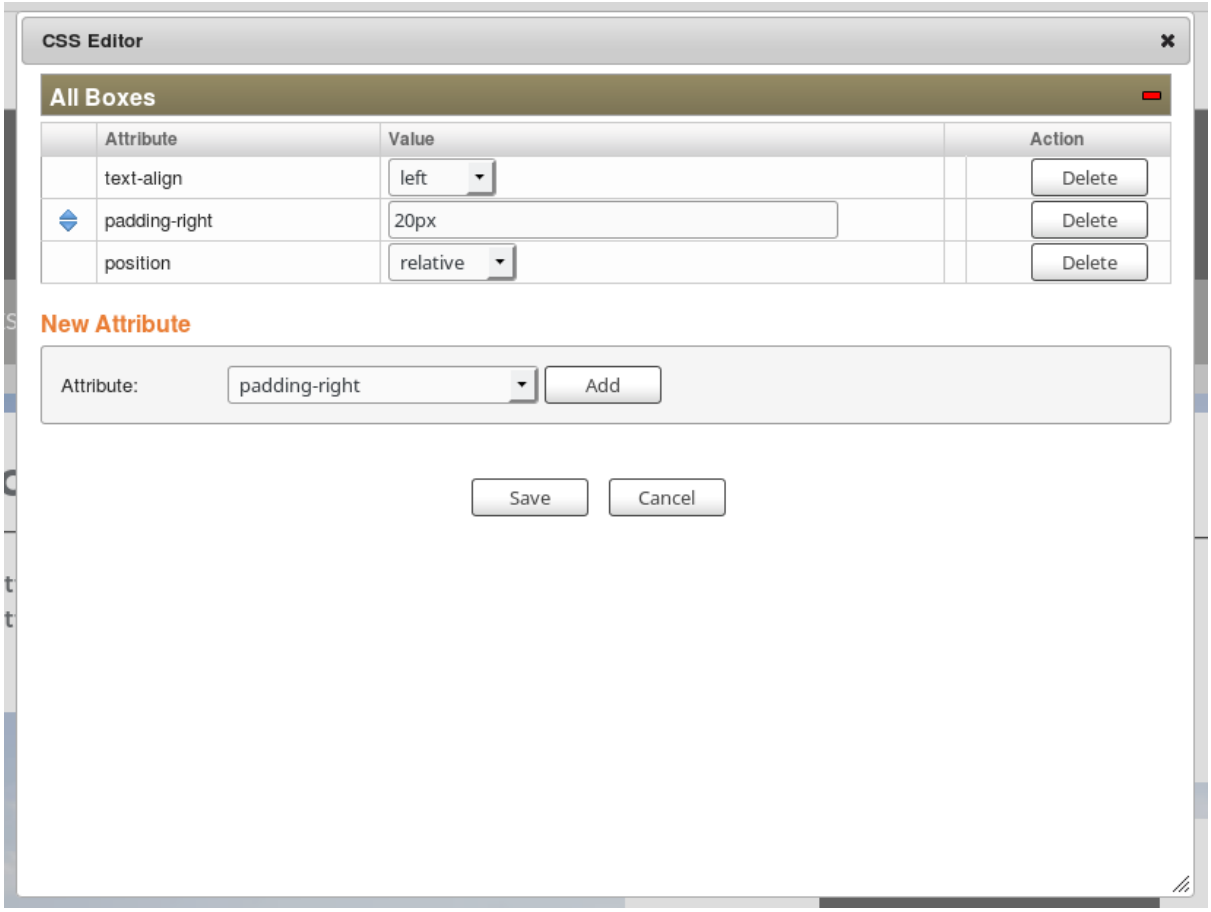
Once an element has been selected, a window shows up where you can edit the CSS attributes and add new ones. You can add new attributes as for example *background-color*, *font-size*, *position* etc.In addiction you can change the attributes of all elements of the same type.



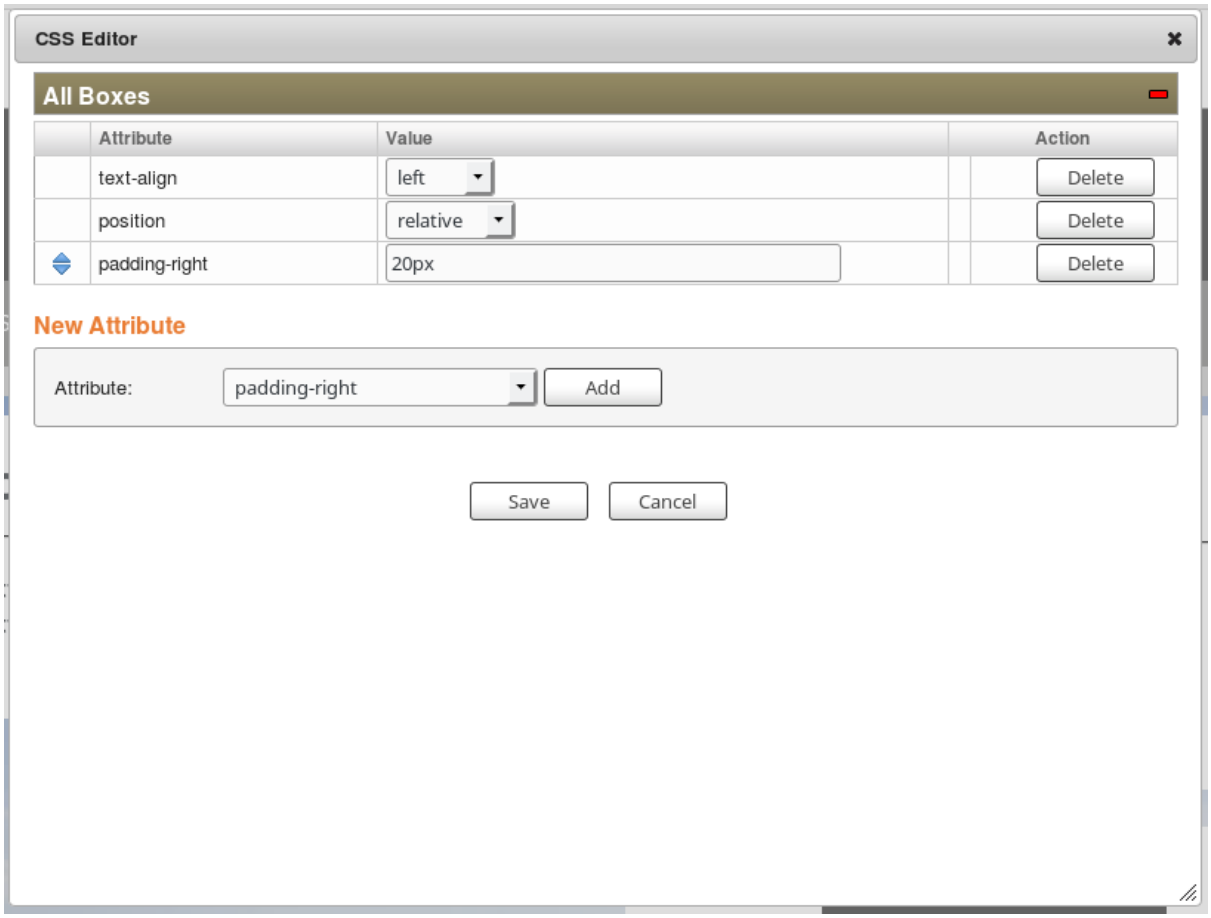
As you can see on the picture above, the EasyWeb CSS Editor contains two fields on which you can change attributes. One of them is the selected welcome box (displayed as **Left - Box 1**) and the other one is for all other similar elements (displayed as **All Boxes**).

So if you want to change certain attributes for all similar elements you do not need to change them on every single element, you ca change them for all similar elements at once. But please note that the attributes of individual elements are still preferred.

The attributes of an element can also be prioritized among themselves. Therefore click into the empty field beneath the attribute name and drag it up or down. Highest priority has the first attribute in the list and the lowest priority has the last one.



