User manual

erteconnect

Table of contents

Introduction

erteconnect is an automation solution that connects sensors and actors based on events. A variety of technologies that are supported in standard have a focus on retail. But because of its flexible enhancement concept this solution can used in other areas as well, like cyber-physical systems or automation of offices.

Most of the things that are necessary for interactions with customers at the point of sales are available right “out of the box” within erteconnect:

- Interactive Digital Signage on all platforms
- Light control for ceiling and product displays
- Recognition and writing of RFID and NFC tags
- Voice and motion control

The integrated event management engine enables an easy and programming-free setup of consistent actions and action chains. Adjusting content or lighting situations to highlight certain products based on an EAN read via RFID can be setup hassle-free without any programming knowledge.

erteconnect is available on all commonly used platforms, and only requires minimal system resources (e.g. Raspberry Pi).

Modul	Description	Linux	Windows	Android	Apple iOS
api	Event management	x	x		
asset	Device managementt	x	x	x	x
face recognition	Customer Analytics			x	
content	Digital Signage	x	x	x	x
rfid	RFID for Nedap !D Pos	x	x		
caen	RFID for CAEN Reader 1260*	x	x		
m302	NFC for M302 Reader/Writer	x	x		
light	Light control for Ansorg Easy	x	x		
enocean	EnOcean telegrams	x	x		
switch	Energenie switchable outlets	x	x		
listen	Voice control using wit.ai etc.	x	x		
motion	Motion control sensor	x	x		

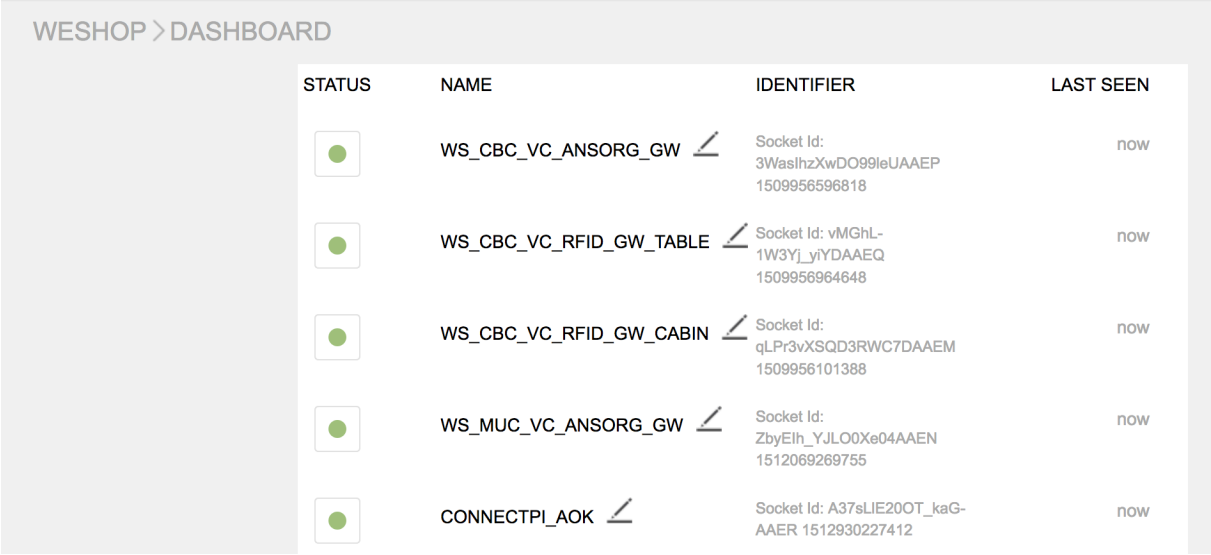
Feature overview

Device management

Dashboard

In device management, the status of erteconnect devices and instances can be managed both remotely and locally. Out cloud platform “Connect” of Erteco Technologies provides easy access to all registered devices. The access URL is http://connect.erteco.de. After login, all options that are included in the subscription are being displayed; in any case, a dashboard icon is displayed to provide access to device management:

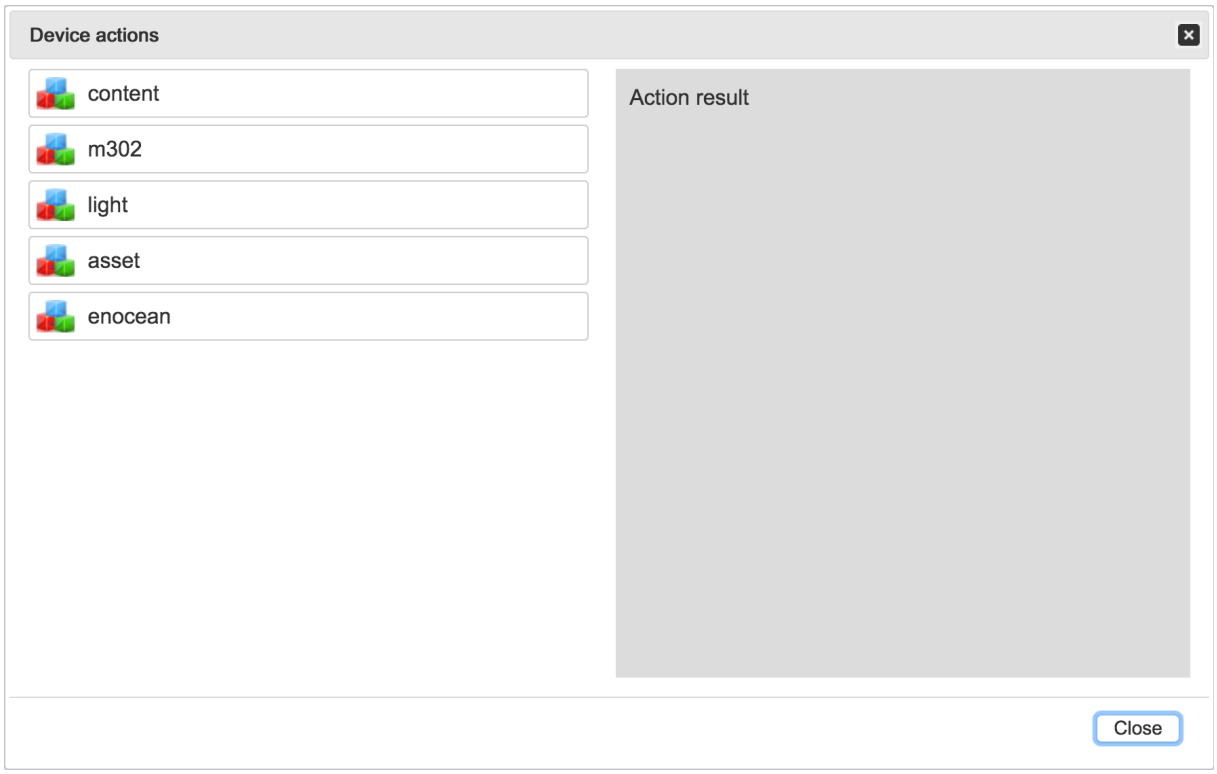
By clicking on this icon, a list of all devices registered for this particular account and their status is being displayed:



This overview shows a devices status as an icon (green or red), the device name which can be changed anytime, a devices socket ID and a time stamp at which the device has been connected to the platform the last time („last seen“).

