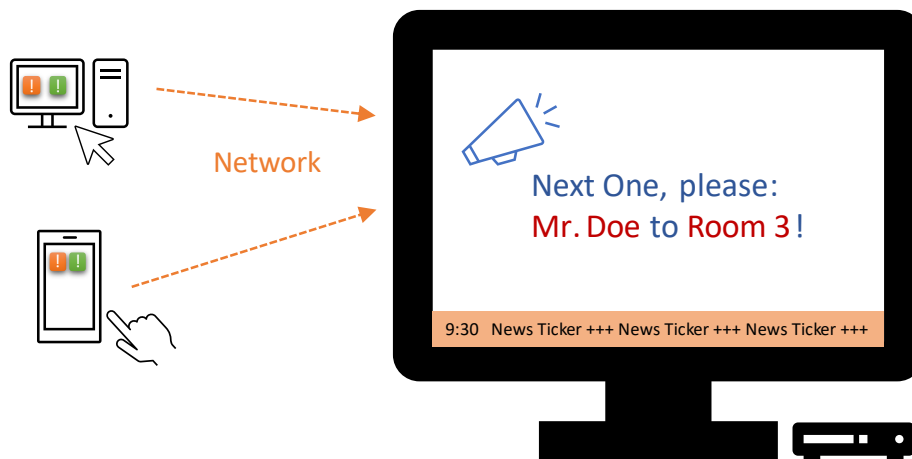


# Remote Control

The digital signage & kiosk software **FrontFace** can be extended by plugins that add additional functionality. This way you can e.g., display special content types, integrate external data sources or systems and add entirely new features to the software.

## Description:

*FrontFace* allows you to play content (playlists) using the built-in scheduling (timers). But if playlists are supposed to be started ad hoc by a user or by an external system, **Remote Control**, a Service Plugin, can be used for exactly this purpose.



The *Remote Control Plugin* provides a web service interface (**REST and SOAP API**) on the player, which can be used to trigger actions on the player. In addition to starting and stopping playlists, also other functions such as changing the system volume level or restarting the PC are available.

The *Remote Control Web Service* can be accessed using a normal browser via HTTP (REST) or from external systems, e.g., building automation systems or self-programmed applications (REST or SOAP).

In addition to that, there is also a separate app for **Windows, Android**, and **iOS** available, the "**Remote Control App**". This one provides a convenient user interface for controlling the *Remote Control Web Service*. Apart from the app there is also an exemplary HTML5 application included, which can be used to control the player through the browser of a smartphone or tablet.

Finally, there is a step-by-step quick start tutorial available for this plugin that shows how to set up a call-up system using FrontFace and the **Remote Control** plugin:

<https://www.mirabyte.com/en/frontface/download.html#tutorials>

## Installation:

Please refer to the user manual of *FrontFace* on detailed information on how to install a plugin.

**Notice:** If you have previously downloaded and installed the trial version of the plugin, you must install the full version over the existing trial version in order to get the full version running. The procedure is exactly the same as for installing a new plugin. If you already used the trial version of the plugin in your project, all settings will remain after installing the full version!

## Installation of the Remote Control App:

- **For Windows 10/11:** Run **FrontFace Remote Control App.msix** from "Other files" in the settings dialog of the plugin in the FrontFace Assistant to install the Windows version of the Remote Control App.
- **For Android:** Either install [FrontFace Remote Control](#) from Google Play or sideload the APK from the "Other files" in the settings dialog of the plugin in the FrontFace Assistant.