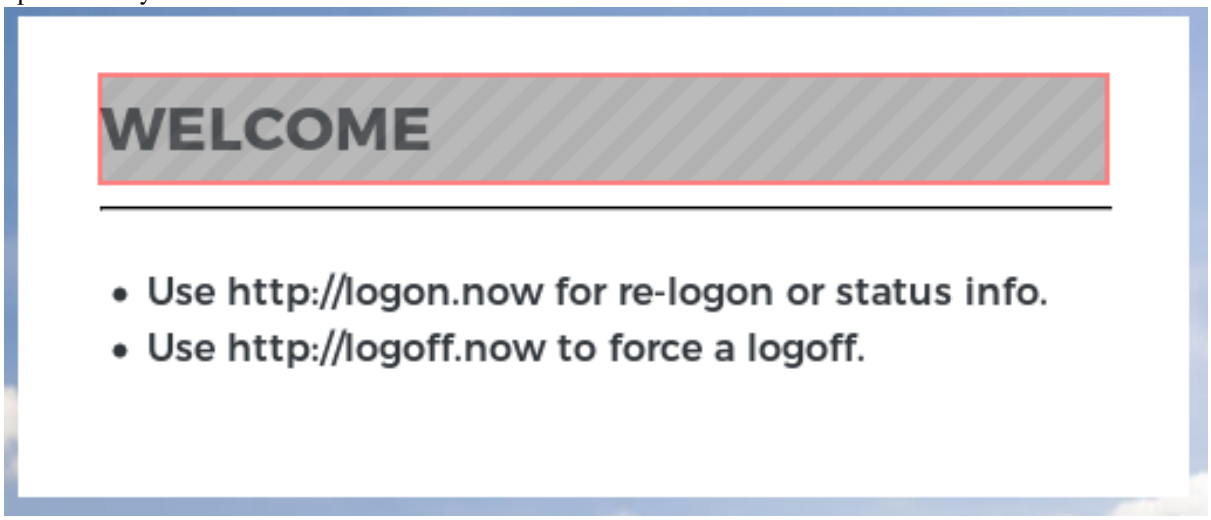
Here you see three attributes defined for the welcome box. Now click into the empty field beneath the attribute name and drag ther attribute to another position.



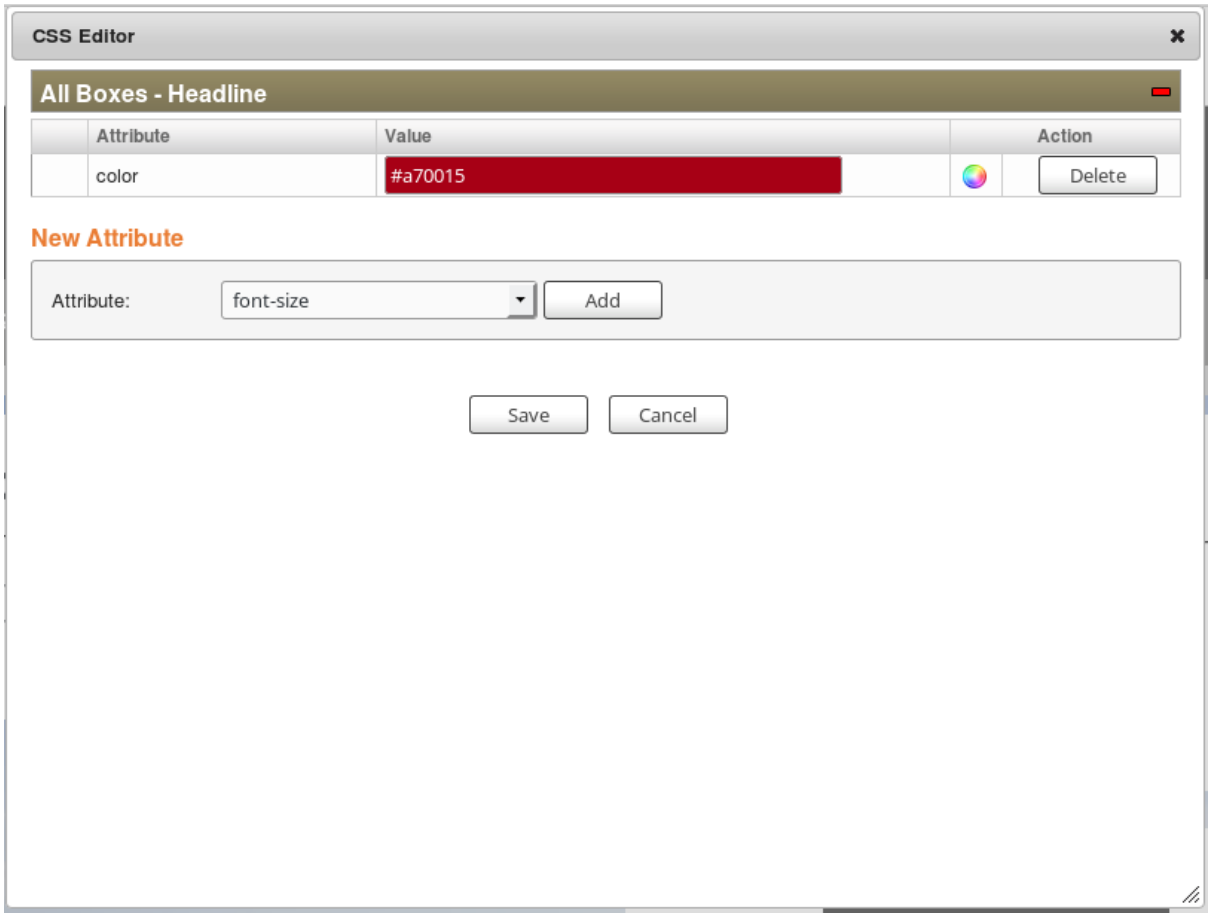
As you can see now, the priority of the attribute **padding-right** has been changed. This attribute is now in last place and therefore has the lowest priority.

10.3.2 Example 1 - Change font color of the header

In this example the headline of the welcome box should be changed. Therefore click on the headline of the box to open the Easyweb CSS Editor.

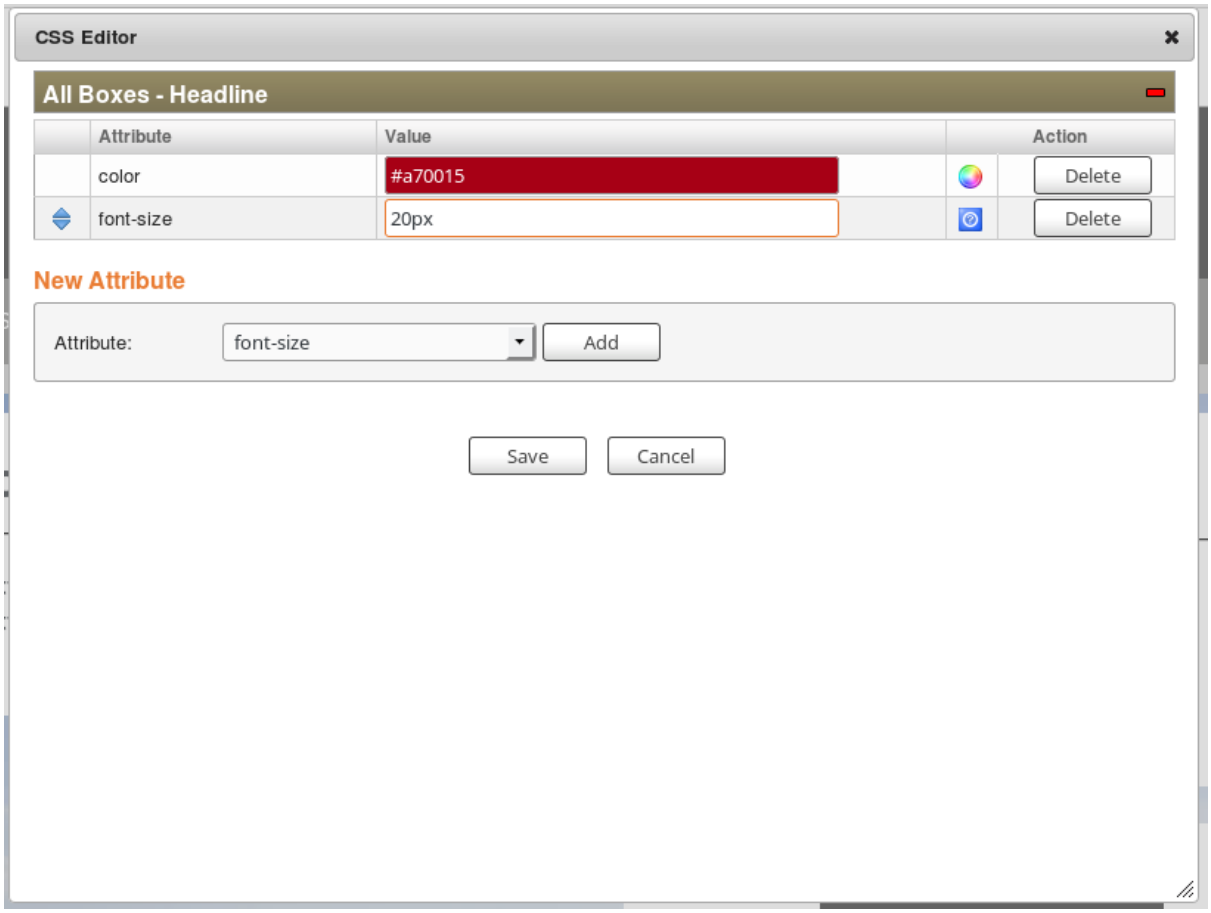


As you can see here, the EasyWeb CSS Editor contains all boxes and therefore all headlines will be changed.

