

Setting up a UDC test connection

URL: <https://docs.grassfish.com/docs/setting-up-a-udc-test-connection>

Archiviert am: 2025-07-17 18:36:35

In order to test the different web services (server and client) and to improve the stability of the spot via local data, it is possible to set up test connections. The corresponding parameters must be defined correctly and in part, have a limited validity.

Note

To secure a correct operation of the spot, the spot developer must ensure that the spot

- is tested against all web services (server and client)
- is transferred to all corresponding player types and the playback is tested (players have different web engines and therefore exhibit differing behavior).

The test connection is set up via the `setLocalTestData` function (see `setLocalTestData`). Below, the determination of the parameters as well as the creation of a local data file under the use of the browser developer tools is explained.

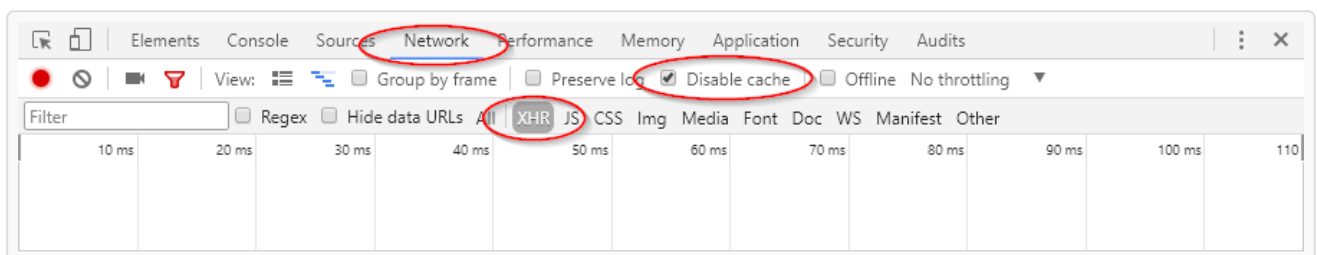
Using developer tools

The required values can be read out via the developer tools of the browser.

Note

We recommend Google Chrome. However, it's possible to use other browsers.

It is recommended, to permanently deactivate the browser cache and to filter the display of the requests as specified in the following image:



Testing of web services

Parameters for the server web service

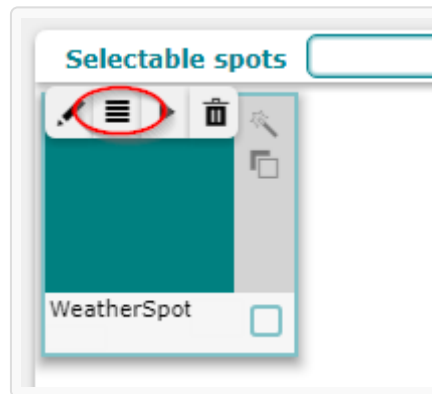
Note

In order to allow for all required parameters to be read out, a corresponding spot with a UDC data source must be available in IXM One.

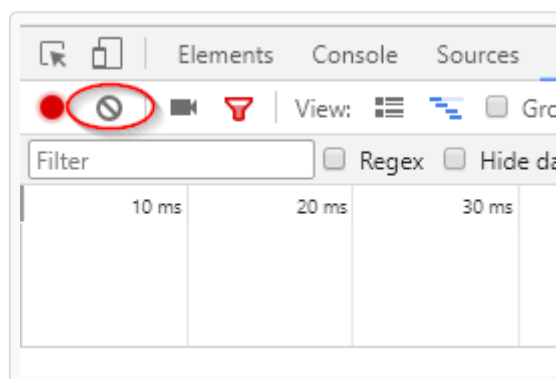
1. IXM One: Log in and open the **Edit spot properties** window of the spot which should be tested under **Content > Playlist> Selectable spots >**



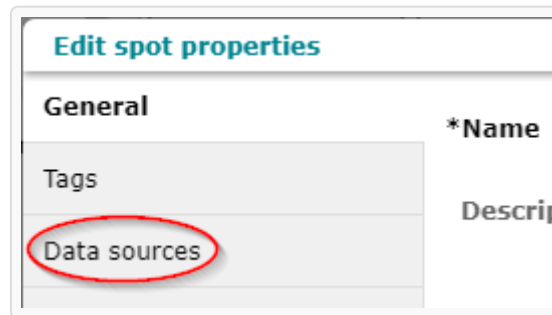
in the tile view of the spot.



