

Configure the server

URL: <https://docs.grassfish.com/docs/en/configure-server-ssp>

Archiviert am: 2025-07-17 18:37:19

This chapter provides step-by-step instructions for the configuration of the Grassfish Server and web service.

Install the web service

You need to install the AdServing API web service to query playout schedules from an external SSP. To install the web service, you must copy its files to the Grassfish Server.

To do so, perform the following steps:

1. To access the web service files, go to the release folder.
2. Copy and paste the AdServing API folder into the **www\lgv2\webservices** folder on the server.
3. To let the web service know where the master.config file is, open the user.config file located in the **AdServing API** folder.
4. Under **add key="MasterConfigDir"**, enter the path to the master.config file as value.
5. Save your changes and close the config file.

Example: configuration of the user.config file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<appSettings>
<add key="MasterConfigDir" value="F:\Grassfish\devfast.grassfish.tv\master.config"/>
</appSettings>
```

Configure the AdServing API

Note

Starting with IXM One version 12.0, you can configure these settings directly in IXM One without having to edit the configuration files. [Learn more...](#)

Follow the instructions for configuring the web service depending on your AdServing API version:

Configure AdServing API earlier than version 1.1.0

To use the web service, you must configure it in the master.config file of the server. Before you do so, consider the following requirements:

- You must configure the master.config file for all individual customers.
- You must configure a valid **Security:SymmetricSecurityKey** as shown in the example below.

To configure the AdServing API, perform the following steps:

1. Open the master.config file of the server.
2. Configure the following settings:
 - **SSP_URL**: Specify the URL of the SSP. That is, the external booking platform.
 - **SpotGroupName**: Specify the name of the spot group to which the media files downloaded from the SSP are uploaded as spots.
 - **ServiceUserName**: Specify the name of the user who uploads the spots and adds them to the playlist.
 - **PlaylistName**: Specify the name of the playlist to which the media files downloaded from the SSP are added as spots.
 - **ExtendSpotValidityInDays**: Specify how long uploaded spots are valid in days. For example, enter 0 if the spot is valid indefinitely or enter 1 if the spot is only valid today. The default value is 0.
 - **MaxLogLevel_File**: Optionally, enter the maximum log level to specify which kind of information you want to log once for all customers:
 - 1 for Error
 - 2 for Warning
 - 3 for Info
 - 4 for Debug
 - 5 for Trace

Note on logging

If you need additional Microsoft logs for debugging or further insight, you can optionally set logging to "Warning", but we recommend setting it to "Critical" in general to avoid performance issues.

Example: configuration of the master.config file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<configuration>
  <appSettings type="global">
    <add key="Security:SymmetricSecurityKey" value="INSERT-A-KEY-HERE" />
  </appSettings>
  <appSettings type="AdServing">
    <add key="AdServing:Customer:dev:SSP_URL" value="http://ads.ooh.digital/adserver.php?nid=S2AD&pid=" />
    <add key="AdServing:Customer:dev:PlaylistName" value="SSP"/>
    <add key="AdServing:Customer:dev:SpotGroupName" value="SSP"/>
    <add key="AdServing:Customer:dev:ServiceUserName" value="admin"/>
    <add key="AdServing:Customer:dev:ExtendSpotValidityInDays" value="1"/>
    <add key="MaxLogLevel_File" value="3"/>
  </appSettings>
</configuration>
```

```
</appSettings>  
</configuration>
```

Note

Note that you must restart the application pool that contains the AdServing API every time you change its settings in the master.config file. You'll learn how to create the application pool in chapter [Configure the IIS web server](#).



Configure AdServing API version 1.1.0 or later

To use the AdServing API web service, you must configure the master.config file of the server and a customer specific SSP config file.