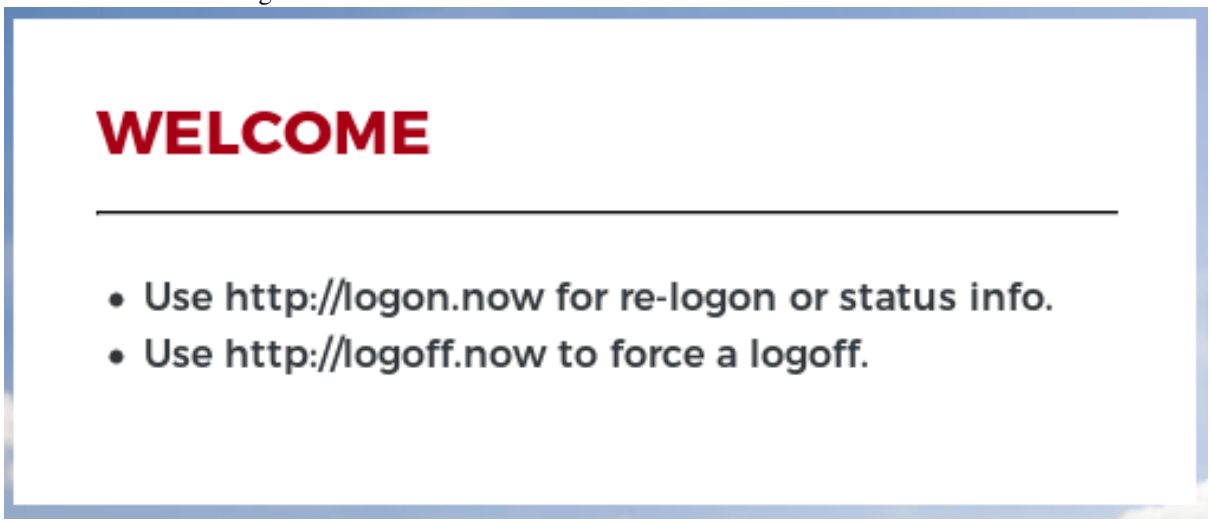


In this example we will change the font color and the font size of the headline. To change the font color, you can either enter the color code of the desired color or use the color circle beneath the **Delete** button.

To change the font size, also add the appropiate attribute (**font-size**). Then enter a font size in pixel (e.g. 20px).



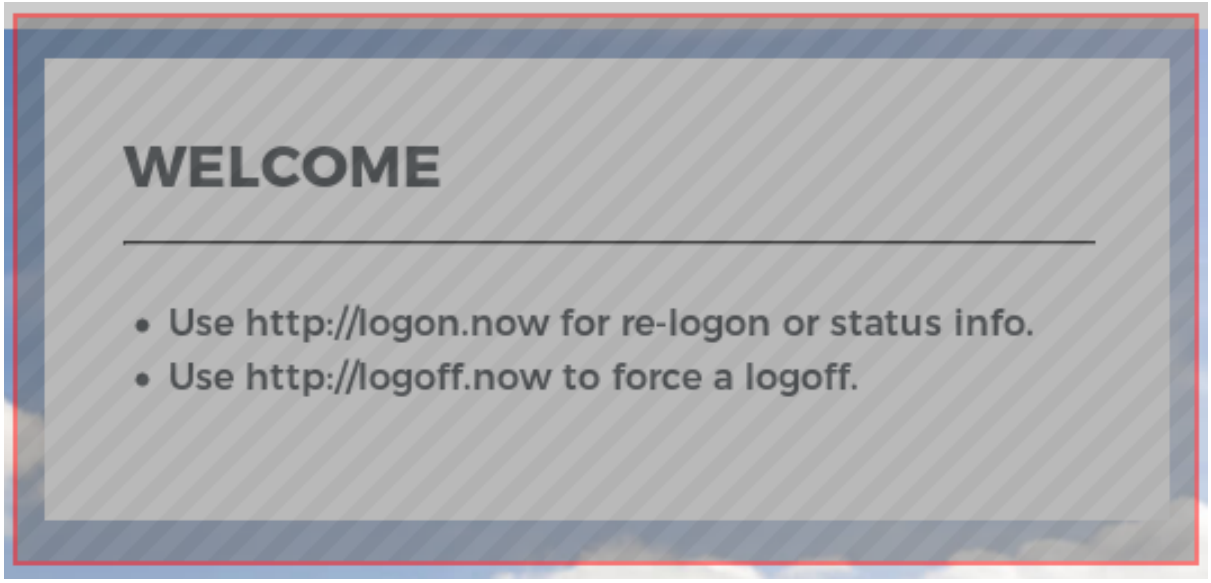
As you can see in this example red was selected as font color. The font size has been set to 20px. These settings will lead to the following result.



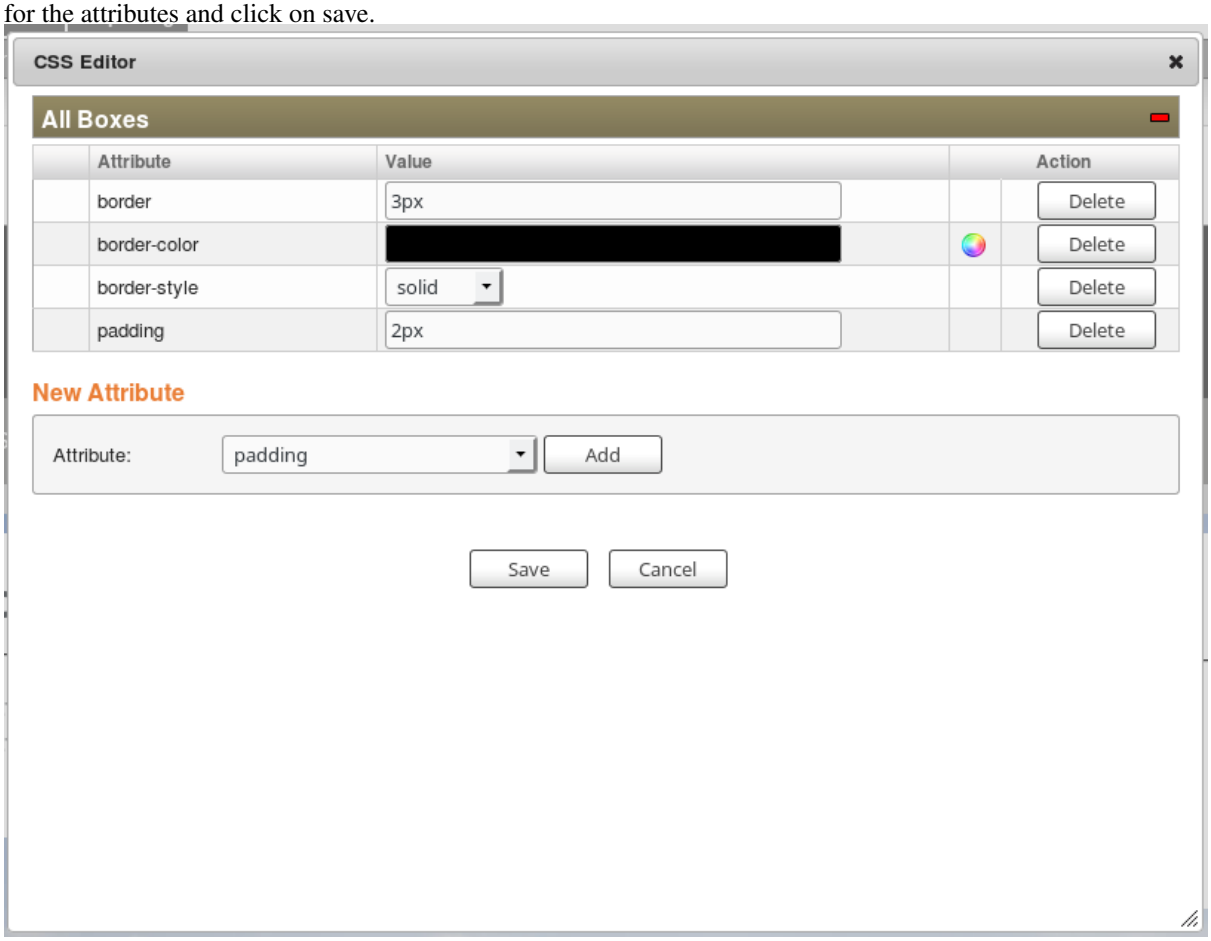
10.3.3 Example 2 - Change border of login boxes

In this example, a own border for all boxes will be defined. Therefore click on the whole box to open the EasyWeb CSS Editor again.

Again, only attributes in the field **All Boxes** are changed.