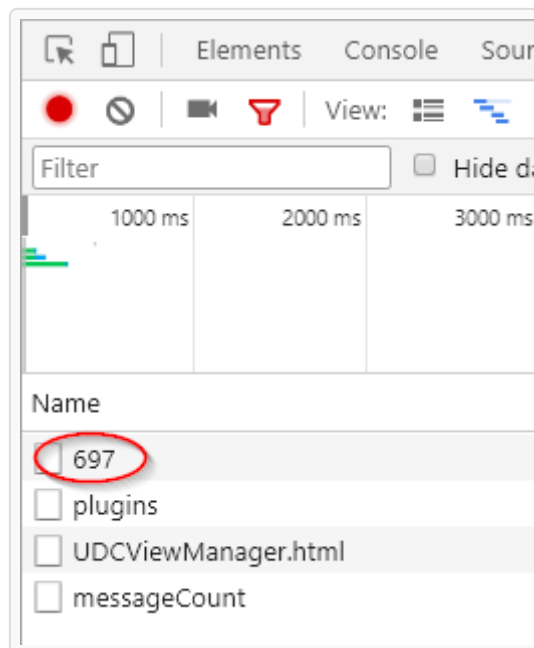
2. Developer tools: Open the developer tools of the browser via the keyboard shortcut CTRL+SHIFT+I (for Windows operating systems only).
3. Developer tools: Switch to the **Network** tab.
4. Developer tools: Delete existing requests by clicking on the icon as illustrated in the image below.



5. IXM One: In the **Edit spot properties** window select the **Data sources** tab.

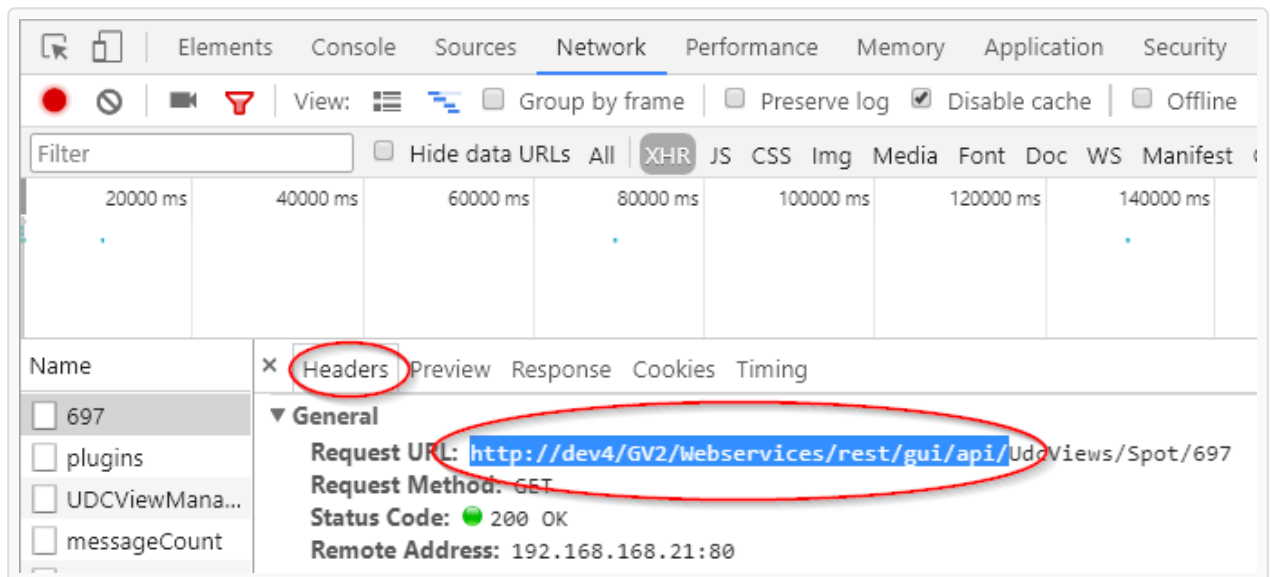


6. Developer tools: Click on the displayed number (Id), in order to display the corresponding details on the right.



7. Developer tools: In the pane **Details** switch to the **Headers** tab.

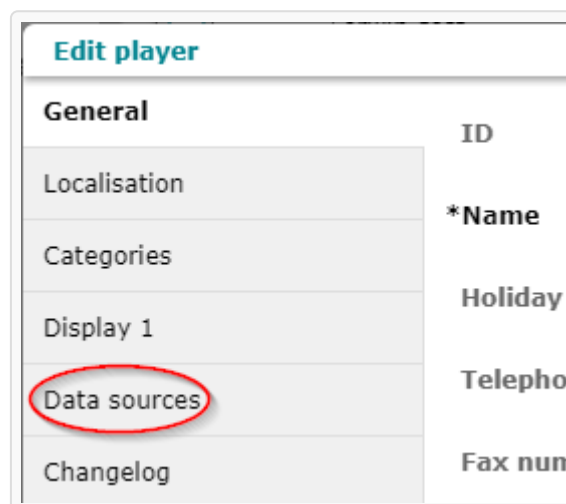
8. Developer tools: Select the request URL shown including „api/“ and copy the highlighted value to the clipboard with CTRL+C.



