HCL Digital Experience

HDX-DEV-100

Digital Data Connector for Beginners Lab

Edition August 2023

Giancarlo Giannini - giancarlo.giannini@hcl.com

Herbert Hilhorst - Herbert.hilhorst@hcl.com

Tim Golledge - tim.golledge@hcl.com

Table of Contents

Author(s)

This document was created by the following Subject Matter Experts:

Herbert Hilhorst
Company: HCL

Bio

Herbert Hilhorst is an HCL Digital Experience (DX) Technical Advisor at HCL.

Contact: herbert.hilhorst@hcl.com

Tim Golledge
Company: HCL

Bio

Tim Golledge is an HCL Digital Experience (DX) Technical Advisor at HCL.

Contact: tim.golledge@hcl.com

Giancarlo Giannini
Company: HCL

Bio

Giancarlo Giannini is an HCL Digital Experience (DX) Technical Advisor at HCL.

Contact: giancarlo.giannini@hcl.com

Introduction

This hands-on lab guides you, as a Developer, in understanding how to use Digital Data Connector (DDC for short) to manage JSON feeds that come from external REST APIs into HCL Digital Experience (DX) platform in a nice configurable way using Web Content. You will use the example of https://www.weatherapi.com and use their REST APIs with JSON results. This is a common standard to integrate any external data. Digital Data Connector may also work with XML and any other kind of format. This will be part of more advanced labs.

In this DX developer lab, you play the role of Gene, a developer for the fictitious Woodburn Studio company.

Gene Hayes, Developer, based in Chicago (USA)

As a Web Developer, you will first discover a weather report page component and learn to configure it. Then you will learn the Weather REST API that has been used for this. And then, you will learn how to build this customizable page component using Digital Data Connector, set up a global configuration to manage the host and default values and add user configuration to allow a business user to manage what and how to show easily. Optionally, you will learn how to authorize external access.

Prerequisites

1. Completion of “HDX-INTRO” and “HDX-BU-100” courses, as well as the previous lessons under “HDX-DEV-100” course, including the labs
2. Optional completion of “HDX-ADM-100” to have a better understanding of the security aspects
3. Your own account on https://www.weatherapi.com. Instructions are provided to get this.
4. Access to download the Lab Resources. In the same place where you have found this lab, you will find corresponding resources which you may download and unzip in your Desktop. This helps you to run the lab more easily, and you may later replace it by your own ones.

You will be using the following user IDs and passwords:

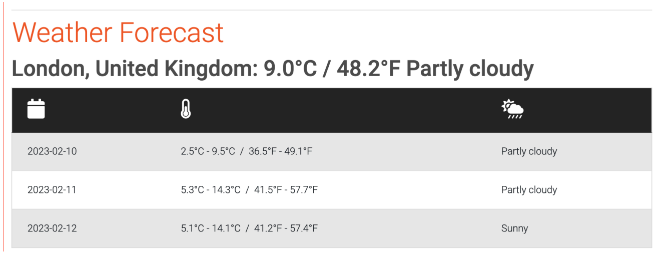
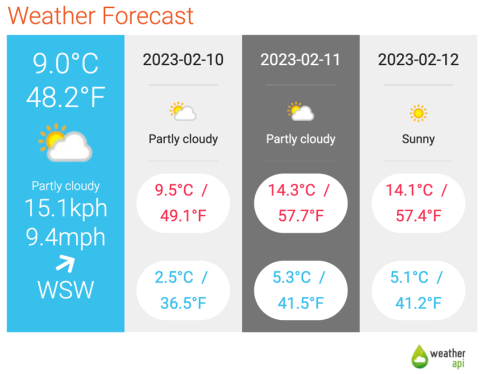
Purpose	User	Password
SoFy Login	Your official email id	Your password
SoFy Solution Console Login	sol-admin	Get it from SoFy
DX Login Developer (Gene Hayes)	ghayes	HCL-Dem0
DX Solution Console Login Administrator	wpsadmin	wpsadmin

Lab Overview

In this lab, you will discover how to use HCL Digital Data Connector (DDC) as integration tool for any external data source that exposes its data through a set of XML/JSON Rest API.

To learn how to use DDC, you will work learn using it with public REST APIs from **Weather API**: https://www.weatherapi.com. WeatherAPI.com provides access to free weather and geo data via a JSON/XML restful API. It allows developers to create desktop, web and mobile applications using this data very easy.

The lab scenario is to create a Weather Forecast application (Page Component) powered by these APIs. The Business User will be able to configure location, number of forecast days and select from multiple presentations. The final output of this lab will look as follows:

Table UI	Card UI
	

Weather Forecast

London, United Kingdom: 9.0°C / 48.2°F Partly cloudy

2023-02-10	2.5°C - 9.5°C / 36.5°F - 49.1°F	Partly cloudy
2023-02-11	5.3°C - 14.3°C / 41.5°F - 57.7°F	Partly cloudy
2023-02-12	5.1°C - 14.1°C / 41.2°F - 57.4°F	Sunny

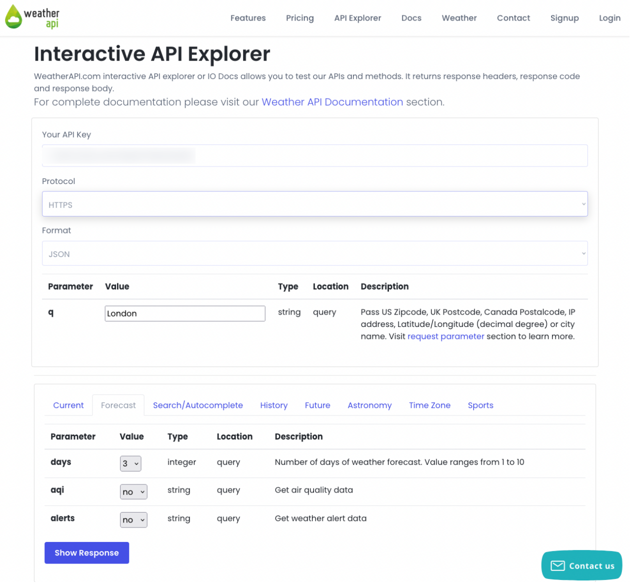
Weather Forecast

9.0°C 48.2°F	2023-02-10	2023-02-11	2023-02-12
Partly cloudy	Partly cloudy	Partly cloudy	Sunny
15.1kph 9.4mph	9.5°C / 49.1°F	14.3°C / 57.7°F	14.1°C / 57.4°F
↗ WSW	2.5°C / 36.5°F	5.3°C / 41.5°F	5.1°C / 41.2°F

In this lab, there are five main and additional optional parts. These are shortly introduced now.

Part 1: Discover Weather APIs

You will discover the Weather REST APIs and test them. This will help you understand how they work and how you may use them in your site.



Interactive API Explorer

WeatherAPI.com interactive API explorer or IO Docs allows you to test our APIs and methods. It returns response headers, response code and response body.

For complete documentation please visit our Weather API Documentation section.

Your API Key

[redacted]

Protocol

HTTPS

Format

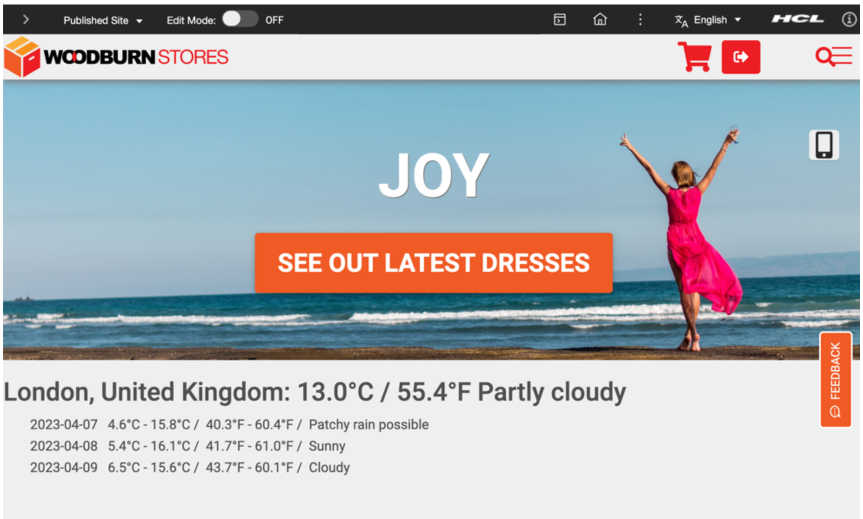
JSON

Parameter	Value	Type	Location	Description
q	London	string	query	Pass US Zipcode, UK Postcode, Canada Postalcode, IP address, Latitude/Longitude (decimal degree) or city name. Visit request parameter section to learn more.

Parameter	Value	Type	Location	Description
days	3	integer	query	Number of days of weather forecast. Value ranges from 1 to 10
aqi	no	string	query	Get air quality data
alerts	no	string	query	Get weather alert data

Part 2: Setup your first DDC integration

Then, you create a new London Weather page component to add to an existing Joy page that shows the current weather and three forecast days. For this, you will create a WCM library with all the required artifacts to use the Weather API in HCL DX and to create a List Definition content item with the right configuration.

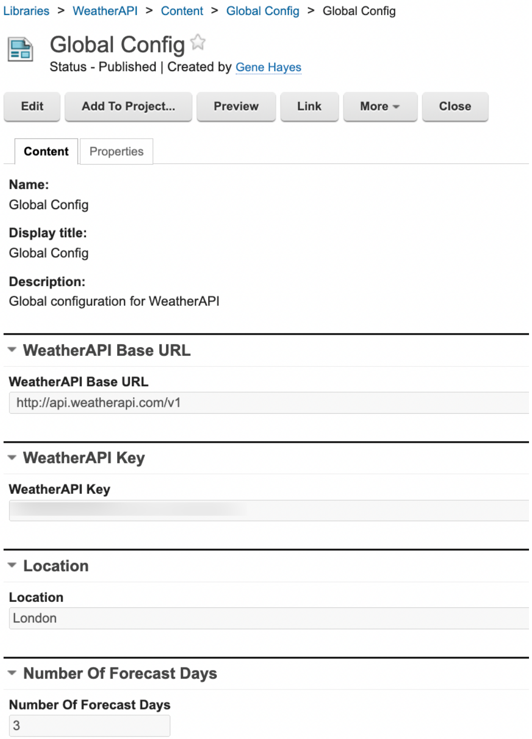


London, United Kingdom: 13.0°C / 55.4°F Partly cloudy

2023-04-07 4.6°C - 15.8°C / 40.3°F - 60.4°F / Patchy rain possible
2023-04-08 5.4°C - 16.1°C / 41.7°F - 61.0°F / Sunny
2023-04-09 6.5°C - 15.6°C / 43.7°F - 60.1°F / Cloudy

Part 3: Add Global Configuration

In this part, you will add a Global Configuration to manage some Global Parameters, like the Weather API host URL, the Weather API Key (required to work with that), some default values, for example city, N° of forecast days, language, etc. You will create a global config content template and a new content, and you will test your application again by using these "global" parameters.


Libraries > WeatherAPI > Content > Global Config > Global Config
Edit
Add To Project...
Preview
Link
More
Close
Content
Properties

Global Config

Status - Published | Created by Gene Hayes

Name:
Global Config

Display title:
Global Config

Description:
Global configuration for WeatherAPI

WeatherAPI Base URL

WeatherAPI Base URL: http://api.weatherapi.com/v1

WeatherAPI Key

WeatherAPI Key: [redacted]

Location

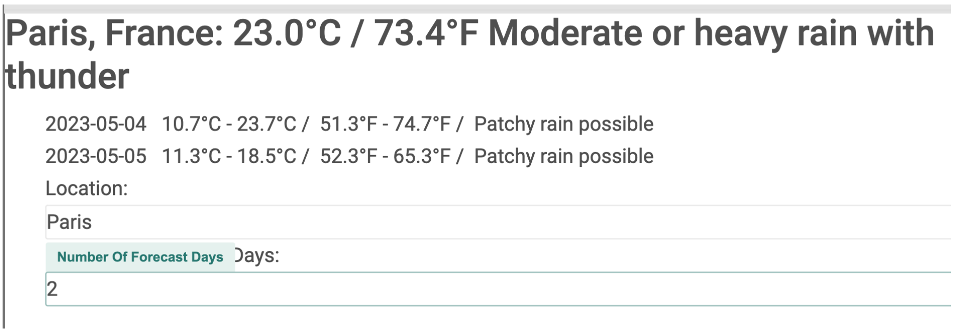
Location: London

Number Of Forecast Days

Number Of Forecast Days: 3

Part 4: Add User Configuration

Then you want your business user to control what to show by adding a configuration of the location and the number of forecast days that override the default values from the Global Parameters. You will add this to your List Definition content template, make it editable by them in the corresponding presentation template, update your content and test it.



Paris, France: 23.0°C / 73.4°F Moderate or heavy rain with thunder

2023-05-04 10.7°C - 23.7°C / 51.3°F - 74.7°F / Patchy rain possible

2023-05-05 11.3°C - 18.5°C / 52.3°F - 65.3°F / Patchy rain possible

Location:

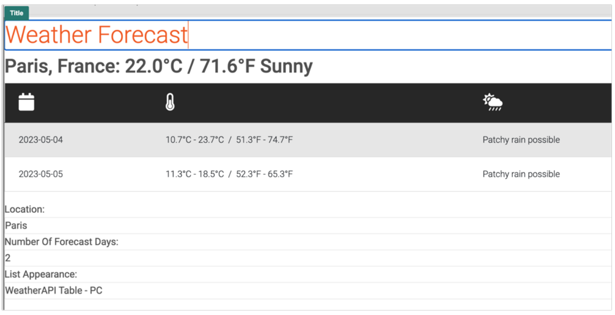
Paris

Number Of Forecast Days:

2

Part 5: Add List Appearance Options

And finally, you will add different ways to show the weather data and allow a business user to select from them. You will create a couple of variations of the list presentation, managed by Personalization Components, with a table and card UI, and then you will extend the List Definition content to allow the user to select between these variations.



Weather Forecast

Paris, France: 22.0°C / 71.6°F Sunny

2023-05-04	10.7°C - 23.7°C / 51.3°F - 74.7°F	Patchy rain possible
2023-05-05	11.3°C - 18.5°C / 52.3°F - 65.3°F	Patchy rain possible

Location:

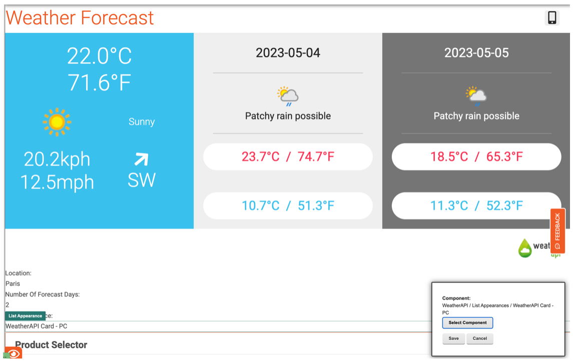
Paris

Number Of Forecast Days:

2

List Appearance:

WeatherAPI Table - PC



Weather Forecast

22.0°C 71.6°F Sunny 20.2kph 12.5mph SW

2023-05-04 Patchy rain possible 23.7°C / 74.7°F 10.7°C / 51.3°F

2023-05-05 Patchy rain possible 18.5°C / 65.3°F 11.3°C / 52.3°F

Location:

Paris

Number Of Forecast Days:

2

List Appearance:

WeatherAPI Card - PC

Component:
WeatherAPI / List Appearances / WeatherAPI Card - PC

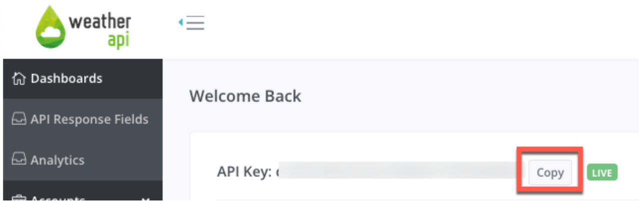
Optional Part 6: Authorize External Access

While this is typically an administrator task, you may optionally learn how to setup the signer certificate to allow use https and the HTTP Outbound Connection (AJAX Proxy) for the Weather API to overcome the CORS specification to authorize access of the external APIs.

Part 1: Discover the Weather APIs

You will discover the Weather REST APIs and test them. This will help you understand how they work and how you may use them in your site.

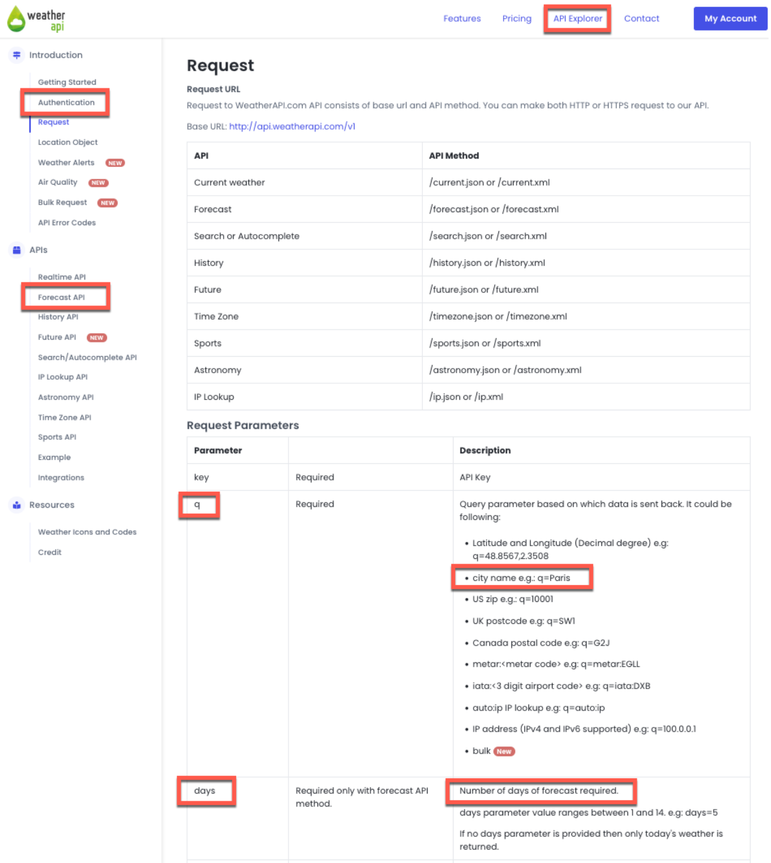
1. To start working with the Weather APIs you must register & sign-up for them. You can do this using this link: https://www.weatherapi.com/signup.aspx. This gives you access to a personal Weather APIs Key, required to work with these APIs, and access to the APIs documentation and "Sandbox" (API Explorer) to understand how to use and work with the APIs. Once done, copy your WeatherAPI **API Key**. Click **Copy**.



Welcome Back

API Key: [redacted]

2. Then go to the Weather API documentation https://www.weatherapi.com/docs/ to learn how to work with the Weather APIs. You want to be able to show the forecast weather data for a specific location, so go to the **APIs – Forecast API page**. See that the location is managed with a **q** parameter and the number of forecast days by a **forecastday** parameter. Then go the API Explorer to test it. Click **API Explorer**.



Request

Request URL

Request to WeatherAPI.com API consists of base url and API method. You can make both HTTP or HTTPS request to our API.

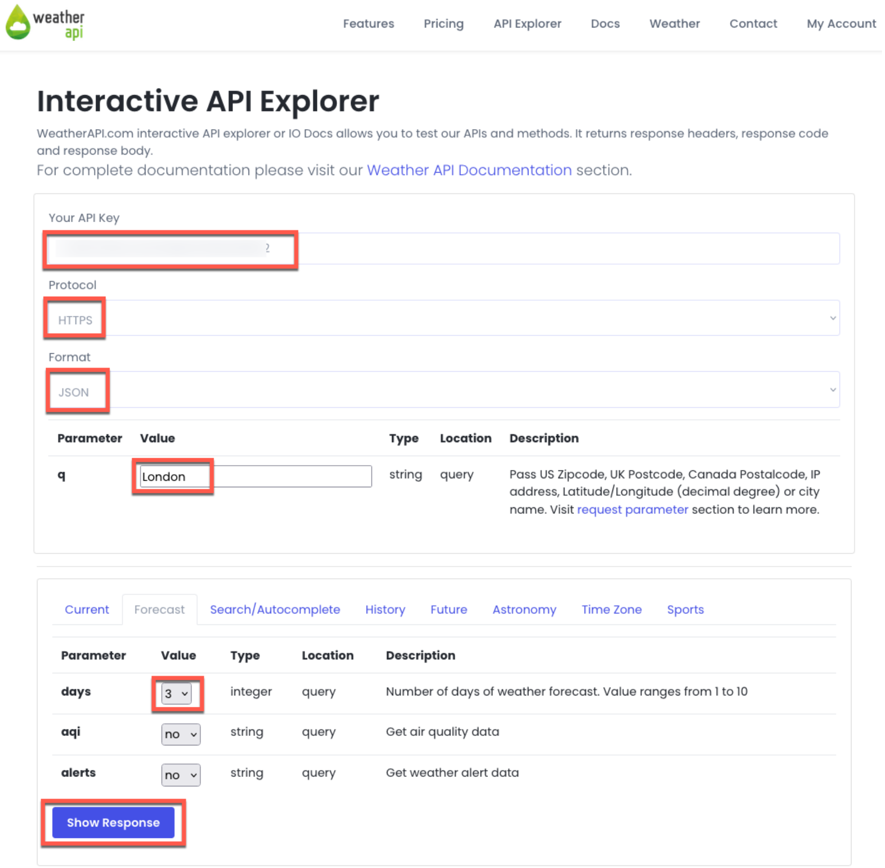
Base URL: http://api.weatherapi.com/v1

API	API Method
Current weather	/current.json or /current.xml
Forecast	/forecast.json or /forecast.xml
Search or Autocomplete	/search.json or /search.xml
History	/history.json or /history.xml
Future	/future.json or /future.xml
Time Zone	/timezone.json or /timezone.xml
Sports	/sports.json or /sports.xml
Astronomy	/astronomy.json or /astronomy.xml
IP Lookup	/ip.json or /ip.xml

Request Parameters

Parameter		Description
key	Required	API Key
q	Required	Query parameter based on which data is sent back. It could be following: • Latitude and Longitude (Decimal degree) e.g: q=48.8567,2.3508 • city name e.g.: q=Paris • US zip e.g.: q=10001 • UK postcode e.g: q=SW1 • Canada postal code e.g: q=G2J • metar:<metar code> e.g: q=metar:EGLL • iata:<3 digit airport code> e.g: q=iata:DXB • auto:ip IP lookup e.g: q=auto:ip • IP address (IPv4 and IPv6 supported) e.g: q=100.0.0.1 • bulk New
days	Required only with forecast API method.	Number of days of forecast required. days parameter value ranges between 1 and 14. e.g: days=5 If no days parameter is provided then only today's weather is returned.

3. Here you can easily test all the APIs. Test the forecast API. Click **Forecast**. Then enter your **API Key**, change the protocol to HTTPS, as your DX server is using https as well, select the format **JSON**, set the location parameter **q** to **London** in this lab (but feel free to change a different location), set the **days** parameter to **3** for 3 forecast days and keep the others to **no**. Then click **Show Response**.



Interactive API Explorer

WeatherAPI.com interactive API explorer or IO Docs allows you to test our APIs and methods. It returns response headers, response code and response body.

For complete documentation please visit our Weather API Documentation section.

Your API Key: [redacted]

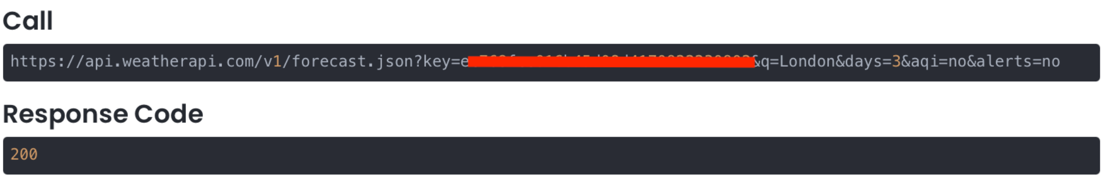
Protocol: HTTPS

Format: JSON

Parameter	Value	Type	Location	Description
q	London	string	query	Pass US Zipcode, UK Postcode, Canada Postalcode, IP address, Latitude/Longitude (decimal degree) or city name. Visit request parameter section to learn more.

Parameter	Value	Type	Location	Description
days	3	integer	query	Number of days of weather forecast. Value ranges from 1 to 10
aqi	no	string	query	Get air quality data
alerts	no	string	query	Get weather alert data

4. This gives you the response with on the head the full WeatherAPI forecast REST API URL that has this format *https://api.weatherapi.com/v1/forecast.json?key=<your-key>&q=<your-Location>&days=<your-N°-of-days>&aqi=no&alerts=no* and response code (HTTP 200 means a successful call):



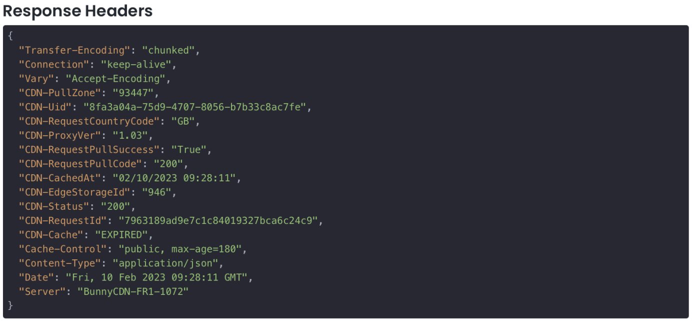
Call

```
https://api.weatherapi.com/v1/forecast.json?key=e[redacted]&q=London&days=3&aqi=no&alerts=no
```

Response Code

```
200
```

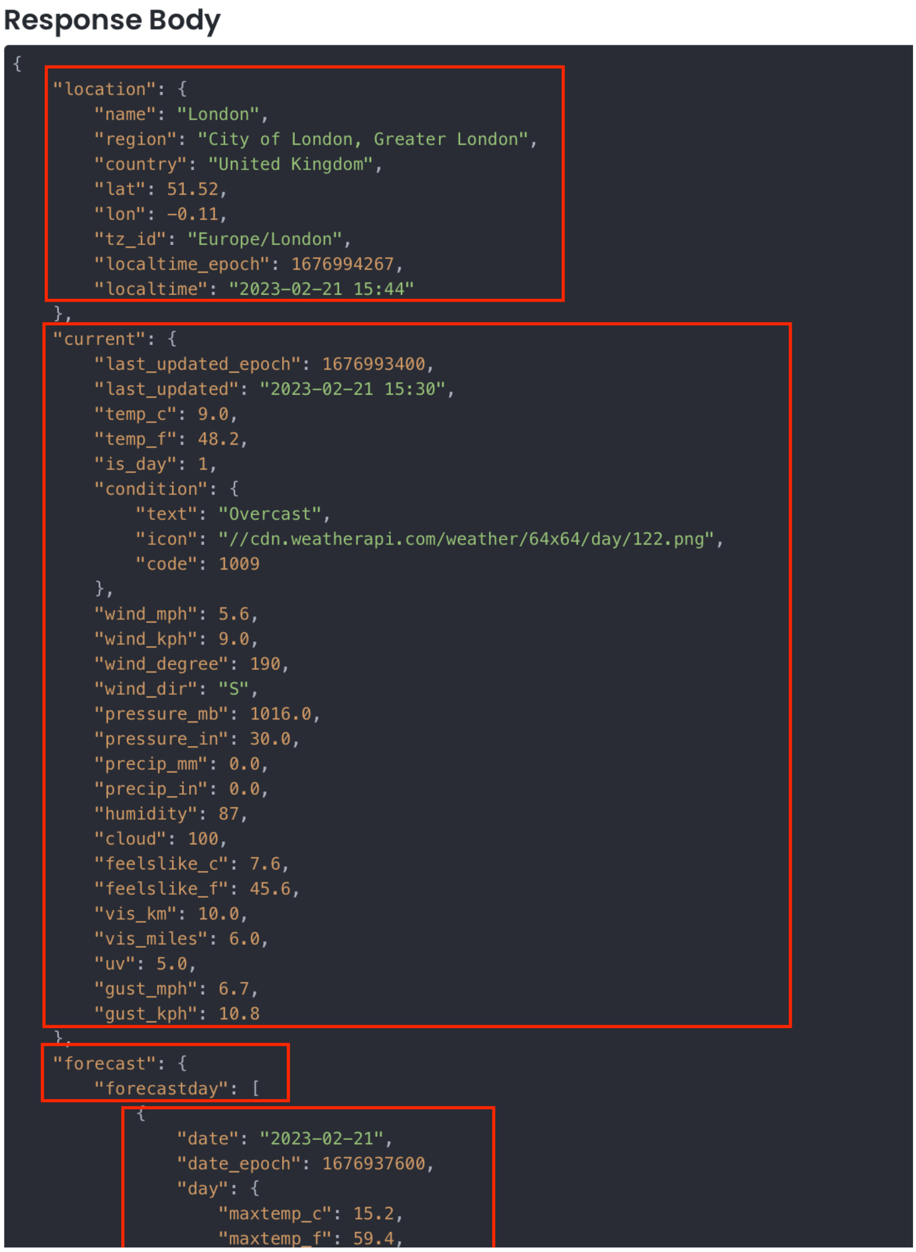
5. And below you will find the Response Header.



