Configure linked lists

URL: https://docs.grassfish.com/grassfish/docs/create-linked-lists

Archiviert am: 2025-07-17 18:39:28

As a spot developer, you can configure a spot with linked lists that are dynamically populated based on the values of other lists. This means dependent dropdown lists, where selecting items from one list affects the available options in another. The lists are dynamically populated with information from a webservice. All you need is the webservice's URL.

> ***Note***
>
> *Learn more about creating dropdown lists in HTML Wizard spots.*

Best practices

Before you configure your linked lists, note the following recommendations:

- Specify the dependencies of each linked list correctly, so it can fetch the right data.
- Verify that the API endpoints (URLs) return the expected data format.
- Initialize linked list values to zero and handle cases where dependent data may not be available.

Configure linked lists

To create linked lists, perform the following steps:

1. Specify the spot element with the following parameters:

Setting	Description
Id	Specify the ID of the element. The ID must be unique. Example: **marketLinkedList**
DataType	Specify the type of data that the element contains. Example: **linkedList**

DisplayName	Specify the display name of the element. This name appears in the HTML Wizard. Example: **Market**

2. Configure each list in the element with the following parameters:

Setting	Description
Id	Specify the ID of the list. The ID must be unique. Example: **market**
DisplayName	Specify the display name of the list. This name appears in the HTML Wizard. Example: **Market**
Url	Specify the URL of the webservice (API endpoint) that provides data for the list items. Example: **https://example.com/markets**
Dependencies	Specify the IDs of other lists that this list depends on, separated by commas. Example: **["market", "modelGroup"]**
Response	Use the response mapping to extract the correct data from the API response. Example
Entry	Specify the entry point in the API response where the list items are located. Example
Mapping	Specify the following settings to map the response data to the list item properties: ◦ **Id**: field name for the item's ID. ◦ **Value**: field name for the item's name. ◦ **Name**: field name for the item's display name. Example

Value	Initialize the value field to hold the selected values for each linked list. Optionally, use **null**.
Options	Set additional options like **Required**.

Example configuration

Each spot configuration is specified in a JSON structure. The following example is a configuration snippet with linked lists:

```
{
 "Document": {
   "Elements": [
     {
       "Id": "marketLinkedList",
       "DataType": "linkedList",
       "DisplayName": "Market",
       "Lists": [
         {
           "Id": "market",
           "DisplayName": "Market",
           "Url": "https://example.com/markets",
           "Dependencies": [],
           "Response": {
             "Entry": "",
             "Mapping": {
               "Id": "market",
               "Value": "market",
               "Name": "market"
             }
           }
         }
       ],
       "Value": {
         "market": null
       },
       "Options": {
         "Required": true
       }
     },
     {
       "Id": "modelRange",
       "DataType": "linkedList",
       "DisplayName": "Model Range",
       "Lists": [
         {
           "Id": "modelGroup",
           "DisplayName": "Model Group",
           "Url": "https://example.com/markets/{market}",
           "Dependencies": ["market"],
           "Response": {
             "Entry": "ranges",
```

```
              "Mapping": {
                "Id": "id",
                "Value": "name",
                "Name": "name"
              }
            }
          },
          {
            "Id": "subModelGroup",
            "DisplayName": "Sub Model Group",
            "Url": "https://example.com/markets/{market}/ranges/{modelGroup}",
            "Dependencies": ["market", "modelGroup"],
            "Response": {
              "Entry": "series",
              "Mapping": {
                "Id": "id",
                "Value": "name",
                "Name": "name"
              }
            }
          }
        ],
        "Value": {
          "modelGroup": null,
          "subModelGroup": null
        },
        "Options": {
          "Required": true
        }
      }
    ]
  }
}
```