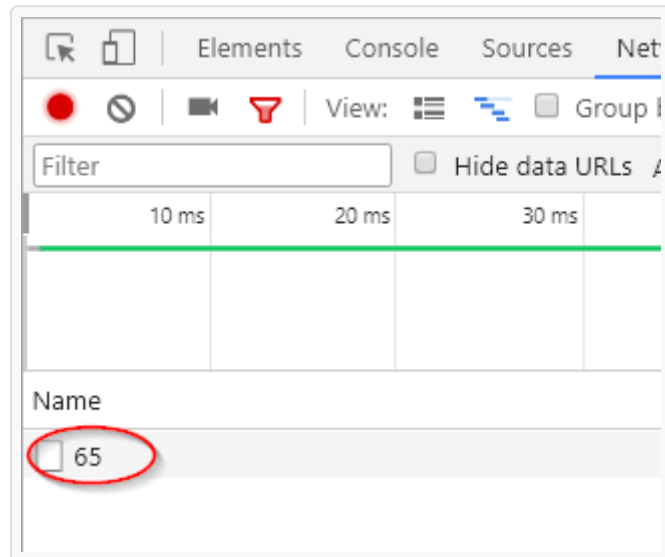
9. Code: Replace the value of `DebugServer.url` with the value saved to the clipboard.

locationId

1. IXM One: Open the **Edit player** window of the player which should be tested under **Channels > Location > [Name of the location] > Edit player**.
2. Developer tools: Open the developer tools via the keyboard shortcut `CRTL+SHIFT+I`, switch to the **Network** tab and delete the existing requests.
3. IXM One: Click on the **Data sources** tab.



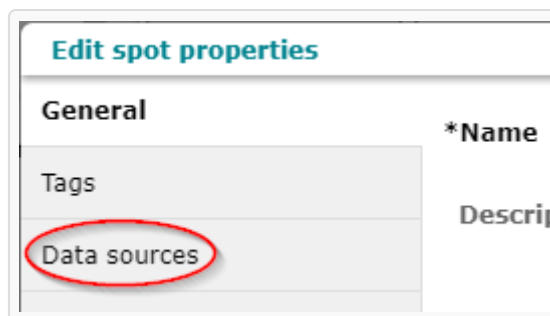
4. Developer tools: Identify the displayed value (Id).



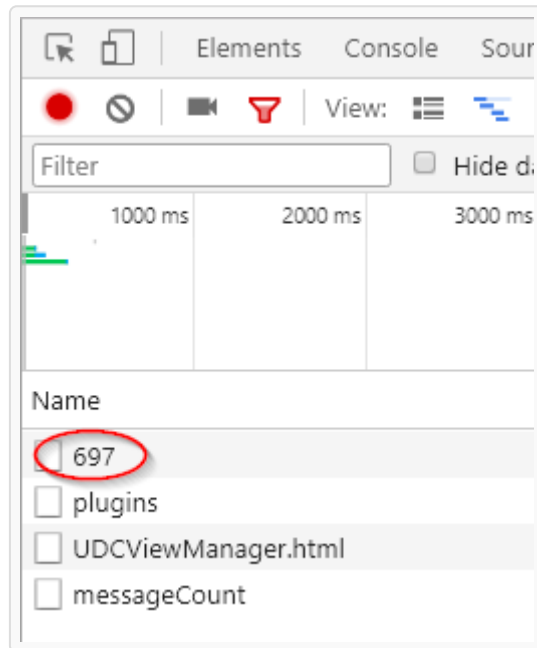
5. Code: Use the value for `DebugServer.locationId` (see `DebugServer`).

spotId

1. IXM One: Open the spot properties.
2. Developer tools: Open the developer tools via the keyboard shortcut `CRTL+SHIFT+I`, switch to the **Network** tab and delete the existing requests.
3. IXM One: In the **Edit spot properties** window click on the **Data sources** tab.