Response Headers

```
{
  "Transfer-Encoding": "chunked",
  "Connection": "keep-alive",
  "Vary": "Accept-Encoding",
  "CDN-PullZone": "93447",
  "CDN-Uid": "8fa3a04a-75d9-4707-8056-b7b33c8ac7fe",
  "CDN-RequestCountryCode": "GB",
  "CDN-ProxyVer": "1.03",
  "CDN-RequestPullSuccess": "True",
  "CDN-RequestPullCode": "200",
  "CDN-CachedAt": "02/10/2023 09:28:11",
  "CDN-EdgeStorageId": "946",
  "CDN-Status": "200",
  "CDN-RequestId": "7963189ad9e7c1c84019327bca6c24c9",
  "CDN-Cache": "EXPIRED",
  "Cache-Control": "public, max-age=180",
  "Content-Type": "application/json",
  "Date": "Fri, 10 Feb 2023 09:28:11 GMT",
  "Server": "BunnyCDN-FR1-1072"
}
```

6. And the most important part, the Response Body with details on the location, the current weather at that location, and then a forecast.forecastday list of forecast details for each day.



Response Body

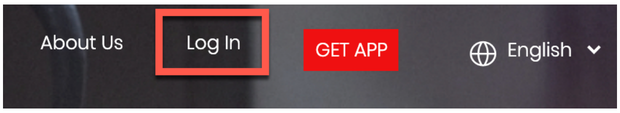
```
{
    "location": {
        "name": "London",
        "region": "City of London, Greater London",
        "country": "United Kingdom",
        "lat": 51.52,
        "lon": -0.11,
        "tz_id": "Europe/London",
        "localtime_epoch": 1676994267,
        "localtime": "2023-02-21 15:44"
    },
    "current": {
        "last_updated_epoch": 1676993400,
        "last_updated": "2023-02-21 15:30",
        "temp_c": 9.0,
        "temp_f": 48.2,
        "is_day": 1,
        "condition": {
            "text": "Overcast",
            "icon": "//cdn.weatherapi.com/weather/64x64/day/122.png",
            "code": 1009
        },
        "wind_mph": 5.6,
        "wind_kph": 9.0,
        "wind_degree": 190,
        "wind_dir": "S",
        "pressure_mb": 1016.0,
        "pressure_in": 30.0,
        "precip_mm": 0.0,
        "precip_in": 0.0,
        "humidity": 87,
        "cloud": 100,
        "feelslike_c": 7.6,
        "feelslike_f": 45.6,
        "vis_km": 10.0,
        "vis_miles": 6.0,
        "uv": 5.0,
        "gust_mph": 6.7,
        "gust_kph": 10.8
    },
    "forecast": {
        "forecastday": [
            {
                "date": "2023-02-21",
                "date_epoch": 1676937600,
                "day": {
                    "maxtemp_c": 15.2,
                    "maxtemp_f": 59.4,
```

You have successfully discovered the Weather APIs and are now ready to use them.

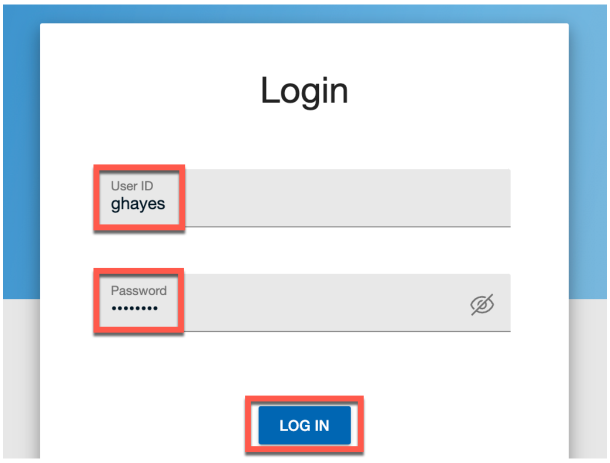
Part 2: Setup your first DDC integration

Then, you create a new London Weather page component to add to an existing Joy page that shows the current weather and three forecast days. For this, you will create a WCM library with all the required artifacts to use the Weather API in HCL DX and to create a List Definition content item with the right configuration.

1. First log on to the DX server as Gene Hayes. Click **Log in**.



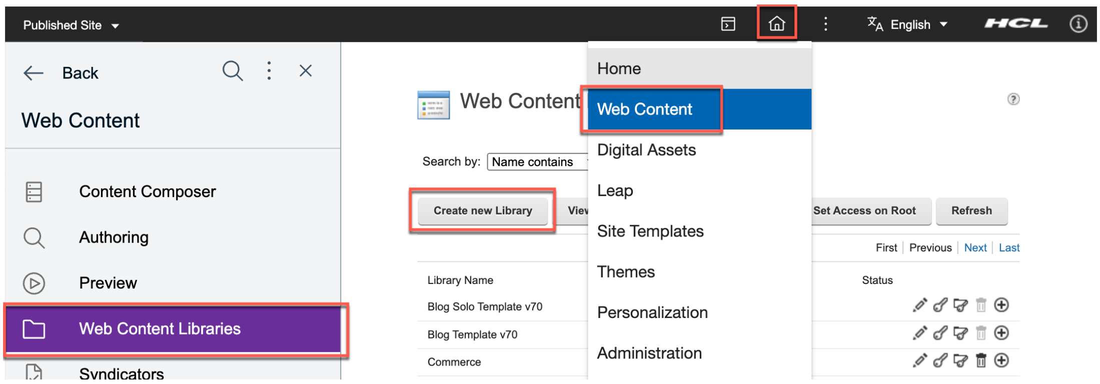
2. You will start creating a new WCM Library. First log in as Gene (User ID: **ghayes**, Password: **HCL-Dem0** – if you do not have these, you may use the default administrator account with User ID: **wpsadmin,** Password: **wpsadmin)**



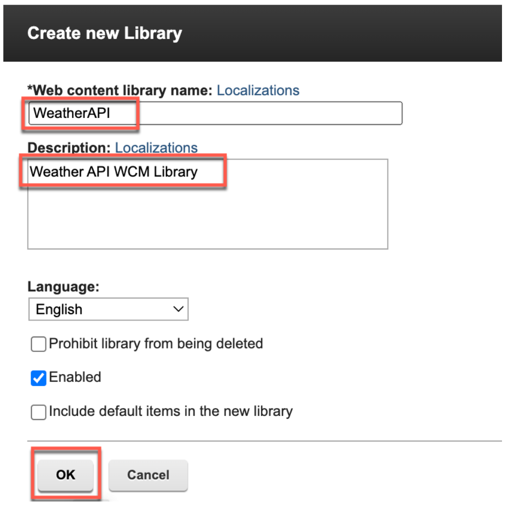
User ID ghayes

Password ••••••••

3. Navigate to: **Practitioner Studio → Web Content → Web Content Libraries**. Then click **Create new Library**:



4. Enter the library name **WeatherAPI** (and optionally a **Description)** and click **OK**.



Create new Library

*Web content library name: Localizations
WeatherAPI

Description: Localizations
Weather API WCM Library

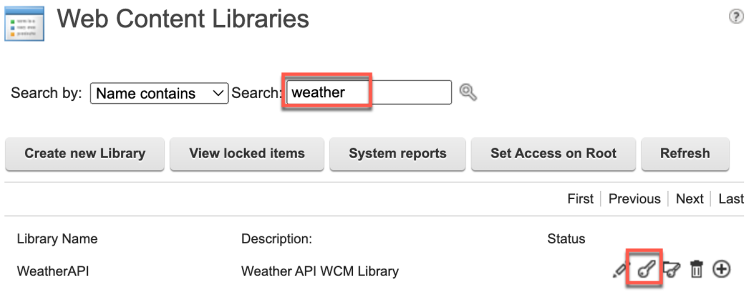
Language:
English

□ Prohibit library from being deleted

☑ Enabled

□ Include default items in the new library

5. Then provide access to all users to your new library. Use **Search by** to locate your new library and click the **Set Permission** icon.

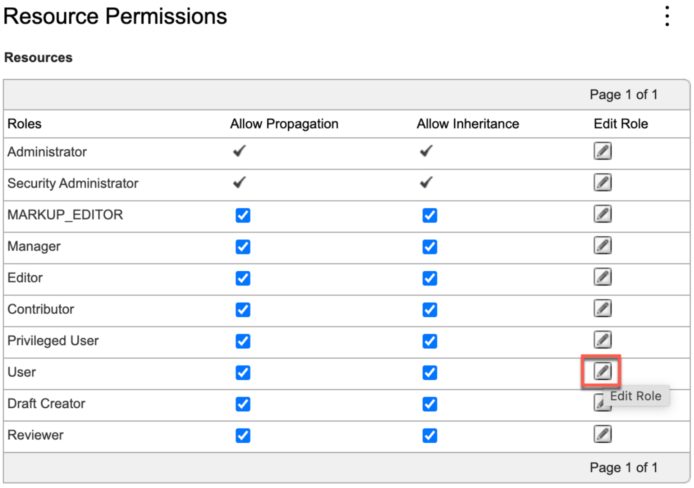


Web Content Libraries

Search by: Name contains Search: weather

Library Name	Description:	Status
WeatherAPI	Weather API WCM Library	

6. Click **Users**.



Resource Permissions

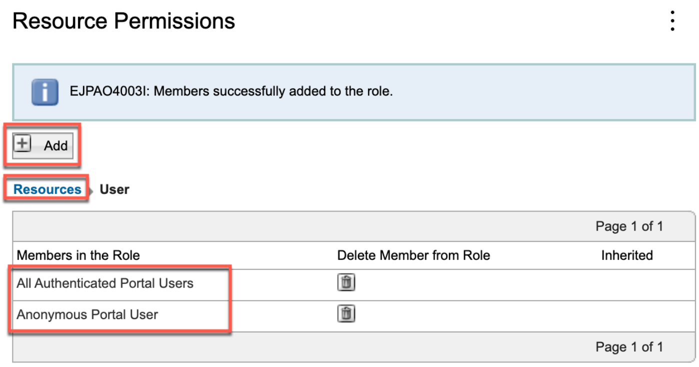
Resources

Page 1 of 1

Roles	Allow Propagation	Allow Inheritance	Edit Role
Administrator	✓	✓	✎
Security Administrator	✓	✓	✎
MARKUP_EDITOR	☑	☑	✎
Manager	☑	☑	✎
Editor	☑	☑	✎
Contributor	☑	☑	✎
Privileged User	☑	☑	✎
User	☑	☑	✎
Draft Creator	☑	☑	✎
Reviewer	☑	☑	✎

Page 1 of 1

7. And click **Add** to add both **Anonymous** and **All Authenticated Portal Users**. Then click **Resources**.



Resource Permissions

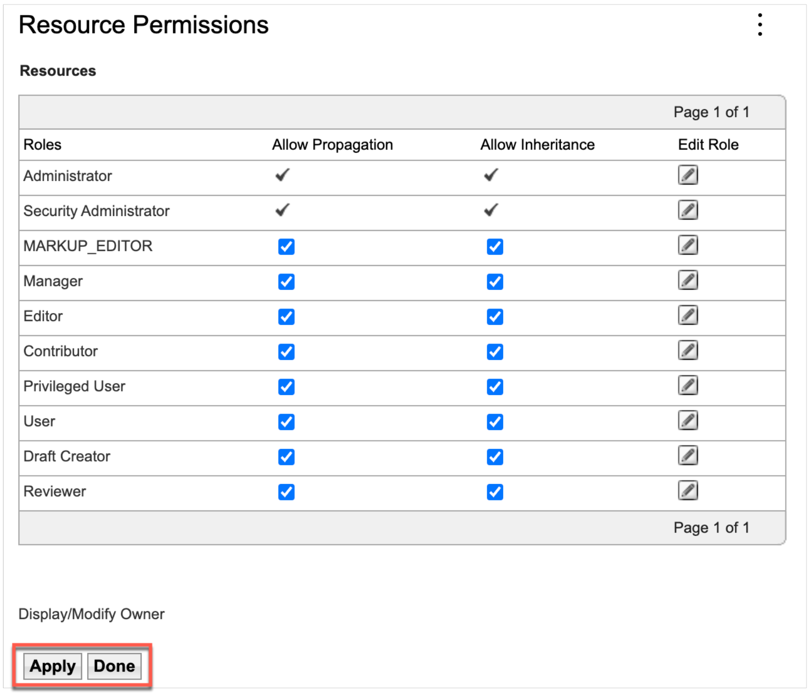
EJPAO4003I: Members successfully added to the role.

Resources › User

Members in the Role	Delete Member from Role	Inherited
All Authenticated Portal Users	🗑	
Anonymous Portal User	🗑	

Page 1 of 1

8. Click **Apply**, **OK** and **Done** to get your updates applied.



Resource Permissions

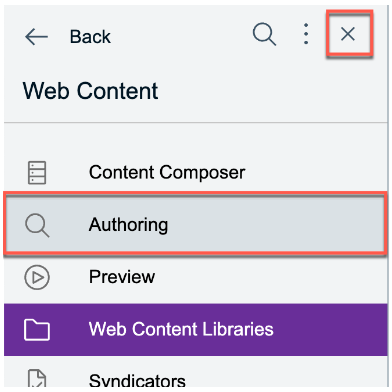
Resources

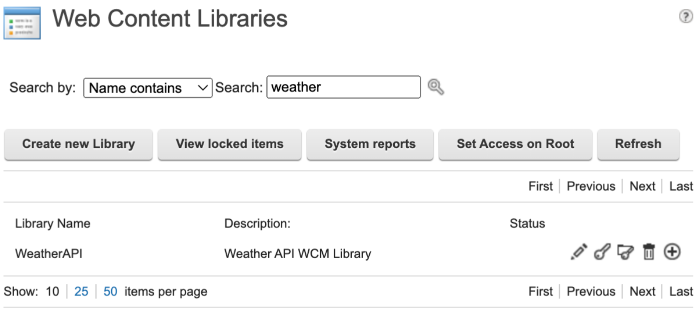
Roles	Allow Propagation	Allow Inheritance	Edit Role
Administrator	✓	✓	✎
Security Administrator	✓	✓	✎
MARKUP_EDITOR	☑	☑	✎
Manager	☑	☑	✎
Editor	☑	☑	✎
Contributor	☑	☑	✎
Privileged User	☑	☑	✎
User	☑	☑	✎
Draft Creator	☑	☑	✎
Reviewer	☑	☑	✎

Page 1 of 1

Display/Modify Owner

9. Then navigate to **Authoring** and close the sidebar.





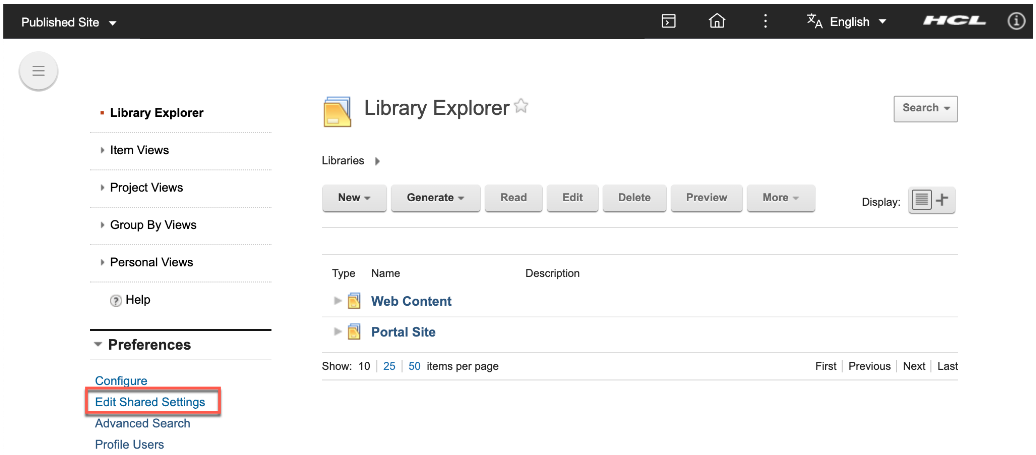
Web Content Libraries

Search by: Name contains Search: weather

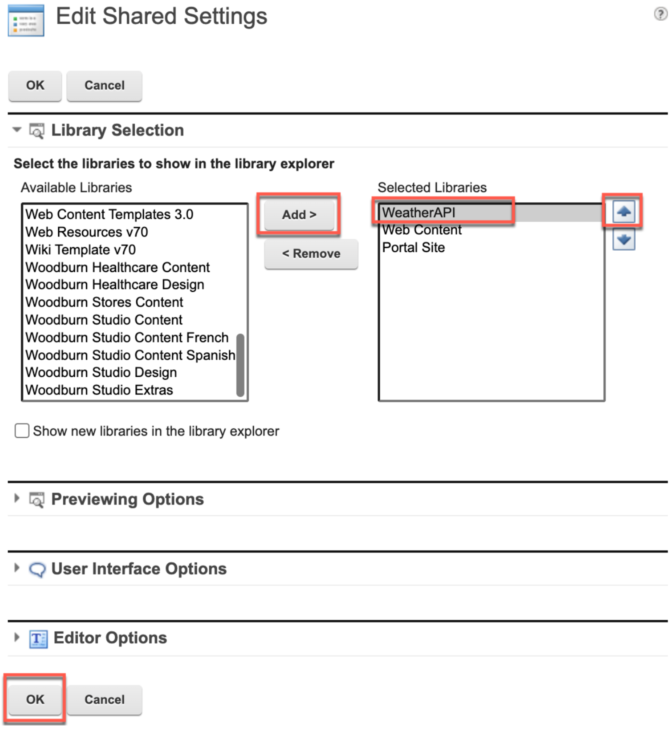
Library Name	Description:	Status
WeatherAPI	Weather API WCM Library	

Show: 10 | 25 | 50 items per page

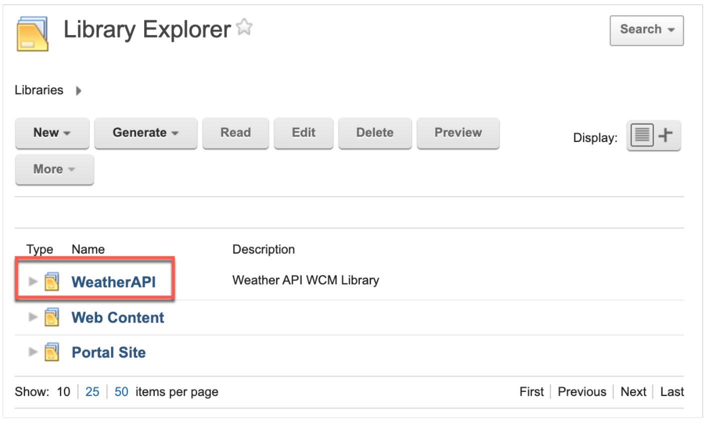
10. Now add the new library the Library explorer. Click **Edit Shared Settings.**



11. Then under **Library Selection**, select your new **WeatherAPI** library underthe Available Libraries and then click **Add >** to move it to the list of the **Selected Libraries**. Then use the arrow button to move it up to the first position. Then click **OK** to save your settings.

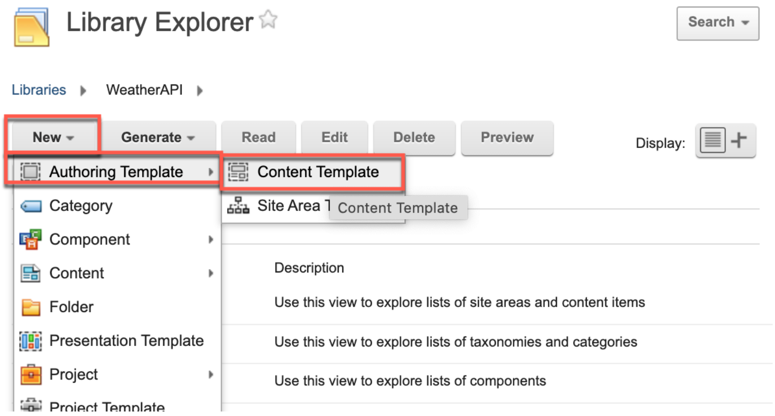


12. Your library now appears in the Library Explorer. Click **WeatherAPI** to see more details.



Type	Name	Description
	WeatherAPI	Weather API WCM Library
	Web Content	
	Portal Site	

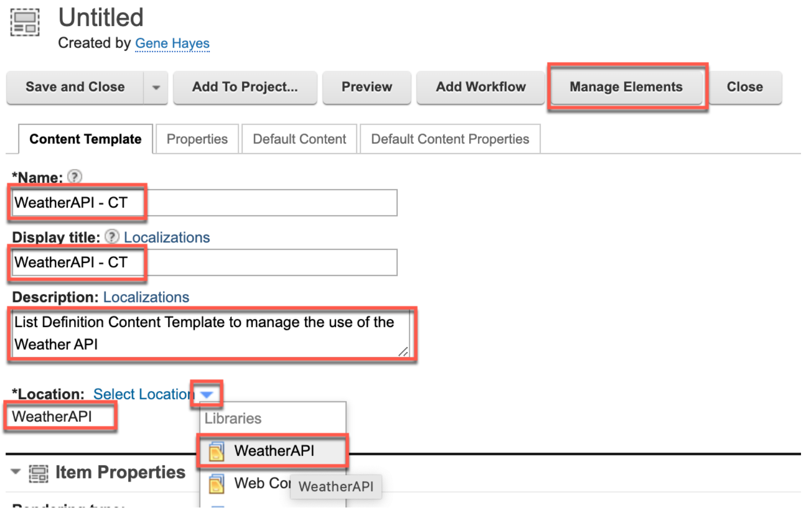
13. Then create a Content Template in your library to manage the Weather API DDC List Definition. Click **New – Authoring Template – Content Template**.



14. Then **Name** it (e.g **WeatherAPI - CT**), give a clear **Description** (e.g. **List Definition Content Template to manage the use of the Weather API**). Then add it to your new library. Click the arrow down next to **Select Location** and select your **WeatherAPI** library.

> Notice that if you created this content template directly under Authoring Templates view, it would be set automatically.

Then add the elements. Click **Manage Elements**.



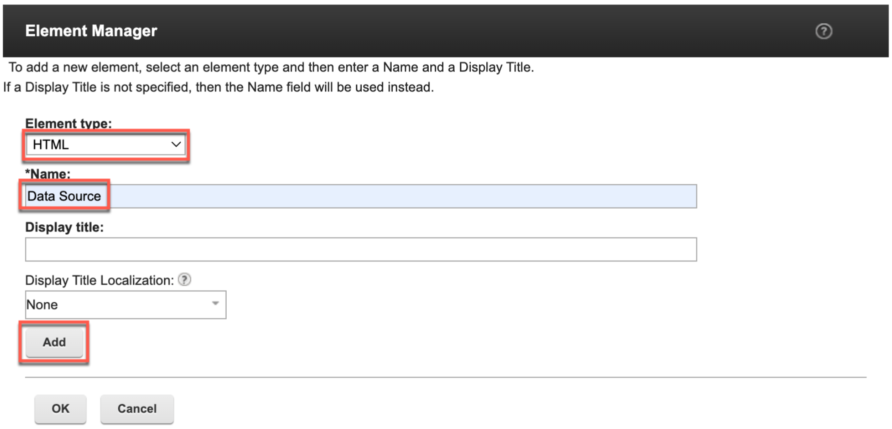
*Name: WeatherAPI - CT

Display title: WeatherAPI - CT

Description: List Definition Content Template to manage the use of the Weather API

*Location: WeatherAPI

15. Here you manage the List Definition elements. First add a **Data Source** HTML element that is used to enter the URL of the REST API. The type is HTML to avoid any HTML encoding problems. Select the **Element type HTML** and **name** it **Data Source**. Then click **Add**.



Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

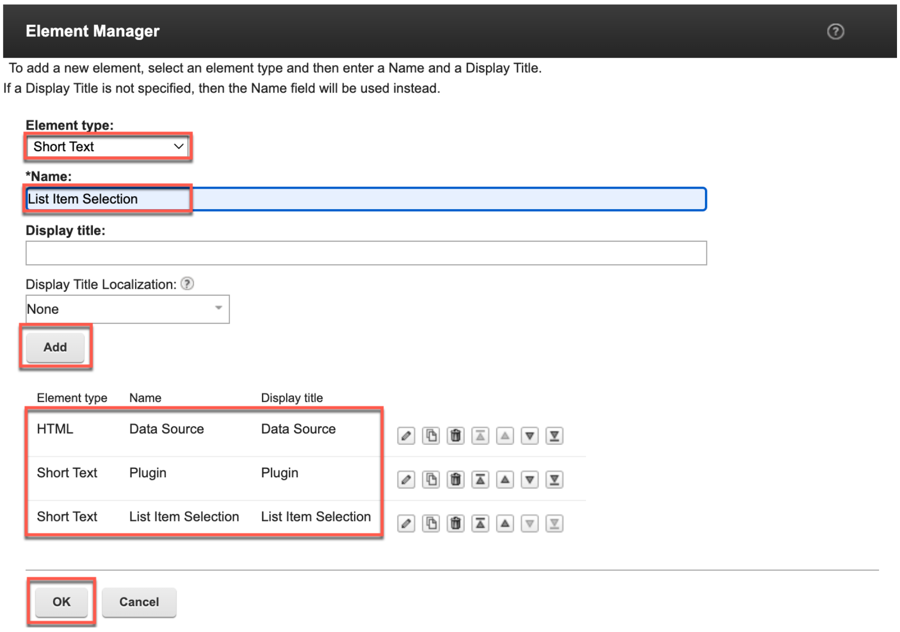
Element type: HTML

*Name: Data Source

Display title:

Display Title Localization: None

16. Then add a **Plugin Short Text** element that allows to define the DDC Plugin. And add the **List Item Selection Short Text** element that allows to select what list items to show. You should then have these three elements. Click **OK** to save.



Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

Element type: Short Text

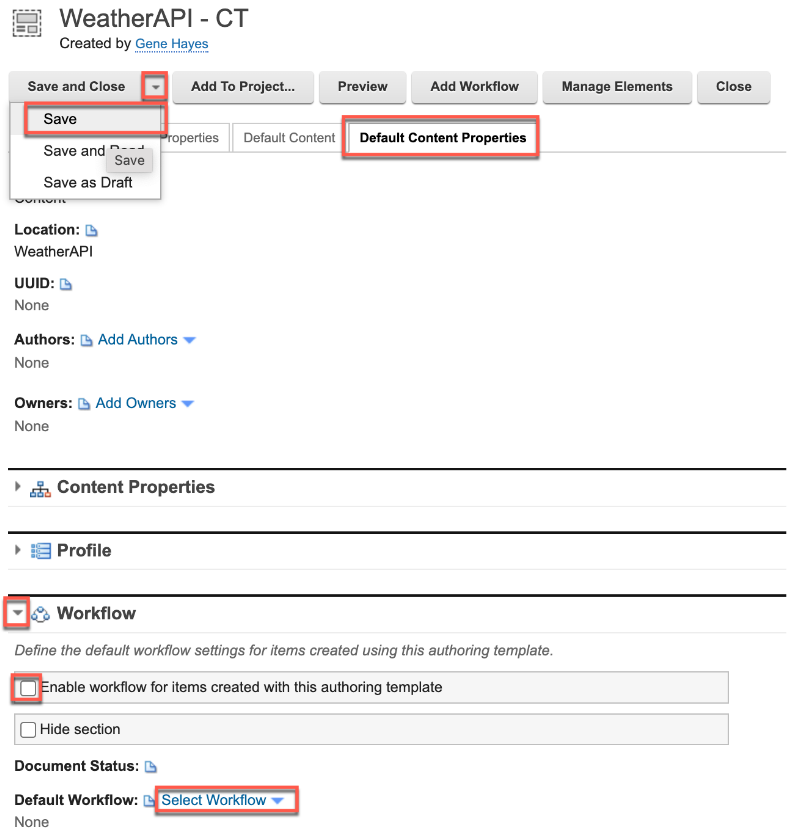
*Name: List Item Selection

Display title:

Display Title Localization: None

Element type	Name	Display title
HTML	Data Source	Data Source
Short Text	Plugin	Plugin
Short Text	List Item Selection	List Item Selection

17. For your List Definition content for the WeatherAPI, you may want to specify a default workflow or simply remove it, to make it easier to manage. Click on the **Default Content Properties** tab, expand **Workflow** and **disable** the **Workflow**. Optionally, if you wish to use a workflow, you may do that using the **Select Workflow** and you could choose for example an existing workflow like **Web Content - Express Workflow**. Then save it. Expand **Save and Close** and click **Save**.