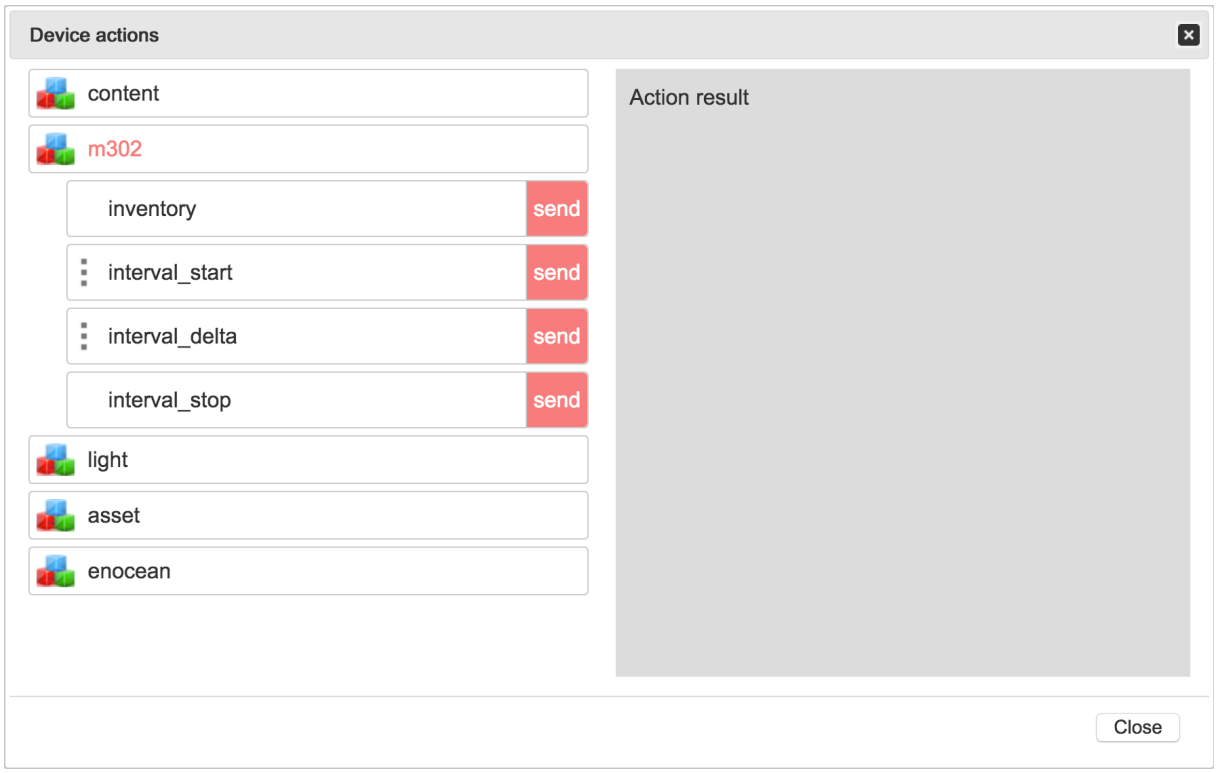
Actions

All functions within modules can be executed directly from within the device overview. The corresponding view can be opened by clicking the (green) status icon:



On this pages' left side there is an overview of all modules that are currently installed on this particular device.

By clicking a module name a list of commands or actions of this module is being opened:

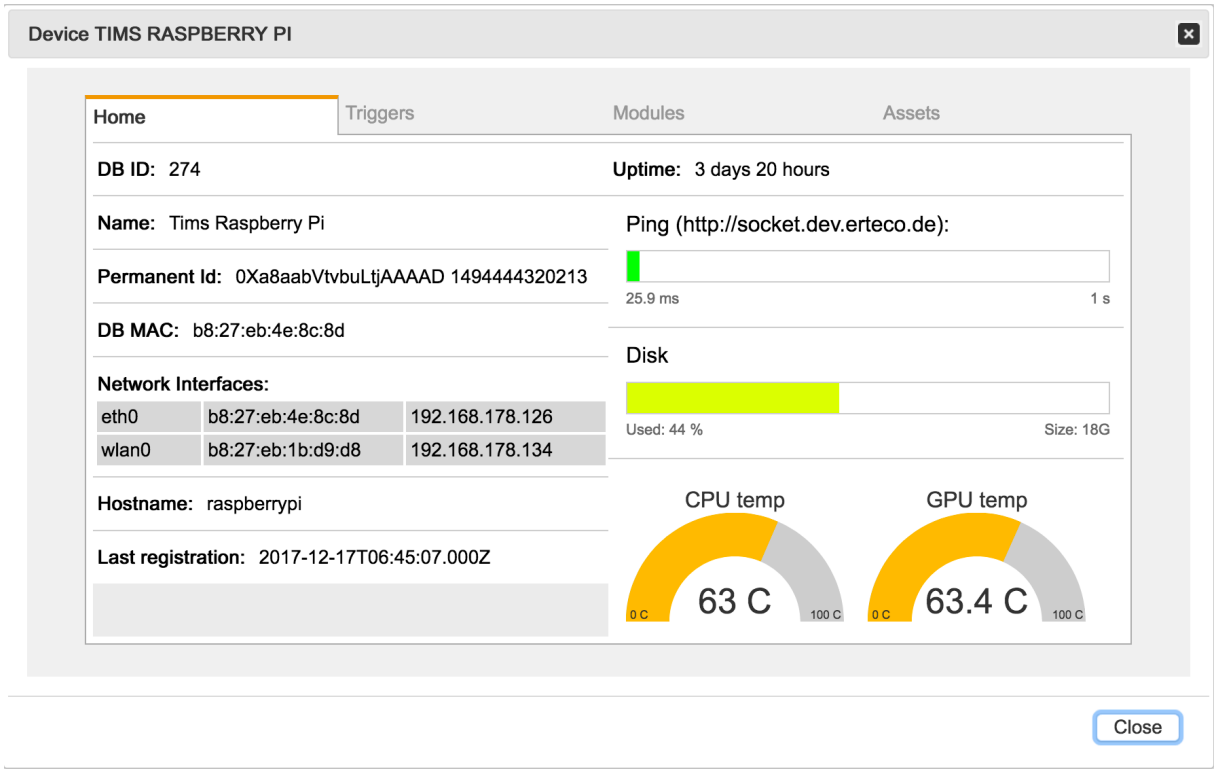


Three small vertical dots to the left of an action indicate that there are parameters that can or have to be supplied for this action. An action is being sent and immediately executed locally once „send“ was clicked. All results of this action execution are displayed in the “Action result” area of this window:

A complete list and explanation of all actions, their parameters and possible results can be found in the respective modules API documentation.

Device information

Detailed information regarding an erteconnect device can be accessed by clicking on its device name in the dashboard. The same information is also available locally within the same network as the device itself by entering the URL http://[Name_or_IP_of_device]:10080/client in an internet browser:



The information displayed is giving an overview of the devices' status and can be helpful for diagnostics purposes in case of support for this erteconnect instance. Additionally, this window gives access to the event management and management of local file and media assets (see next chapter).

Event management

Event management in erteconnect provides an easy way to start actions as reactions to events, or to create event process chains by building consecutive event reactions, without any big programming effort.

Events

erteconnect is built following an object-oriented paradigm, therefore events are always tied to an object. Objects are usually equal to an active erteconnect module. Object “rfid“ can, for example, create an event “inventory“ once an inventory cycle was finished, or objet “enocean“ could throw event „data-known“ when an EnOcean telegram of a known device was received.
The only exception to this rule is object “generic” with event “generic”, which can be used for custom-created events.
A complete list of object at the time of writing of this manual is available as an addendum to this manual; the most up to date version of this object/event list is always available online in the login area of connect.erteco.de.