To do so, perform the following steps:

1. Open the master.config file of the server.
2. Go to the setting **AdServing:Customer:<CUSTOMER-ID>:SSP_CONFIG** and specify the path to the customer specific SSP config file.
3. Optionally, specify the following general settings in the master.config file:
 - **AdServing:DownloadWorkerIntervalMs**: Specify the interval for checking for new media files that must be downloaded and imported into IXM One in milliseconds. The default value is 1000 ms. If you specify less than 0 milliseconds, the AdServing API stops working.
 - **MaxLogLevel_Console**: Enter the maximum console log level to specify which kind of information you want to log:
 - 1 for Error
 - 2 for Warning
 - 3 for Info
 - 4 for Debug
 - 5 for Trace
 - **MaxLogLevel_File**: Enter the maximum file log level to specify which kind of information you want to log:
 - 1 for Error
 - 2 for Warning
 - 3 for Info
 - 4 for Debug
 - 5 for Trace

Note on logging

If you need additional Microsoft logs for debugging or further insight, you can optionally set logging to Warning, but we recommend setting it to Critical in general to avoid performance issues.

Example: configuration of the master.config file

```
<appSettings type="AdServing">
  <add key="AdServing:Customer:27:SSP_CONFIG"
    value="\\gfstor002.grassfish.local\GFServers\Server\01\configs\ssp-config-27.json"/>
  <add key="AdServing:DownloadWorkerIntervalMs" value="1500"/>
  <add key="MaxLogLevel_Console" value="5"/>
  <add key="MaxLogLevel_File" value = "5" />
</appSettings>
```

Note

Note that you must restart the application pool that contains the AdServing API every time you change its settings in the *master.config* file. You'll learn how to create the application pool in chapter [Configure the IIS web server](#).

Configure SSP parameters

You must configure the SSP parameters like the SSP's URL in the SSP config file. The file has the following structure:

```
{
  "sspConfigurations": [
    "sspConfiguration": {
      "sspPriority": int,
      "sspUrl": string,
      "cutMediaUrlBeforeSubString": string,
      "sspRequestHeaders": Dictionary<string, string>,
      "sspCategoryId": int,
      "sspPlaylistId": int,
      "sspSpotGroupId": int,
      "sspServiceUserId": int,
      "extendSpotValidityInDays": int
    }
  ]
}
```

Note

The file must contain at least one SSP config object that specifies the URL, IDs, etc. For multiple SSPs, you need to create multiple SSP config objects.