WeatherAPI - CT

Created by Gene Hayes

Location:
WeatherAPI

UUID:
None

Authors: Add Authors
None

Owners: Add Owners
None

Content Properties

Profile

Workflow

Define the default workflow settings for items created using this authoring template.

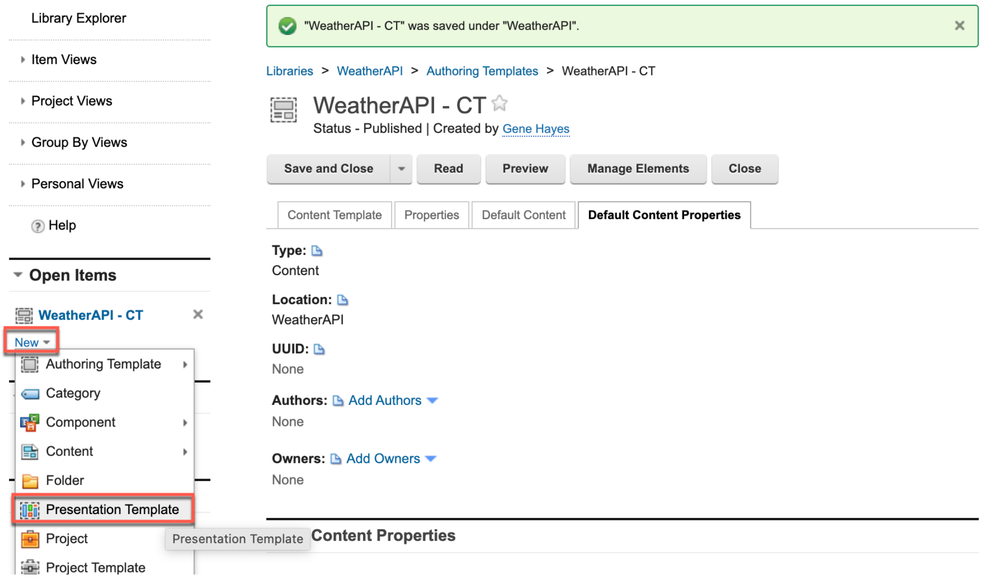
☐ Enable workflow for items created with this authoring template

☐ Hide section

Document Status:

Default Workflow: Select Workflow
None

18. Then create the corresponding WeatherAPI List Definition Presentation Template. Under the Open Items, click **New** and then **Presentation Template**. You may also close the notification that your WeatherAPI – CT was correctly saved.



"WeatherAPI - CT" was saved under "WeatherAPI".

Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Created by Gene Hayes

Type:
Content

Location:
WeatherAPI

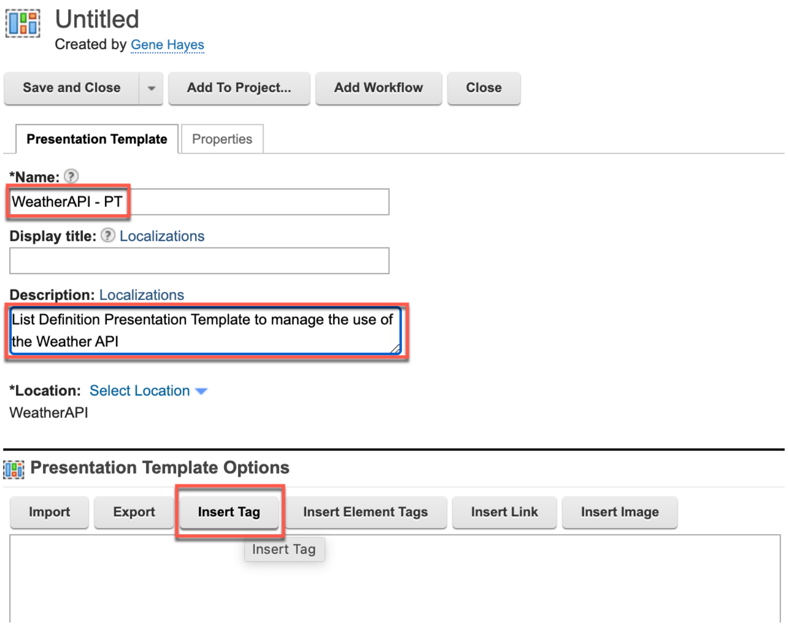
UUID:
None

Authors: Add Authors
None

Owners: Add Owners
None

Content Properties

19. **Name** it **WeatherAPI – PT** and add a clear **Description**, like **List Definition Presentation Template to manage the use of the Weather API**. Then under Presentation Template Options, establish the right List Rendering Context starting from the Data Source, Plugin and List Item Selection attributes of the List Definition. This is managed with the tag *[Plugin:ListRenderingContext action="set" extension-id="<Plugin>" attribute="source=<Data Source>" attribute="listItemSelection=<List Item Selection>"]*. Create this now. First add the Plugin:ListRenderingContext tag. Click **Insert Tag**.



Untitled

Created by Gene Hayes

*Name: WeatherAPI - PT

Display title:

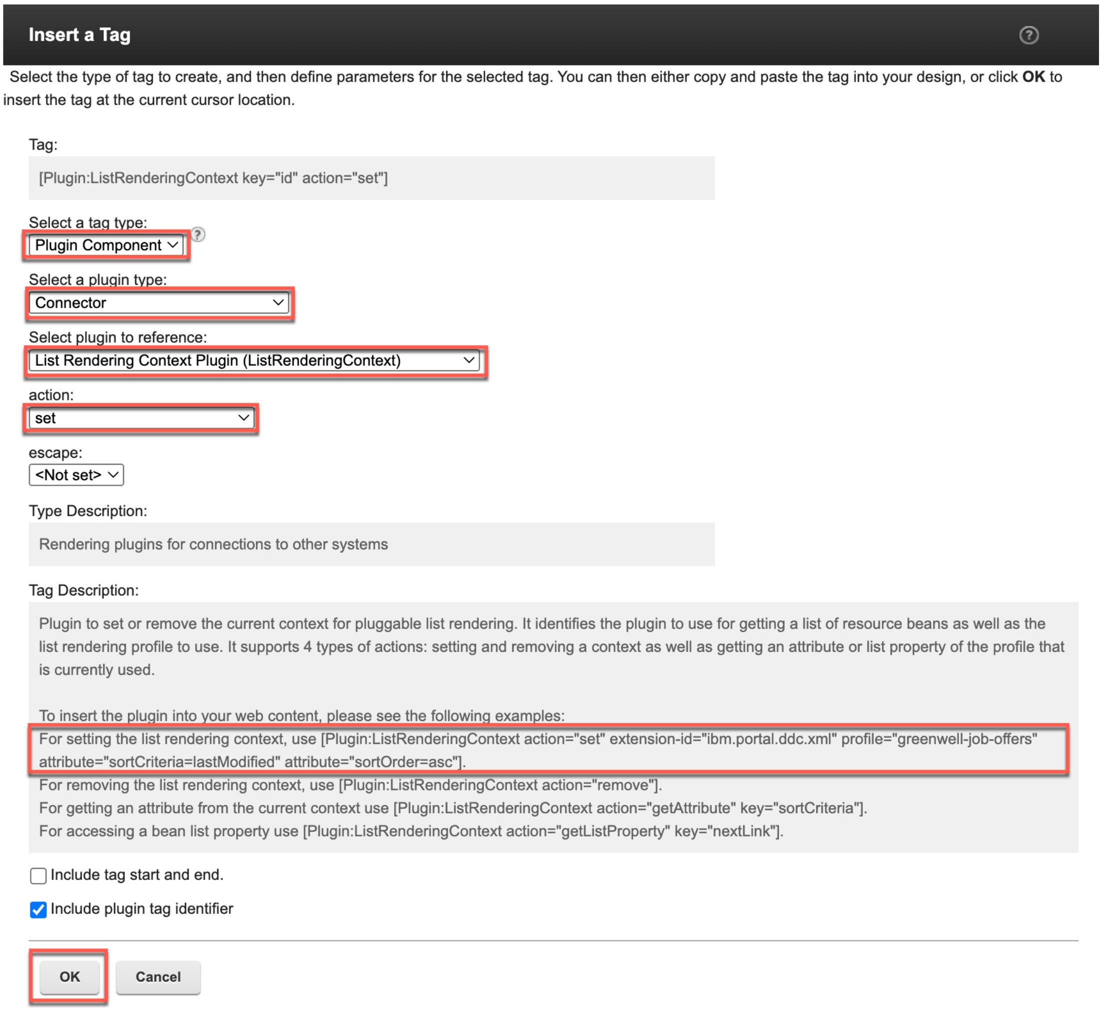
Description: List Definition Presentation Template to manage the use of the Weather API

*Location: Select Location

WeatherAPI

Presentation Template Options

20. Then under **Select a tag type**, select **Plugin Component**, then under **Select a plugin type**, **Connector** (for DDC), select under **Select plugin to reference List Rendering Context (ListRenderingContext),** ensure the **action** is set to **set**, see under the Tag Description what parameters are typically used for a set action and click **OK**.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

[Plugin:ListRenderingContext key="id" action="set"]

Select a tag type: Plugin Component

Select a plugin type: Connector

Select plugin to reference: List Rendering Context Plugin (ListRenderingContext)

action: set

escape: <Not set>

Type Description:

Rendering plugins for connections to other systems

Tag Description:

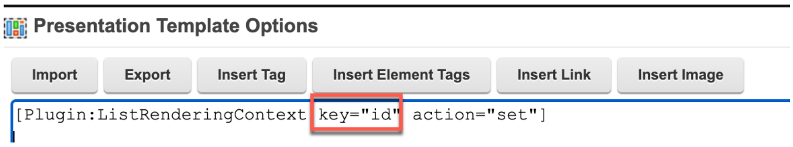
Plugin to set or remove the current context for pluggable list rendering. It identifies the plugin to use for getting a list of resource beans as well as the list rendering profile to use. It supports 4 types of actions: setting and removing a context as well as getting an attribute or list property of the profile that is currently used.

To insert the plugin into your web content, please see the following examples:
For setting the list rendering context, use [Plugin:ListRenderingContext action="set" extension-id="ibm.portal.ddc.xml" profile="greenwell-job-offers" attribute="sortCriteria=lastModified" attribute="sortOrder=asc"].
For removing the list rendering context, use [Plugin:ListRenderingContext action="remove"].
For getting an attribute from the current context use [Plugin:ListRenderingContext action="getAttribute" key="sortCriteria"].
For accessing a bean list property use [Plugin:ListRenderingContext action="getListProperty" key="nextLink"].

□ Include tag start and end.

☑ Include plugin tag identifier

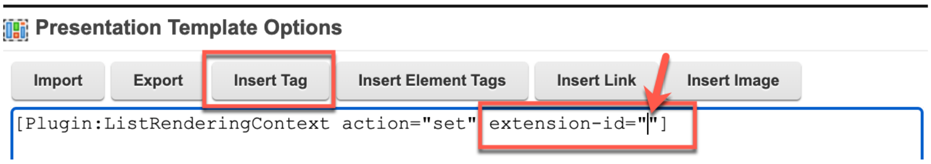
21. Then replace the key="id" to extension-id and the two attributes to set. First set this extension-id, which is defined by your WeatherAPI List Definition with the Plugin element. Delete **key="id"** first.



Presentation Template Options

```
[Plugin:ListRenderingContext key="id" action="set"]
```

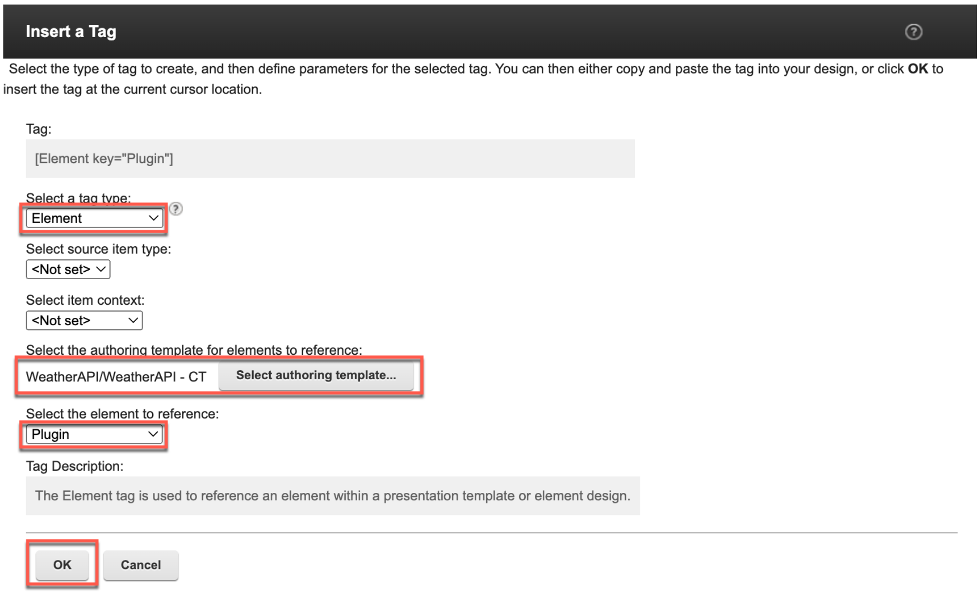
22. Then add **extension-id=""** after the action="set" parameter, put your cursor between "" and click **Insert Tag**.



Presentation Template Options

```
[Plugin:ListRenderingContext action="set" extension-id=""]
```

23. Then select the **tag type Element**, **authoring template** your **WeatherAPI/WeatherAPI – CT**, **Plugin element to reference** and click OK.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:
[Element key="Plugin"]

Select a tag type:
Element

Select source item type:
<Not set>

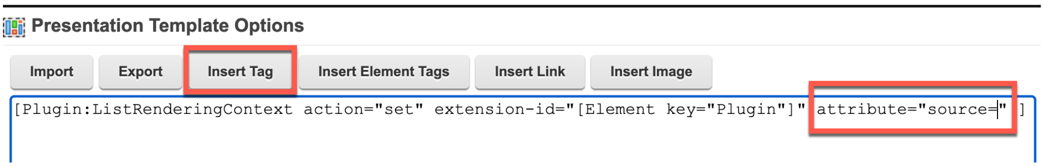
Select item context:
<Not set>

Select the authoring template for elements to reference:
WeatherAPI/WeatherAPI - CT

Select the element to reference:
Plugin

Tag Description:
The Element tag is used to reference an element within a presentation template or element design.

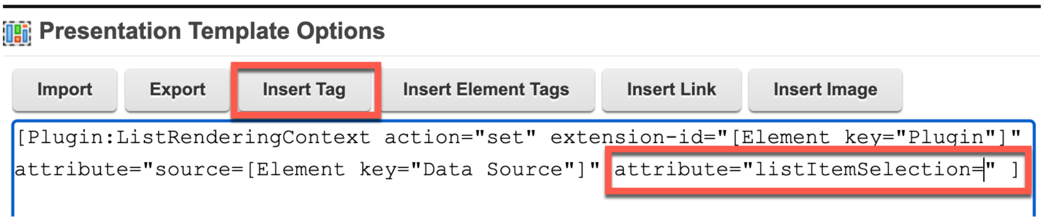
24. Then add the source attribute which is set to your Data Source element. Add **attribute="source="** and click **Insert Tag.**



Presentation Template Options

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]" attribute="source=" ]
```

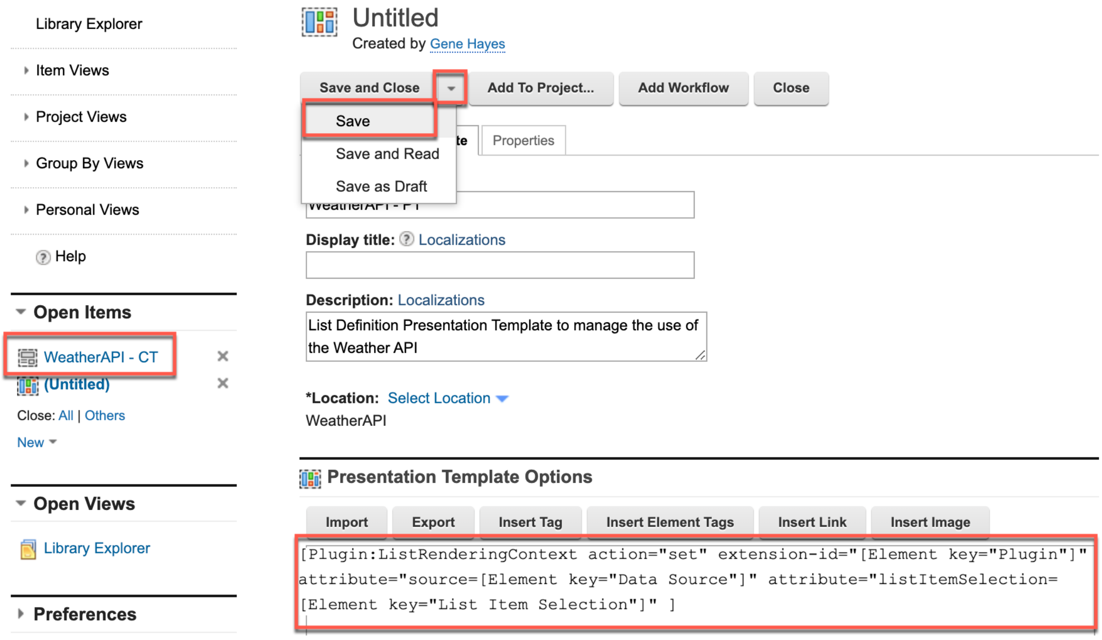
25. And in the same way as in step 22, select element, your library, this time the **Data Source** element and click **OK**. Then add the second attribute for the List Item Selection. Add **attribute="listItemSelection="** and use the tag helper (with **Insert Tag**) to insert the right element tag for the **List Item Selection** element.



Presentation Template Options

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]"
attribute="source=[Element key="Data Source"]" attribute="listItemSelection=" ]
```

26. Check your code and then save it, using the drop down next to **Save and Close** and **Save**. Then select **your WeatherAPI – CT** to make this new presentation template your default presentation template for your content template.



WeatherAPI - PT

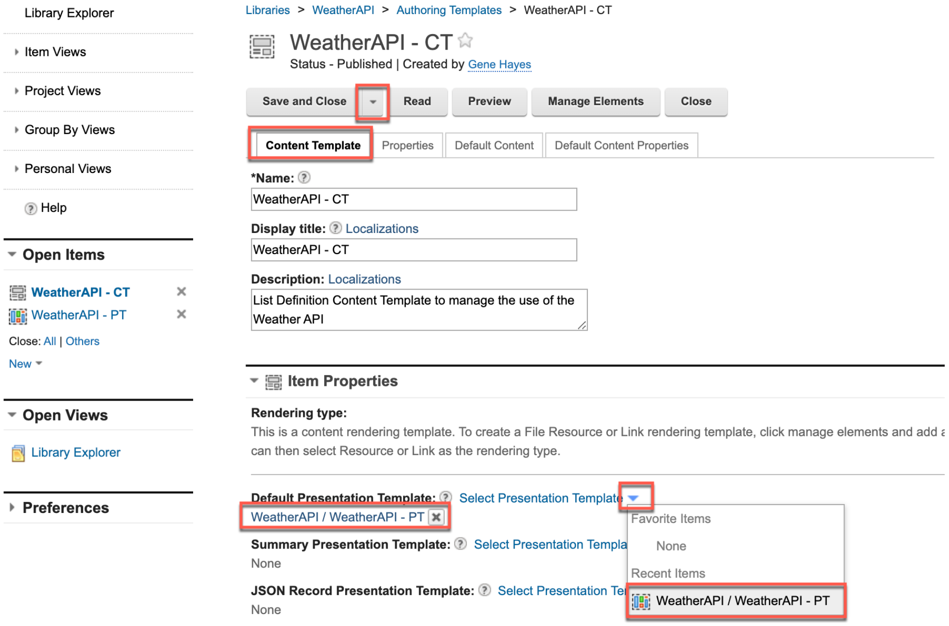
Display title: Localizations

Description: Localizations
List Definition Presentation Template to manage the use of the Weather API