Event creation

Events can be created within erteconnect in two ways:

1. By executing a command or action in erteconnect, because erteconnect modules usually throw pre-defined event after their execution.
2. By calling a specific URL of the erteconnect instance; this call can be issued either as GET (in the easiest case by entering the corresponding URL into an internet browsers’ address bar), or as a POST. In both cases the format of this call is as follows:
   http://[Name oder IP]:10080/trigger?source=[Objekt]&event=[Ereignis].

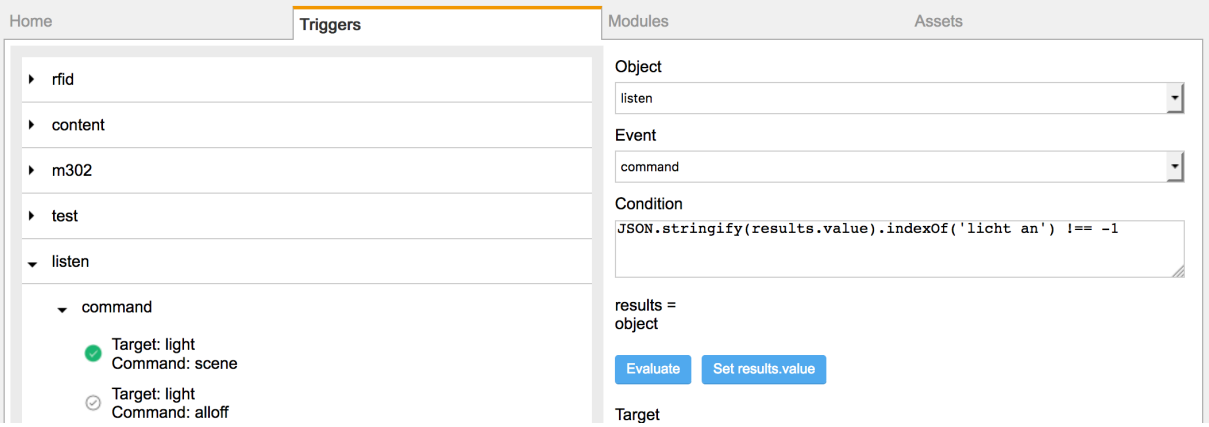
Usually there is event-specific data that is being attached to events, which is giving additional information regarding that specific event. This can be, for example, information regarding the originating device and component that created an EnOcean telegram that was received by erteconnect. That data is encapsulated in a JSON object named “results”, which is attached to all events. There is no standardized structure to this “results” object; it always depends on a specific object, event and event instance. Also, error messages are being transmitted within “results”, and can be handled appropriately by receivers.

If an event is created manually by issuing a web service call via URL, a “results” object can be supplied either as URL parameter (only for GET), or as a JSON object within a message body with format application/json.

A complete overview of all potential structures and contents of “results” objects can be found in all modules’ API documentation.

Conditions

Reactions to events can be dependent on considtions. In order to have most flexibility, conditions are being written in JavaScript. Conditions have to be written as statements that result either in “true” (= 1) or “false” (= 0 or <> 1).



There is one exception to this rule, and that is condition “*“, which means that a condition is being executed in any case, independent of any properties or values in the “results” object.
To support creating conditions there is a feature to test a condition statement before saving it. For this test evaluation to work, the “results” object to check against has to be specified first by clicking “Set results.value”; once this is done, clicking “Evaluate” executes the condition statement and shows its result.

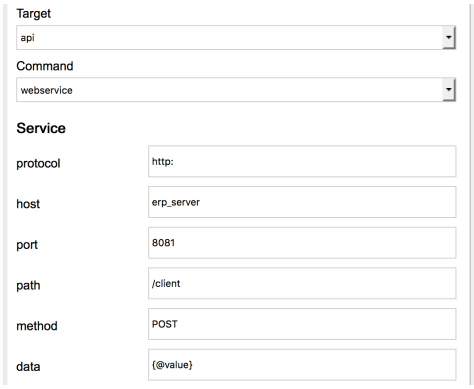
Variables

Both in condition statements and in action parameters you can use properties and values provided by an event in its “results” object during runtime.

Value of properties of a “results” object are being referenced by entering [@name of property].

Example: action “push” on object “content” displays contents of an URL on a signage screen. This action has a parameter called “url”, that specifies the URL to be displayed. If you use „http://google.com/?q=[@pure_uri]“ as parameter value for “url”, its part “[@pure_uri]” is being replaced at runtime with the “results” objects’ value of property “pure_uri”. In case that property has a value of “urn:epc:id:sgtin:4054352.059592.1”, the URL to show would be: http://google.com?q=urn:epc:id:sgtin:4054352.059592.1.
If there are multiple properties with that same name, they are replaced one after another. If in the example above a “results“ object contains property “pure_uri“ twice, one with value „059592.1“ and another with „059592.2“, and parameter “url” would be specified as “http://someserver/?param1=[@pure_uri]&param2=[@pure_uri]”, the resulting URL to call and show would be http://someserver/?param1=059592.1&param2=059592.2.

Aside from using property values from a “results” object one can also use complete objects. This is especially helpful if more than one specific value to supposed to be transferred.
Example: when action “inventory” on object “rfid” is executed, all data of all tags are supposed to be transferred to an ERP server.
This can be done using action “webservice” on object “api”, which is executing a web service call. Data to be transferred can be specified in parameter “data“, in this example with value “{@value}“. The curly brackets indicate that in this case the pattern is not replaced by a property value, but by a complete object from within “results”.

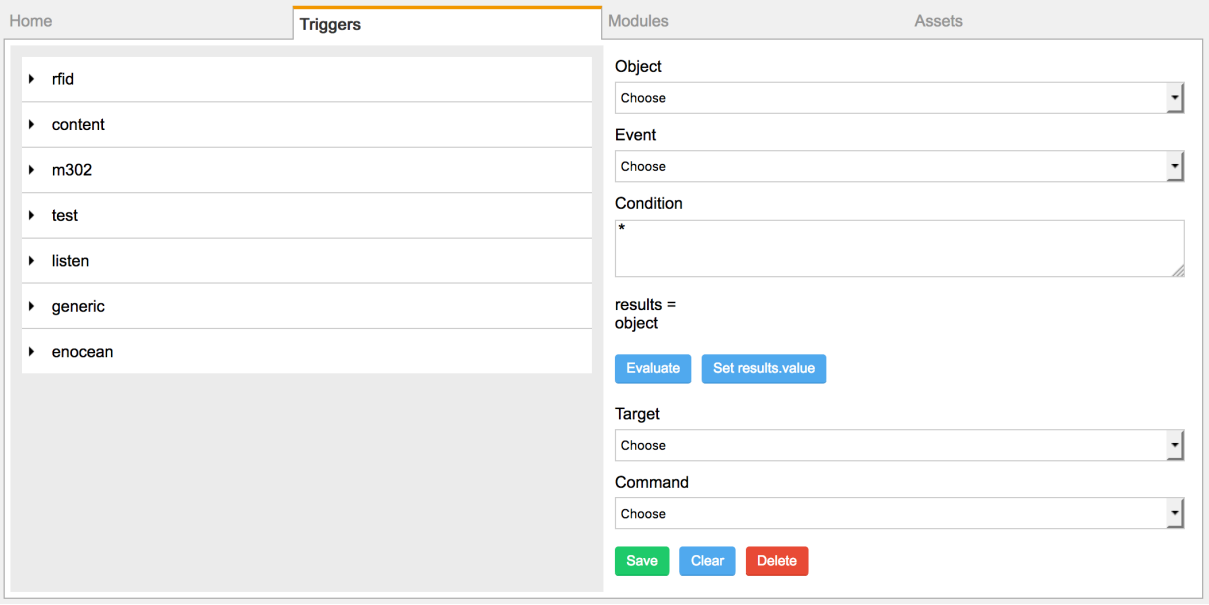


Trigger

A complete set of instructions how to react to an event is called “trigger”. Triggers always consist of

- an object and event, for which a reactions is supposed to be executed,
- a condition,
- a target object and action, that should be executed,
- and parameters for that action execution.