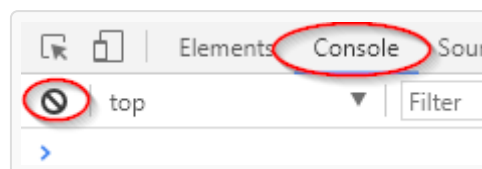
4. Developer tools: Identify the displayed value (Id).



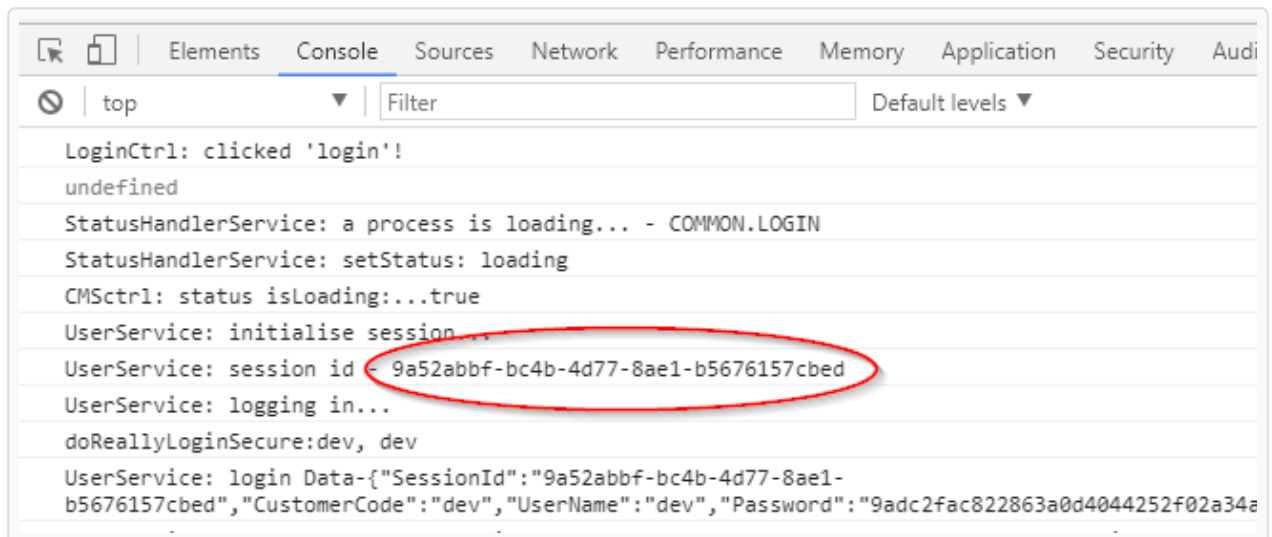
5. Code: Use the above value for `DebugServer.spotId` (see `DebugServer`).

sessionId

1. IXM One: Open IXM One or if there is already one instance of IXM One running, close the current session via the log out icon in the top right. The login window is shown.
2. Developer tools: Open the developer tools of the browser.
3. Developer tools: Switch to the **Console**.
4. Developer tools: Delete the existing values (see image):



5. IXM One: Log in to IXM One.
6. Developer tools: Highlight the session key and copy it to the clipboard.



7. Code: Replace the value of `DebugServer.sessionId` with the value copied to the clipboard.

Note

The value for `DebugServer.sessionId` is only valid for the current session. For this reason, the window should be not closed during testing. The value must be renewed for every additional login as well as the above described process must be repeated.

Parameters for the client web service

Note

In order to allow for all required parameters to be read out, a corresponding spot with a UDC data source must be available in IXM One. It must be also be available in a playlist which has been transferred to a player.

Ip

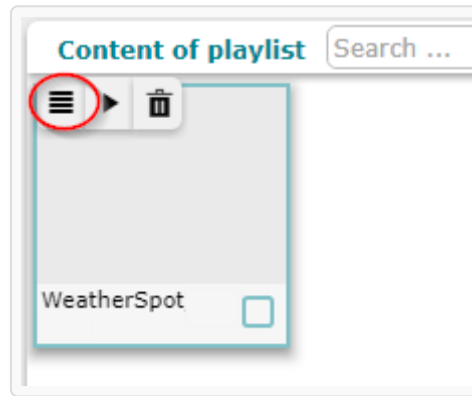
1. Player: Identify the IP address by pressing the F8 key.
2. Code: Use the displayed value for `DebugClient.ip`.