*Location: Select Location
WeatherAPI

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]"
attribute="source=[Element key="Data Source"]" attribute="listItemSelection=
[Element key="List Item Selection"]" ]
```

27. Go to the **Content Template** section and under **Item Properties**, **Default Present Template**, use the drop down to select your new presentation template **WeatherAPI/WeatherAPI – PT**. Then save it again.



*Name:
WeatherAPI - CT

Display title: Localizations
WeatherAPI - CT

Description: Localizations
List Definition Content Template to manage the use of the Weather API

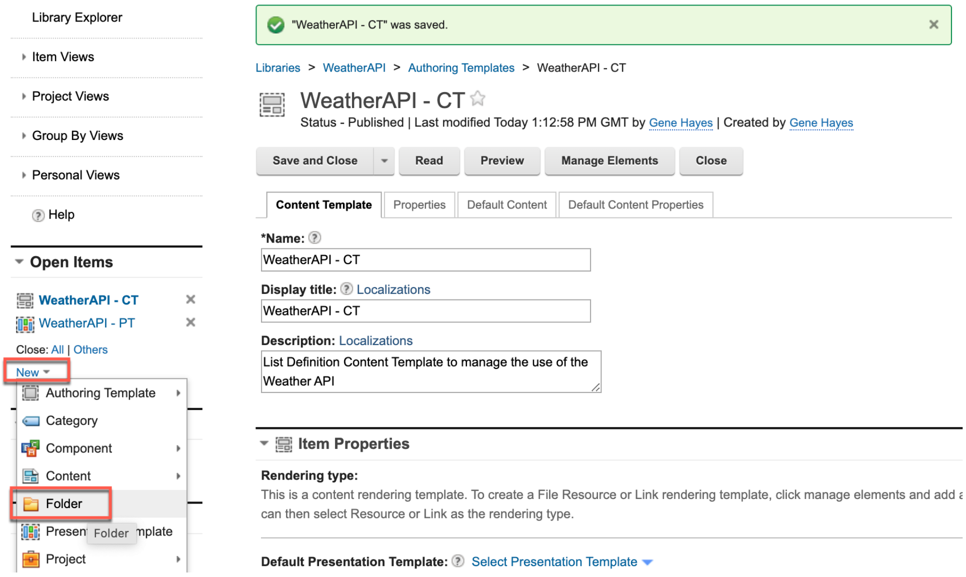
Rendering type:
This is a content rendering template. To create a File Resource or Link rendering template, click manage elements and add a can then select Resource or Link as the rendering type.

Default Presentation Template: Select Presentation Template
WeatherAPI / WeatherAPI - PT

Summary Presentation Template: Select Presentation Templa
None

JSON Record Presentation Template: Select Presentation Te
None

28. You have set the List Rendering Context in your presentation template. Now you need to use this to generate a list. This is done using a Personalization Component. As you will create multiple list presentations, you will first create a new folder under components to place them all, called List Appearances. Click **New** – **Folder**.



"WeatherAPI - CT" was saved.

Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Last modified Today 1:12:58 PM GMT by Gene Hayes | Created by Gene Hayes

*Name: WeatherAPI - CT

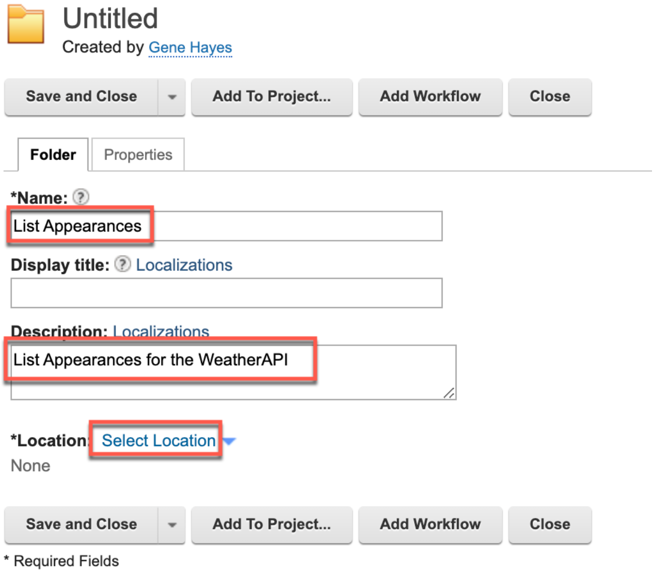
Display title: Localizations WeatherAPI - CT

Description: Localizations List Definition Content Template to manage the use of the Weather API

Rendering type:
This is a content rendering template. To create a File Resource or Link rendering template, click manage elements and add a can then select Resource or Link as the rendering type.

Default Presentation Template: Select Presentation Template

29. Enter the name **List Appearances**, a good description, like **List Appearances for the WeatherAPI**, and click **Select Location** to select the right place to create this folder.



Untitled

Created by Gene Hayes

*Name: List Appearances

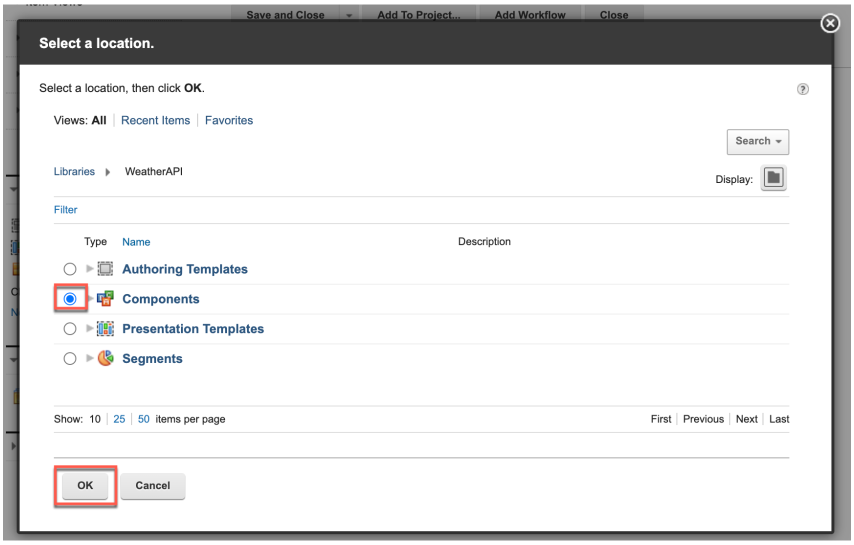
Display title: Localizations

Description: Localizations List Appearances for the WeatherAPI

*Location: Select Location
None

* Required Fields

30. Then select **Components** and click **OK**.



Select a location.

Select a location, then click **OK**.

Views: **All** | Recent Items | Favorites

Libraries ▸ WeatherAPI

Display:

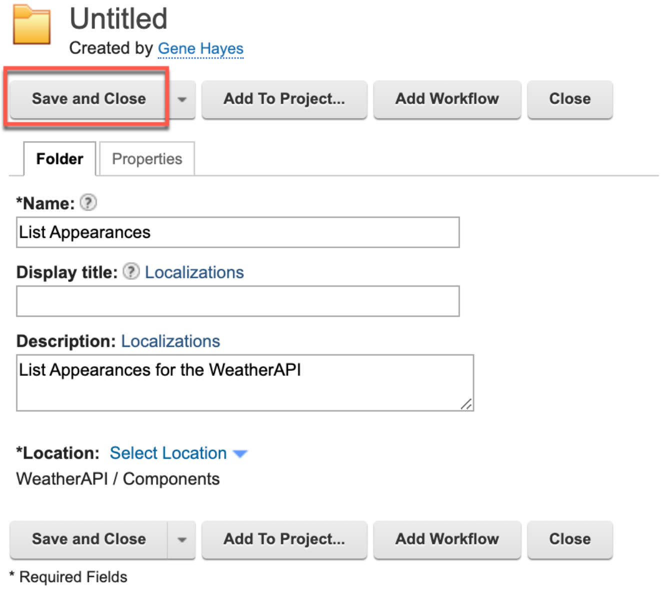
Filter

Type	Name	Description
○	Authoring Templates	
●	Components	
○	Presentation Templates	
○	Segments	

Show: 10 | 25 | 50 items per page

First | Previous | Next | Last

31. And then save and close it, as you will not make any changes to this item. Click **Save and Close**.



Untitled

Created by Gene Hayes

***Name:** List Appearances

Display title: Localizations

Description: Localizations

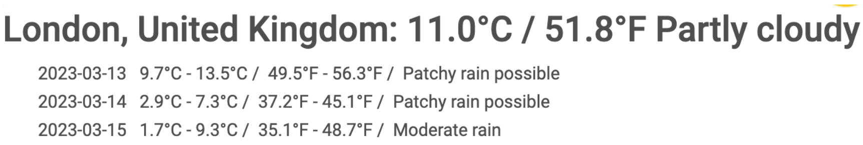
List Appearances for the WeatherAPI

***Location:** Select Location

WeatherAPI / Components

* Required Fields

32. Then create your first WeatherAPI List Appearance Personalization Component that will look like



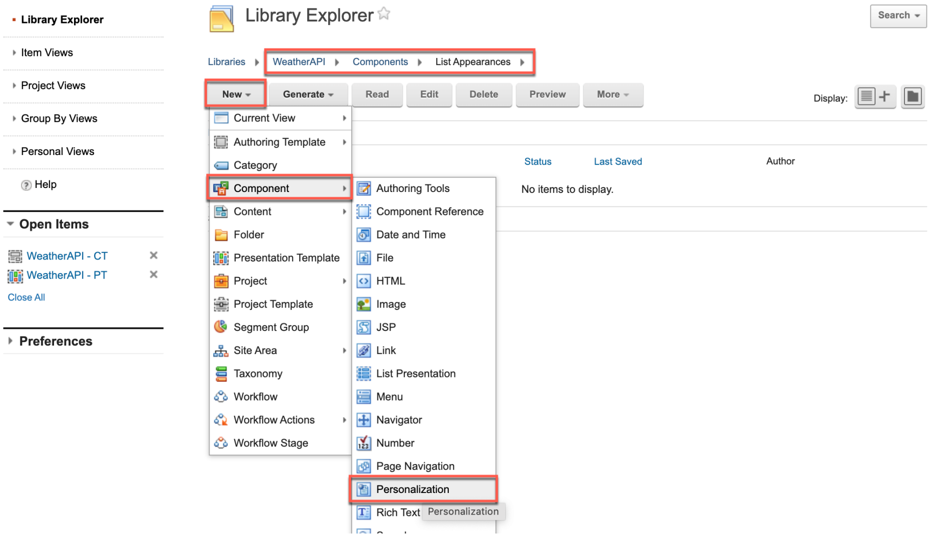
London, United Kingdom: 11.0°C / 51.8°F Partly cloudy

2023-03-13 9.7°C - 13.5°C / 49.5°F - 56.3°F / Patchy rain possible

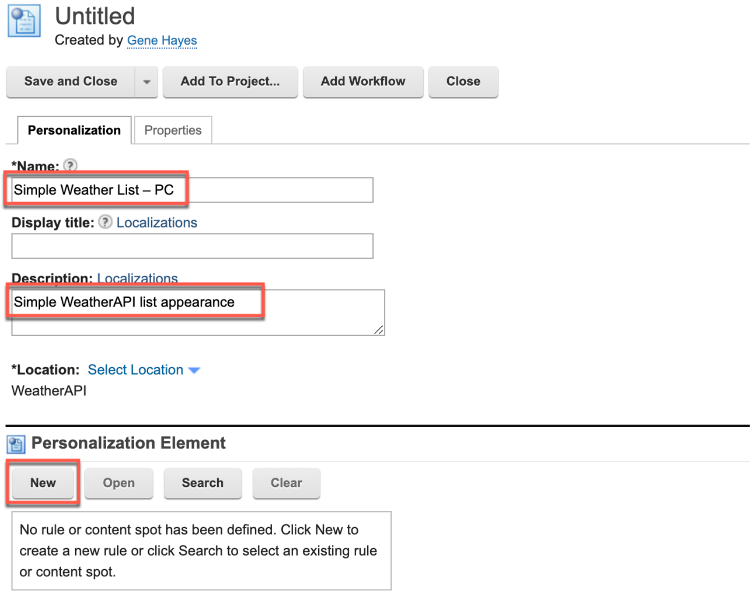
2023-03-14 2.9°C - 7.3°C / 37.2°F - 45.1°F / Patchy rain possible

2023-03-15 1.7°C - 9.3°C / 35.1°F - 48.7°F / Moderate rain

33. Then go to your new folder. Under the Library Explorer, click **WeatherAPI** – **Components** – **List Appearance**, and then **New - Component** and **Personalization**.



34. Name it **Simple Weather List - PC** and add a clear description, like **Simple WeatherAPI list appearance**. To make this work with your list, you need to create a Pluggable Resources resource collection. Click **New**.



*Name: Simple Weather List – PC

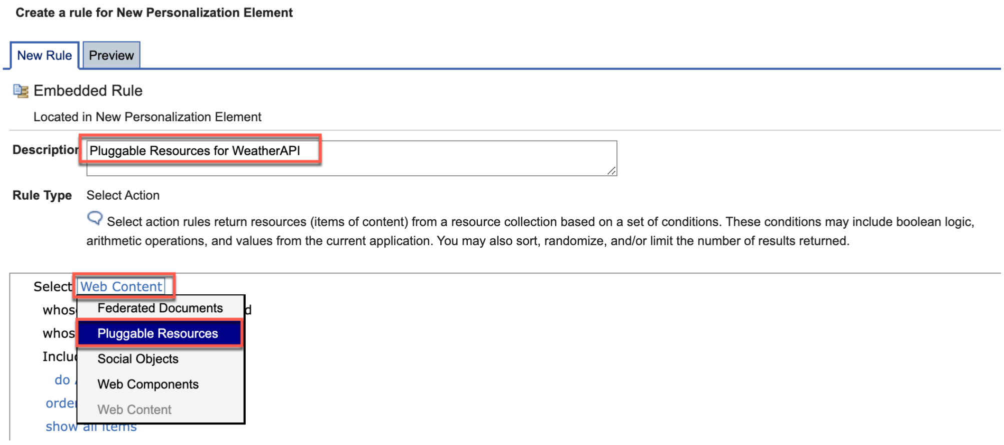
Display title:

Description: Simple WeatherAPI list appearance

*Location: WeatherAPI

No rule or content spot has been defined. Click New to create a new rule or click Search to select an existing rule or content spot.

35. This opens the Personalization Editor where you configure an embedded rule. Give it a description, like **Pluggable Resources for WeatherAPI**. Click **Web Content** and from the dropdown menu and select **Pluggable Resources**.



Create a rule for New Personalization Element

Embedded Rule

Located in New Personalization Element

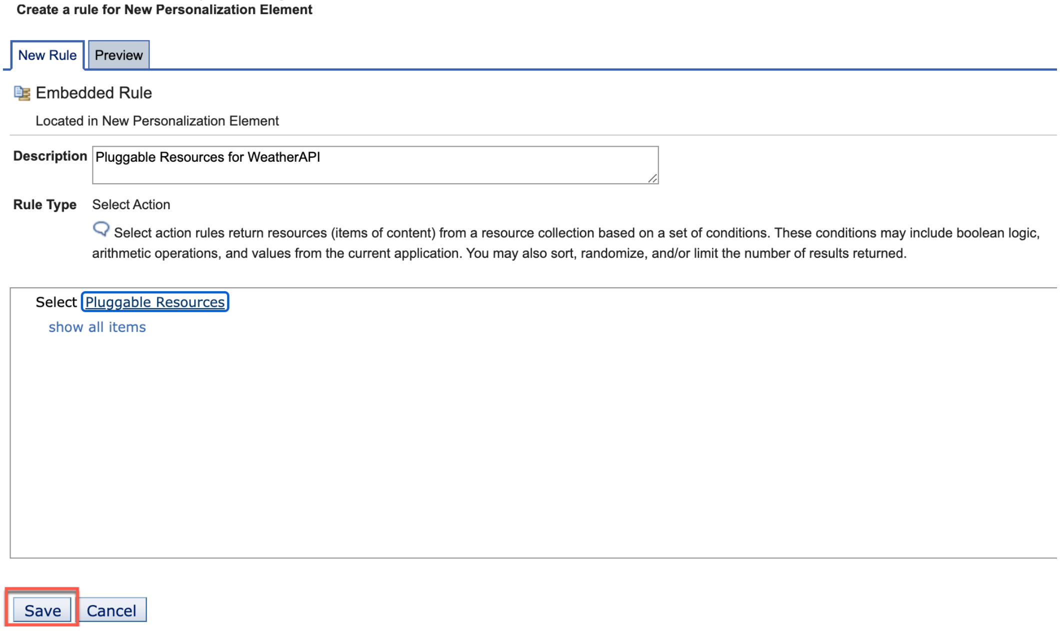
Description Pluggable Resources for WeatherAPI

Rule Type Select Action

Select action rules return resources (items of content) from a resource collection based on a set of conditions. These conditions may include boolean logic, arithmetic operations, and values from the current application. You may also sort, randomize, and/or limit the number of results returned.

Select Web Content

36. It is automatically configured to show all items. Now save it. Click **Save**.



Create a rule for New Personalization Element

Embedded Rule

Located in New Personalization Element

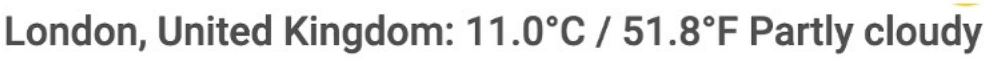
Description Pluggable Resources for WeatherAPI

Rule Type Select Action

Select action rules return resources (items of content) from a resource collection based on a set of conditions. These conditions may include boolean logic, arithmetic operations, and values from the current application. You may also sort, randomize, and/or limit the number of results returned.

Select Pluggable Resources
show all items

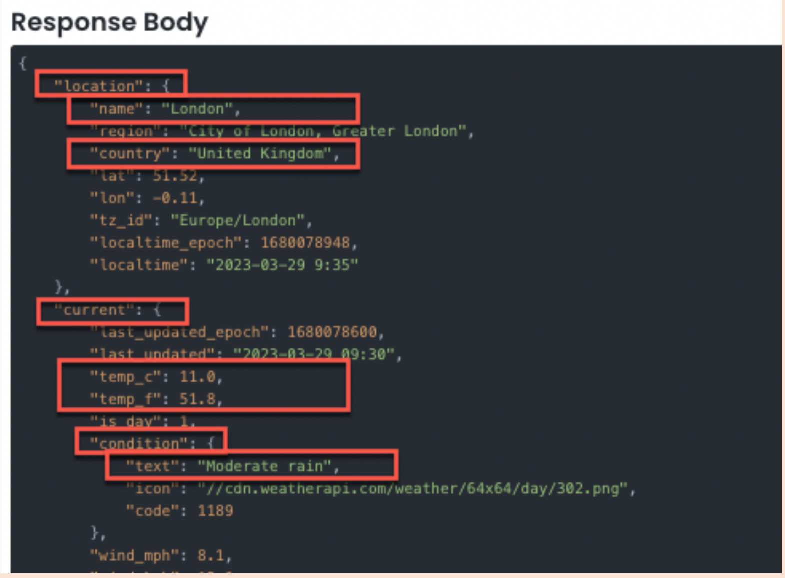
37. It shows your embedded rule. Now you may define how the list will show. Like a Menu Component, your Personalization Component has a List Presentation Markup with a header, result design, footer, separator, and no result section. Start with the header to show some weather information on your location, like current temperature in degrees Celsius and Fahrenheit and the current condition text:



London, United Kingdom: 11.0°C / 51.8°F Partly cloudy

38. Under **List Presentation Markup Header** of your list appearance, you may add this with a special tag again.

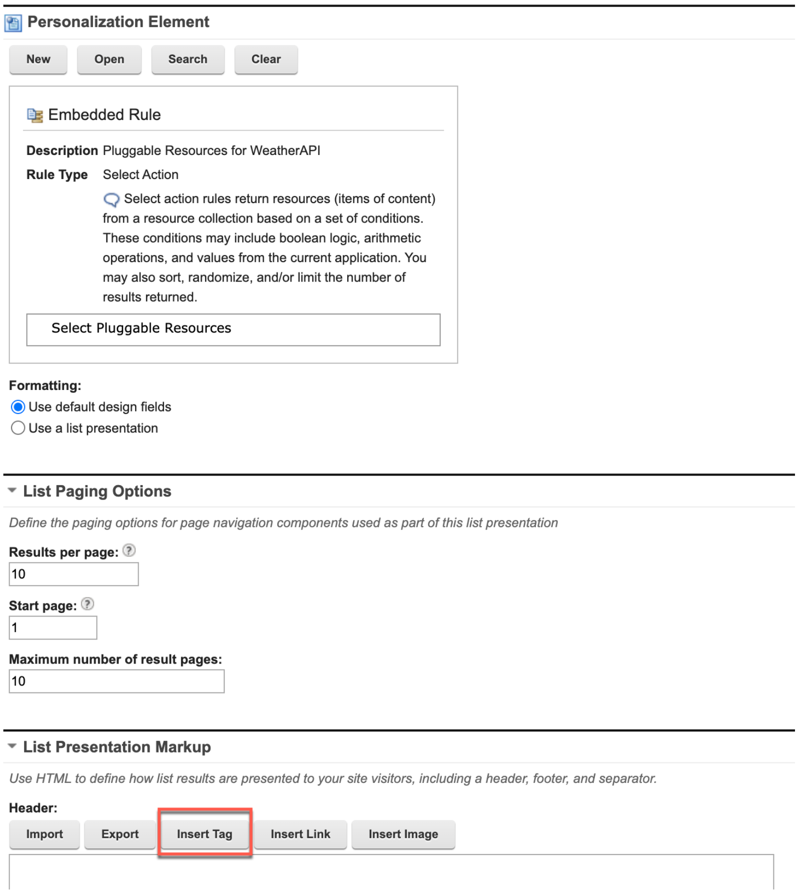
> Now Remember that the WeatherAPI forecast REST API call gives a response body with access to a set of data, in this format. It shows for example the location name (accessible with location.name), current temperature in degrees Celsius (current.temp_c) and Fahrenheit (current.temp_f) and current condition in text form (current.condition.text). See the top of your Response Body from part 1.



Response Body

```
{
  "location": {
    "name": "London",
    "region": "City of London, Greater London",
    "country": "United Kingdom",
    "lat": 51.52,
    "lon": -0.11,
    "tz_id": "Europe/London",
    "localtime_epoch": 1680078948,
    "localtime": "2023-03-29 9:35"
  },
  "current": {
    "last_updated_epoch": 1680078600,
    "last_updated": "2023-03-29 09:30",
    "temp_c": 11.0,
    "temp_f": 51.8,
    "is_day": 1,
    "condition": {
      "text": "Moderate rain",
      "icon": "//cdn.weatherapi.com/weather/64x64/day/302.png",
      "code": 1189
    },
    "wind_mph": 8.1,
```

Click **Insert Tag**.



Personalization Element

Embedded Rule

Description Pluggable Resources for WeatherAPI

Rule Type Select Action

Select action rules return resources (items of content) from a resource collection based on a set of conditions. These conditions may include boolean logic, arithmetic operations, and values from the current application. You may also sort, randomize, and/or limit the number of results returned.

Select Pluggable Resources

Formatting:

● Use default design fields

○ Use a list presentation

List Paging Options

Define the paging options for page navigation components used as part of this list presentation

Results per page: 10

Start page: 1

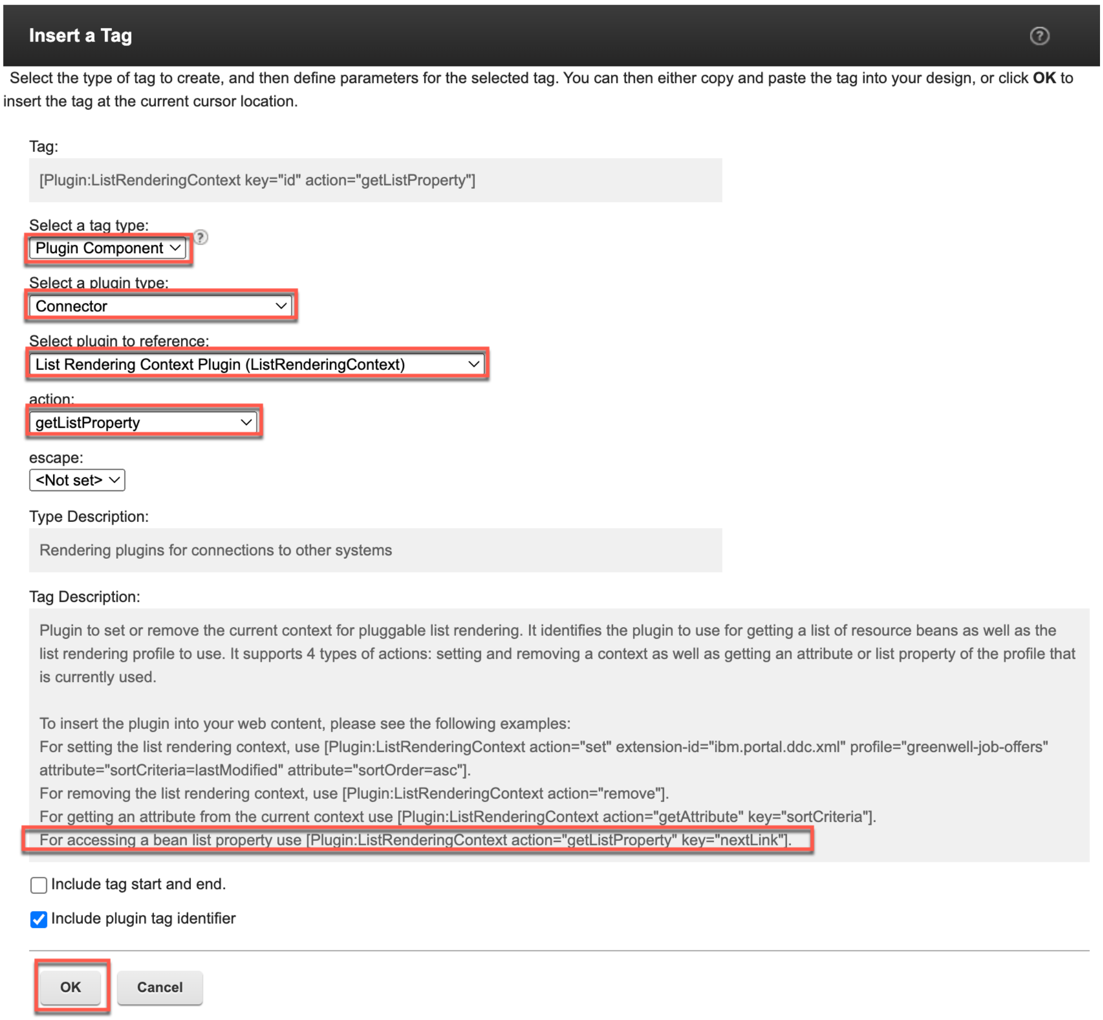
Maximum number of result pages: 10

List Presentation Markup

Use HTML to define how list results are presented to your site visitors, including a header, footer, and separator.

Header:

39. To display any of these properties, you use a getListProperty action. You will first add the location name property. You refer to this with location.name. Select again **Plugin Component** as **tag type**, **Connector** as **plugin type**, **List Rendering Context Plugin** and this time a **getListProperty action**. Click **OK** to save.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:
[Plugin:ListRenderingContext key="id" action="getListProperty"]

Select a tag type: Plugin Component

Select a plugin type: Connector

Select plugin to reference: List Rendering Context Plugin (ListRenderingContext)

action: getListProperty

escape: <Not set>

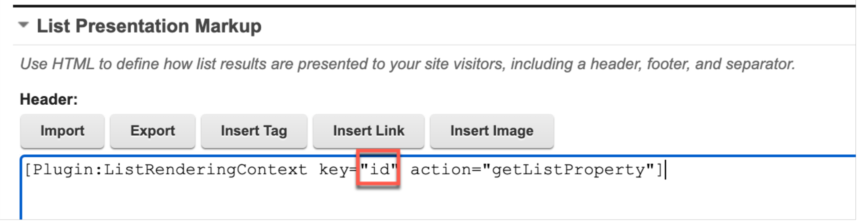
Type Description:
Rendering plugins for connections to other systems

Tag Description:
Plugin to set or remove the current context for pluggable list rendering. It identifies the plugin to use for getting a list of resource beans as well as the list rendering profile to use. It supports 4 types of actions: setting and removing a context as well as getting an attribute or list property of the profile that is currently used.

To insert the plugin into your web content, please see the following examples:
For setting the list rendering context, use [Plugin:ListRenderingContext action="set" extension-id="ibm.portal.ddc.xml" profile="greenwell-job-offers" attribute="sortCriteria=lastModified" attribute="sortOrder=asc"].
For removing the list rendering context, use [Plugin:ListRenderingContext action="remove"].
For getting an attribute from the current context use [Plugin:ListRenderingContext action="getAttribute" key="sortCriteria"].
For accessing a bean list property use [Plugin:ListRenderingContext action="getListProperty" key="nextLink"].

☐ Include tag start and end.
☑ Include plugin tag identifier

40. It sets the key to "id".



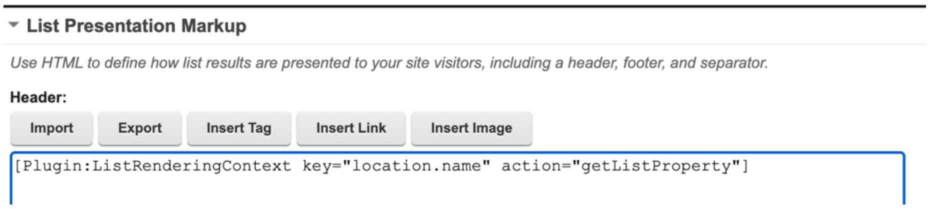
List Presentation Markup

Use HTML to define how list results are presented to your site visitors, including a header, footer, and separator.

Header:

```
[Plugin:ListRenderingContext key="id" action="getListProperty"]
```

41. Now update this **id** to the **location.name** to make it show the name of the location.



List Presentation Markup

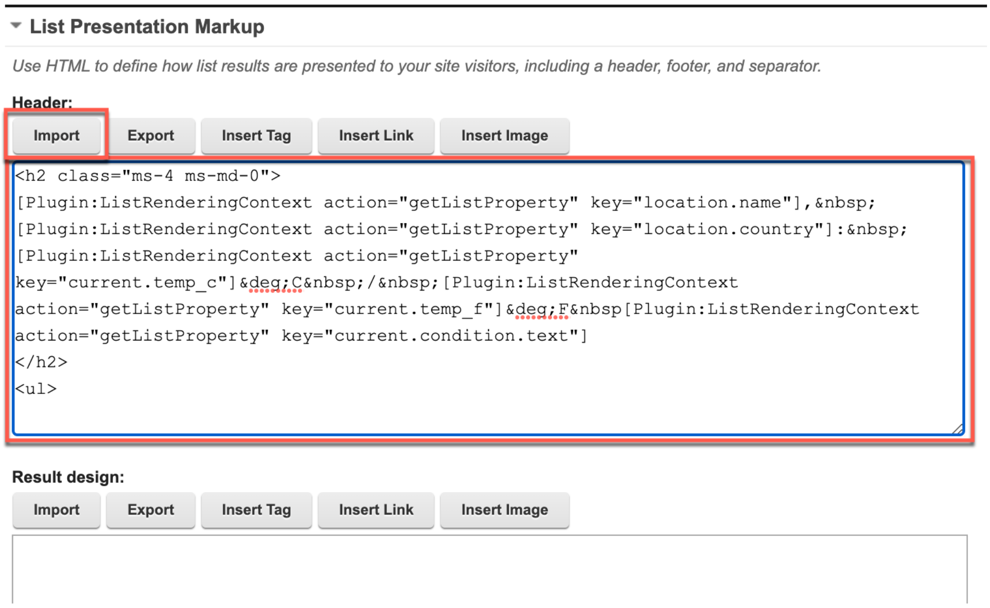
Use HTML to define how list results are presented to your site visitors, including a header, footer, and separator.

Header:

```
[Plugin:ListRenderingContext key="location.name" action="getListProperty"]
```

42. In the same way, add the location country (location.country), current temperature in degrees Celsius (current.temp_c), current temperature in Fahrenheit (current.temp_f) and current condition in text format (current.condition.text). Also add some formatting. To show the list of details for the forecast days, end your header with a list starting unordered list tag <ul>. You may also use **Import** to import the header from the downloaded resources.

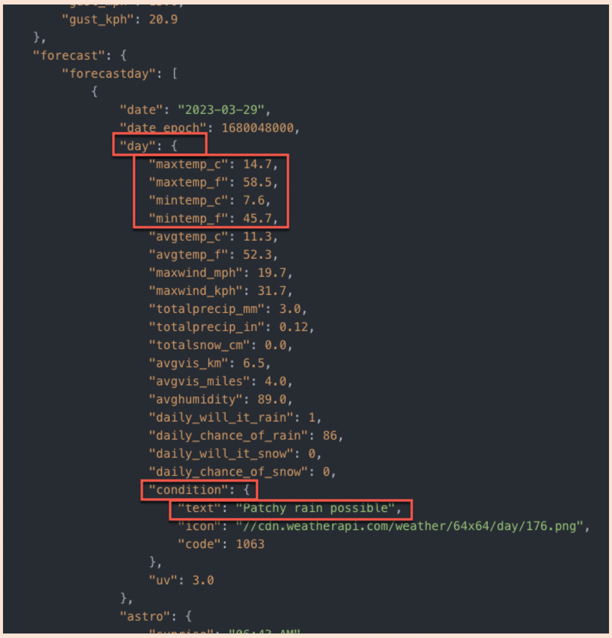
```
<h2 class="ms-4 ms-md-0">

[Plugin:ListRenderingContext action="getListProperty" key="location.name"],&nbsp;[Plugin:ListRenderingContext action="getListProperty" key="location.country"]:&nbsp;[Plugin:ListRenderingContext action="getListProperty" key="current.temp_c"]&deg;C&nbsp;/&nbsp;[Plugin:ListRenderingContext action="getListProperty" key="current.temp_f"]&deg;F&nbsp[Plugin:ListRenderingContext action="getListProperty" key="current.condition.text"]

</h2>

<ul>
```



```
<h2 class="ms-4 ms-md-0">
[Plugin:ListRenderingContext action="getListProperty" key="location.name"],&nbsp;
[Plugin:ListRenderingContext action="getListProperty" key="location.country"]:&nbsp;
[Plugin:ListRenderingContext action="getListProperty"
key="current.temp_c"]&deg;C&nbsp;/&nbsp;[Plugin:ListRenderingContext
action="getListProperty" key="current.temp_f"]&deg;F&nbsp[Plugin:ListRenderingContext
action="getListProperty" key="current.condition.text"]
</h2>
<ul>
```

43. For each forecast day, you want to show the date, minimum and maximum temperatures in degrees Celsius and Fahrenheit and the condition in text format, using the response body again.

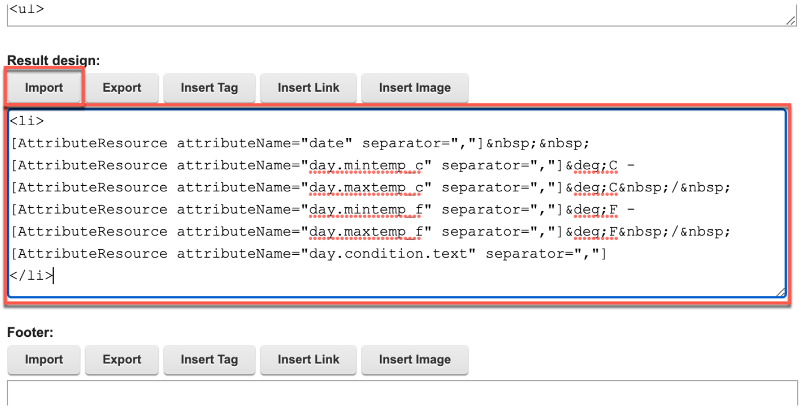
> This is how this data appears in the response body from the WeatherAPI forecast REST API call.