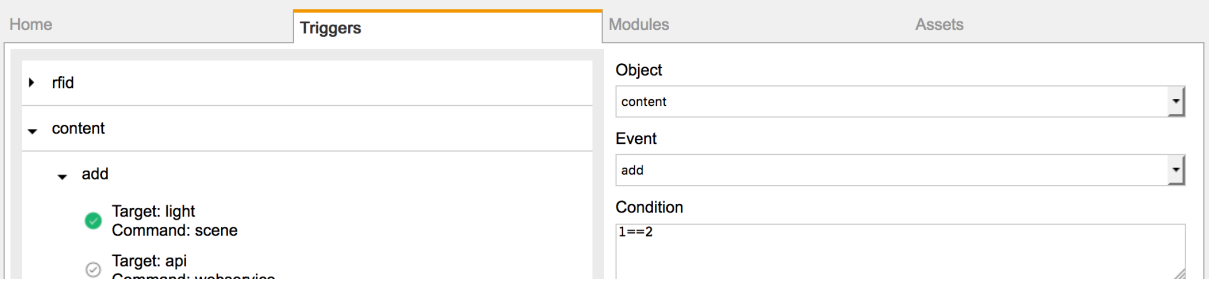
The event management interface is accessible through tab “Triggers” on a erteconnect instances’ detail page:



In the intial view (or once button “Clear” was pressed) a new trigger can be created. First the object of the event to react to has to be specified. Once this is done the list of events is updated and an event can be selected.
The applies to reactions specified in a trigger: first the target object has to be selected, which fills the list of available actions. Once an action has been selected its parameters are being listed below.

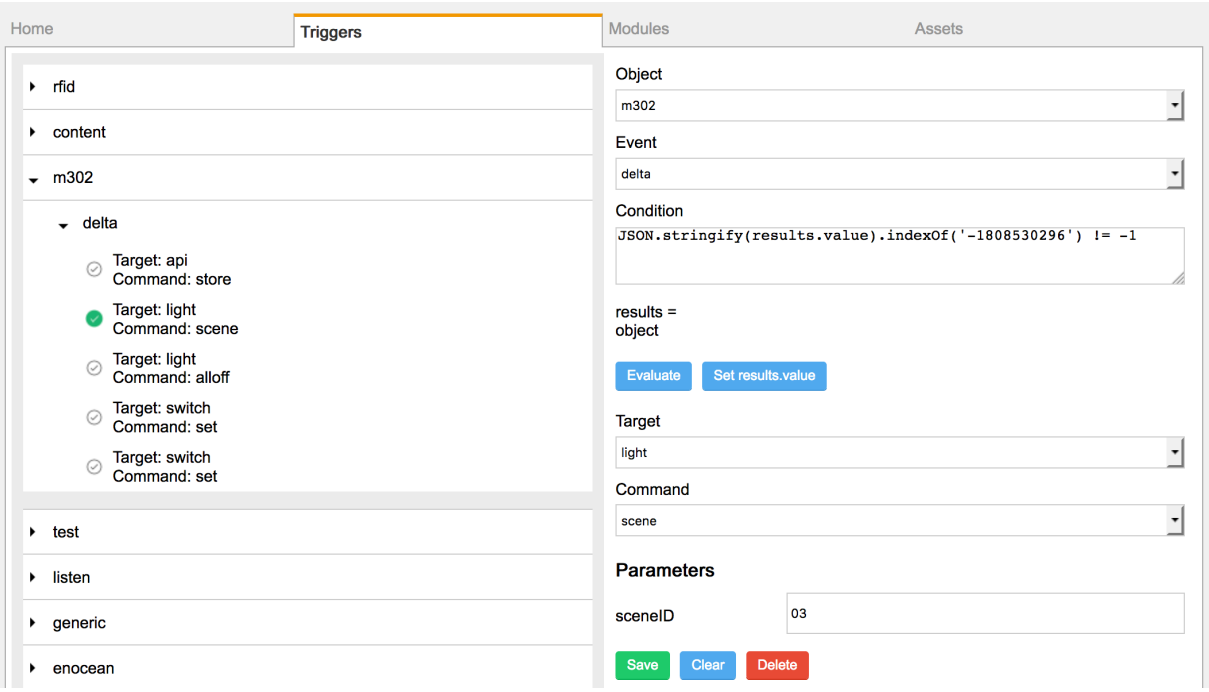
Complete triggers can be save by clicking button “Save“. Immediately after saving, triggers are active and are being executed (of course dependent on their condition), if their corresponding event is thrown.
There is no option or flag to deactivate a trigger. To keep a trigger, but don’t have it executed when its event occurs, its condition can be set to “1==2”, which always results in false and thus always stops execution of this trigger.

Existing triggers are displayed as a hierarchy on the windows’ left side. On top there objects, underneath their events that triggers have been maintained for.



Once a trigger has been selected on the left, its details are displayed on the right, and the tick left to it is colored green.
As long as a trigger is selected, this part of the hierarchy is always open and visible.
A trigger is being unselected once another trigger is selected or button “Clear” is clicked.

An example for a trigger and its meaning:



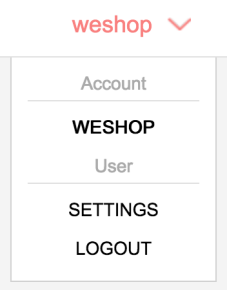
Object “m302”, which is a NFC reader module, creates event “delta” anytime a NFC tag is being read. The events “results” object contains tag IDs which can be used to identify the tagged products.
In this example, once event “delta” of object “m302” occurs, a condition is executed. This condition checks if a specific tag ID is found within the “reults” object. Because the JavaScript function “indexOf” returns -1 if that value is not being found, and its position within “results” if it is, this statement is true if that tag ID is found.
When that tag ID was found within this events “results”, action “scene” with parameter “03” is executed on object “light” (Ansorg Easy light control module). This results in a specific light scene to be set, which can put the detected product into the lime light.

Server

User interface

After logging into http://connect.erteco.de one first sees an overview page:

On this page, in the central section all options licensed to the current account are being displayed, in this case PLAYSISTS und DASHBOARD & STATISTICS.



In the upper right the logged in users name is being displayed. Clicking on this name opens a menu, showing the accounts this user is part of, with the option to change the account for this session (only applicable if there are multiple accounts assigned). In “SETTINGS” a new password can be set.

In the lower right corner is a link to this “USER GUIDE”, which is available online both in English and German.

As soon as an option has been selected in the middle, a navigation path is displayed (as “breadcrumb navigation”) in the upper left, right underneath the erteconnect logo. This navigation enables direct access along the current navigation structure.