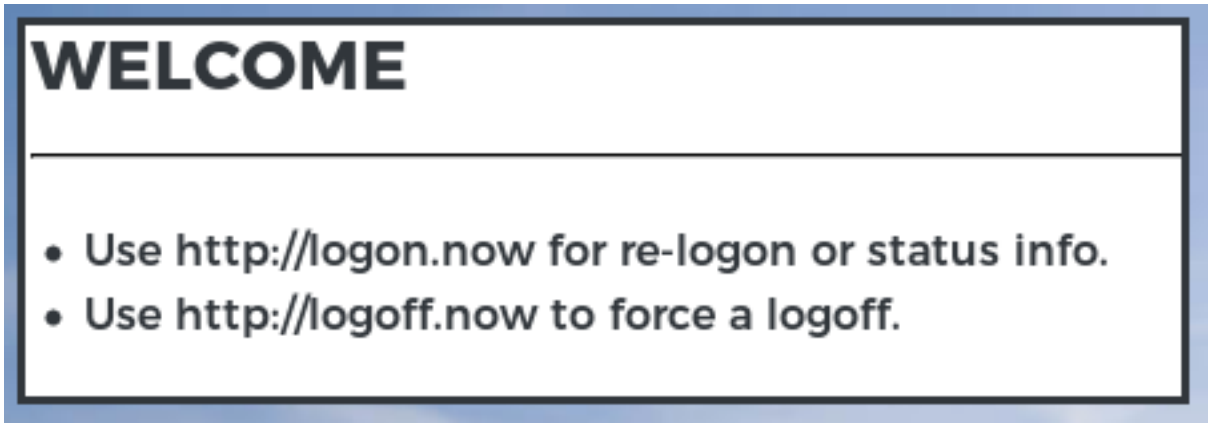


In this example, a solid, black border with 3px border-width and 2 px distance from the box is defined. Therefore add the required attributes (**border-color**, **border-width**, **border-style** and **padding**). Etner the required values for the attributes and click on save.



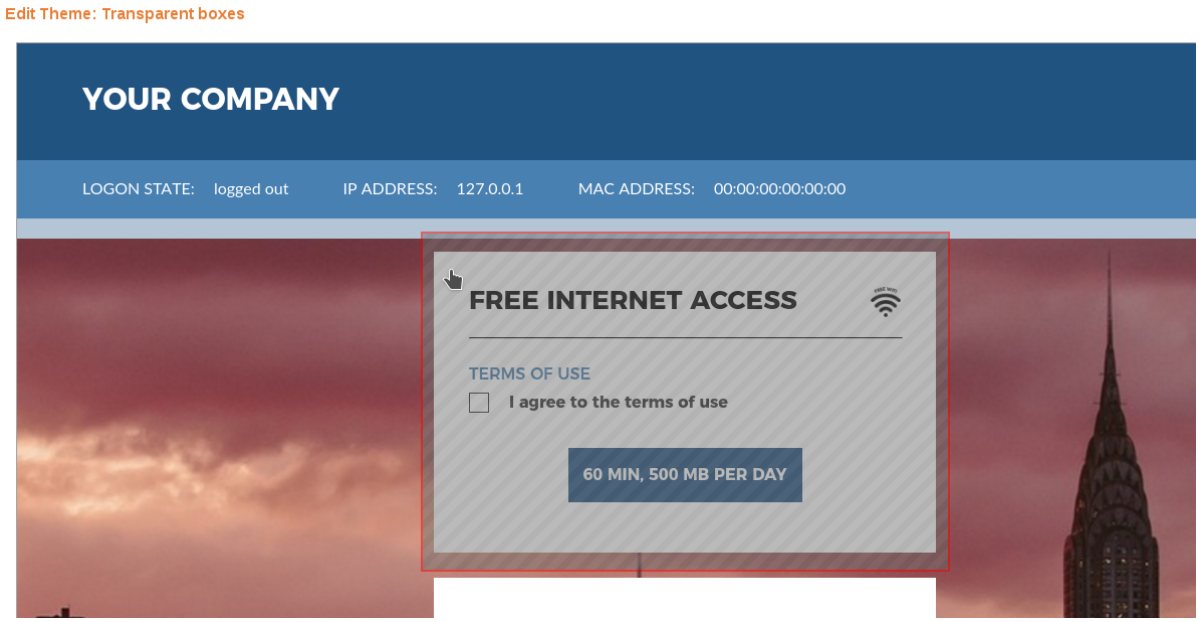
The attributes for this example are displayed above. The settings will lead to the following result for all boxes.

As you can see, a solid black border with 2px width and px distance from the box has been created.

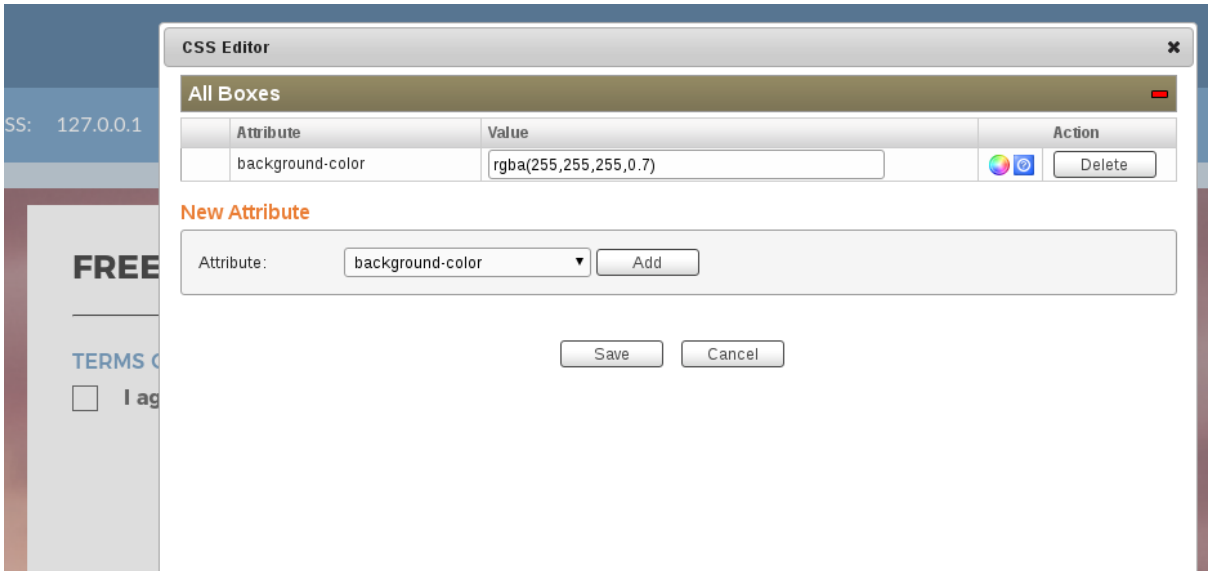


10.3.4 Example 3 - Make boxes transparent

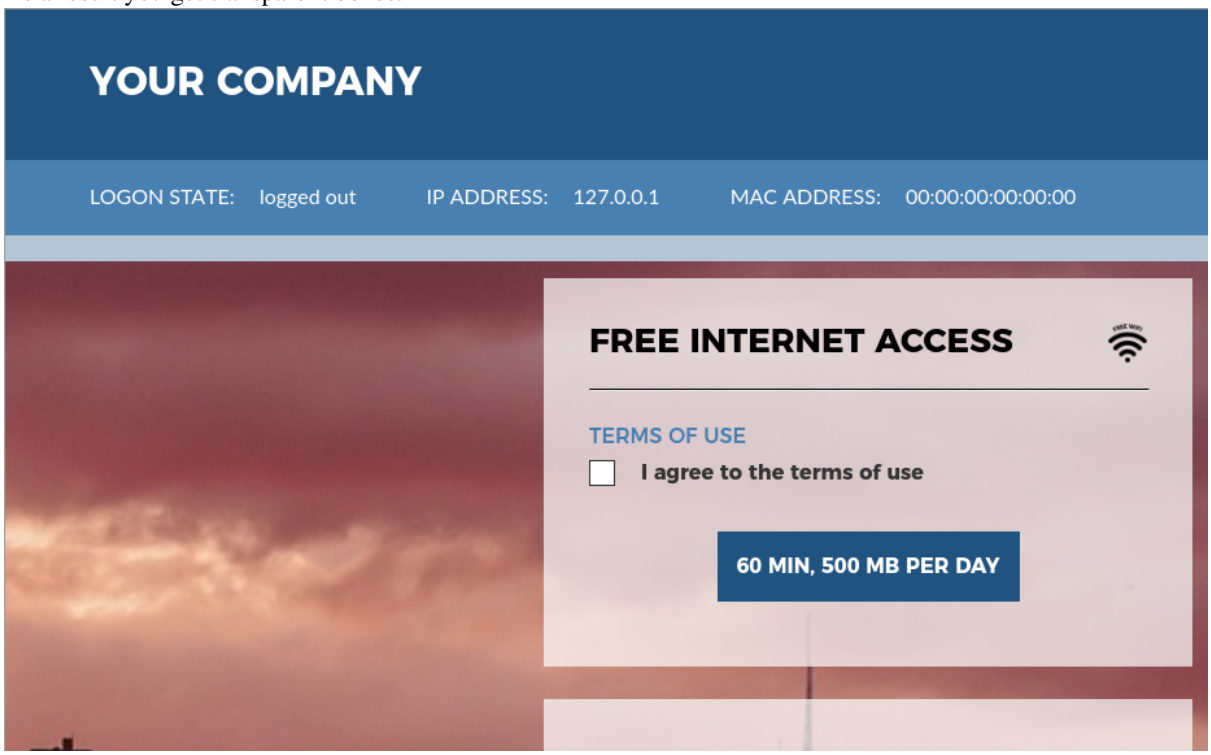
First select a box



Add the CSS property **background-color** and set as value for example `rgba(255,255,255,0.7)`. The first 3 values are the color in the RGB format - in this case *white* - and the last value is the opacity where *1.0* means opaque and *0.0* menas 100% transparent. It depends on the used background image which value to choose. Note that darker and homogenic backgrounds work better with transparency.