```
        "gust_kph": 20.9
    },
    "forecast": {
        "forecastday": [
            {
                "date": "2023-03-29",
                "date_epoch": 1680048000,
                "day": {
                    "maxtemp_c": 14.7,
                    "maxtemp_f": 58.5,
                    "mintemp_c": 7.6,
                    "mintemp_f": 45.7,
                    "avgtemp_c": 11.3,
                    "avgtemp_f": 52.3,
                    "maxwind_mph": 19.7,
                    "maxwind_kph": 31.7,
                    "totalprecip_mm": 3.0,
                    "totalprecip_in": 0.12,
                    "totalsnow_cm": 0.0,
                    "avgvis_km": 6.5,
                    "avgvis_miles": 4.0,
                    "avghumidity": 89.0,
                    "daily_will_it_rain": 1,
                    "daily_chance_of_rain": 86,
                    "daily_will_it_snow": 0,
                    "daily_chance_of_snow": 0,
                    "condition": {
                        "text": "Patchy rain possible",
                        "icon": "//cdn.weatherapi.com/weather/64x64/day/176.png",
                        "code": 1063
                    },
                    "uv": 3.0
                },
                "astro": {
```

You will manage that in the result design. You may use the same approach as above or copy this code or import it from the downloaded resources into your **Result Design**.

```
<li>

[AttributeResource attributeName="date" separator=","]&nbsp;&nbsp;

[AttributeResource attributeName="day.mintemp_c" separator=","]&deg;C - [AttributeResource attributeName="day.maxtemp_c" separator=","]&deg;C&nbsp;/&nbsp;

[AttributeResource attributeName="day.mintemp_f" separator=","]&deg;F - [AttributeResource attributeName="day.maxtemp_f" separator=","]&deg;F&nbsp;/&nbsp;

[AttributeResource attributeName="day.condition.text" separator=","]

</li>
```



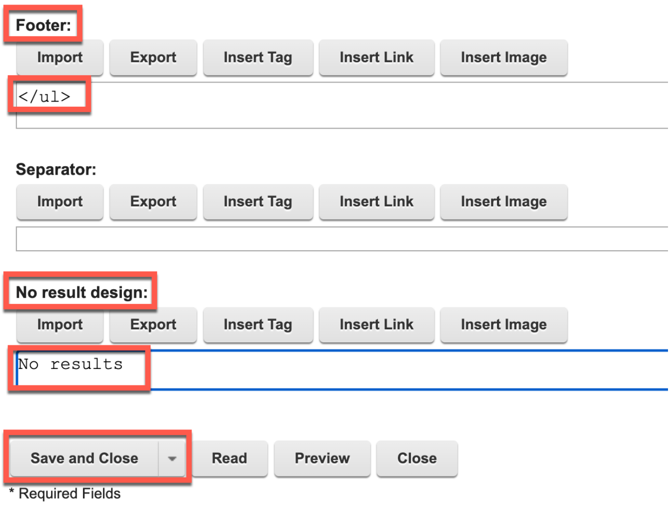
```
<ul>
```

```
<li>
[AttributeResource attributeName="date" separator=","]&nbsp;&nbsp;
[AttributeResource attributeName="day.mintemp_c" separator=","]&deg;C -
[AttributeResource attributeName="day.maxtemp_c" separator=","]&deg;C&nbsp;/&nbsp;
[AttributeResource attributeName="day.mintemp_f" separator=","]&deg;F -
[AttributeResource attributeName="day.maxtemp_f" separator=","]&deg;F&nbsp;/&nbsp;
[AttributeResource attributeName="day.condition.text" separator=","]
</li>
```

44. In the **Footer** section, you need to close the list of forecast day details using

```
</ul>
```

45. There is no separator needed here. Add an information to show if there is no forecast data in the **No result design** section, like **No Results.** Then click **Save and Close** to save your first **List Appearance**.



Footer:

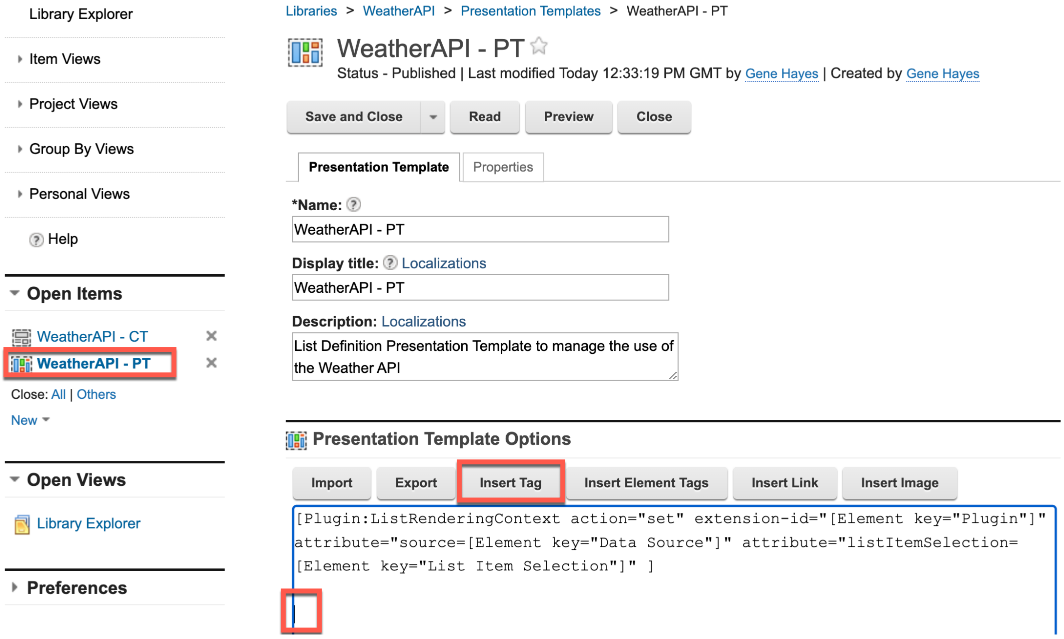
```
</ul>
```

Separator:

No result design:

```
No results
```

46. Now that you have a List Appearance, add this to your Presentation Template. Click under Open Items (if you kept it open), **WeatherAPI – PT**, move your cursor under the current code in Presentation Template Options and click **Insert Tag**.



WeatherAPI - PT

Status - Published | Last modified Today 12:33:19 PM GMT by Gene Hayes | Created by Gene Hayes

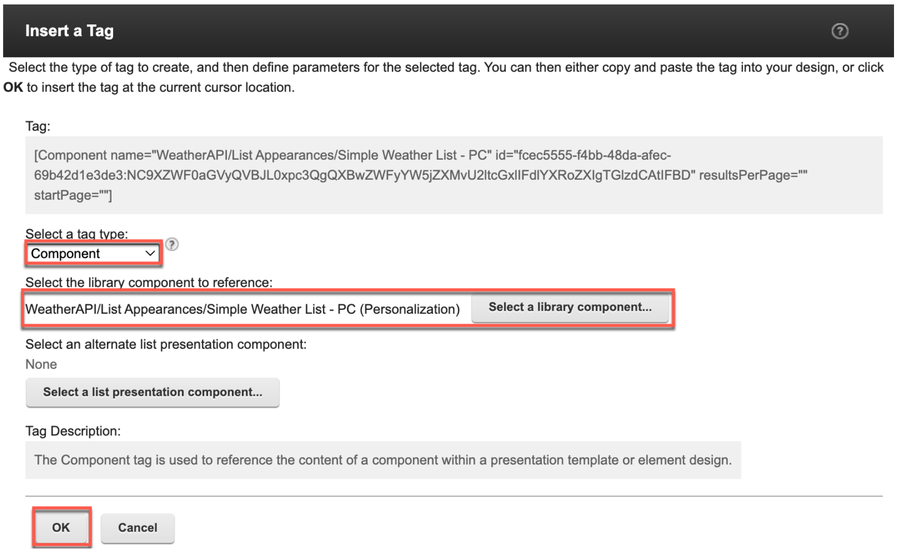
*Name: WeatherAPI - PT

Display title: Localizations WeatherAPI - PT

Description: Localizations List Definition Presentation Template to manage the use of the Weather API

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]"
attribute="source=[Element key="Data Source"]" attribute="listItemSelection=
[Element key="List Item Selection"]" ]
```

47. Select **Component** as **tag type**, select your new Simple Weather List – PC component and click **OK**.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

[Component name="WeatherAPI/List Appearances/Simple Weather List - PC" id="fcec5555-f4bb-48da-afec-69b42d1e3de3:NC9XZWF0aGVyQVBJL0xpc3QgQXBwZWFyYW5jZXMvU2ltcGxlIFdlYXRoZXIgTGlzdCAtIFBD" resultsPerPage="" startPage=""]

Select a tag type: Component

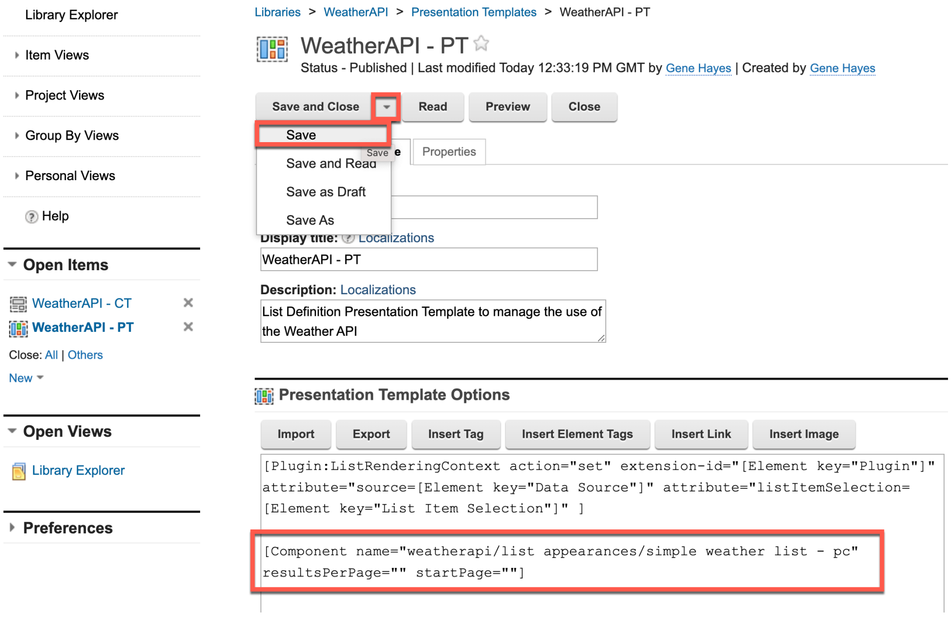
Select the library component to reference:
WeatherAPI/List Appearances/Simple Weather List - PC (Personalization)

Select an alternate list presentation component:
None

Tag Description:
The Component tag is used to reference the content of a component within a presentation template or element design.

Select a library component...
Select a list presentation component...
OK
Cancel

48. And click **Save and Close - Save** to save your changes.



Libraries > WeatherAPI > Presentation Templates > WeatherAPI - PT

WeatherAPI - PT

Status - Published | Last modified Today 12:33:19 PM GMT by Gene Hayes | Created by Gene Hayes

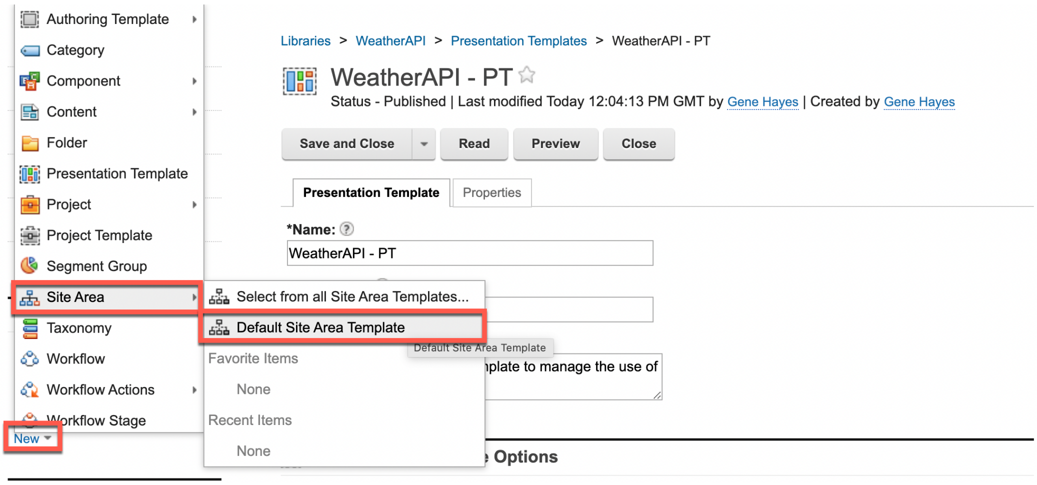
Display title: Localizations
WeatherAPI - PT

Description: Localizations
List Definition Presentation Template to manage the use of the Weather API

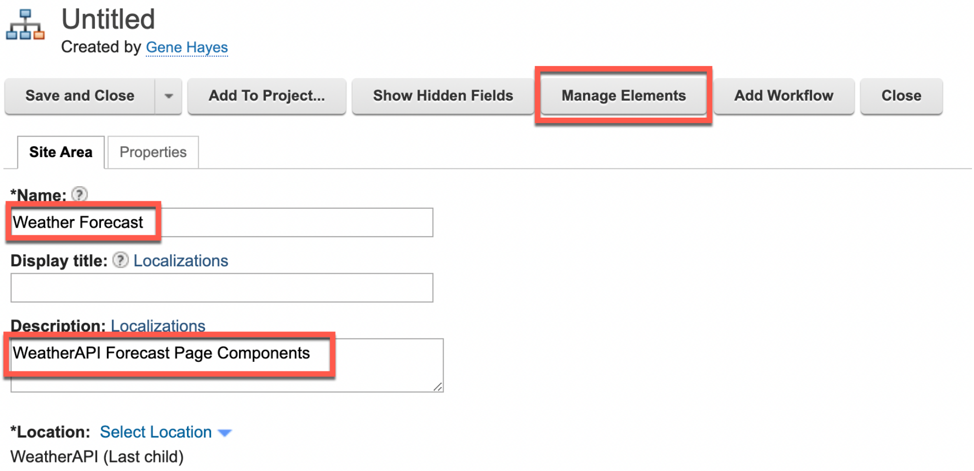
Presentation Template Options

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]"
attribute="source=[Element key="Data Source"]" attribute="listItemSelection=
[Element key="List Item Selection"]" ]

[Component name="weatherapi/list appearances/simple weather list - pc"
resultsPerPage="" startPage=""]
```

49. You want your business users to select a WeatherAPI component from the toolbar. You manage this be creating a site area that is configured to show in the toolbar. Click **New** – **Site Area** and **Default Site Area Template**.



50. **Name** it, e.g. **Weather Forecast**, add a **description**, e.g. **WeatherAPI Forecast Page Components** and click **Manage Elements** to add a nice preview image. You may use the one from the downloaded resources.



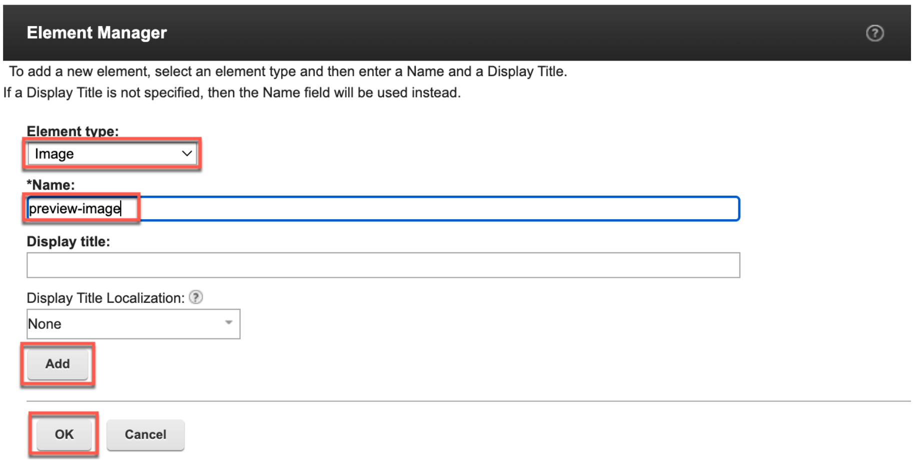
*Name: Weather Forecast

Display title:

Description: WeatherAPI Forecast Page Components

*Location: WeatherAPI (Last child)

51. Select **Image** as **Element type**, **name** it **preview-image**, click **Add** and then **OK**.



To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

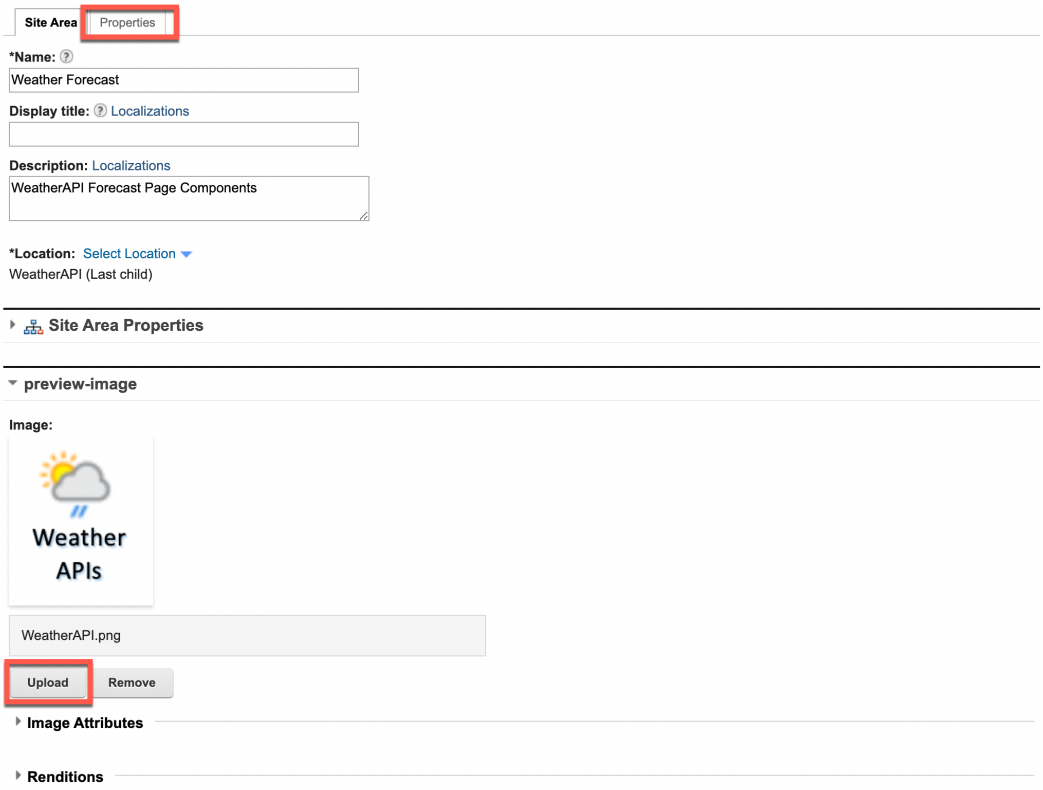
Element type: Image

*Name: preview-image

Display title:

Display Title Localization: None

52. Then use **Upload** to add a nice icon. You may use the WeatherAPI.png that you downloaded. To make this appear under the toolbar for your users, you need to add a special keyword ibm.portal.toolbar.NewContent that you will find under properties. Click **Properties**.



*Name: Weather Forecast

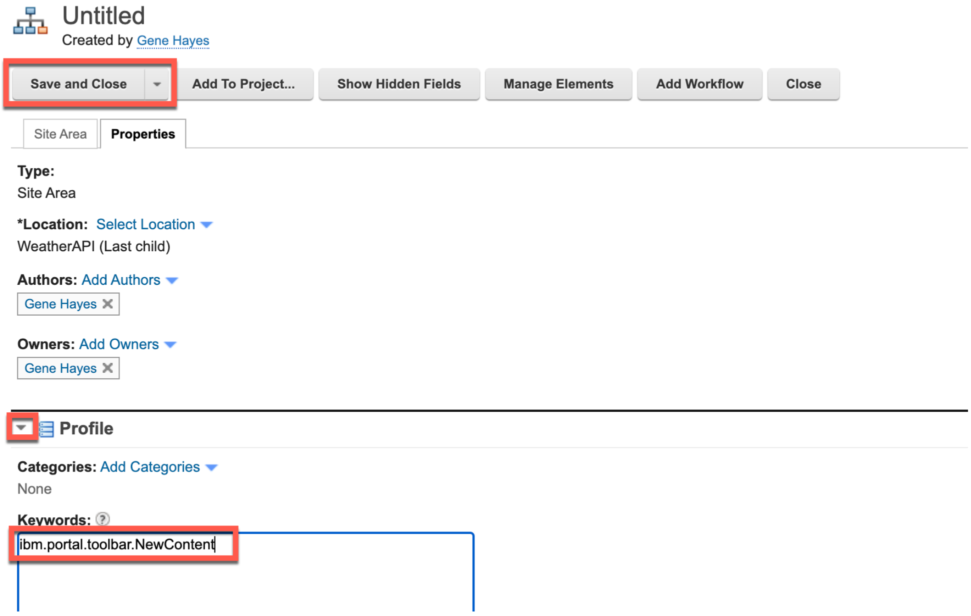
Display title: Localizations

Description: Localizations
WeatherAPI Forecast Page Components

*Location: Select Location
WeatherAPI (Last child)

Image: WeatherAPI.png

53. Open the **Profile** section and add **ibm.portal.toolbar.NewContent** in **Keywords**. Then click **Save and Close**.



Type:
Site Area

*Location: Select Location
WeatherAPI (Last child)

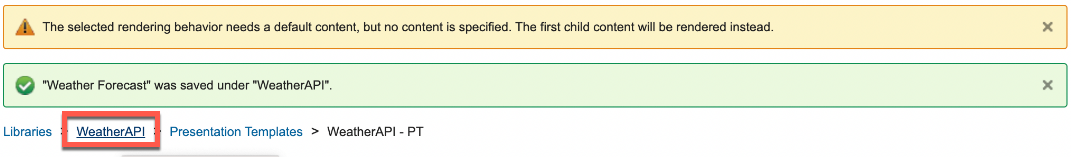
Authors: Add Authors
Gene Hayes

Owners: Add Owners
Gene Hayes

Categories: Add Categories
None

Keywords: ibm.portal.toolbar.NewContent

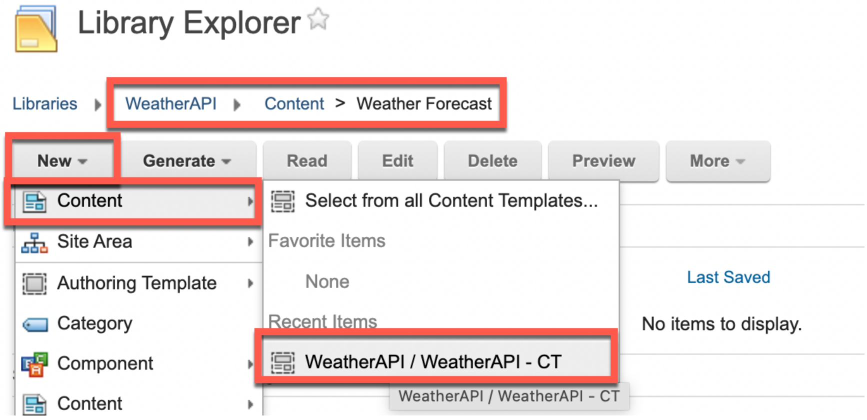
54. Then create a List Definition content for the Weather Forecast, based on your content template. Go to your new site area first. Click **WeatherAPI** library.



The selected rendering behavior needs a default content, but no content is specified. The first child content will be rendered instead.

"Weather Forecast" was saved under "WeatherAPI".

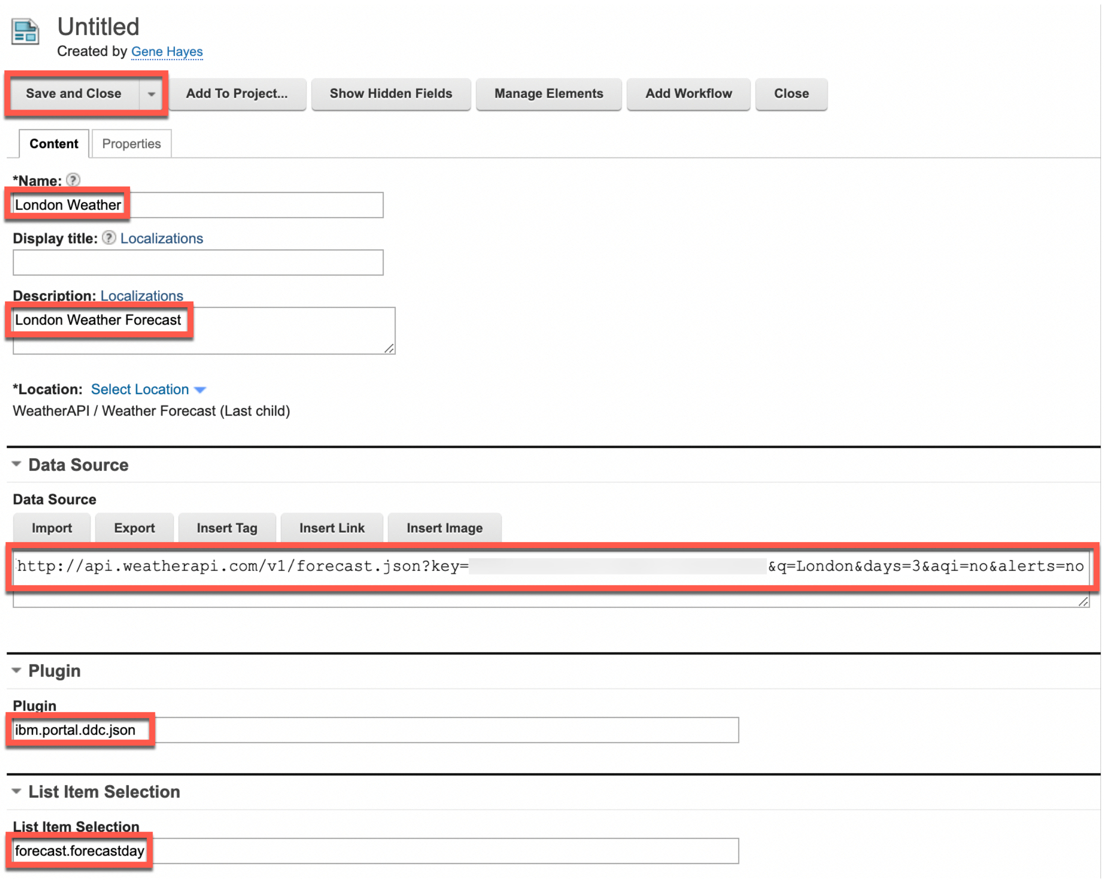
55. Then navigate to **Content** – **Weather Forecast** and click **New** – **Content** – **WeatherAPI/WeatherAPI – CT**.



56. Enter a **Name London Weather** and **Description London Weather Forecast**. For the **Data Source** attribute, enter the WeatherAPI URL defined above and use your own Weather API KEY: **http://api.weatherapi.com/v1/forecast.json?key=<your-api-key>&q=London&days=3&aqi=no&alerts=no.**

> Notice that you use http here. If you want to use https, you need to authorize external access, by setting up a proxy and managing the certificates, explained in optional Part 6. In some other cases, optional part 6 is also needed.

As this is a JSON API, use **ibm.portal.ddc.json** for the **Plugin**. And set the **List Item Selection** option to **forecast.forecastday** to point to the right place in the JSON result. Optionally, you may add a preview image for this London Forecast as well, in the same way as for the Site Area. Then click **Save and Close**.



Untitled
Created by Gene Hayes

*Name: London Weather

Display title: Localizations

Description: Localizations
London Weather Forecast

*Location: Select Location
WeatherAPI / Weather Forecast (Last child)

Data Source

Data Source