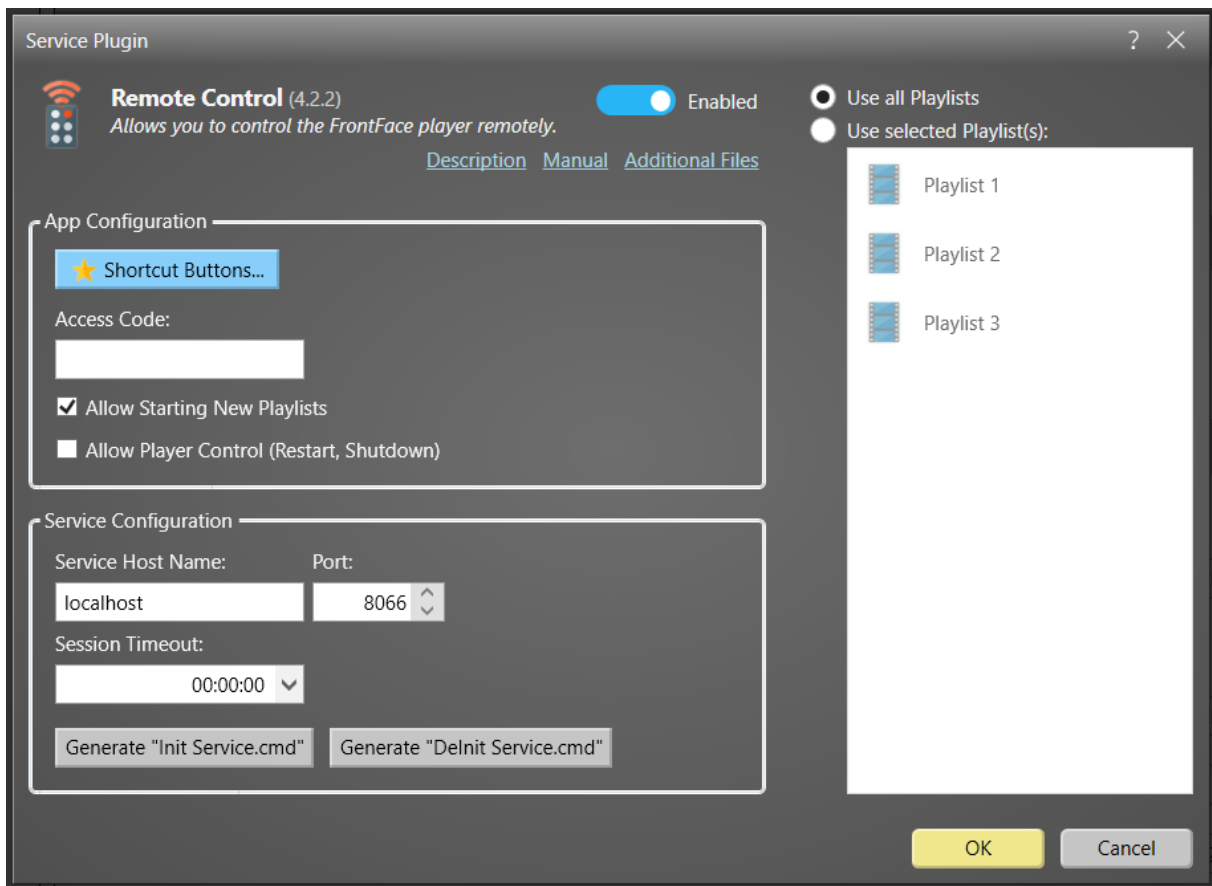


- **For iOS (or MacOS 12.3+ with Apple M processor):** Install [FrontFace Remote Control](#) from the Apple Play Store.



## Setup and Usage:

After you have installed the plugin in your project, you must add it to the desired player in the "Services" section under "Player" in the FrontFace Assistant (the "Show advanced player settings..." option must be activated for this!). After that you can select which playlist(s) can be started by Remote Control in the in the settings dialog of the plugin. You can either select individual playlists or touch menus or simply all playlists and touch menus from your FrontFace project:



### App Configuration:

You can use the **"Shortcut Buttons..."** button to optionally configure quick access buttons in the app. These buttons are displayed in the app and can be used, for example, to start playlists (with and without querying placeholders) or for other functions.

Optionally, you can store an access code ("password") as protection for accessing the remote control, define a period of time for the expiry of a valid session and specify whether playlists can be started and the player PC controlled via the Remote Control App (shut down, restart, system volume).

### Service Configuration:

You can specify the host name of the player PC (usually "localhost", otherwise player name or IP address) and the service port (8066 by default). In most cases, these settings do not need to be touched.

With the **Generate "Init Service.cmd"** and **Generate DeInit Service.cmd** buttons, you then generate two batch files that you then must execute on the player PC (!) with administrator rights in order to install or to install the service uninstall (only necessary if you no longer want to use the Remote Control Plugin on the player!).

The initialization of the service is only required once so that the *FrontFace Player App* can receive commands via the remote control web service. If you are using a firewall (in addition to or as an alternative to the Windows firewall), you must also open the port mentioned above manually.

Once you have initialized the remote control web service on the player PC with the batch file, you can publish your project as usual on the player using the FrontFace Assistant. After publishing, the service is ready to be used.