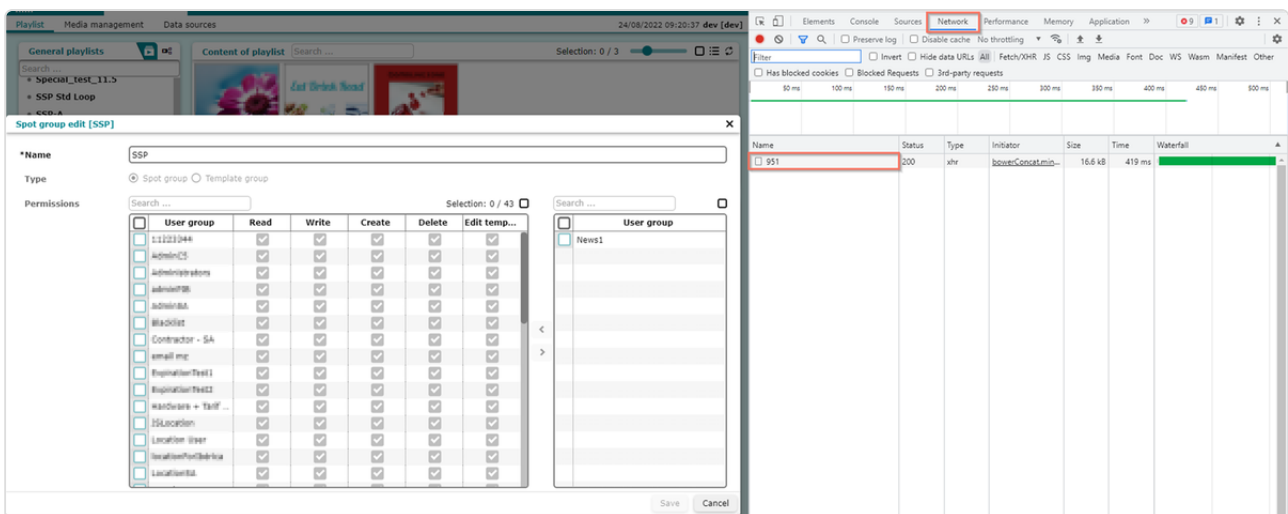
Get the IDs

Before you can configure the SSP config file, you need to get the IDs of IXM One category, playlist, spot group, and SSP user for each SSP. You'll enter these IDs in the file.

When the system uses IDs instead of names, you can simply change the names of playlists or spot groups in IXM One without affecting the ID and SSP. This also means that the same SSP admin user works for multiple SSP providers.

To get the IDs, perform the following steps:

1. In IXM One, open the editing window of the category, playlist, spot group, or user whose ID you need.
2. Open the Developer Tools in your browser. For example, in Google Chrome go to **Customize and control Google Chrome > More Tools > Developer Tools**.
3. In the Developer Tools window, select the **Network** tab.
4. Read the ID from the list and write it down for the file configuration in the next chapter.
5. Repeat steps 1 through 4 until you have the required IDs of the category, playlist, spot group, and user.



Ensure that your configuration is valid

Your configuration is validated with the help of validation rules. If one or more of these validation rules are violated, the AdServing API logs an error message and stops.

The following rules apply:

- Multiple **sspConfiguration** elements within one file must not contain the same values for: **SspPriority**, **SspUrl**, and **SspCategoryId**. Ensure that these elements have different values in the config file.
- None of the specified configuration parameters must be missing. This excludes the optional parameters **CutMediaUrlBeforeSubString**, **SspRequestHeaders**, and **ExtendSpotValidityInDays**.
- All specified configuration parameters must have a value.

- None of the specified configuration parameters must have a value that doesn't match the specified data type.
- The basic syntax of the JSON must be valid.

Configure the config file

To configure the SSP parameters in the SSP config file, perform the following steps:

1. Open the SSP config file.
2. Specify the following settings:

Setting	Data type	Default value	Description
SspPriority	int	0	Specify the priority of the SSP. The lower the value, the higher the priority.
SspUrl	string	Mandatory setting (no default value)	Specify the URL of the SSP. Optionally, you can add a query string. Example: <i>https://my.ssp.com/?param=value</i>
CutMediaUrl BeforeSubString	string	None	Enable this setting to cut the media URL before the specified substring. This means that only the substring and everything after it is used. This is necessary if all requests use the same media URL.
SspRequestHeaders	Dictionary <key, value>	None	Optionally, configure custom request headers. This can be useful, for example, for authentication.
SspCategoryId	int	Mandatory setting (no default value)	Specify the ID of IXM One category. If a player requests a new category, the SSP config are used for the category. IXM One category with ID 1 is the default.

SspPlaylistId	int	Mandatory setting (no default value)	Specify the ID of the playlist. The system assigns default values for the following settings: General and screen player settings If the ID belongs to a playlist: If the ID belongs to a playlist:
SspSpotGroupId	int	Mandatory setting (no default value)	Specify the ID of the spot group. Downloaded SSP content:
SspServiceUserId	int	Mandatory setting (no default value)	Specify the ID of the user. The specified user is used:
ExtendSpotValidityInDays	int	0	Specify how long a spot is valid. If you set a value greater than 0: This is based on the time:

Example: configuration of the SSP config file

```
{
  "SspConfigurations": [
    {
      "SspPriority": 1,
      "SspUrl": "https://test.grassfish.tv/testdata/ssp/units/",
      "CutMediaUrlBeforeSubString": "?",
      "SspRequestHeaders": {
        "hs-auth": " api-key 1234567890"
      },
      "SspCategoryId": 1593,
      "SspPlaylistId": 259,
      "SspSpotGroupId": 958,
      "ExtendSpotValidityInDays": 3,
      "SspServiceUserId": 1489
    },
    {
      "SspPriority": 2,
      "SspUrl": " https://test.grassfish.tv/random/ssp/schedule/",
      "CutMediaUrlBeforeSubString": "?",
      "SspCategoryId": 1592,
      "SspPlaylistId": 263,
      "SspSpotGroupId": 957,
      "ExtendSpotValidityInDays": 3,
      "SspServiceUserId": 1489
    }
  ]
}
```

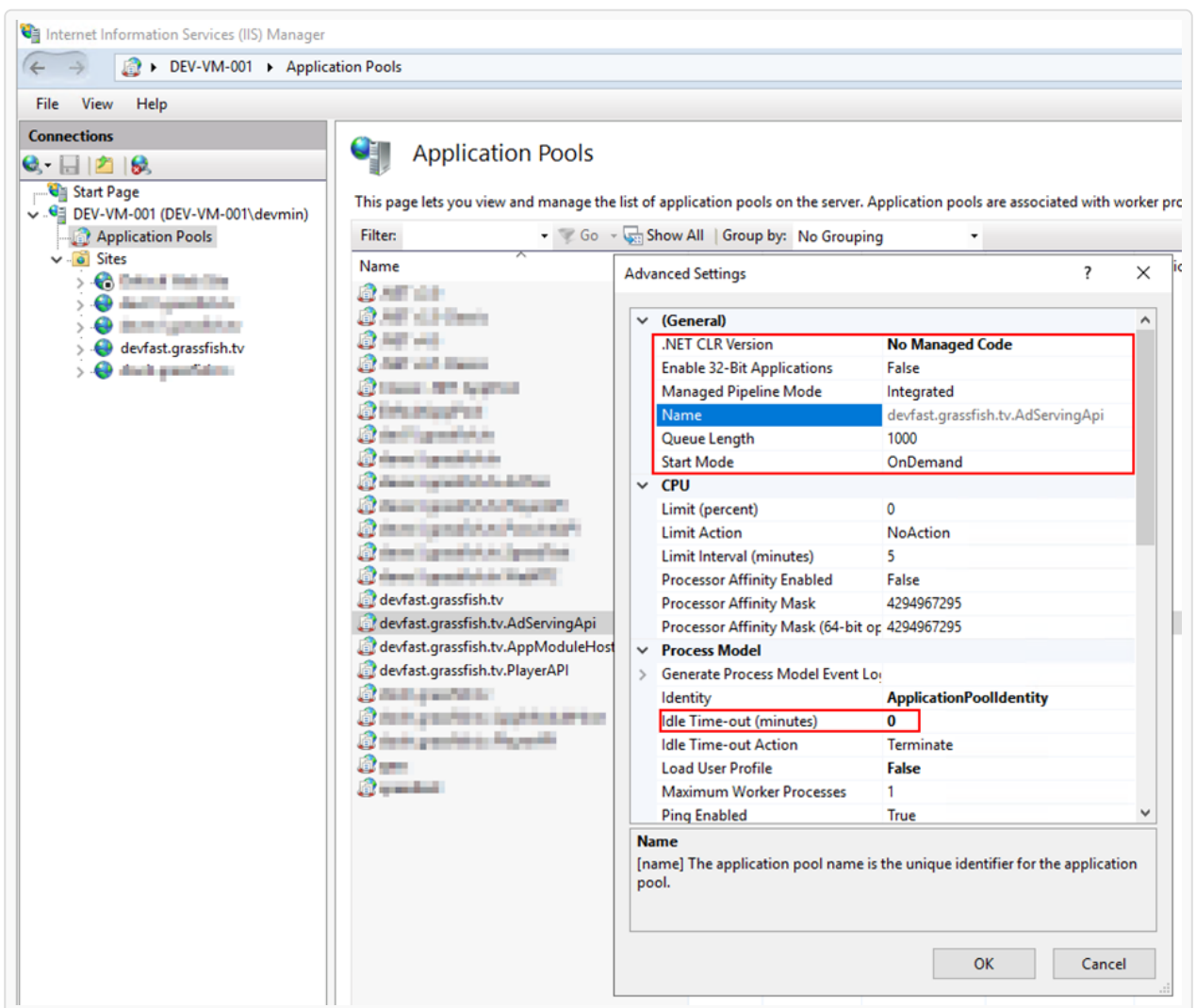
```
}
```

```
]
```

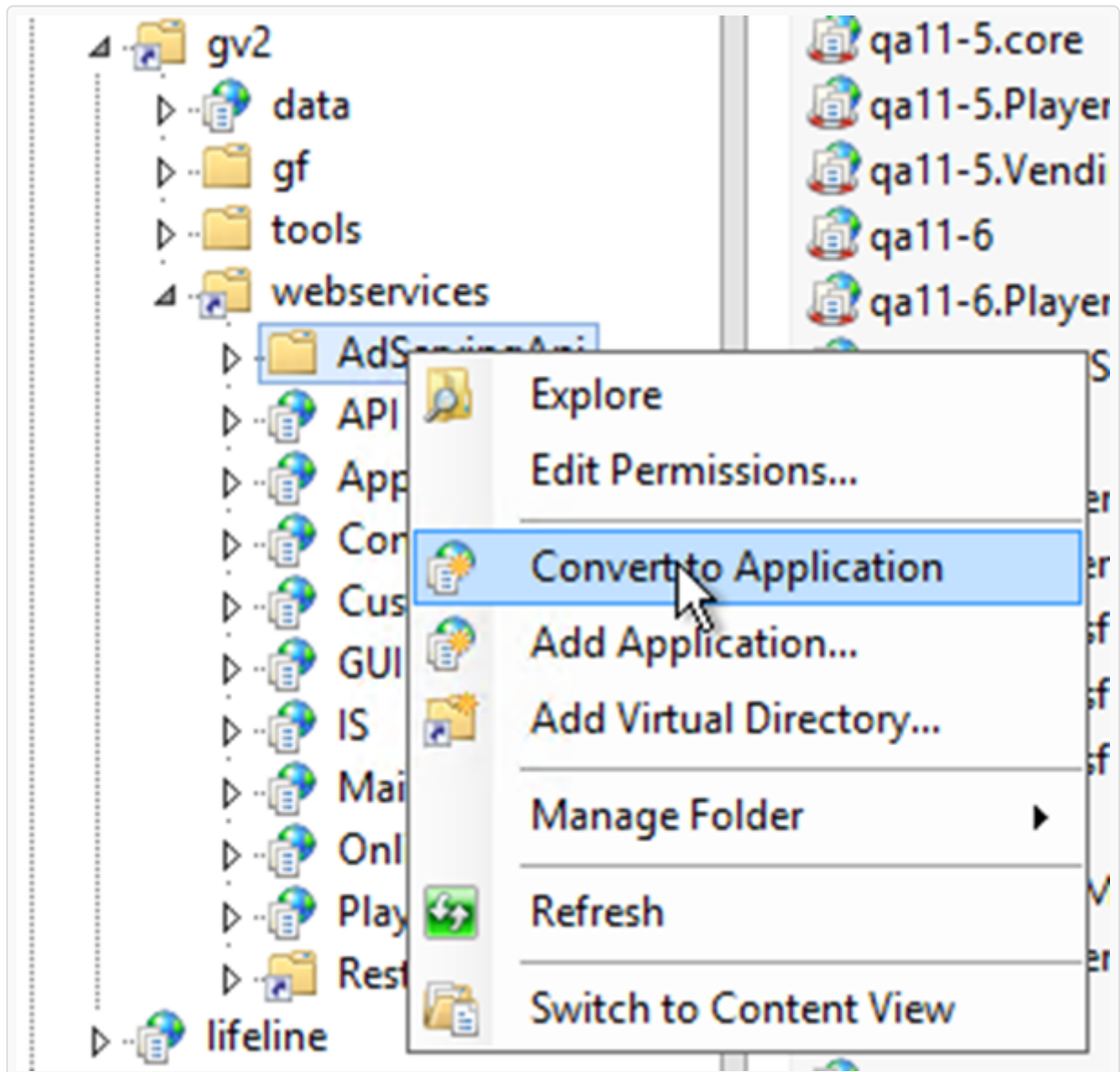
Configure the IIS web server

You must configure the Internet Information Services (IIS) web server for the Grassfish web service. You'll create a new application pool for the AdServing API which allows you to restart it independently in the future. To do so, perform the following steps:

1. To create a new application pool, open the IIS Manager.
2. In the IIS Manager, right-click on **Application Pools** and select **Add Application Pool**.
3. Specify the application pool details as required and click **OK**.
4. Right-click on the application pool and select **Advanced Settings**.
5. Set the **Idle Time-out** to **0** minutes.