```
http://api.weatherapi.com/v1/forecast.json?key=[redacted]&q=London&days=3&aqi=no&alerts=no
```

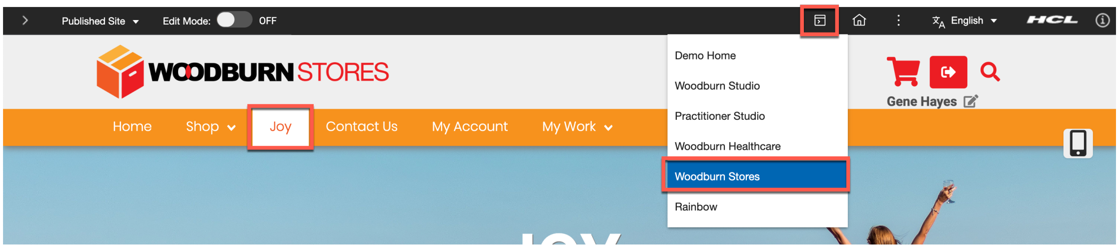
Plugin

Plugin: ibm.portal.ddc.json

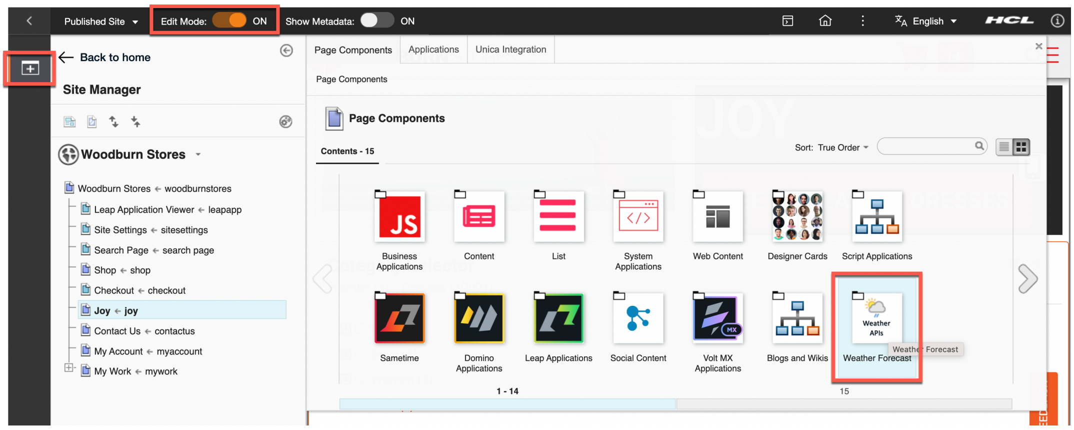
List Item Selection

List Item Selection: forecast.forecastday

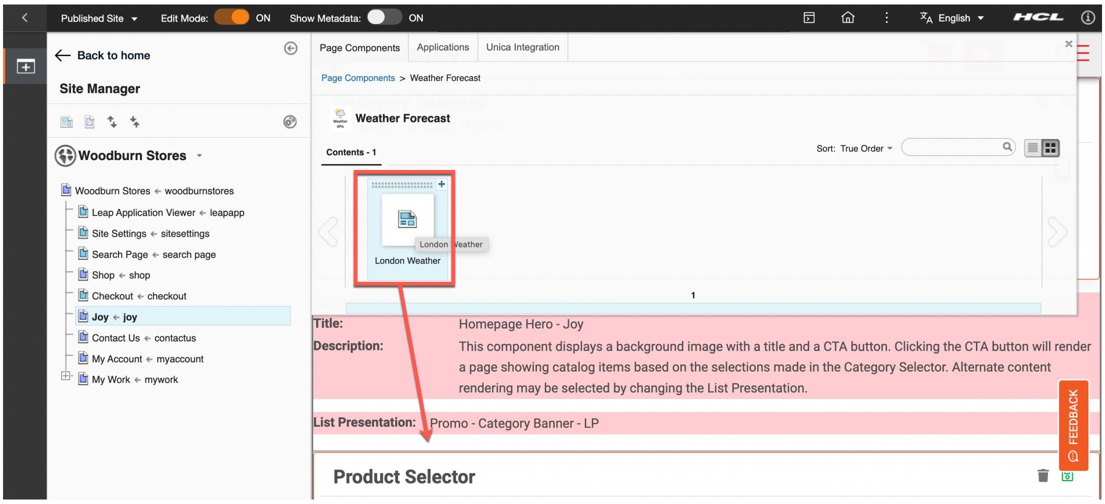
57. Now test it on a page in a DX site. Navigate to **Woodburn Stores** demo site and **Joy** page.



58. Then switch on the **Edit Mode**, click **Add page components and applications** and click your new **Weather Forecast** icon.



59. Drag & drop your **London Weather** page component right under the first page component **Homepage Hero - Joy.**

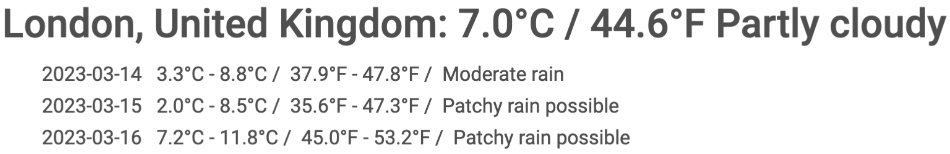


Title: Homepage Hero - Joy

Description: This component displays a background image with a title and a CTA button. Clicking the CTA button will render a page showing catalog items based on the selections made in the Category Selector. Alternate content rendering may be selected by changing the List Presentation.

List Presentation: Promo - Category Banner - LP

60. Switch off the edit mode and it now shows. If it shows no result, you likely do not have access to the APIs, and need to authorize external access, as described in optional part 6.



London, United Kingdom: 7.0°C / 44.6°F Partly cloudy

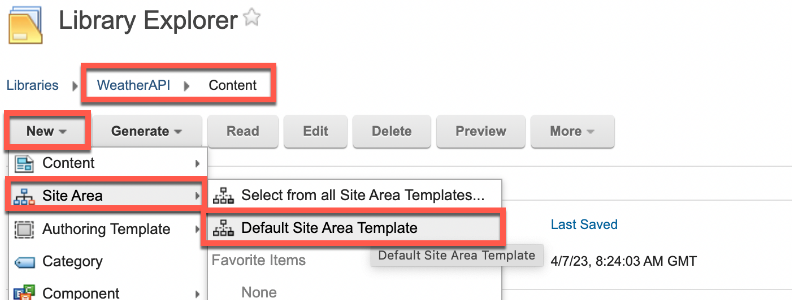
2023-03-14 3.3°C - 8.8°C / 37.9°F - 47.8°F / Moderate rain
2023-03-15 2.0°C - 8.5°C / 35.6°F - 47.3°F / Patchy rain possible
2023-03-16 7.2°C - 11.8°C / 45.0°F - 53.2°F / Patchy rain possible

Congratulations! You successfully created your first DDC component for the WeatherAPI.

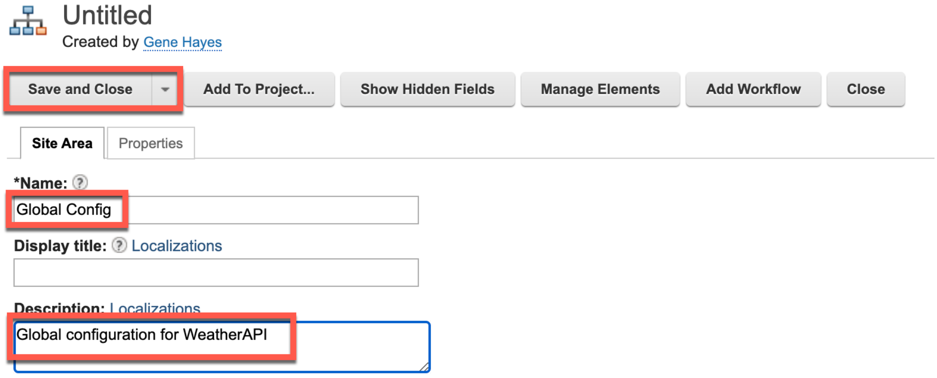
Part 3: Add Global Configuration

In this part of the lab, you will add a Global Configuration related to the Weather APIs integration to manage some Global Parameters, like the Weather API host URL, the Weather API Key (required to work with that) and some default values, for example, for the location and number of forecast days. If any of these changes, you only need to update it centrally.

1. You first create a Global Config site area in your library to store your global config content. As before, navigate to **WeatherAPI** - **Content**, click **New** - **Site Area** based on the **Default Site Area Template**.



2. Name it **Global Config.** Optionally give it a description, like **Global configuration for WeatherAPI**. Click **Save and Close**.



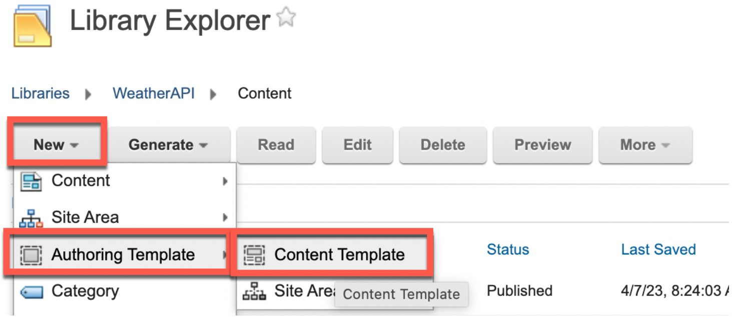
*Name: Global Config

Display title: Localizations

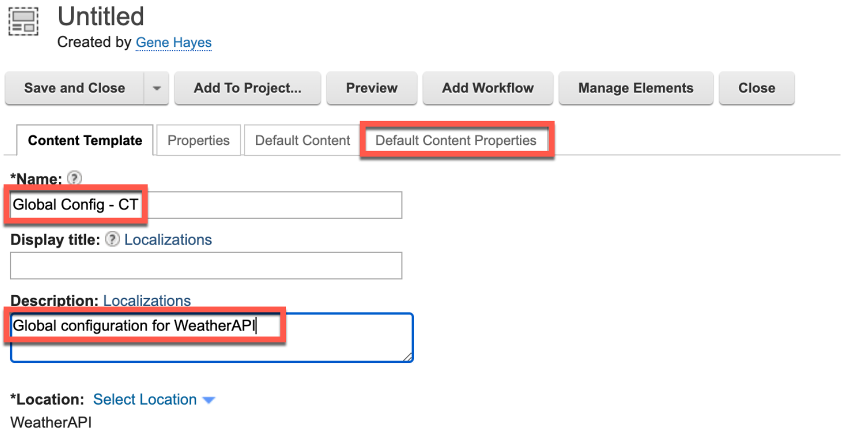
Description: Localizations

Global configuration for WeatherAPI

3. Then create the Global Config Content Template. Click **New** – **Authoring Template** – **Content Template**.



4. Name it **Global Config – CT** and give it a description, like **Global configuration for WeatherAPI**. Then remove the workflow. Click the **Default Content Properties**.



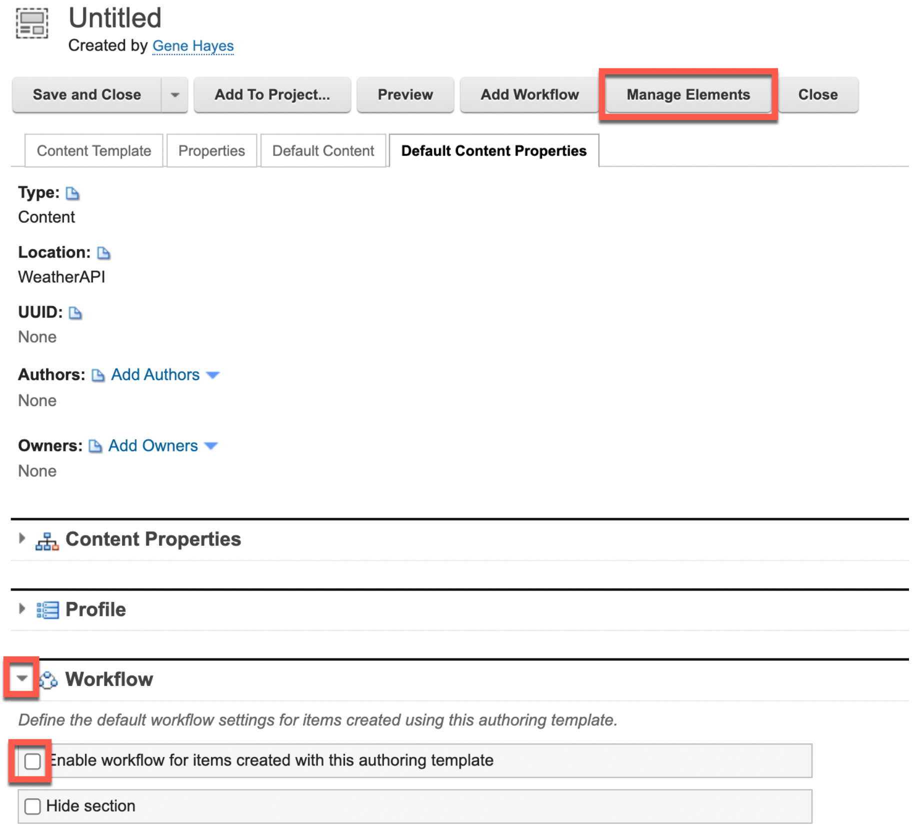
*Name: Global Config - CT

Display title: Localizations

Description: Localizations
Global configuration for WeatherAPI

*Location: WeatherAPI

5. Open the Workflow section and uncheck the Enable **workflow for items created with this authoring template**. Then add your elements. Click **Manage Elements**.



Type: Content

Location: WeatherAPI

UUID: None

Authors: None

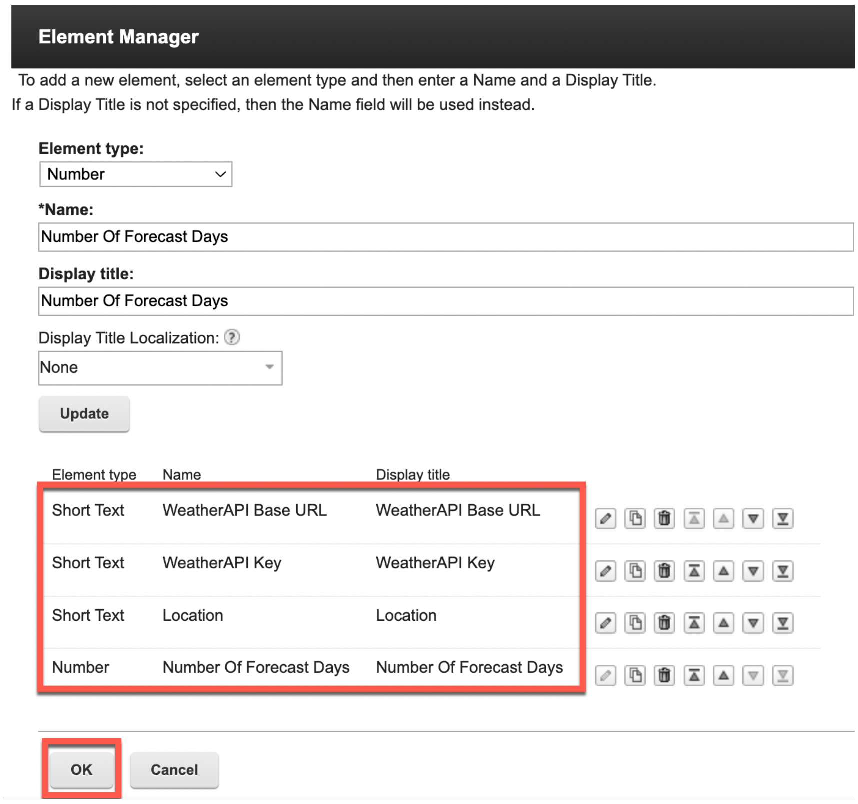
Owners: None

Define the default workflow settings for items created using this authoring template.

□ Enable workflow for items created with this authoring template

□ Hide section

6. And define the following list of elements **WeatherAPI Base URL**, **WeatherAPI Key**, **Location** short texts and **Number Of Days** number, and click **OK**.



Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

Element type: Number

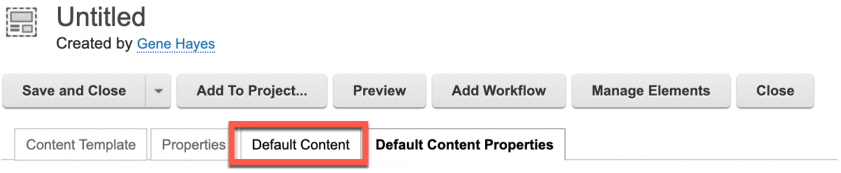
***Name:** Number Of Forecast Days

Display title: Number Of Forecast Days

Display Title Localization: None

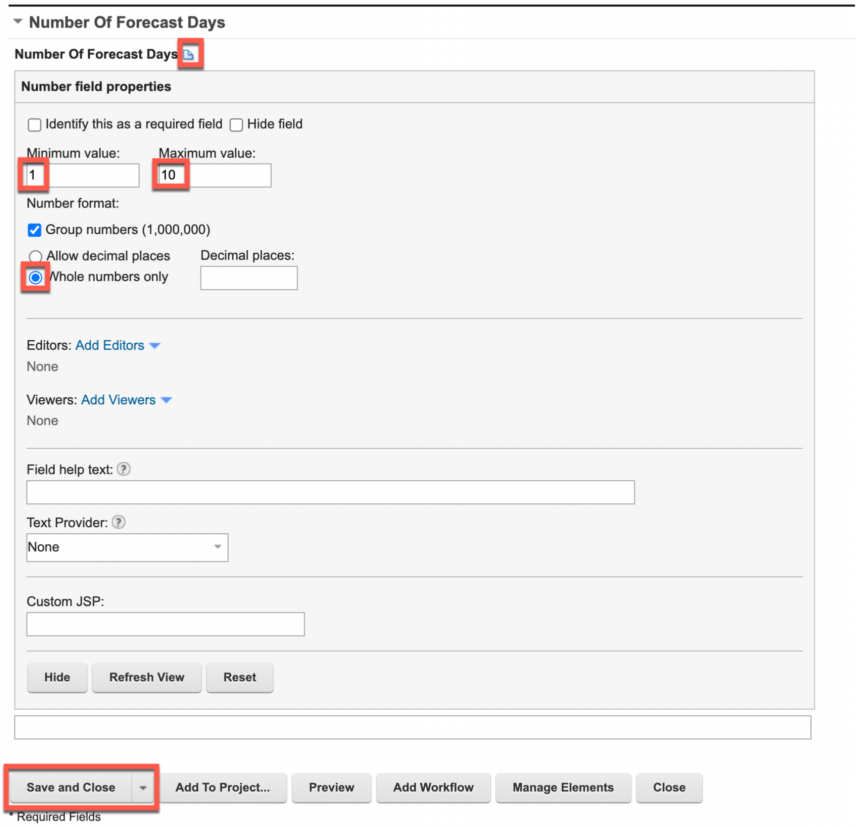
Element type	Name	Display title
Short Text	WeatherAPI Base URL	WeatherAPI Base URL
Short Text	WeatherAPI Key	WeatherAPI Key
Short Text	Location	Location
Number	Number Of Forecast Days	Number Of Forecast Days

7. If you look at the Weather APIs Explorer, you may notice that the number of forecast days must be between 1 and 10 days. You may configure this as a validation rule. Click **Default Content**.



Untitled
Created by Gene Hayes

8. Then open the field properties of **Number Of Forecast Days**, set the **Minimum value** to **1** and **Maximum Value** to **10**, **check** the **Whole number only** radio button and then **Save and Close**.



Number Of Forecast Days

Number Of Forecast Days

Number field properties

□ Identify this as a required field □ Hide field

Minimum value: 1

Maximum value: 10

Number format:

☑ Group numbers (1,000,000)

○ Allow decimal places

● Whole numbers only

Decimal places:

Editors: Add Editors

None

Viewers: Add Viewers

None

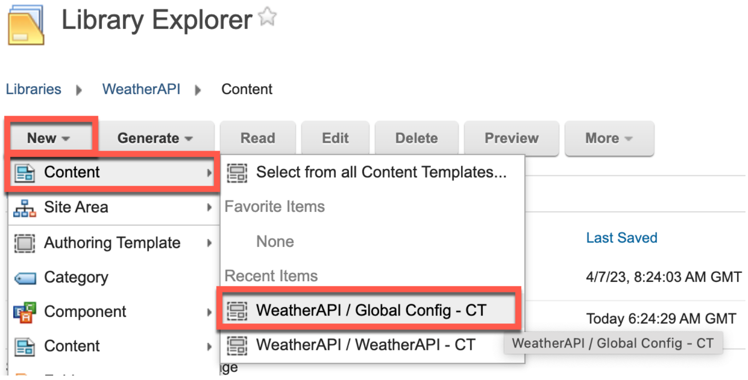
Field help text:

Text Provider: None

Custom JSP:

* Required Fields

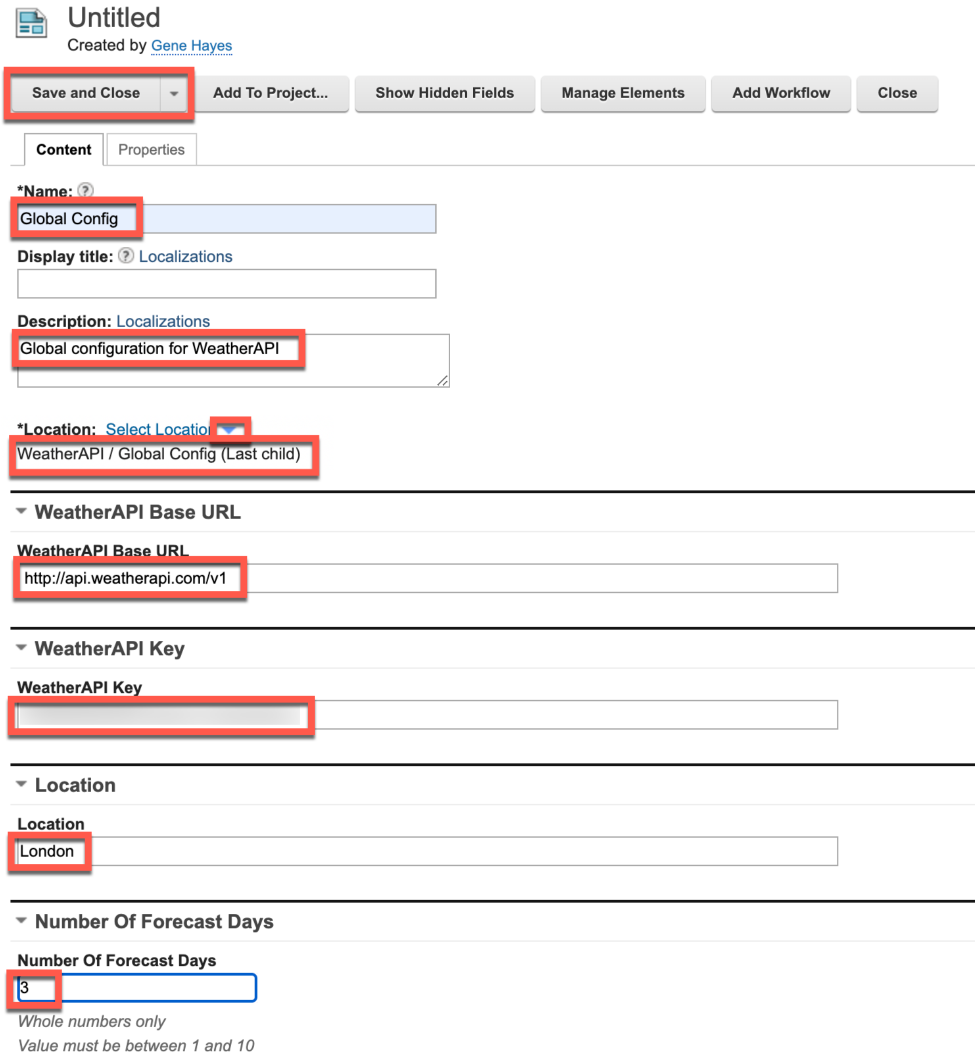
9. Then create the global config Content, based on this Content Template. Click **New** – **Content** – **WeatherAPI / Global Config – CT**.



Library Explorer

Libraries ▸ WeatherAPI ▸ Content

10. **Name** it **Global Config** and give it a clear **Description**, like **Global configuration for WeatherAPI**, and select **WeatherAPI / Global Config** as the **Location**. Then enter the base URL, you may get from https://www.weatherapi.com/docs/#intro-request: **http://api.weatherapi.com/v1**. Again, if you wish to use https, you should refer to part 7 in this lab. Add your **WeatherAPI Key**, default **Location**, e.g. **London** and enter **3** forecast days. Then click **Save and Close**.



*Name: Global Config

Display title: Localizations

Description: Localizations
Global configuration for WeatherAPI

*Location: Select Location
WeatherAPI / Global Config (Last child)

WeatherAPI Base URL

WeatherAPI Base URL: http://api.weatherapi.com/v1

WeatherAPI Key

WeatherAPI Key:

Location

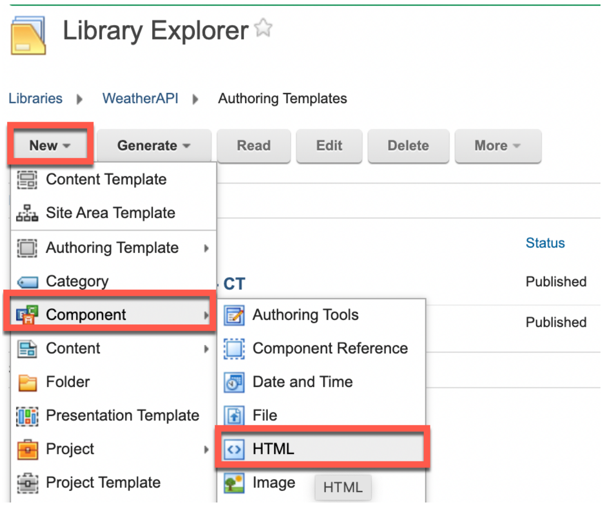
Location: London

Number Of Forecast Days

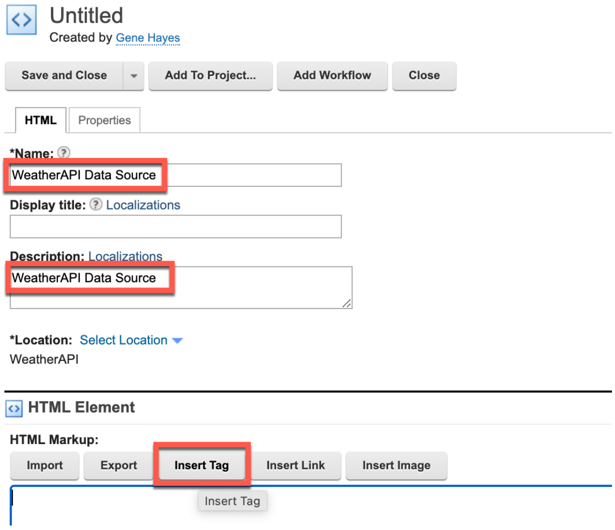
Number Of Forecast Days: 3

Whole numbers only
Value must be between 1 and 10

11. Now you need to update the data source to use these global configuration parameters. This is currently managed in an HTML element in each content. To make it easier to manage this for all content, you will manage this in a shared HTML Component that you will reference in your content instead. First create this Data Source HTML Component. Click **New** – **Component** – **HTML**.



12. **Name** it **WeatherAPI Data Source** and give it a **Description**. Then create the data source. Its format is *<WeatherAPI Base URL>/forecast.json?key=<WeatherAPI Base Key>&q=<Location&days=<Number of Forecast Days>&aqi=no&alerts=no*. Begin with adding the WeatherAPI API Base URL from your global config content. Click **Insert Tag**.



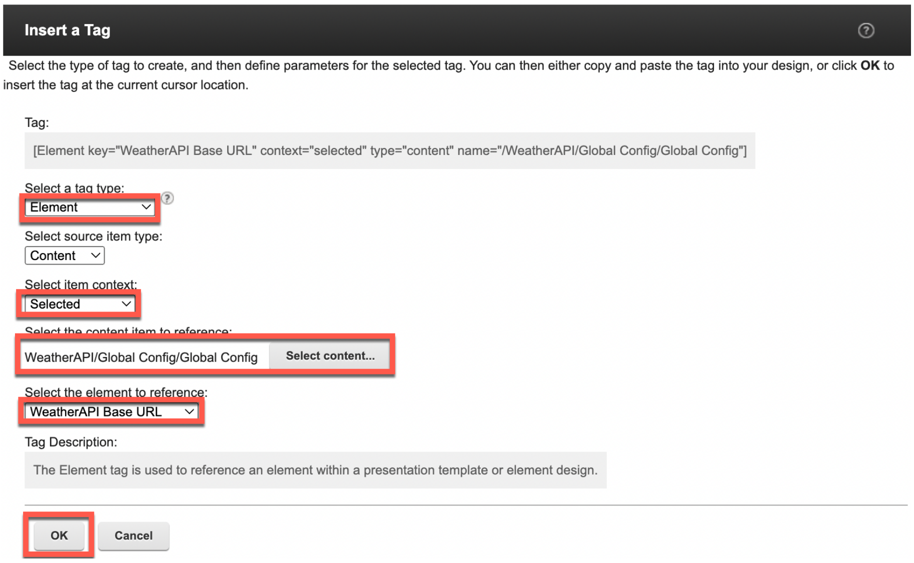
*Name: WeatherAPI Data Source

Display title: Localizations

Description: Localizations WeatherAPI Data Source

*Location: Select Location
WeatherAPI

13. Select **Element tag type**, then select **Selected** as **item context**, your **WeatherAPI/Global Config/Global Config content**, the **WeatherAPI Base URL element to reference** and click OK.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

[Element key="WeatherAPI Base URL" context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]

Select a tag type: Element

Select source item type: Content

Select item context: Selected

Select the content item to reference: WeatherAPI/Global Config/Global Config

Select the element to reference: WeatherAPI Base URL

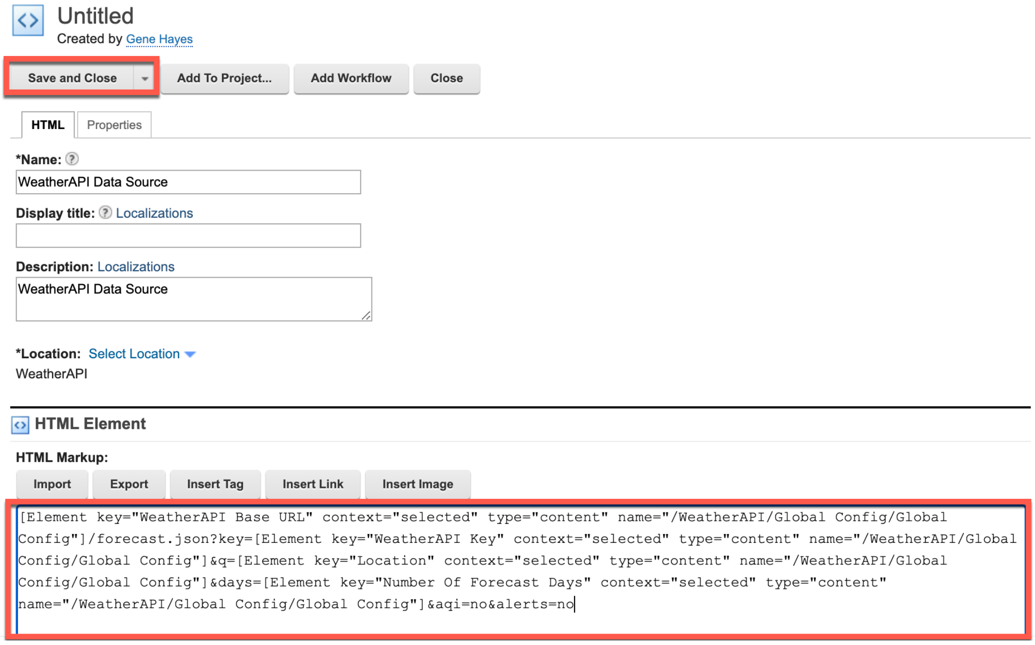
Tag Description:

The Element tag is used to reference an element within a presentation template or element design.

14. In the same way, construct the full URL. You may also copy it from here or from the downloaded resources.

[Element key="*WeatherAPI Base URL*" context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]/forecast.json?key=**[Element key="*WeatherAPI Key*" context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]**&q=**[Element key="*Location*" context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]**&days=**[Element key="*Number Of Forecast Days*" context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]**&aqi=no&alerts=no

Then click **Save and Close**.



Untitled

Created by Gene Hayes

*Name: WeatherAPI Data Source

Display title: Localizations

Description: Localizations WeatherAPI Data Source

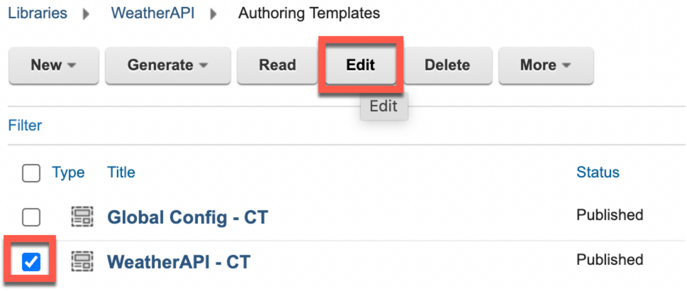
*Location: Select Location WeatherAPI

HTML Element

HTML Markup:

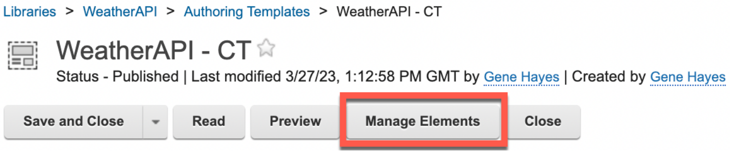