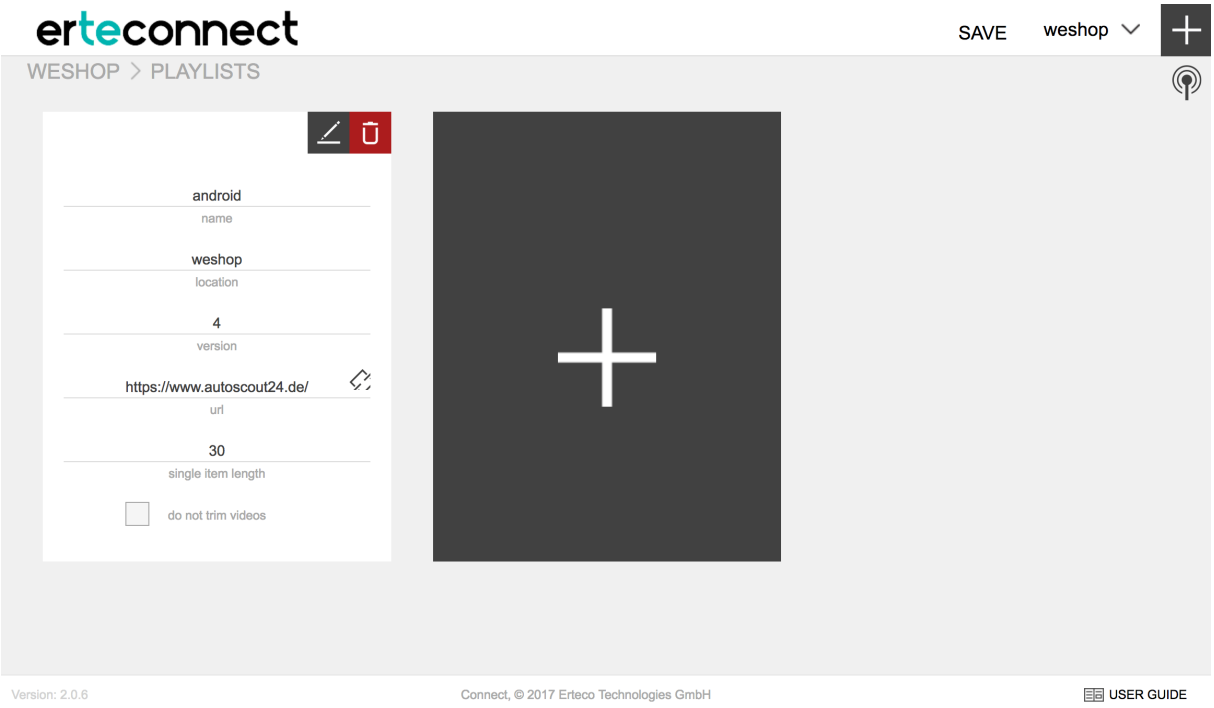
Playlists

In many cases automation scenarios include screen output to display what is happening and other information. Within erteconnect there is module „content“, which provides easy access to basic features for Digital Signage.

One corner stone of digital signage systems is management of playlists, within content items (pictures and videos) can be defined, and which are assigned to erteconnect instances. In erteconnect, playlists are maintained exclusively server-side. Assigning playlists to devices, on the other hand, happens directly to specific erteconnect instances.

Creating playlists

Playlist management can be access by clicking  on the entry page. In playlist management there first is an overview of all playlists that have been created for this account so far.



Aside from the aforementioned “breadcrumb” navigation path there are also other additional items displayed in the upper right corner:

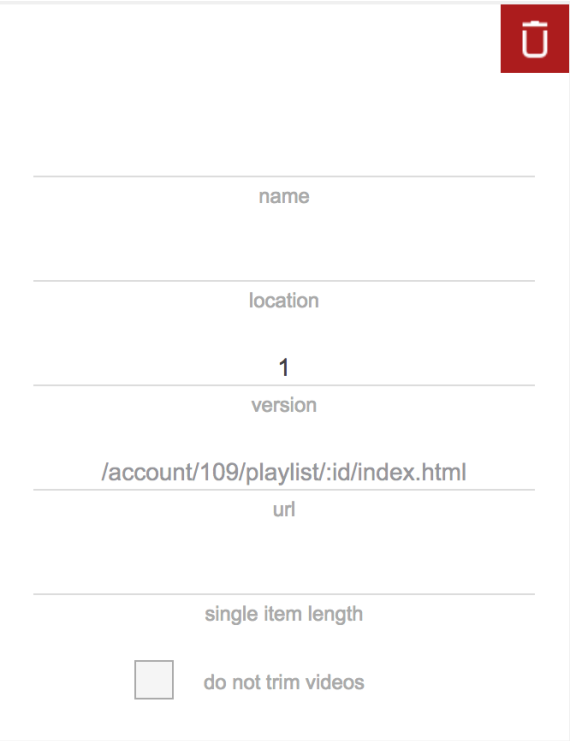
SAVE

A button to save current changes (both for playlists and for playlist items)

A button to create new playlists or playlist items

A button to create Beacon-specific playlists (currently only available on mobile platforms Android and Apple iOS)

A new playlist can be created either by clicking on the big plus tile in the center of the screen, of the small one on the upper right. A new playlist tile without content is being displayed, where that playlists' properties can be maintained:



Name

The name for this playlist; this can be chosen freely. Only upper- and lowercase letters, numbers and some special characters may be used.

Location

A location where this playlist is being used; this can be chosen freely, similar to the name. There is no assignment of playlists to devices or instances based on this criterion.

Version

The playlist version; this is maintained automatically by the server, increasing by 1 every time this playlist is being saved.

URL

The web address for this playlist, which can be used to test a playlist before it is rolled out to signage devices. Additionally, if a playlists or other content to be displayed is already saved at a different web server, the default value (in grey) can

be overwritten manually. In that case any content within this erteconnect playlist is being ignored, and instead content pulled from the URL specified.

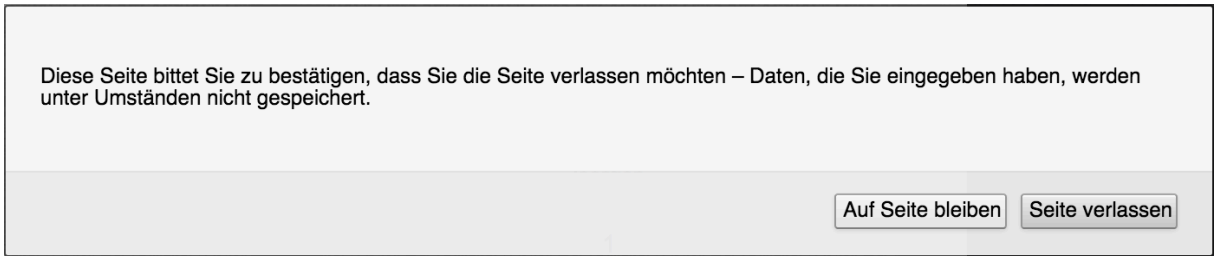
Single Item Length

Time in seconds that an individual item within this playlist is being displayed. This time is the same for all items in a specific playlist; the only exception are videos (see below).

Do Not Trim Videos

If there are videos within this playlist, this flag determines whether they are played for their full length or not, before the next element is displayed. In case this flag is not set, videos are being interrupted after the time in seconds specified above, and the next playlist items comes up.

Once all information regarding a playlist has been maintained, a new playlist can be saved by clicking button „SAVE“. If changes have not been saved when trying to leave the page, erteconnect is asking whether or not to stay and save:



Once a playlist has been created and saved, clicking its pencil symbol open a maintenance page:

In this maintenance page all playlist elements are displayed in a small preview. In case of a newly created playlist this overview is initially empty, with the exception of a big plus tile.