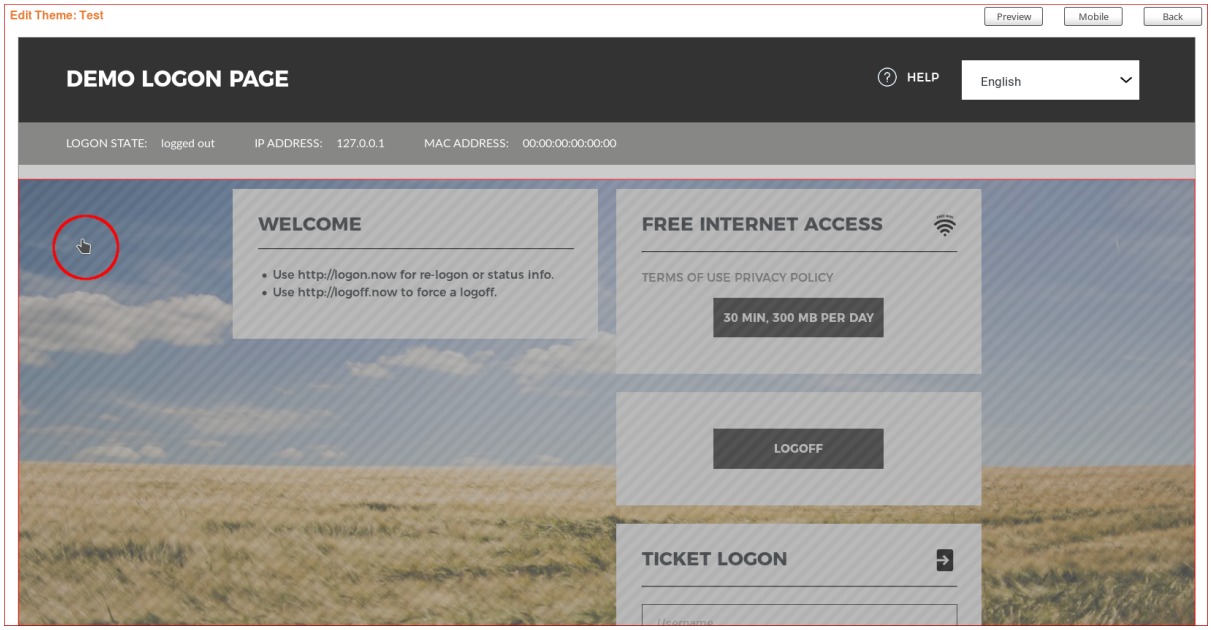
As a result you get transparent boxes.



10.3.5 Example 4 - Change the background image

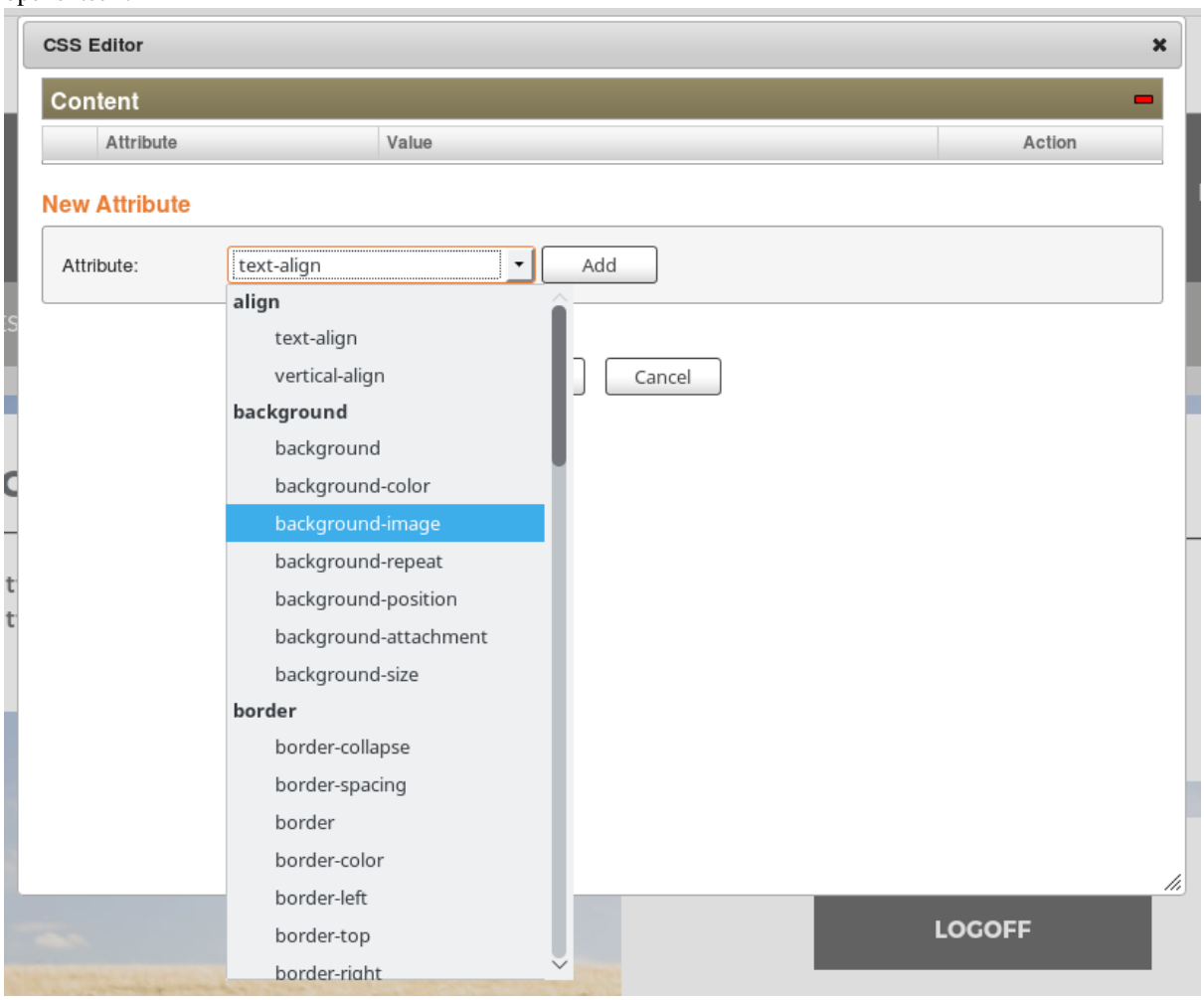
Attention: This is for older systems up to version 8.0 or for old landing page styles. If you use verion 17.0 or higher with a new *Metro* style you can use the new background chooser (page 251).