```
[Element key="WeatherAPI Base URL" context="selected" type="content" name="/WeatherAPI/Global Config/Global
Config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected" type="content" name="/WeatherAPI/Global
Config/Global Config"]&q=[Element key="Location" context="selected" type="content" name="/WeatherAPI/Global
Config/Global Config"]&days=[Element key="Number Of Forecast Days" context="selected" type="content"
name="/WeatherAPI/Global Config/Global Config"]&aqi=no&alerts=no
```

15. Then you need to update your WeatherAPI – CT Content Template to be able to select this component. Select it and click **Edit**.


Libraries ▸ WeatherAPI ▸ Authoring Templates

	Type	Title	Status
□		Global Config - CT	Published
☑		WeatherAPI - CT	Published

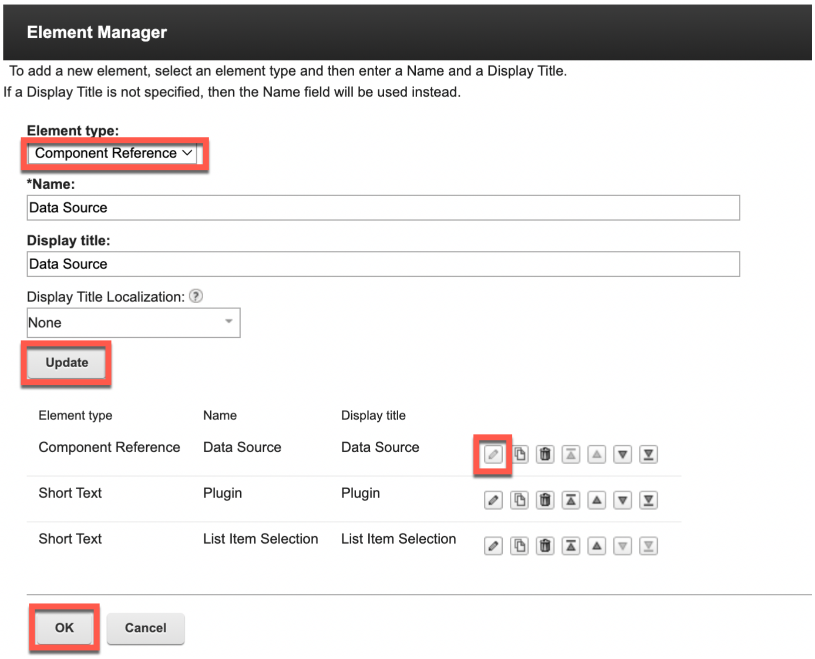
16. Click **Manage Elements**.


Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Last modified 3/27/23, 1:12:58 PM GMT by Gene Hayes | Created by Gene Hayes

17. Then click the **edit button** of the **Data Source** element, change the **Element type** to **Component Reference**, click **Update** and **OK**.


Element Manager

To add a new element, select an element type and then enter a Name and a Display Title.
If a Display Title is not specified, then the Name field will be used instead.

Element type: Component Reference

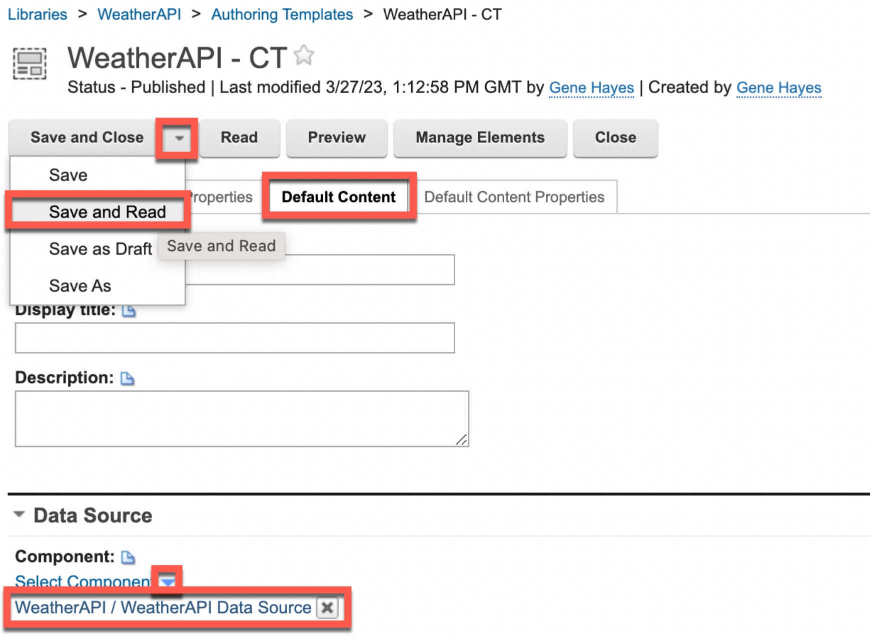
*Name: Data Source

Display title: Data Source

Display Title Localization: None

Element type	Name	Display title
Component Reference	Data Source	Data Source
Short Text	Plugin	Plugin
Short Text	List Item Selection	List Item Selection

18. Then update this element and set its default value to your new component. Click **Default Content** tab, select the **WeatherAPI / WeatherAPI Data Source component,** and **Save and Read**.



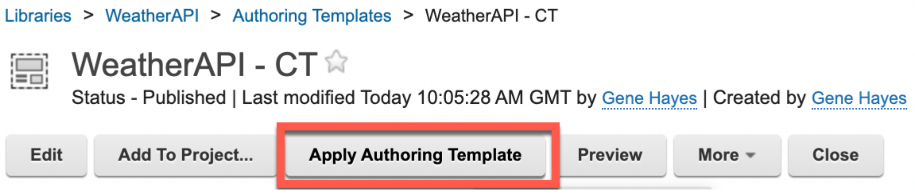
Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Last modified 3/27/23, 1:12:58 PM GMT by Gene Hayes | Created by Gene Hayes

Component: WeatherAPI / WeatherAPI Data Source

19. As this allows you to apply these changes to your existing two contents. Click **Apply Authoring Template**.

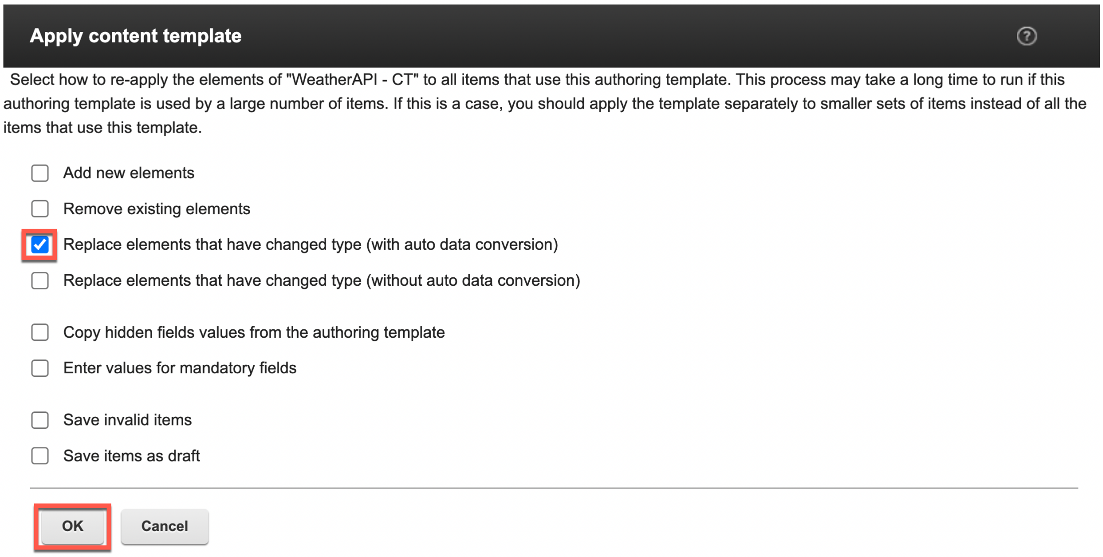


Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Last modified Today 10:05:28 AM GMT by Gene Hayes | Created by Gene Hayes

20. You want to replace the current HTML by this default component. **Check Replace elements that have changed type (with auto data conversion)** and click **OK**.



Apply content template

Select how to re-apply the elements of "WeatherAPI - CT" to all items that use this authoring template. This process may take a long time to run if this authoring template is used by a large number of items. If this is a case, you should apply the template separately to smaller sets of items instead of all the items that use this template.

☐ Add new elements

☐ Remove existing elements

☑ Replace elements that have changed type (with auto data conversion)

☐ Replace elements that have changed type (without auto data conversion)

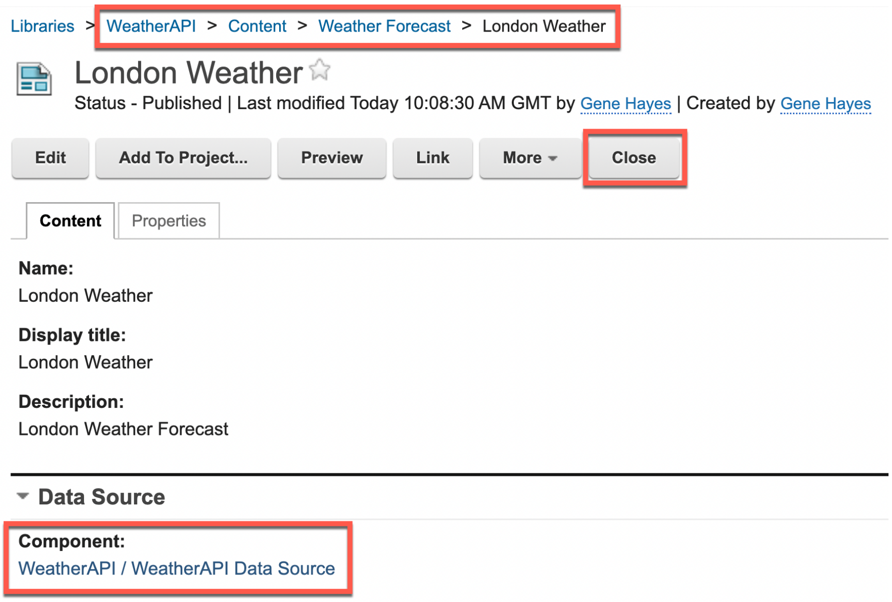
☐ Copy hidden fields values from the authoring template

☐ Enter values for mandatory fields

☐ Save invalid items

☐ Save items as draft

21. Check if your contents are updated correctly. Open your **WeatherAPI** – **Content** – **Weather Forecast** – **London Weather** content and check that the **Data Source** element now points to your new component. Then close it. Click **Close**.



Libraries > WeatherAPI > Content > Weather Forecast > London Weather

London Weather

Status - Published | Last modified Today 10:08:30 AM GMT by Gene Hayes | Created by Gene Hayes

Name:
London Weather

Display title:
London Weather

Description:
London Weather Forecast

Data Source

Component:
WeatherAPI / WeatherAPI Data Source

22. You may also check your Joy page and notice it still shows the same.

Congratulations! You successfully added a global configuration to your data source.

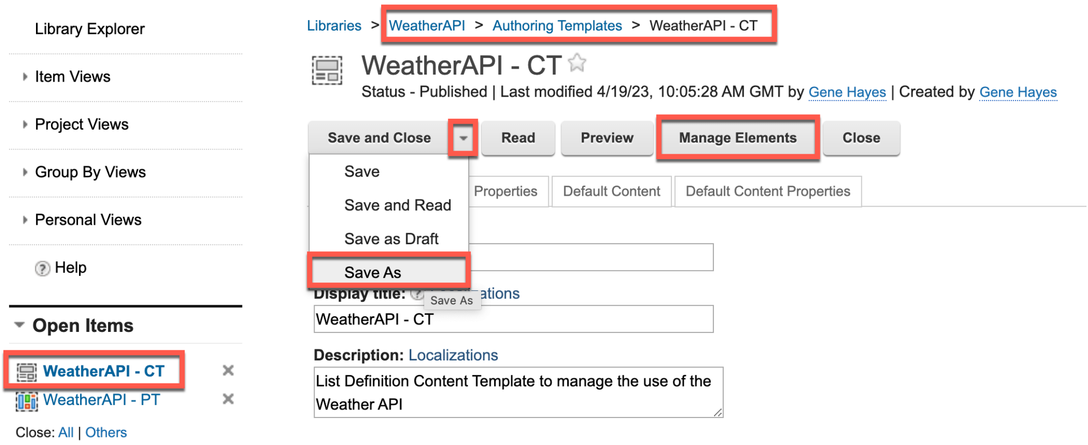
Part 4: Add User Configuration

Then you want your business user to control what to show by adding a configuration of the location and the number of forecast days that override the default values from the Global Parameters. You will add this to your List Definition content template, make it editable by them in the corresponding presentation template, update your content and test it.

1. First update your List Definition Content Template, the WeatherAPI – CT. In the Authoring Portlet, edit the **WeatherAPI – Authoring Templates – WeatherAPI – CT** Content Template.

> You may want to make a backup of the existing **WeatherAPI - CT** Content template using the **Save As** option and then open the original again.

Then add the new elements. Click **Manage Elements**.



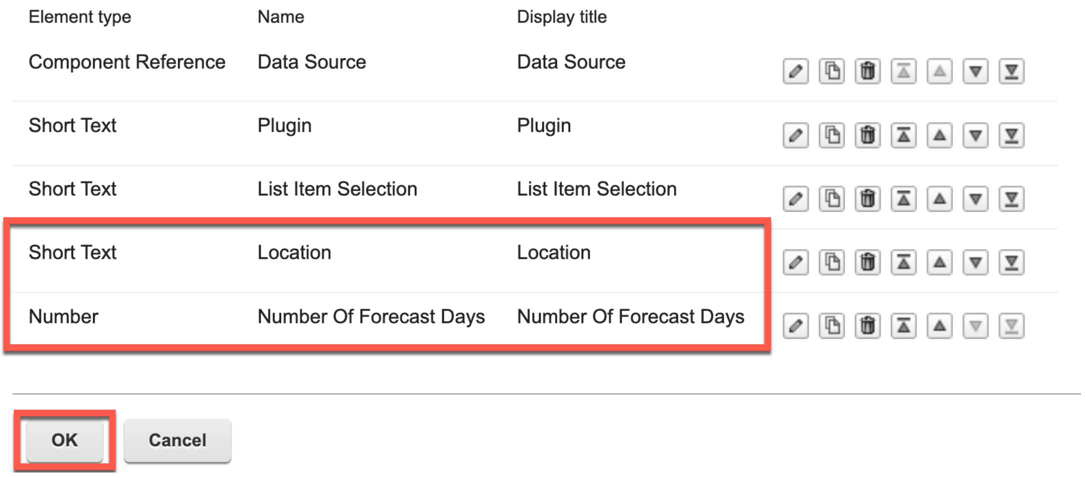
WeatherAPI - CT

Status - Published | Last modified 4/19/23, 10:05:28 AM GMT by Gene Hayes | Created by Gene Hayes

Display title: WeatherAPI - CT

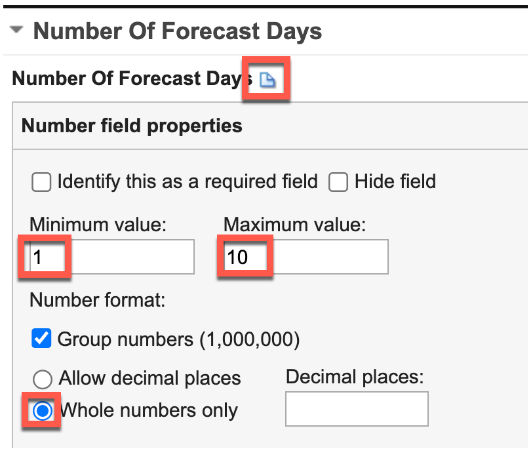
Description: List Definition Content Template to manage the use of the Weather API

1. Add the new elements **Location (short text)** and **Number Of Forecast Days (number)** and click **OK** to save the updates.



Element type	Name	Display title
Component Reference	Data Source	Data Source
Short Text	Plugin	Plugin
Short Text	List Item Selection	List Item Selection
Short Text	Location	Location
Number	Number Of Forecast Days	Number Of Forecast Days

2. Then, add the same validation for Number Of Forecast Days attribute, as you did in the Global Config Content Template. Under **Default Content**, set the **minimum value** to **1**, **maximum** to **10** and **use whole numbers** for **Number Of Forecast Days**.



Number Of Forecast Days

Number Of Forecast Days

Number field properties

□ Identify this as a required field □ Hide field

Minimum value: 1

Maximum value: 10

Number format:

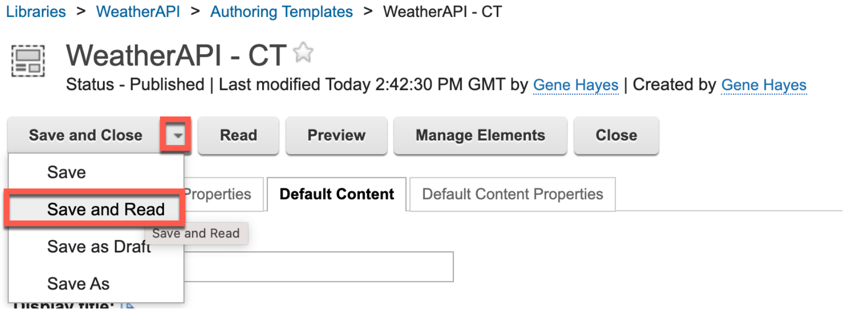
☑ Group numbers (1,000,000)

○ Allow decimal places

● Whole numbers only

Decimal places:

3. Then click **Save and Read** as this allows you to update the existing content.

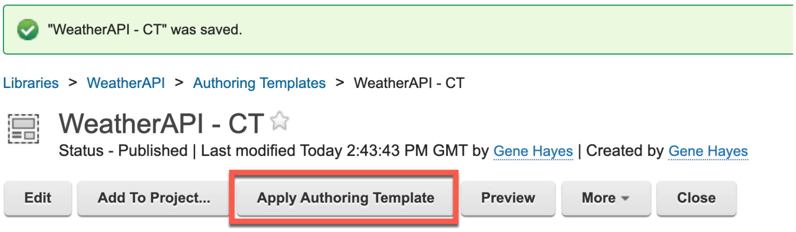


Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Last modified Today 2:42:30 PM GMT by Gene Hayes | Created by Gene Hayes

4. Now apply these changes to your existing content. Click **Apply Authoring Template**.



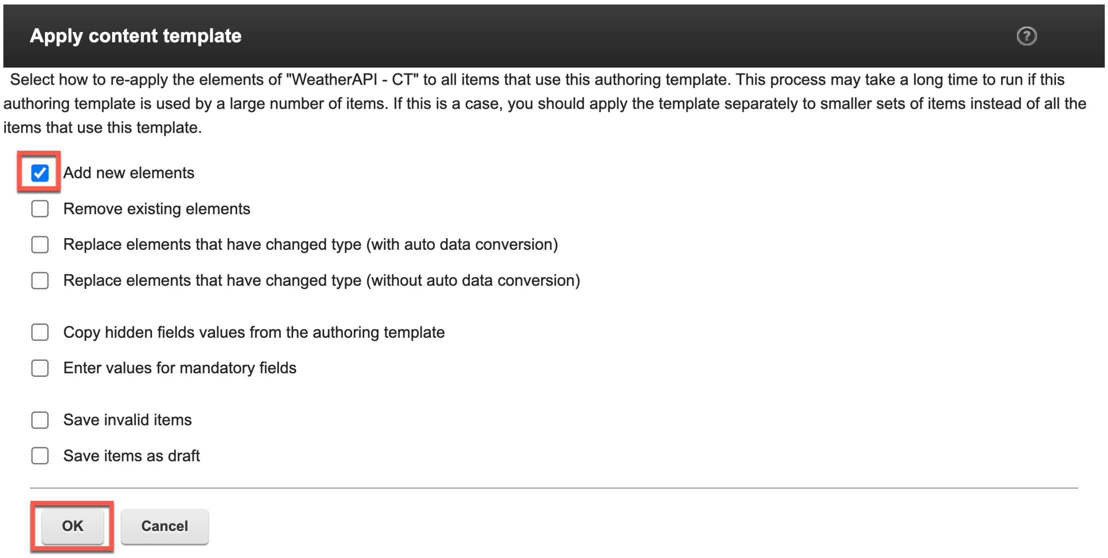
"WeatherAPI - CT" was saved.

Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

Status - Published | Last modified Today 2:43:43 PM GMT by Gene Hayes | Created by Gene Hayes

5. Then select **Add new elements** and click **OK**.



Apply content template

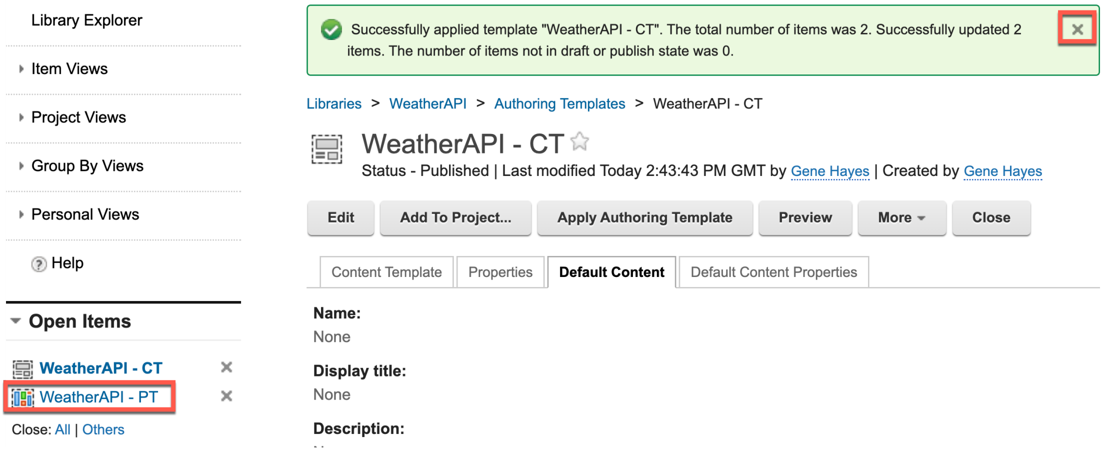
Select how to re-apply the elements of "WeatherAPI - CT" to all items that use this authoring template. This process may take a long time to run if this authoring template is used by a large number of items. If this is a case, you should apply the template separately to smaller sets of items instead of all the items that use this template.

☑ Add new elements
□ Remove existing elements
□ Replace elements that have changed type (with auto data conversion)
□ Replace elements that have changed type (without auto data conversion)

□ Copy hidden fields values from the authoring template
□ Enter values for mandatory fields

□ Save invalid items
□ Save items as draft

6. Notice it updated your two contents, one for the page component and the one you added to the Joy page. Now make them editable for the business user when they are in edit mode. **Edit the WeatherAPI – PT** Presentation Template.



Successfully applied template "WeatherAPI - CT". The total number of items was 2. Successfully updated 2 items. The number of items not in draft or publish state was 0.

Libraries > WeatherAPI > Authoring Templates > WeatherAPI - CT

WeatherAPI - CT

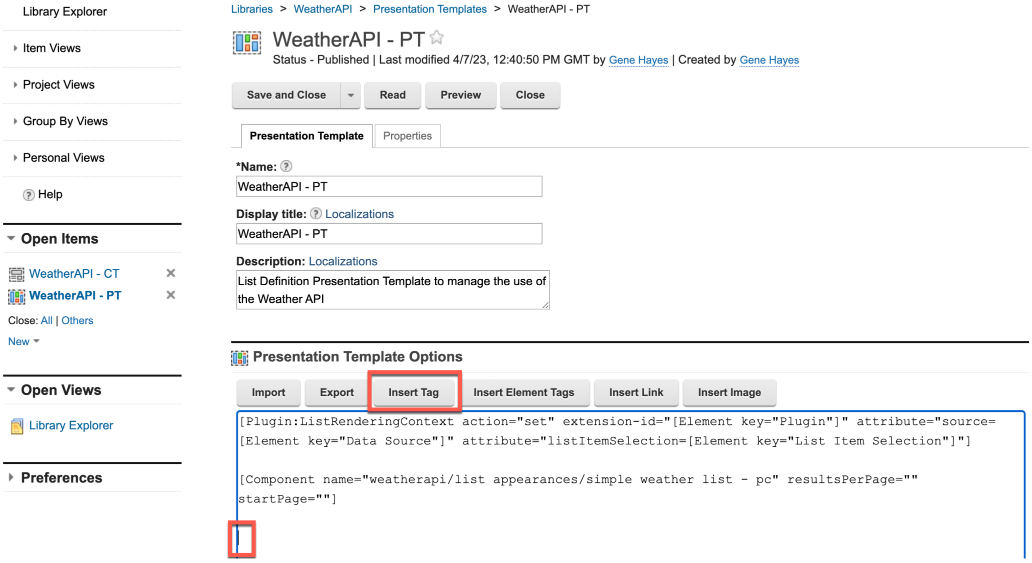
Status - Published | Last modified Today 2:43:43 PM GMT by Gene Hayes | Created by Gene Hayes

Name:
None

Display title:
None

Description:

7. First add a tag to manage code for edit mode only. Go below you code and click **Insert Tag**.



Libraries > WeatherAPI > Presentation Templates > WeatherAPI - PT

WeatherAPI - PT

Status - Published | Last modified 4/7/23, 12:40:50 PM GMT by Gene Hayes | Created by Gene Hayes

*Name: WeatherAPI - PT

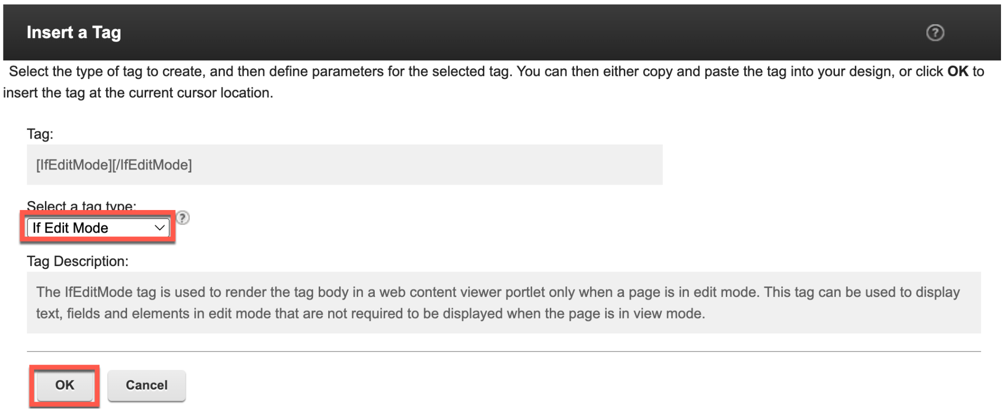
Display title: Localizations WeatherAPI - PT

Description: Localizations List Definition Presentation Template to manage the use of the Weather API

Presentation Template Options

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]" attribute="source=
[Element key="Data Source"]" attribute="listItemSelection=[Element key="List Item Selection"]"]

[Component name="weatherapi/list appearances/simple weather list - pc" resultsPerPage=""
startPage=""]
```