A new playlist element can be created by clicking this plus tile or the one in the upper right corner, similar to playlist management. Now content can be added into this element by clicking “Search…”. It is possible to create multiple playlist elements right after another.

When creating playlist elements, interactions can be defined. These interactions are being executed when a playlist element is being tapped on during display. Interactions can either be websites that are specified in “action url” and opened upon tap, or apps that are pre-installed locally and launched on tap (only available for Android platforms.

To display a website , the following format should be used:

http://localhost:8080?action=play&url=[URL to display]&duration=[seconds]

In this case tapping a playlist element during playback opens the website specified in parameter “url” for as long as there has been no interaction for “seconds” seconds. If, for example, the time has been set to 20 seconds, the website that was opened can be browsed and scrolled and navigated in, and once 20 seconds passed without any such interaction, erteconnect restarts the current playlist.

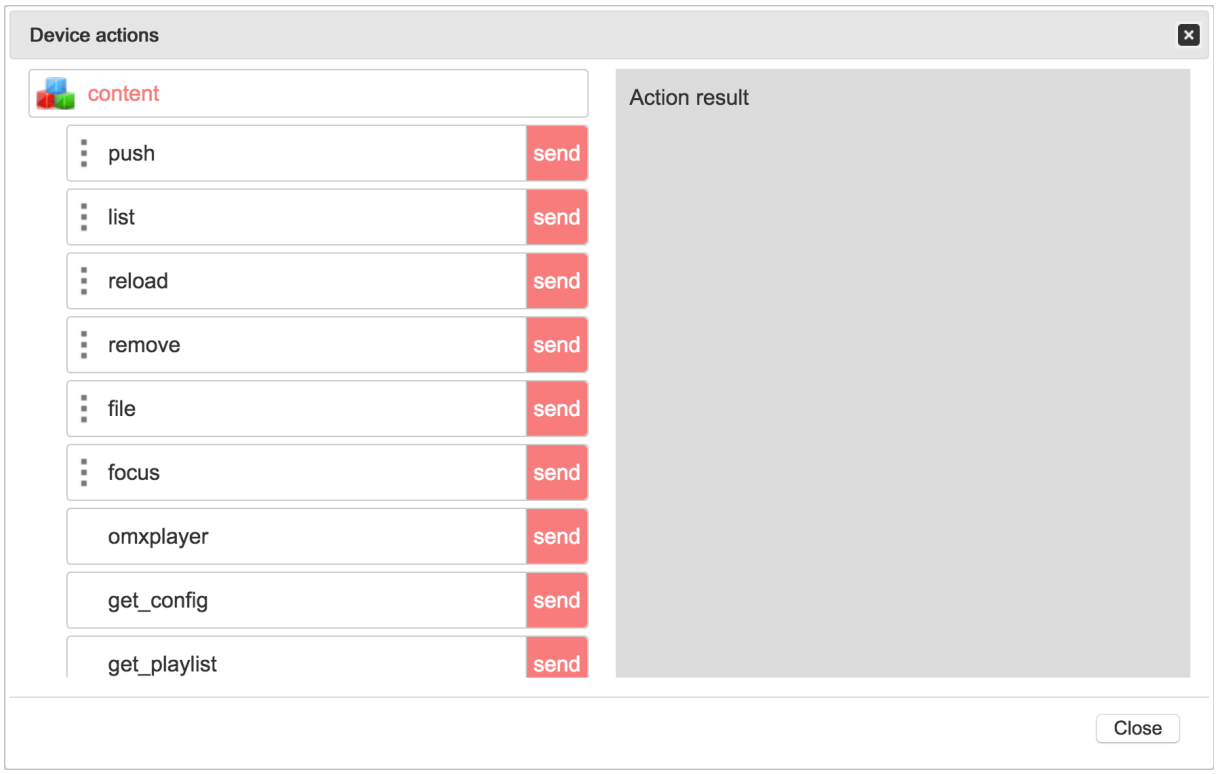
Starting locally installed apps for erteconnect running on Android platforms is being explained in chapter “Mobile - Android & Apple“.

Once all corresponding playlist elements, i.e. content, have been added to a playlist, and potential changes to basic playlist information have been done (in the lower section), all changes need to be saved by clicking “SAVE“. Otherwise all changes are lost (see above)!

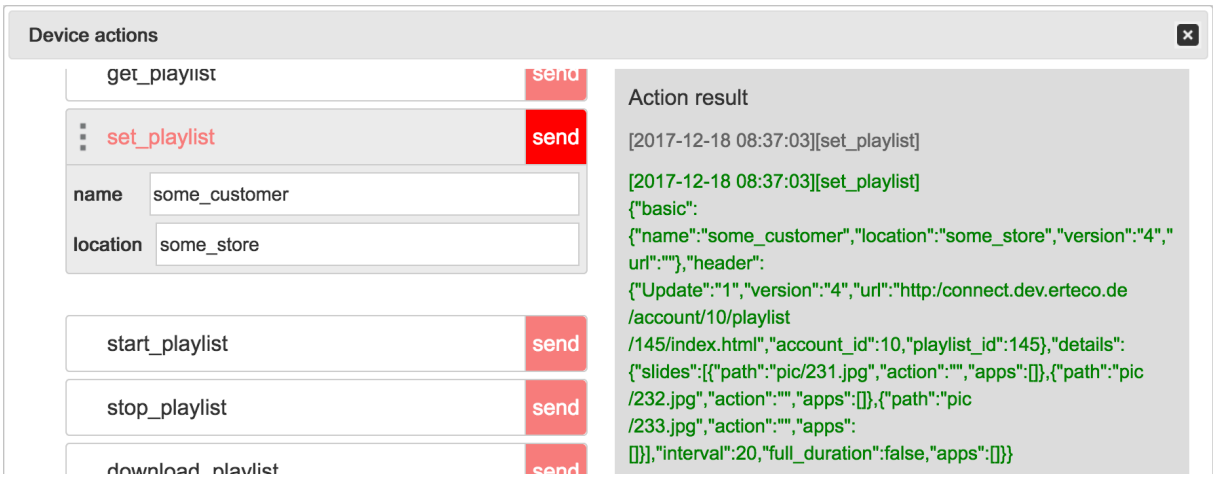
Assigning playlists

Assigning playlists to erteconnect devices is done through device management, and on mobile platforms right in the erteconnect player app.

To assign playlists to devices one has to navigate back to the entry page, and click on tile „DASHBOARD & STATISTICS“. A page with a list of devices available in this account is opened, and in that list the status icon left of the device name, and then “content” have to be clicked.



Dependent on screen size and resolution one has to scroll down to action “set_playlist”, and is opening it by clicking on its action name:



In this view, one can provide both the name and location of the playlist to be played on the device as parameters. By clicking “send” this action is being sent to erteconnect, and its results are displayed in “Action result” on the right side right after execution.