6. Right-click on the web service folder and select **Convert to Application**.



7. Open the **Advanced Settings** of the web service and assign it to the new application pool.

8. Save your changes and close the IIS Manager.

Install the UDC importer

Note

You need the UDC importer if you use a DOOH add-on version earlier than 2.5.1 and if you use the Qt Player. If you use add-on version 2.5.1 or later or a different player, you can skip this chapter.

The DOOHPlaybackImporter UDC plugin transmits booked spot instances to the Qt Player. Note that the importer only does so for players that have been online.

Note

For DOOHPlaybackImporter version 2.6 or earlier you must install the file *Dapper.dll* in addition to the *DoohPlaybackImporter.dll*. Copy and paste both files from the release folder to `\GVServer2\UdcService\Plugins` on the server and restart the UDC service.

Activate the UDC importer

To activate the UDC importer on the server, perform the following steps:

1. Log in to the server admin account of IXM One.
2. Go to **Administration > Global > Customer management**.
3. Double-click on the customer.
4. In the **Edit client** window, switch to the **Data importer** tab.
5. Select **Assigned** for the DOOHPlaybackImporter.
6. Click **Save** to save your changes.

Edit client

General

Search ...

Number of items: 3

Data importer

Name	Imported	Version	Assigned	Share
DOOH Playback Importer	12/02/2020 1...	2.6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ExchangeCalendarImpor...	17/02/2020 0...	0.11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Generic Foreca Weather...	27/01/2023 1...	1.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Configure the UDC importer

You must configure the UDC importer with the configuration file that is located on the server. To do so, perform the following steps:

1. Open the DoohUdcImporterConfig.json file.
2. Change the following settings according to your requirements:

Setting	Default value	Description
BufferPercentage	10	Specify the percentage
OfflineIntervalMin	30	Specify the maximum interval If a player is offline
OfflineIntervalMinSpecialDistributions	4320 (=3 days)	Specify the maximum interval
LogDbQueryResultForAllSiBoxes	false	Set to true to log queries
ExtendedLogging	false	Set to true to log extended data

3. Save your changes and close the config file.

Example: configuration of the DoohUdcImporterConfig.json file

```
{
  "BufferPercentage": 10,
  "OfflineIntervalMin": 360
}
```

API means application programming interface. An API consists of multiple definitions and protocols for developing and integrating application software. An API is an interface that allows independent applications to communicate with each other and exchange data.

UDC stands for Universal Data Controller. The UDC importer imports external data into the IXM system. UDC plugins specify which data should be imported and how it should be transformed for use by spots and players.