The service end points are:

REST: <http://name-oder-ip-des-players:8066/FrontFace/rest/>

SOAP: <http://name-oder-ip-des-players:8066/FrontFace/>

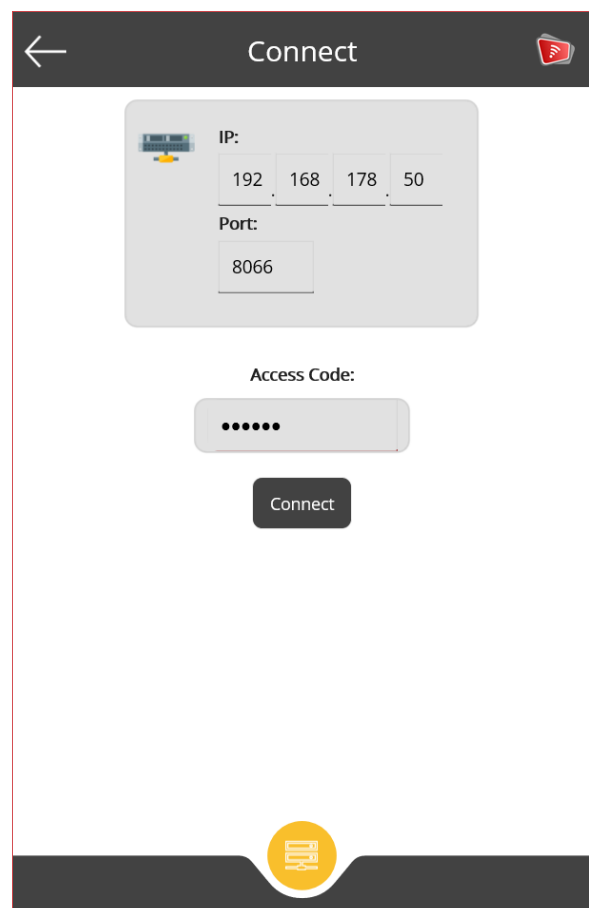
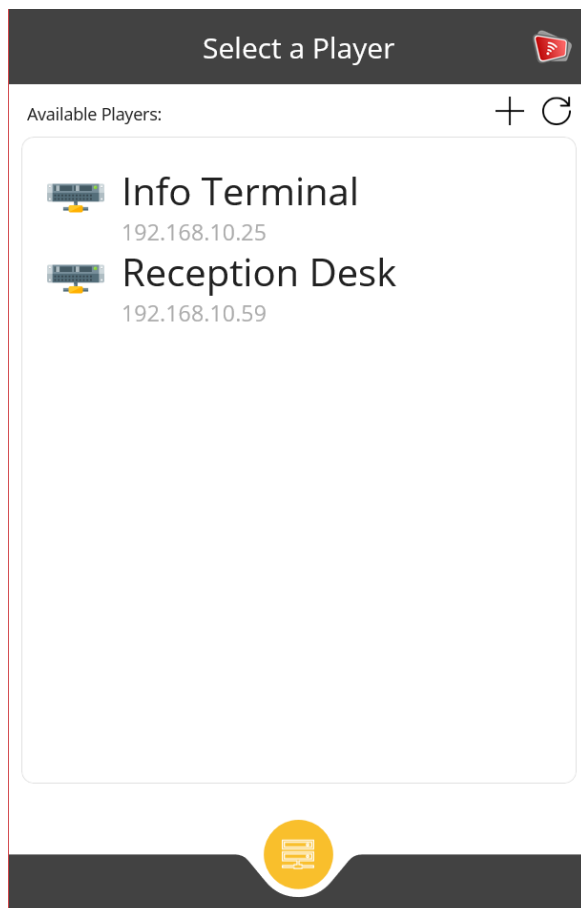
You can also enter these URLs in a normal browser to access the help pages with service description and the methods that are available. A quick overview of the available methods can be found in the [addendum of this manual](#).

### Using the Remote Control App:

In addition to using the Remote Control web service via the REST/SOAP API, you can also use the mentioned apps (for Windows, Android or iOS).

To do this, install the supplied Windows app (**FrontFace Remote Control App.msix** from "Other files" in the settings dialog of the plugin in the FrontFace Assistant) or the mobile apps from the respective app stores ("FrontFace Remote Control App") on your smart phone or tablet.