Starting playlist

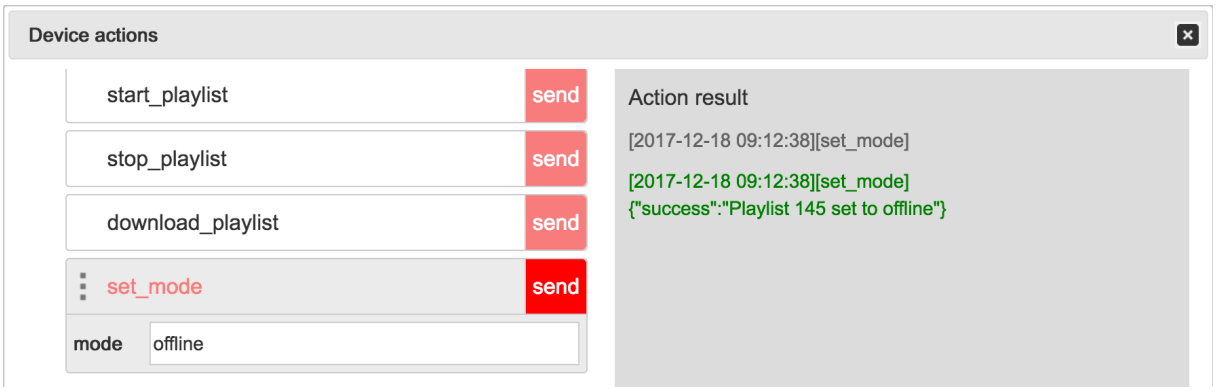
Playlists can either be started and played straight from the server, or synchronized to a local erteconnect device and played offline from there to save network bandwidth.

Starting and playing a playlist from a server is the preferred method in most cases. Not only are changes to playlists synchronized automatically, but because of the caching feature of Chrome / Chromium, which is used for displaying content in erteconnect, bandwidth usage is reduced as well.

To start a playlist locally one has to navigate to “Device actions“ again by clicking the status icon next to the device name, open module “content”, scroll down to “start_playlist“, and click “send” next to it:

To start a playlist offline, one first has to synchronize all content items from a playlist to a local erteconnect device using action “download_playlist”:

Now playback of that playlist could be started offline immediately, but it is recommended to first switch the mode of erteconnect to offline:



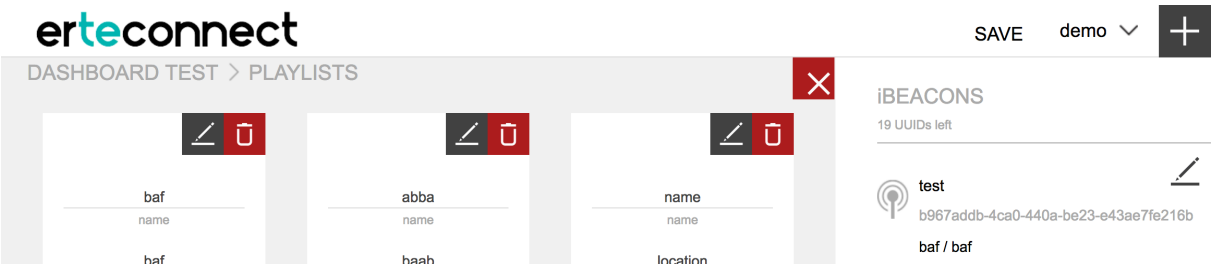
From now on even after a reboot of this erteconnect instance, playback of this playlist starts automatically in offline mode.

Another option to directly start offline playback of a playlist is to use action “play_offline” of module “content”. The downside here is that in this case offline playback is only valid while this erteconnect instance is running. After a reboot or restart the current playlist is always started according to the “mode” (online or offline) in settings.

Beacon-Playlists

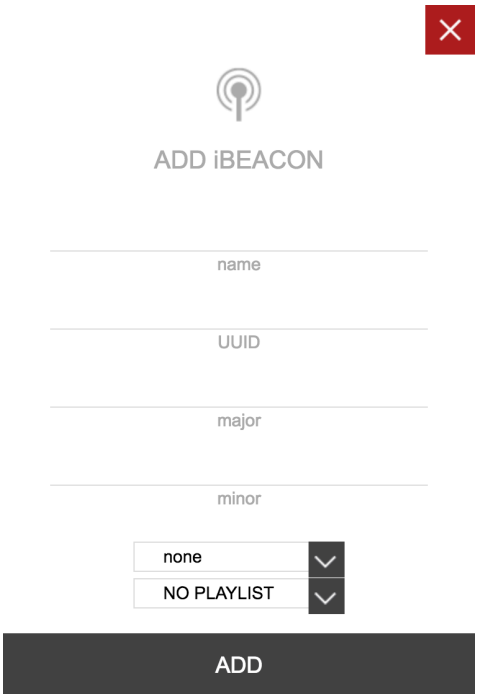
In addition to a specific assignment of a playlist to a erteconnect device one can also define playlists to be started automatically once a certain Beacon has been received by an erteconnect instance (currently only available for mobile platforms Android and Apple iOS).

The beacon playlist settings are accessible by clicking on in the playlist overview page. This opens a new section on the right side that displays a list of currently setup beacons:



Because of technical restrictions on some mobile client platforms, in sum there are only 20 different beacon UUIDs that can be specified.

Beacon playlists can be created by clicking button + in the lower area of the „iBEACONS“ section.
A window appears where basic information regarding the beacon initiating the playlist playback can be maintained:



Name

A free text to differentiate this beacon playlist from the others in this list; this can be maintained freely.

UUID

The UUID of the beacons that should initiate the playback. Ideally this value is copied from a beacon scanner, since it usually is quite complex in its structure.

Major / Minor

These values are also transferred from a beacon together with its UUID, and provide means to differentiate within a single UUID. Because of this the limit of 20 UUIDs to track on a single account is not relevant, since by using minor / major values in addition the number of playlists to start is sufficient for most cases.

Matching class

This value can and should be left at its default value of “none” (standard). Other values could be used to enable specific recognition mechanisms, however these are usually not applicable.

Show playlist

Selection of a playlist to start once those specific beacon and minor / major values has been received on an erteconnect instance.

Creation of a beacon playlist is finalized by clicking on button „ADD“.

Existing list items can be changed or deleted anytime by clicking ✓ in „iBEACONS“, and choosing the corresponding button on the bottom.

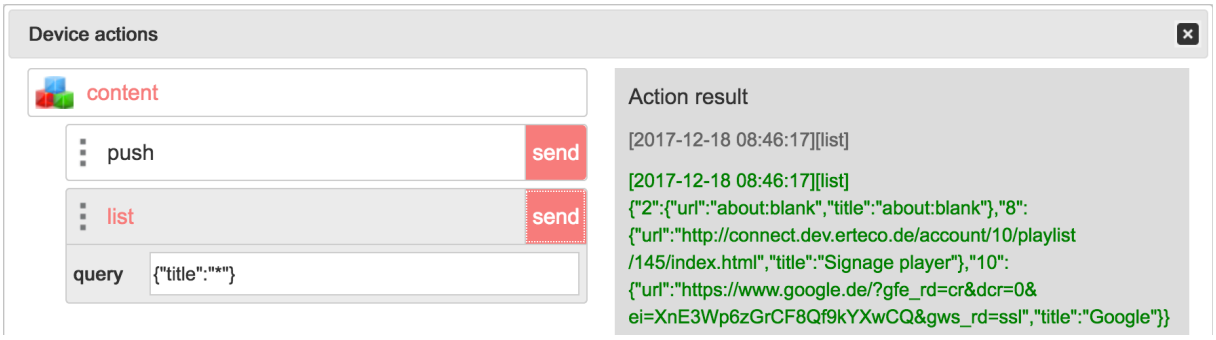
Showing individual pages

With erteconnect one can also use a device as player for digital signage without using playlists from the cloud. This can be done by sending a websites' URL directly to Chrome / Chromium, which in turn is opened and displayed as a new tab.

To show a websites' URL, action „push“ in „Device actions“ is being used:

In field „url“ the URL of the website to display is entered. After clicking “send“ the result of this action is displayed on the right side, which contains both the ID and URL of the tab that just has been opened as a confirmation.