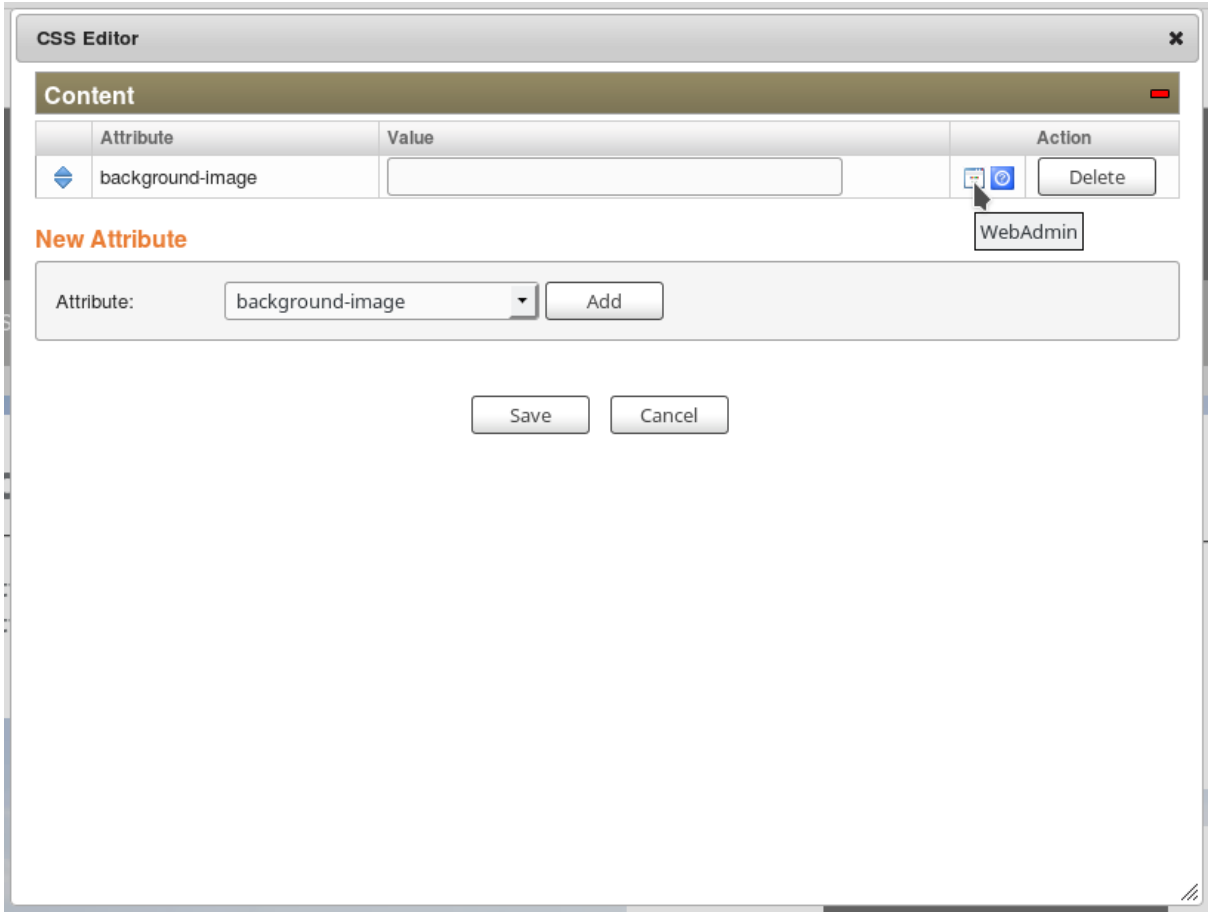
In this example we will change the background-image of the default logon page. Click on the background image of the page to open the EasyWeb CSS Editor.



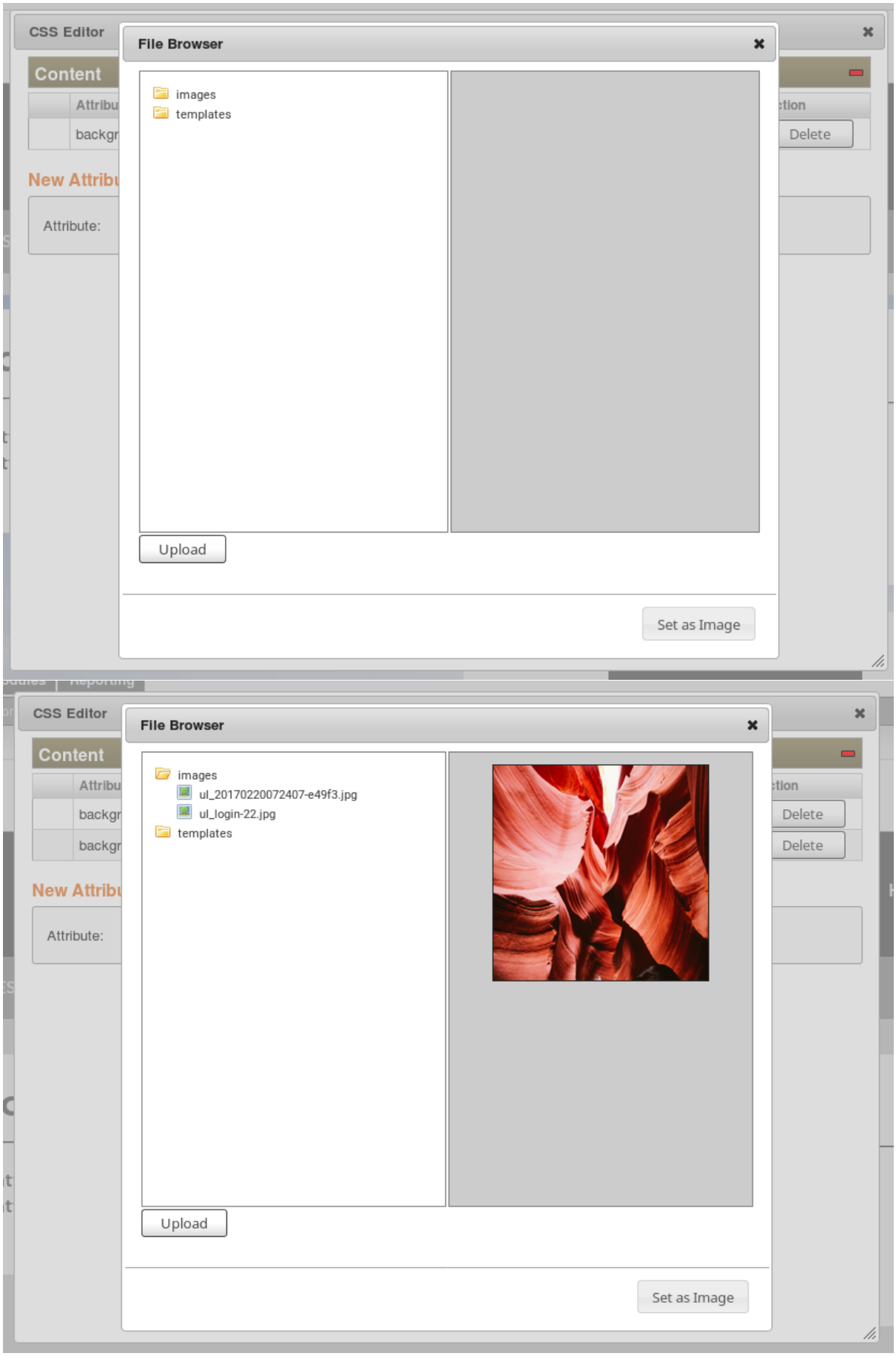
Only attributes in the field **Content** will be changed.

Add a new attribute with the name **background-image**. Now click on the Filesystem icon and a File browser opens itself.