Please make sure that the device on which you want to use the app is in the same local network as your player PC(s) to be controlled. After starting the app for the first time, you may be asked for permission to access the local network. This must be allowed to use the app.

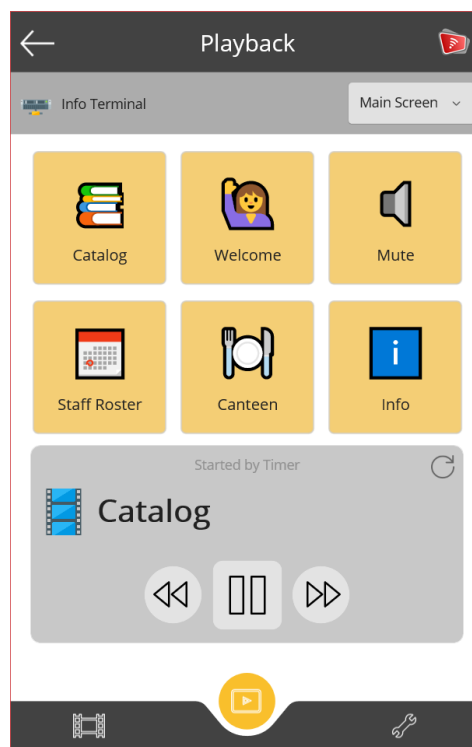


To control a player PC, you must first connect to it. The app usually automatically lists all player PCs that have the Remote Control Plugin installed and that are on the same local network. You can use the refresh button to start the search for player PCs again at any time. If a player PC is not found but can definitely be reached from the app, you can also use the “+”-button to establish a connection manually. To do this, you must enter the IP address and the port used on the following page.

If an access code was stored in the settings dialog of the Remote Control Plugin in the FrontFace Assistant, this must be entered here for a connection as well.

After successfully establishing the connection, the app will display the playlist currently playing on the player PC. This can be controlled using the corresponding control buttons (e.g. pausing or paging forward/backward). If multiple screens are connected to the player PC, the screen selection in the upper right area of the app can be used to select which screen is currently to be controlled.

The configured shortcut buttons are also displayed on this page and can be clicked directly to trigger the configured action. If a playlist is started and any placeholders are defined that have an empty value, an input dialog will appear beforehand in which you can enter the desired placeholder value (e.g. the name of a visitor or similar). Placeholders with pre-assigned values are displayed accordingly without a query being made.



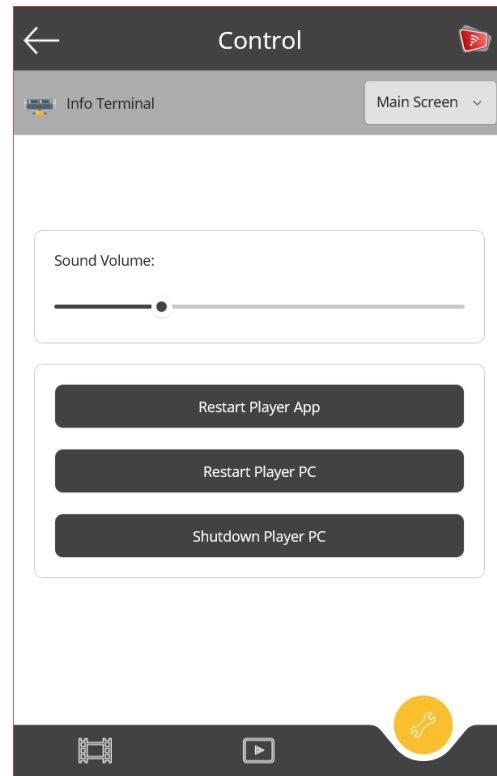
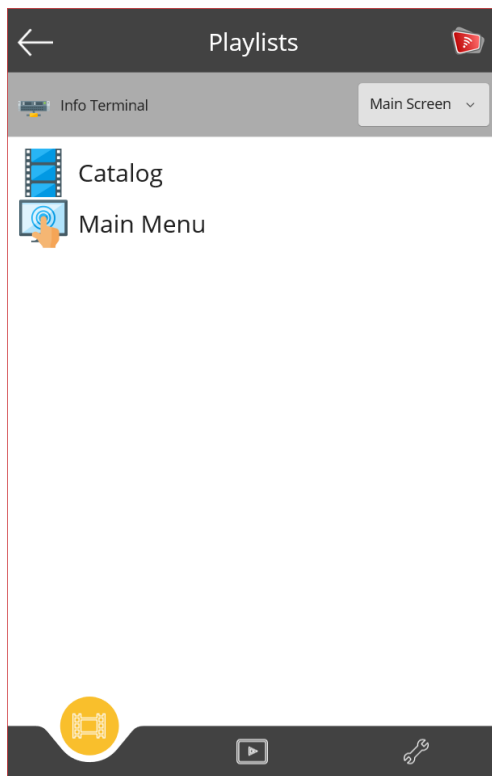
There are two more tabs available from the menu bar at the bottom of the app, which can optionally be deactivated for users, depending on the configuration of the plugin:



Makes it possible to start the playlists available on the player PC (including the assignment of placeholders (syntax: `#{placeholder}`)).

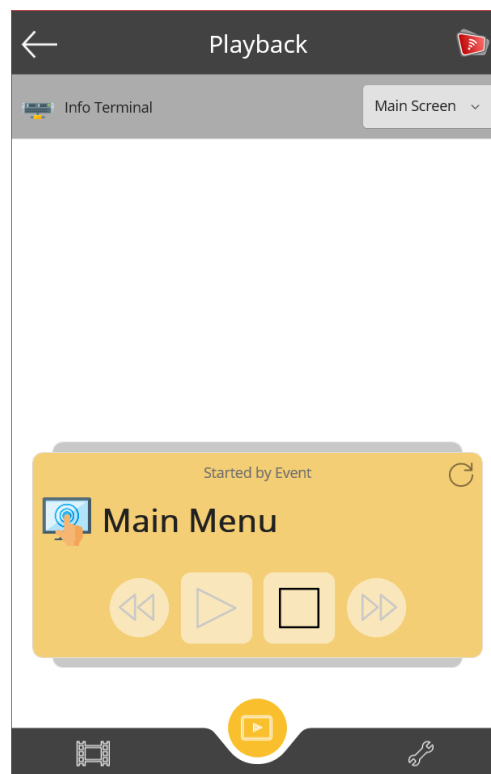


Used to adjust the system volume and other things, for example, to restart or shut down the player PC.



FrontFace differentiates between two types of playlist playback: playlists that are started time-controlled by a timer (FrontFace Assistant) and playlists whose playback is triggered by a so-called "event source". The Remote Control Plugin is such an event source.

Playlists started by a timer are paused by default when another playlist is started by the Remote Control App.



A playlist started by the Remote Control App (or by other event sources) is highlighted in yellow in the "Playback" tab. The playback of such a playlist can also be stopped again. The previously paused playlist started by a timer is then automatically resumed. If a playlist is started by the app while another playlist started by the app or by another event is running, this playlist replaces the previously played playlist. Stopping the playback then always leads to the playlist active by the time control (timer) being continued.

Optionally, when starting a playlist through the app, it can also be specified that the playlist should not be paused in the background. This can be useful if the playlist to be started has a transparent layout, for example, and its content is to be played "over" the previous playlist. For example, a notification or a call system can be implemented in the form of an overlay, with the main playlist continuing to run in the background.

### **Exemplary HTML5 App:**

The included file `FrontFace Remote Control App (HTML5).html` represents a simple showcase HTML5 app that uses the REST Web Service API to control playlists. You can use this HTML page as template or reference for creating your own, custom user interface in order to control the *FrontFace Player App* from any computer, tablet or smartphone that comes with a HTML5 browser.

### **Notice about Identifying and Controlling Playlists:**

Playlists or playlist pages can be referenced by their name or internal ID. You can determine the ID by using the mouse in the *FrontFace Assistant*. Hover over the name of the desired playlist in the list view. A tooltip appears which shows the ID (e.g. PL-X7H-8KA).

Playlists that are started from an event source such as the *Remote Control Plugin* always pause the playlist that is currently started by a timer and continue playing when previously the started playlist finishes or stops. The background playlist will only continue to run if you selected the "Overlay" option for starting the playlist.