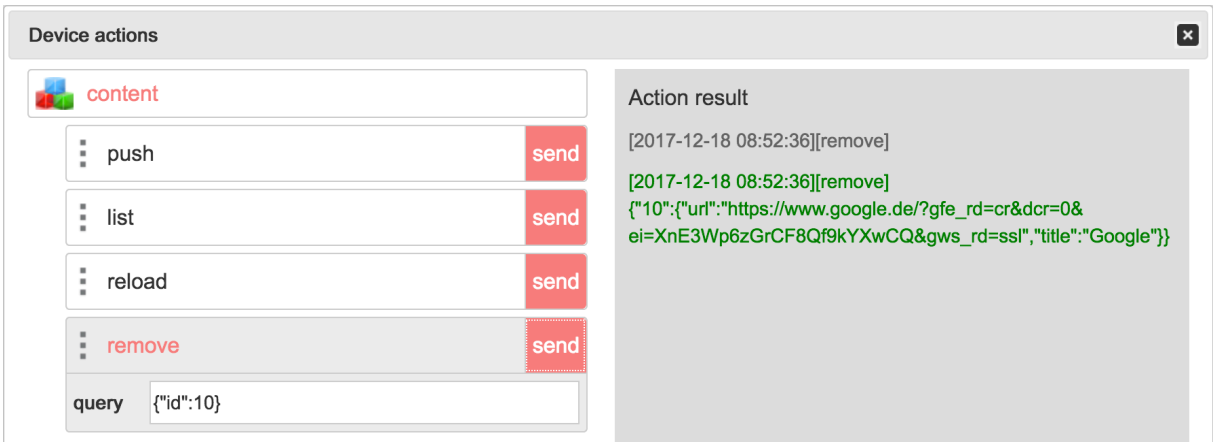
The current status of all tabs within Chrome / Chromium can be displayed by executing action “list“, for example with parameter “query” set to search for all titles:



As a result, a list of all tab matching that query is displayed, containing information about the tab IDs, their URL and title. If the focus hasn't been changed manually,

always the last tab that was opened is the one displayed. In the example above this is Google.

To close a tab, one can use action “remove”. Since there is a chance that more tabs are closed that were intended if the query criterion was too extensive, it is advised to use “id” as criterion when removing tabs:



After execution of this action via “send” the list of tabs that were just closed is being displayed.

Additional actions to manage tabs in Chrome / Chromium in an erteconnect instance are “reload“, to update tab contents while bypassing the local cache, “file“ to display files that are saved locally on an erteconnect instance, and “focus“, to change the currently displayed tab. More in-depth information regarding these actions is available in the API documentation of module “content“.

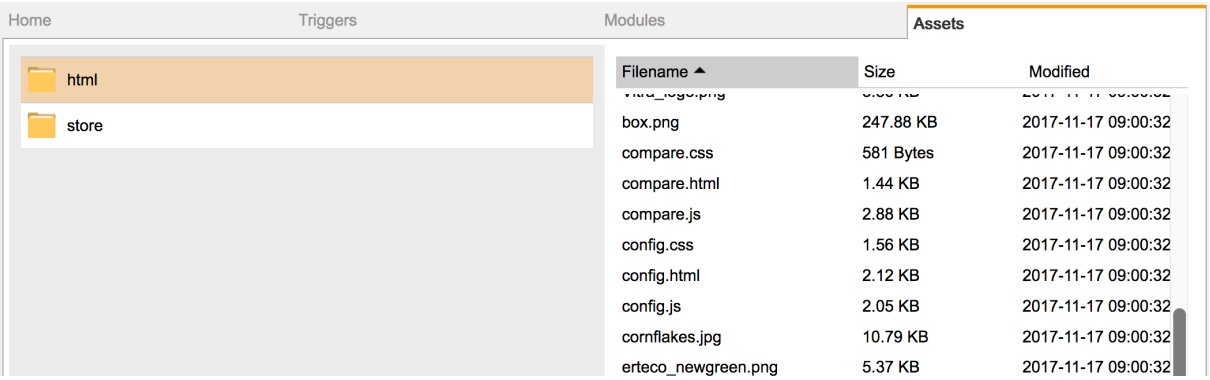
As shown above, most actions that manipulate tabs in an erteconnect instance require a “query” parameter to be specified to identify the corresponding tabs. More information regarding the various queries and query parameter is available online in Googles’ API documentation at https://developer.chrome.com/extensions/tabs#method-query.

Maintaining local assets

So far all options shown to playback content (with the exception of displaying files saved locally) require a separate web server that contains the data to be displayed.

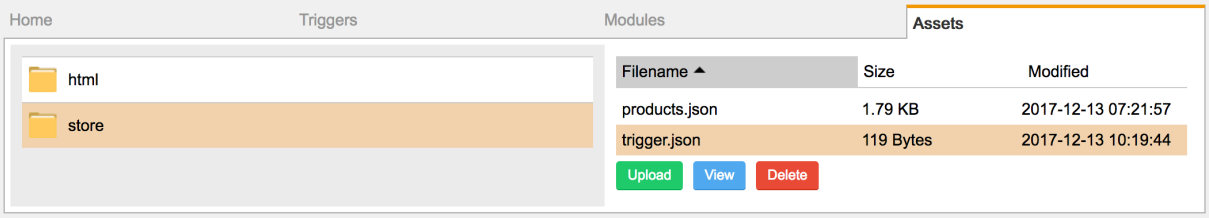
Since erteconnect instances have a built-in web server to handle API traffic via HTTP, one can also use this web server to serve HTML-based content. Maintaining

this content can be done in device management, by clicking the name of an erteconnect instance and selecting tab “Assets“:



On the left side there is an overview of those local folders on the erteconnect device where files can be maintained by a user. Files residing in folder “html“ can be accessed by entering the following URL into an internet browser:
http://[name/IP address of erteconnect instance]:10080/html/[name of file]
Files that are saved in folder “Store“ can be used to provide data access to dynamic web pages, like product data.

Maintaining folder contents is done using button “Upload“, which is always displayed once a folder has been selected, and buttons “View“ and “Delete“, all of which are located at the bottom. The latter two are only displayed once a file has been selected on the right.



Once the necessary files for a solution have been created, they can be displayed using action “push“ in module “content“.

As an example for these kinds of solutions, all necessary files for solution “Smart Shelf” are included in standard deliveries of erteconnect. This solution is based in two NFC readers and provides an overview for product comparisons. The necessary files are:

- html/compare.html (main/entry page to “push“)
- html/compare.css
- html/compare.js

- store/products.json

These files can be used as template for customer-generated and -used scenarios.

Dashboard

An introduction into and overview of features of the dashboard was already provided in chapter “Device management” in “Feature overview”.

Client

Platforms

Desktop - Linux & Windows

Additional information regarding platform-specific issues around erteconnect on Linux and Windows will be provided at a later point of time.

In the meantime, please send your questions regarding these topics to the following mail address: info@erteco.de

Mobile - Android & Apple

On Android and Apple iOS, both of which are usually mobile platforms, there is only a limited feature set of erteconnect available. This feature set is limited to digital signage, with the exception of Face Recognition / Customer Analytics, which is currently only available on Android. As mentioned further up, both mobile platforms are currently the only platforms that react to beacon signals to initiate playback of specified playlists.

Additional information regarding platform-specific issues around erteconnect on Android and Apple iOS will be provided at a later point of time.

In the meantime, please send your questions regarding these topics to the following mail address: info@erteco.de