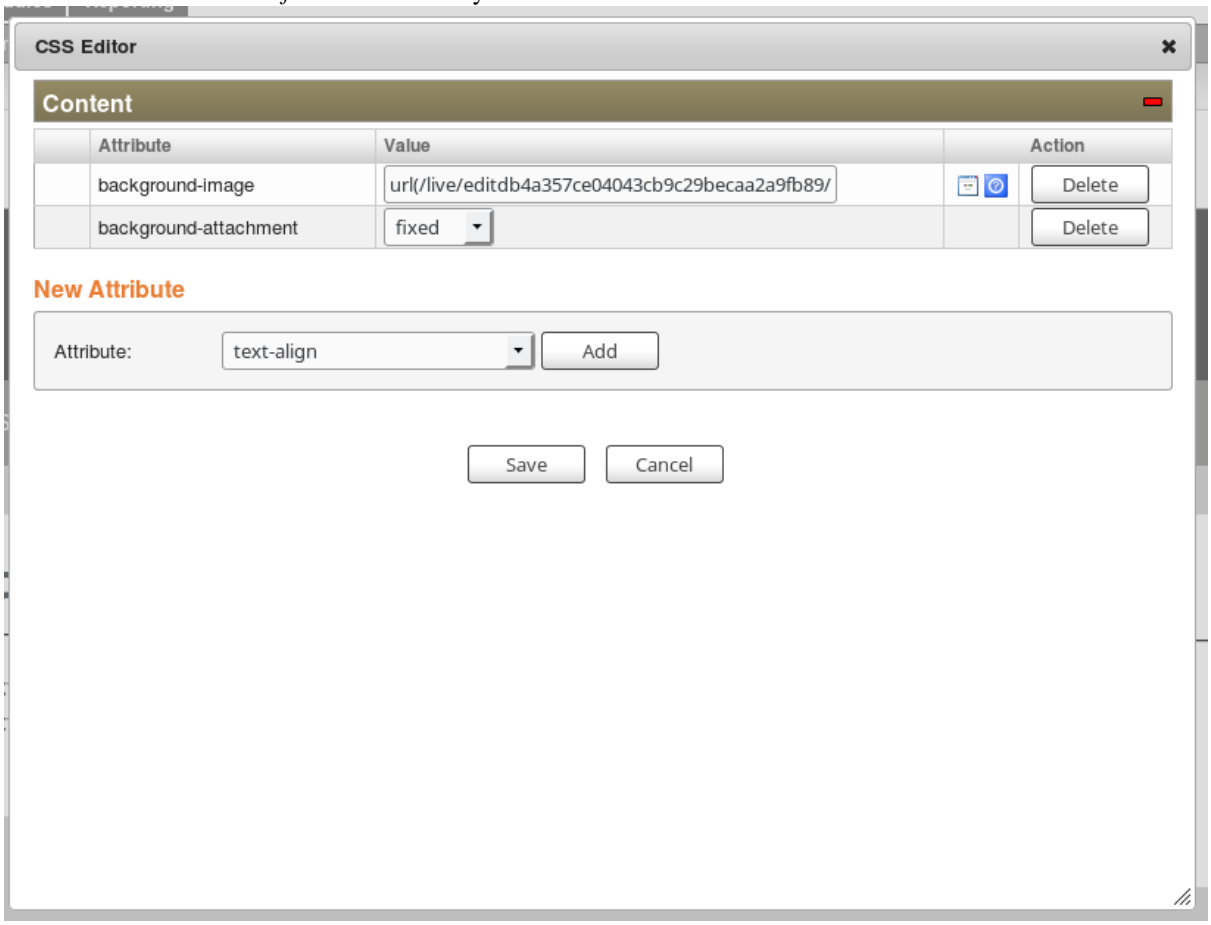


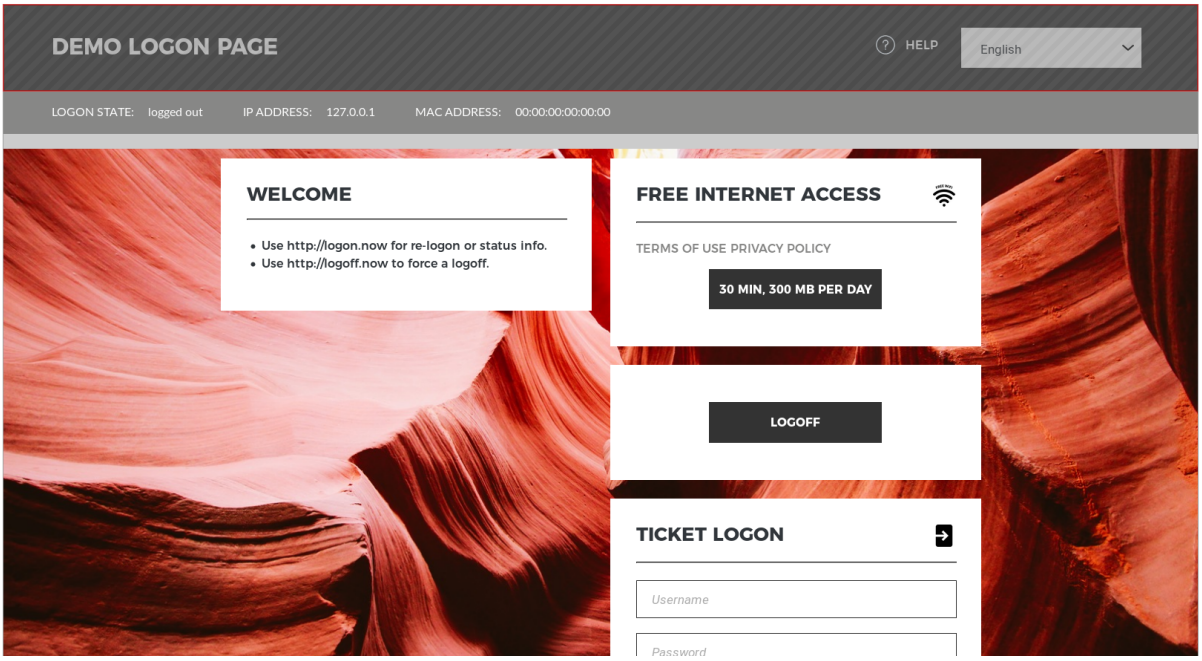


Click the button **Upload** and select the background image you want to use for your login page.


CSS Editor
File Browser
images
templates
Upload
Set as Image
Content
New Attribu
Attribute:
Delete
CSS Editor
File Browser
images
ul_20170220072407-e49f3.jpg
ul_login-22.jpg
templates
Upload
Set as Image
Content
New Attribu
Attribute:
Delete
Delete

After you clicked the button **Set as Image** the new background image will be set. The css property *background-attachement* will be set to *fixed* automatically.





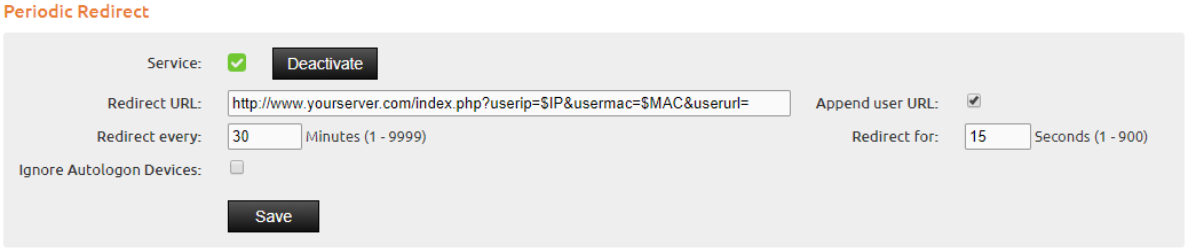
10.4 Periodic Redirect

Hint:

- This feature allows you to periodically redirect client devices on to a customizable URL.
- Usually this feature is used to display product placements or important information.
- Redirects are only possible if a client device does send HTTP requests. HTTPS requests will **not work**.

10.4.1 Configuration

In order to configure the **Periodic Redirect**, activate it in the *WebAdmin* menu **Client Logon / Redirect**. Here you will find the following available settings:



The settings from the screenshot will redirect any client device every **30 minutes** for **15 seconds** of time on to the configured URL. Also the option **Append user URL** was activated. This will add the user URL which is being redirected to the configured page as a **GET parameter** to the redirect itself and allows you, to forward to this page later on. If the URL you want to redirect to does not belong to you, then no GET parameters are required.

Available GET parameters:

- **$IP** - the IP address of the device which will be redirected
- **$MAC** - the MAC address of the device which will be redirected
- the *user URL* (if enabled) - will be directly added to the end of the configured URL. If this is activated, a proper **GET** parameter must be configured at the end of the URL. In the screenshot from above, this was done by using the GET parameter **userurl**.

CHAPTER

ELEVEN

TICKET PRINTER

11.1 Epson TM-T20

This manual describes how to configure and connect to the **ticket printer** series **TM-T20**. The same configuration will also apply to the models **Epson TM88III, TM88IV** and **TM88V**.

Hint:

- A ticket printer can only connect to one IACBOX at a time.
- The **Lite Version** of the IACBOX supports **2 Ticket Printers** while the full version supports **100 Ticket Printers**.

11.1.1 Configuration

Epson ticket printers usually get shipped with the internal IP address **192.168.192.168**. Configure a device (e.g. notebook) to be in the same address range and connect the ticket printer to your notebook by using an ethernet cable. Dont forget to power on the ticket printer. You now can open the ticket printer address in your web browser and access it's basic configuration over it. This menu will also let you set additional settings like a password protection.

The network configuration of the ticket printer can always be accessed by printout it's basic configuration. To do so, power off the ticketprinter. Now hold the **Feed Button** and while doing so, power on the ticket printer. After about 4 seconds, the device will print out it's current configuration.

After determining or adjusting the network settings, you can add the ticket printer in the IACBOX *WebAdmin* menu **Modules / Ticket Printer**.

Hint:

- After changing the *Ticket Printer* configuration, a **Service Restart** is required.

11.1.2 Printer Paper

The *Epson TM-T20* ticket printer supports two different paper sizes. Thermal paper with 80mm width and thermal paper with 57.5mm / 58mm width. To use the paper with a width of 57.5mm / 58mm, a so-called "spacer" is included with your printer. The "spacer" is simply inserted into the paper tray to reduce the 80mm range to a 57.5mm / 58mm range. Then the printer must be configured for the used paper width. By default, the paper width is configured for 80mm paper.

11.1.3 Configuration of the Paper Width

The configuration is done directly on the ticket printer by using the **Feed Button**. The current menu navigation will be printed in realtime.

- Hold the **Feed Button** and turn on the printer. The current configuration of the printer will then be printed.
- Hold down the **Feed Button** for **1 second** or longer. Enter the **mode selection** menu.
- Press the **Feed Button 3 times**, then hold down the **Feed Button** for **1 second** or longer. This will enter the configuration menu.
- Press the **Feed Button 6 times**, then hold down the **Feed Button** for **1 second** or longer. This will enter the *Paper Width* menu. Here you can select the paper width which you want to use for further print-outs.

After the paper width has been configured directly on the ticket printer device, it must also be configured on the IACBOX. Therefore navigate to the *WebAdmin menu* **Modules / Ticket Printer** and configure the desired paper width.

Hint:

- After changing the *Ticket Printer* configuration, a **Service Restart** is required.

CHAPTER

TWELVE

TROUBLESHOOTING

The **Troubleshooting** section will provide quick help for hardware and software-related issues which can appear with certain configurations.