siid

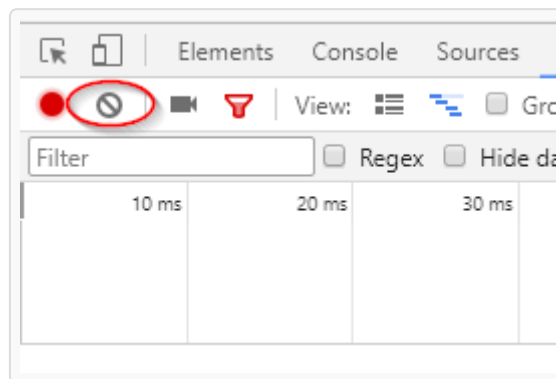
1. IXM One: Log in and open the **Edit display properties** window of the spot instance which should be tested under **Content > Playlist > General playlists > [Name of the playlist] Playlist content >**



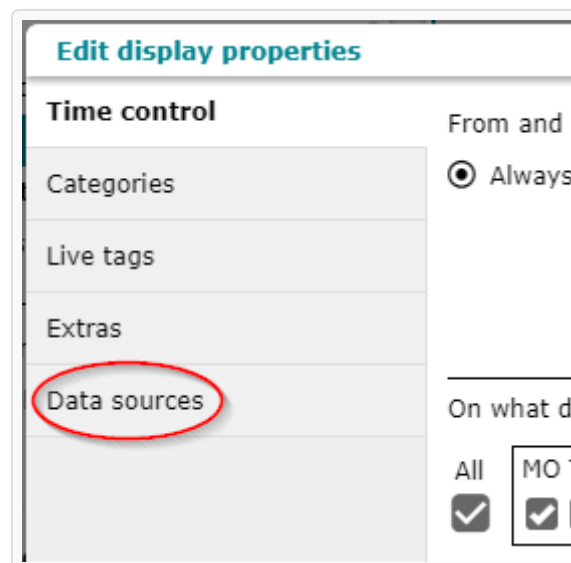
in the tile view of the spot.



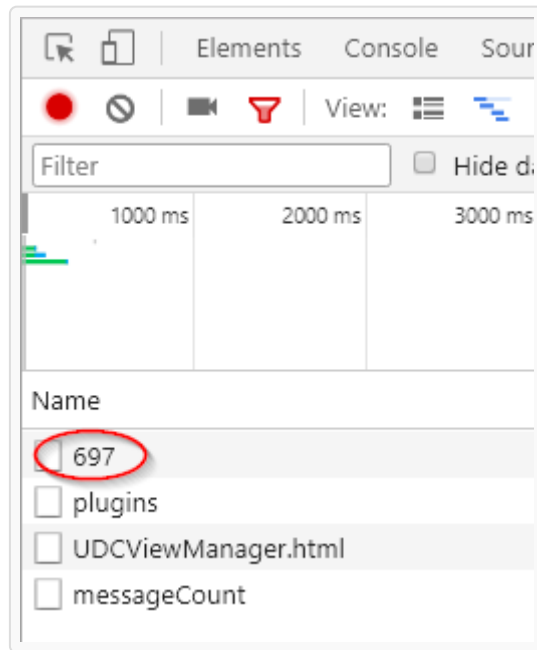
2. Developer tools: Open the developer tools of the browser via the keyboard shortcut CTRL+SHIFT+I (for Windows operating systems only).
3. Developer tools: Switch to the **Network** tab.
4. Developer tools: Delete the existing request by clicking on the icon as shown in the image below.



5. IXM One: In the **Edit display properties** window click on the **Data sources** tab.



6. Developer tools: Identify the displayed value (Id).



7. Code: Use the above value for DebugClient.siid.

Creating local test data

In order to test the spot against local data, establish an initial test connection to the web services, optionally via server or client. For this, the corresponding data sources are accessed.

Following this, complete the steps below:

1. Browser: Open the spot in the browser via the ascInterface.
2. Developer tools: Open the developer tools via the keyboard shortcut CTRL+SHIFT+I.
3. Developer tools: Switch to the **Network** tab.
4. Developer tools: Identify the UDC request:
 - It starts with a distinct description (e.g. "News").
 - If there are two entries available, select the entry with the larger file size (it contains the data).


```
var udc = new GFUdcConnector();
var connection = {
    url: "./udc.json"};
udc.setLocalTestData(connection);
// fetch some data...
```

Using local test data allows for a simple and convenient way to develop the spot against a static server and client solution. Further, it is possible to quickly modify the data and simulate various scenarios.