### **Help and Support:**

If you need help or technical support when using this plugin or the *FrontFace* software, you can contact our support. For more information, please visit: <https://www.mirabyte.com/en/support/>

## ADENDUM

# FrontFace Remote Control REST API

### PlaylistStart

<http://player-name-or-ip:8066/FrontFace/rest/PlaylistStart?>

playlistIdOrName={PLAYLISTIDORNAME}&screen={SCREEN}&overlay={OVERLAY}&startPageIndex={STARTPAGEINDEX}&duration={DURATION}&loopCount={LOOPCOUNT}&placeholders={PLACEHOLDERS}

{PLACEHOLDERS} is a JSON object with the following syntax:

```
&placeholders={"key1": "value1", "key2": "value2"}
```

### PlaylistStop

<http://player-name-or-ip:8066/FrontFace/rest/PlaylistStop?screen={SCREEN}>

### PlayerAppRestart

<http://player-name-or-ip:8066/FrontFace/rest/PlayerAppRestart>

### PlayerGetName

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetName>

### PlayerGetPlaylists

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetPlaylists>

### PlayerGetCurrentPlaylistName

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetCurrentPlaylistName?screen={SCREEN}>

### PlayerGetCurrentPlaylistIsRunning

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetCurrentPlaylistIsRunning?screen={SCREEN}>

### PlayerRestart

<http://player-name-or-ip:8066/FrontFace/rest/PlayerRestart>

### PlayerSetMasterVolume

<http://player-name-or-ip:8066/FrontFace/rest/PlayerSetMasterVolume?level={LEVEL}>

### PlayerGetMasterVolume

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetMasterVolume>

### PlayerShutdown

<http://player-name-or-ip:8066/FrontFace/rest/PlayerShutdown>

### PlaylistPause

<http://player-name-or-ip:8066/FrontFace/rest/PlaylistPause?screen={SCREEN}>

### PlaylistResume

<http://player-name-or-ip:8066/FrontFace/rest/PlaylistResume?screen={SCREEN}>

### PlaylistShowNextPage

<http://player-name-or-ip:8066/FrontFace/rest/PlaylistShowNextPage?screen={SCREEN}>



### **PlaylistShowPageByPageIdOrName**

[http://player-name-or-ip:8066/FrontFace/rest/PlaylistShowPageByPageIdOrName?  
pageIdOrName={PAGEIDORNAME}&screen={SCREEN}](http://player-name-or-ip:8066/FrontFace/rest/PlaylistShowPageByPageIdOrName?pageIdOrName={PAGEIDORNAME}&screen={SCREEN})

### **PlaylistShowPageByPageIndex**

[http://player-name-or-ip:8066/FrontFace/rest/PlaylistShowPageByPageIndex?  
pageIndex={PAGEINDEX}&screen={SCREEN}](http://player-name-or-ip:8066/FrontFace/rest/PlaylistShowPageByPageIndex?pageIndex={PAGEINDEX}&screen={SCREEN})

### **PlaylistShowPreviousPage**

<http://player-name-or-ip:8066/FrontFace/rest/PlaylistShowPreviousPage?screen={SCREEN}>

### **StartSession**

<http://player-name-or-ip:8066/FrontFace/rest/StartSession>

### **StopSession**

<http://player-name-or-ip:8066/FrontFace/rest/StopSession>

### **PlayerGetScreens**

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetScreens>

### **PlayerGetPlaylistsEx**

<http://player-name-or-ip:8066/FrontFace/rest/PlayerGetPlaylistsEx>

### **ShowNotification**

[http://player-name-or-ip:8066/FrontFace/rest/ShowNotification?  
message={MESSAGE}&duration={DURATION}](http://player-name-or-ip:8066/FrontFace/rest/ShowNotification?message={MESSAGE}&duration={DURATION})

## **Protected Calls with Access Code**

In order to protect the API calls with an access code or password, please enter this in the settings dialog of the plugin in the FrontFace Assistant.

With each API call, the two additional HTTP headers "id" and "token" must be transferred with each request. Generate a new, random character string for each API call, which is passed in the "id" field.

The value for "token" is generated according to the following scheme: Concatenate the stored access code with the "id" random value generated for the call to form a character string and calculate the SHA256 hash from this string. The result must then be passed in hexadecimal code in capital letters as a character string.

Please note that each generated token can only be used once for an API call.

## **Session Handling**

A session timeout can optionally be defined in the settings dialog of the plugin in the FrontFace Assistant. This means that a current session can be terminated on the server side if there is a long period of inactivity. The client application can then, for example, prompt the user to reconnect (with access code).

To do this, add a unique, randomly generated session ID to the client request via the additional HTTP header field "client" for each API call. To start and stop the session on the client side use the *StartSession* and *StopSession* methods.