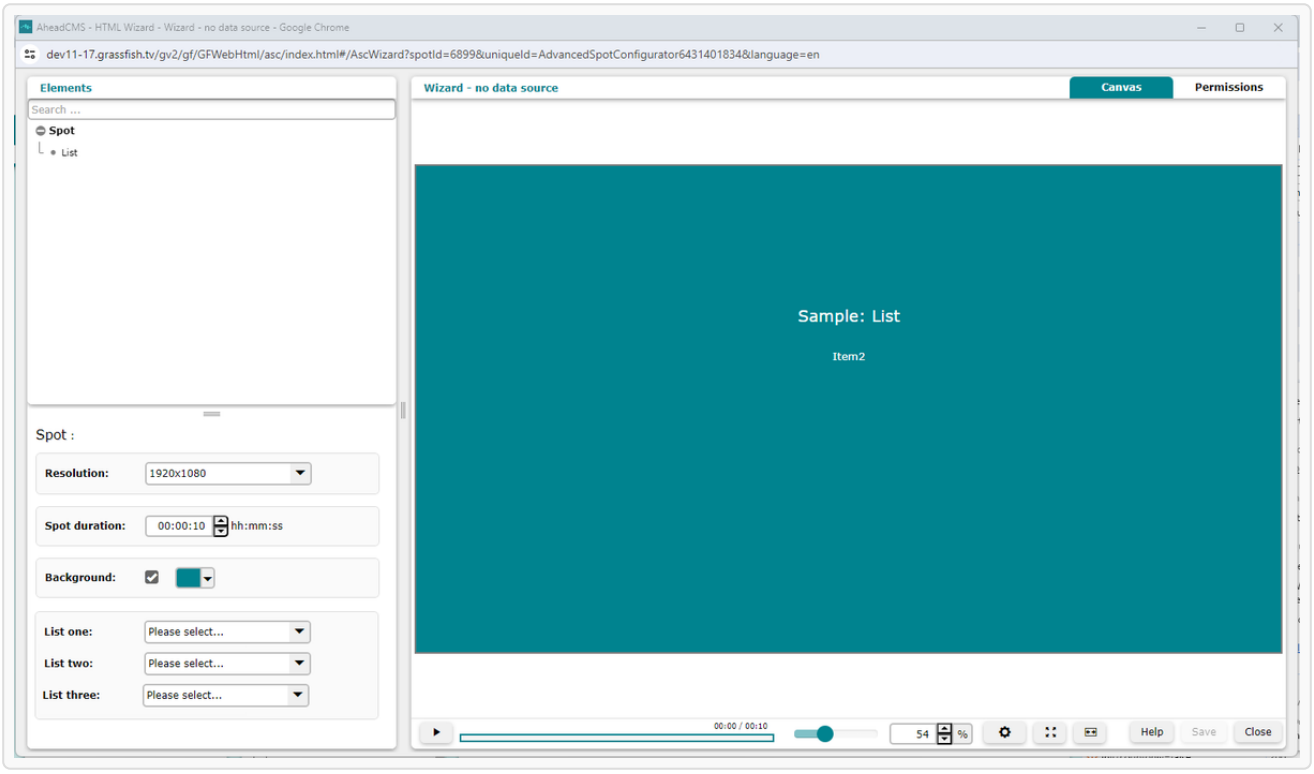
What happens in the example?

The example configures the following elements and linked lists:

- The element **marketLinkedList** contains one linked list:
  - The linked list is called **market**.
  - It fetches its items from `https://example.com/markets`.
  - Since there are no dependencies, it can retrieve data independently from other lists.
- The element **modelRange** contains two linked lists:
  - The first linked list is called **modelGroup**. It depends on the list **market** to fetch its items.
  - The second linked list is called **subModelGroup**. It depends on the lists **market** and **modelGroup** to fetch its items.

Example display

In the HTML Wizard, list lements are displayed as follows:


AheadCMS - HTML Wizard - Wizard - no data source - Google Chrome
dev11-17.grassfish.tv/gv2/gf/GFWebHtml/asc/index.html#/AscWizard?spotId=6899&uniqueId=AdvancedSpotConfigurator6431401834&language=en
Elements
Search ...
Spot
List
Wizard - no data source
Canvas
Permissions
Sample: List
Item2
Spot :
Resolution:
1920x1080
Spot duration:
00:00:10
hh:mm:ss
Background:
List one:
Please select...
List two:
Please select...
List three:
Please select...
00:00 / 00:10
54
%
Help
Save
Close