12.1 The Installation does not start

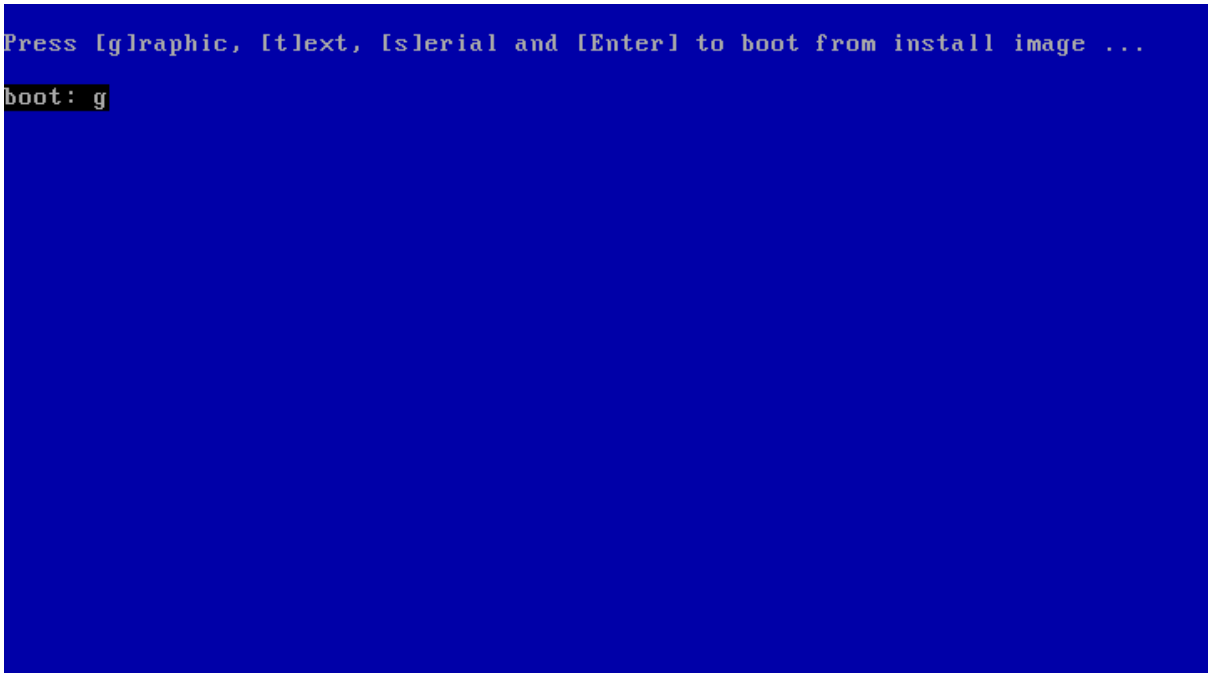
12.1.1 Problem

The USB-stick (or CD) with the IACBOX installation is plugged in, yet the installation does not automatically start when the hardware is being booted.

12.1.2 Solution

- If the hardware does not boot from your installation medium (CD or USB-stick), restart the hardware and use the **F10 hotkey** to open the *BIOS Boot-Selection* while the hardware is booting.
- If the *BIOS Boot-Selection* does not list your installation medium, then the system does not recognize it. From here on either the medium is defective or it was not created properly. First off you may try to re-create the installation medium by following the USB Stick Creation manual step by step. If this does not lead to sucess, then retry these steps with a different USB-stick.
- Some new EFI-based hardware systems do not have a *BIOS Boot-Selection*. If this is the case, the IACBOX installation medium must be configured as **First Boot Device** in the BIOS boot order settings.

When successfully booting from the installation medium, the following mode-selection screen with appear:



On some EFI-Systems, you may directly see the IACBOX installation screen:

For further installation notes refer to the IACBOX Installation manual.

12.2 Migration to Version 17.2

12.2.1 Description

In order to upgrade from IACBOX version 17.0 to 17.2, the **Online Update** will perform a complete system migration. This means that the system will be replaced with a renewed base system - and all settings and tickets will be migrated back to restore the previous state.

Attention:

- **Create a manual backup before starting the Online Update.**
  - **This process can take up to one hour.**
  - While migrating, the IACBOX will restart **multiple times**. The migration is done if you can reach the **WebAdmin** and its version equals **17.2**.
  - The online update may not be interrupted under any circumstances - otherwise the system will likely become inconsistent and must be reinstalled manually.
  - If the automatic migration fails, the old version 17.0 will be restored automatically.
  - After a successfull migration your sytem version will be **17.2** but since the **Online Update** was triggered, further available **patchlevel versions** may be installed automatically after a short delay.
  - 32-bit Hardware is no longer supported and wont receive this update.

12.3 The IACBOX is not starting

12.3.1 Problem

The IACBOX system is not starting anymore.

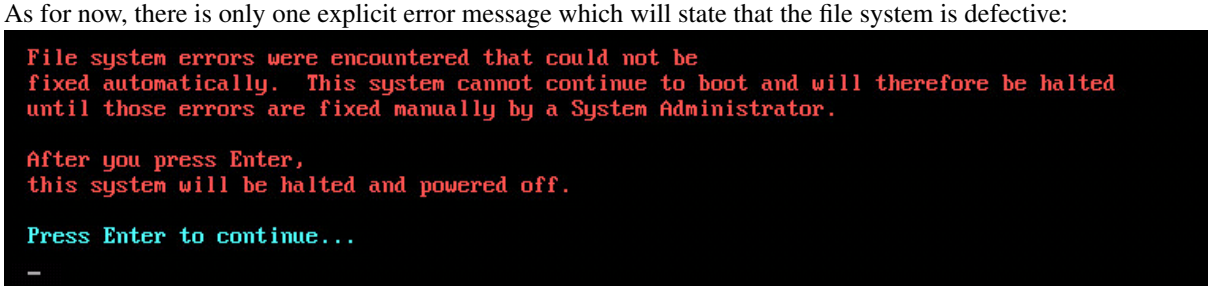
12.3.2 Solution

Hint:

- If the system hangs right after the IACBOX was freshly installed, then it is most likely that the SATA controller setting in the BIOS was not set to **AHCI**. This is being mentioned in the *Basic BIOS settings* section of the IACBOX Installation manual. If this is the case, then the IACBOX must be installed again after the BIOS settings were changed according to the manual.

First of it must be determinated if the IACBOX does randomly **hang** while booting, or if it does report an explicit **error message**. That for, a monitor and keyboard must be plugged in and the system must be restarted. If the system randomly hangs, then this indicates a hardware problem with the most likely reasons of an overheating CPU, a defective memory, or a defective hard drive.

As for now, there is only one explicit error message which will state that the file system is defective:



If this is the case, then the file system was destroyed. This can happen whenever the system does get **shut down unexpectedly** (e.g. power failures). In some cases, this can also indicate a defective hard drive, which definitely

should be checked by an administrator on-site. As for the file system, there is a chance that it can be repaired. The according steps are documented in the Rescue Boot manual.

12.4 Online Update does not work

12.4.1 Problem

The IACBOX **Online Update** does not work.

12.4.2 Solution

First off it must be clarified that the IACBOX does not directly report connection-related errors since this should work at any time with a valid system configuration. Further information to connection-related errors can be found in the **update logs** on the online update WebAdmin page.

Generally there are 3 different types of errors which can appear:

- Rsync error
- Certificate mismatch (general connection error)
- Kernel version too old

Attention:

- For *every update* the IACBOX will create an **update log** which can be accessed at the bottom of the online update page. Some information may only be available in the according log file, so these should always be checked first in case the online update fails.

Rsync Errors

Rsync errors indicate that the IACBOX could connect to the online update servers, but the datastream was interrupted at some point. This usually happens with some kinds of firewall solutions which do intercept SSL traffic. Review all firewall settings and disable features with possibly intercept SSL traffic.

General Connection Errors

These kinds of errors indicate general problems like:

- No internet connection
- The DNS servers dont resolve correctly
- The current system time is invalid

The first 2 points can easily be tested by navigating to the WebAdmin menu **System / Tools** and by performing a **Ping** on to **update.frozentox.org**. This must always return a valid ping reply. If not, then the problem is related to either no internet connection or a faulty DNS server configuration. Also since the IACBOX uses encrypted traffic for updates, it is important that the system has a **valid date and time configuration**. This is required for the initial handshake for certificate-based encryption. If the system time is in the past or future, certificate checks can quickly fail and terminate the connection. As a solution, restore the default NTP server in **Settings / General** to **ntp.frozentux.org** and **restart** the IACBOX before re-trying the update process.

Kernel version too old

Following exemplary lines will be visible in the update log:

- *Upgrade not possible for this kernel version 2.6.35*

- *Minimum required kernel version is 3.2!*
- *Hint: to be able to upgrade to newer version (1) backup your system and (2) reinstall your system using CD/USB installation media version 6.0.8222 and (3) restore afterwards!*

The kernel will only be installed once upon the initial installation of the IACBOX. An automated update to newer versions is not possible since hardware drivers sometimes will be removed, which would break compatibility to alot of older systems. Therefore a manual **System Migration** must be performed. Detailed steps for this are being explained in the according Migration Manual.