8. Then select **tag type If Edit Mode** and click **OK**.



Insert a Tag

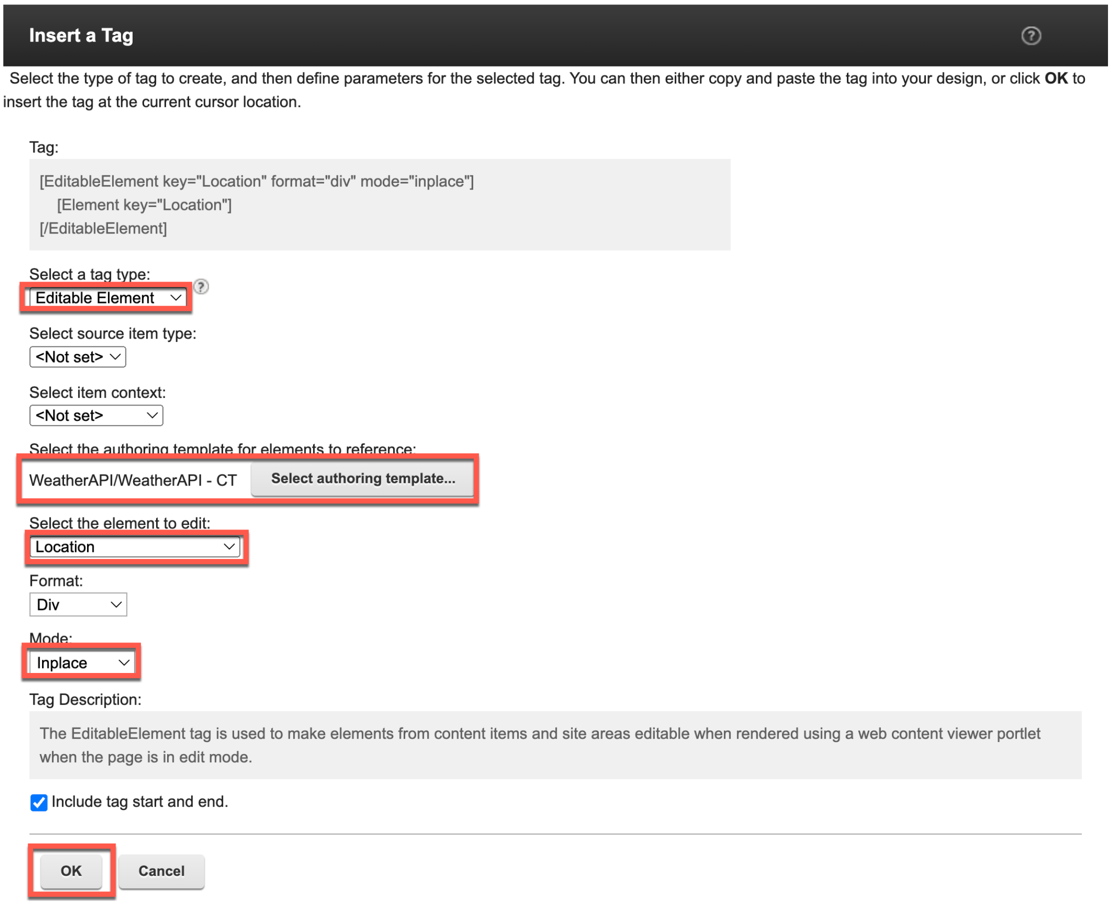
Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:
[IfEditMode][/IfEditMode]

Select a tag type: If Edit Mode

Tag Description:
The IfEditMode tag is used to render the tag body in a web content viewer portlet only when a page is in edit mode. This tag can be used to display text, fields and elements in edit mode that are not required to be displayed when the page is in view mode.

9. Then add an editable element for both location and number of forecast days, again using the Insert Tag. Select **Editable Element tag type**, the **WeatherAPI/WeatherAPI – CT content template**, **Location** and later **Number Of Forecast Days**, **mode Inplace** and **OK**.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:
```
[EditableElement key="Location" format="div" mode="inplace"]
    [Element key="Location"]
[/EditableElement]
```

Select a tag type: Editable Element

Select source item type: <Not set>

Select item context: <Not set>

Select the authoring template for elements to reference: WeatherAPI/WeatherAPI - CT

Select the element to edit: Location

Format: Div

Mode: Inplace

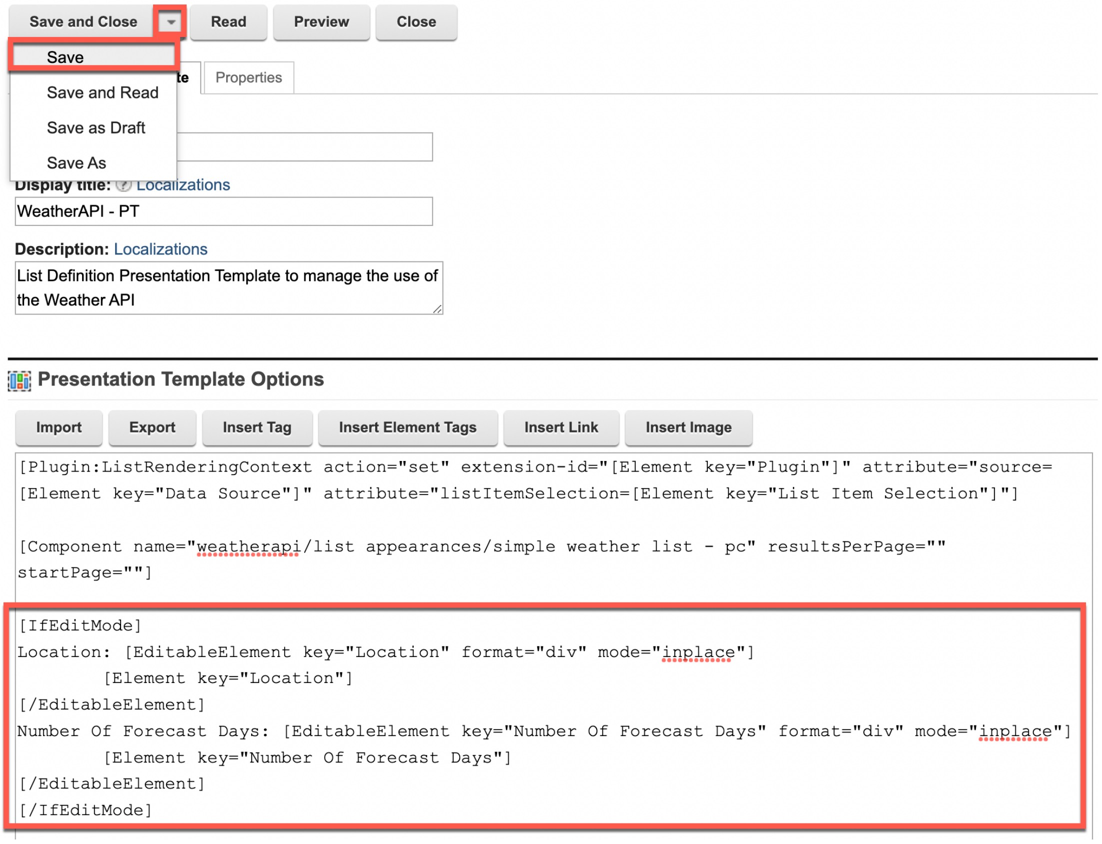
Tag Description:
The EditableElement tag is used to make elements from content items and site areas editable when rendered using a web content viewer portlet when the page is in edit mode.

☑ Include tag start and end.

10. Update it to show what elements you are editing, e.g. add **Location:** before the element tag. You may also get it from the downloaded resources.

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]" attribute="source=[Element key="Data Source"]" attribute="listItemSelection=[Element key="List Item Selection"]"]

[Component name="weatherapi/list appearances/simple weather list - pc" resultsPerPage="" startPage=""]


[IfEditMode]

Location: [EditableElement key="Location" format="div" mode="inplace"]

        [Element key="Location"]

[/EditableElement]

Number Of Forecast Days: [EditableElement key="Number Of Forecast Days" format="div" mode="inplace"]

        [Element key="Number Of Forecast Days"]

[/EditableElement]

[/IfEditMode]
```

This should give you the following. Save it, using **Save**.

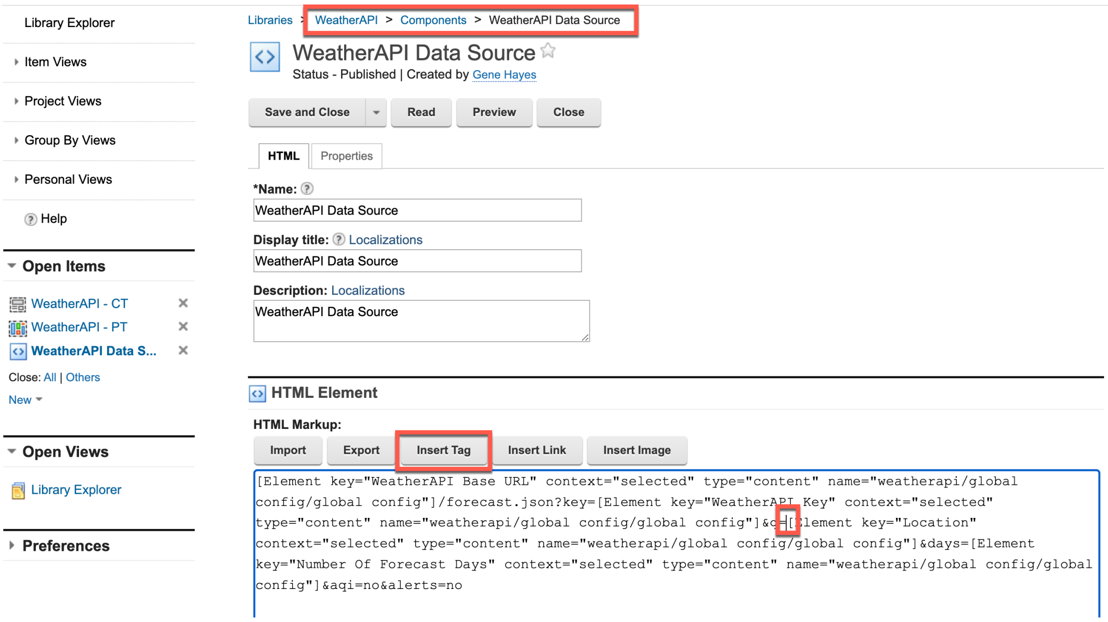


Display title: WeatherAPI - PT

Description: List Definition Presentation Template to manage the use of the Weather API

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]" attribute="source=
[Element key="Data Source"]" attribute="listItemSelection=[Element key="List Item Selection"]"]

[Component name="weatherapi/list appearances/simple weather list - pc" resultsPerPage=""
startPage=""]

[IfEditMode]
Location: [EditableElement key="Location" format="div" mode="inplace"]
        [Element key="Location"]
[/EditableElement]
Number Of Forecast Days: [EditableElement key="Number Of Forecast Days" format="div" mode="inplace"]
        [Element key="Number Of Forecast Days"]
[/EditableElement]
[/IfEditMode]
```

11. Then update the data source HTML component to use the additional parameters. Edit it and update the q parameter for location. Use **Insert Tag** before the Location element to determine if the user configuration is set, or if it should continue to use the default value.



WeatherAPI Data Source

Status - Published | Created by Gene Hayes

*Name: WeatherAPI Data Source

Display title: Localizations WeatherAPI Data Source

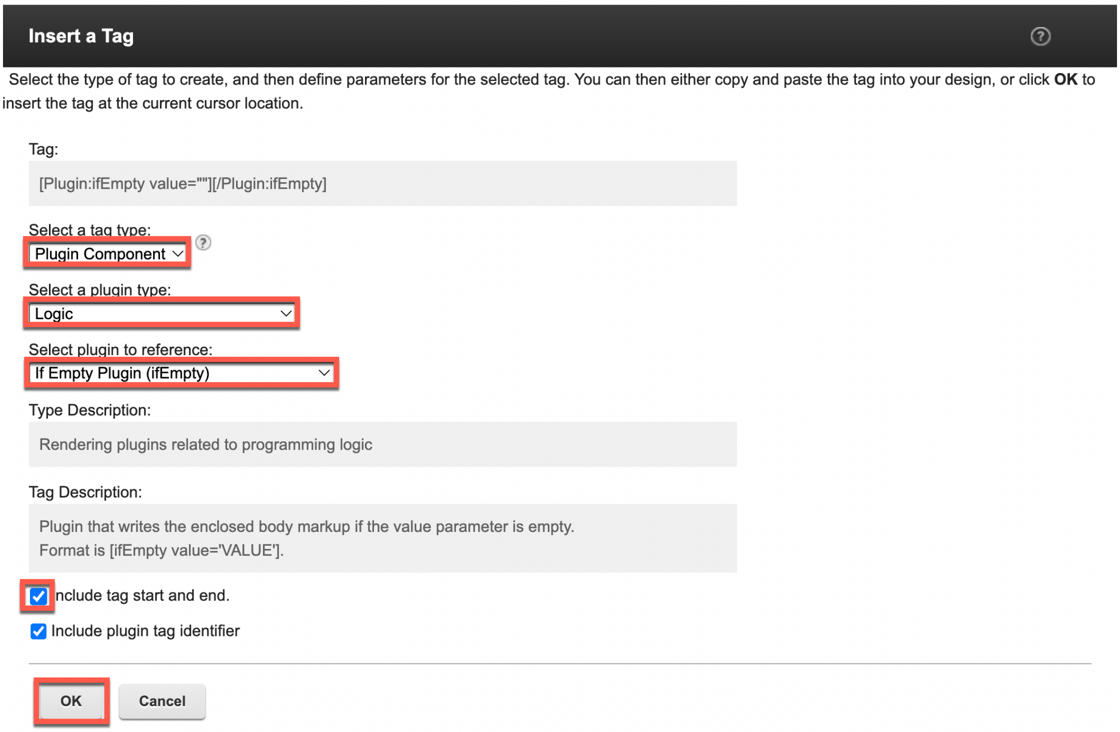
Description: Localizations WeatherAPI Data Source

HTML Element

HTML Markup:

```
[Element key="WeatherAPI Base URL" context="selected" type="content" name="weatherapi/global
config/global config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected"
type="content" name="weatherapi/global config/global config"]&q=[Element key="Location"
context="selected" type="content" name="weatherapi/global config/global config"]&days=[Element
key="Number Of Forecast Days" context="selected" type="content" name="weatherapi/global config/global
config"]&aqi=no&alerts=no
```

12. You can check if the Location element has a value. Select **Plugin Component** as **tag type**, **Logic** as **plugin type**, I**f Empty Plugin (ifEmpty) plugin to reference**, select the **Include tag start and end** and click **OK**.



Insert a Tag

Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

[Plugin:ifEmpty value=""][/Plugin:ifEmpty]

Select a tag type: Plugin Component

Select a plugin type: Logic

Select plugin to reference: If Empty Plugin (ifEmpty)

Type Description:

Rendering plugins related to programming logic

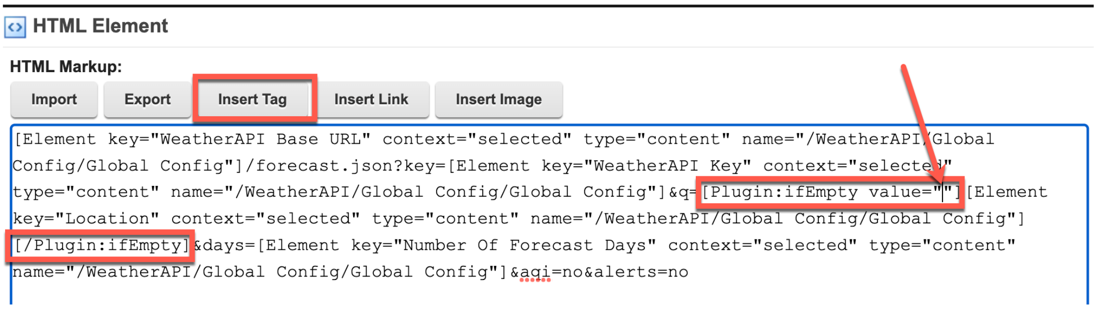
Tag Description:

Plugin that writes the enclosed body markup if the value parameter is empty.
Format is [ifEmpty value='VALUE'].

☑ nclude tag start and end.

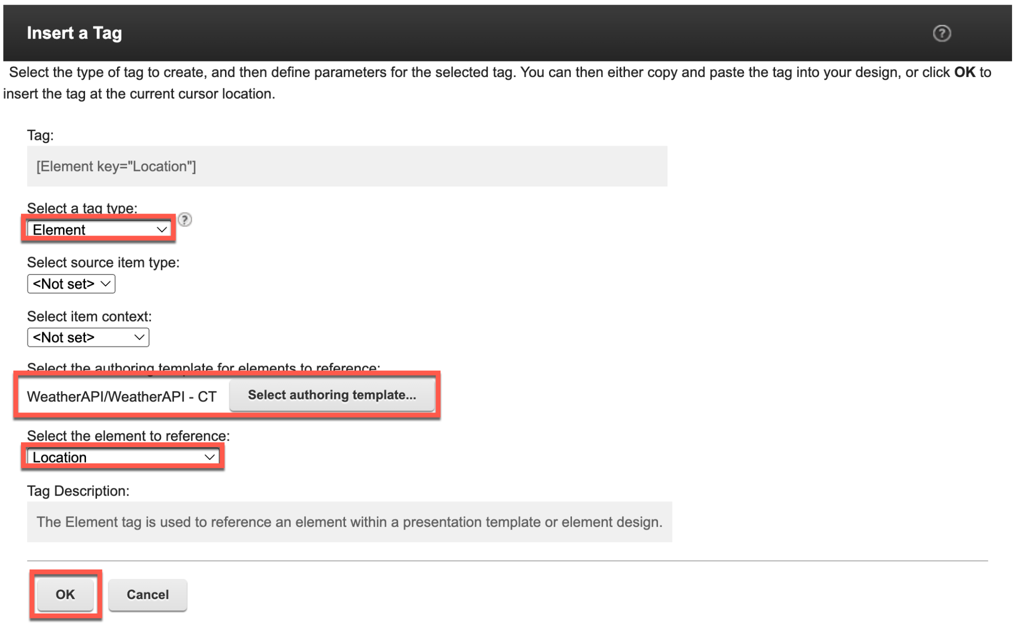
☑ Include plugin tag identifier

13. If it is empty, it can continue using the global configuration location element. Therefore, you need to move the ifEmpty plugin close tag **[/Plugin:ifEmpty]** after this location element tag. Then put your cursor in the value that is now "" to test if the location element is empty and click **Insert Tag**.



```
[Element key="WeatherAPI Base URL" context="selected" type="content" name="/WeatherAPI/Global
Config/Global Config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected"
type="content" name="/WeatherAPI/Global Config/Global Config"]&q=[Plugin:ifEmpty value=""][Element
key="Location" context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]
[/Plugin:ifEmpty]&days=[Element key="Number Of Forecast Days" context="selected" type="content"
name="/WeatherAPI/Global Config/Global Config"]&aqi=no&alerts=no
```

14. Select the **Location** element in your **WeatherAPI – CT** and click **OK**.



Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:
[Element key="Location"]

Select a tag type:
Element

Select source item type:
<Not set>

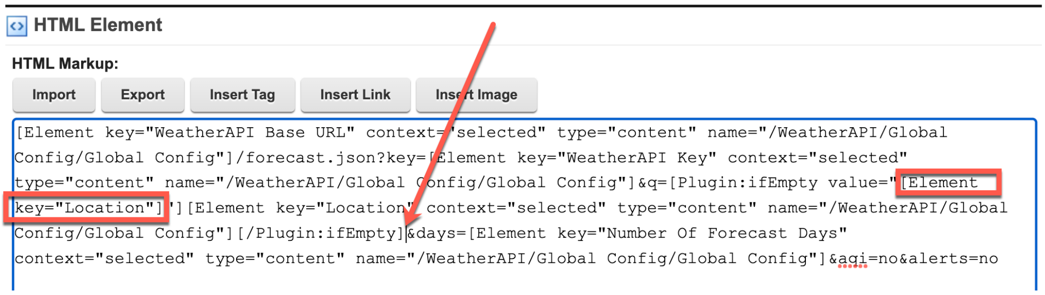
Select item context:
<Not set>

Select the authoring template for elements to reference:
WeatherAPI/WeatherAPI - CT

Select the element to reference:
Location

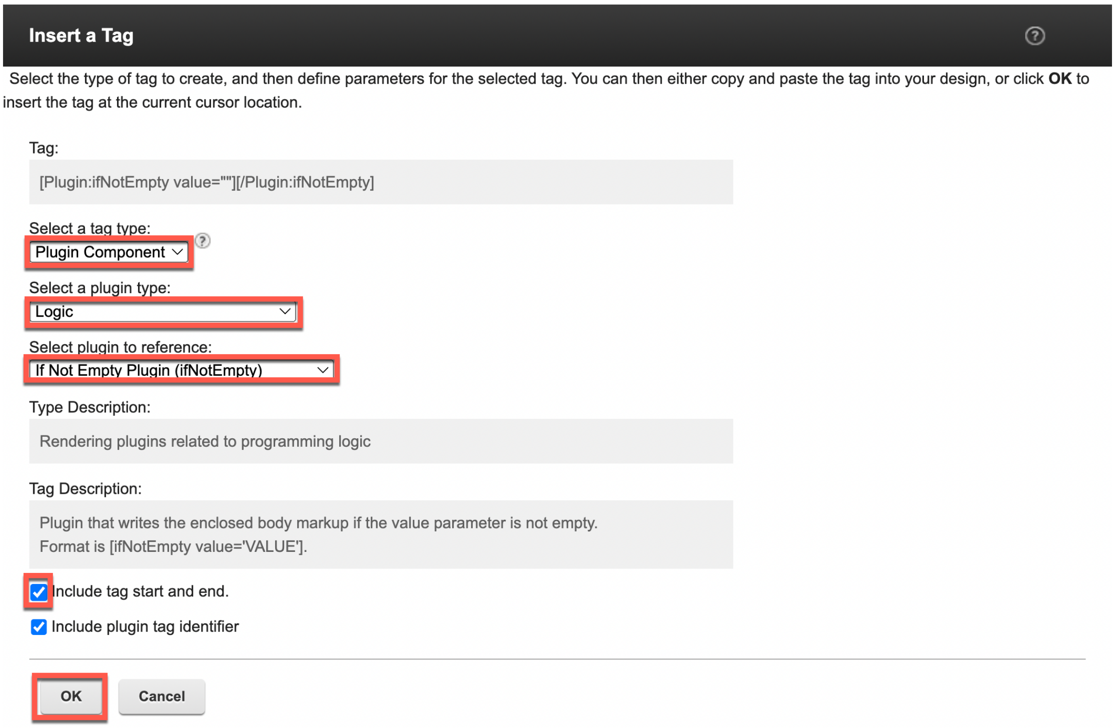
Tag Description:
The Element tag is used to reference an element within a presentation template or element design.

15. Then click **Insert Tag** again to manage the case it is set.



```
[Element key="WeatherAPI Base URL" context="selected" type="content" name="/WeatherAPI/Global
Config/Global Config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected"
type="content" name="/WeatherAPI/Global Config/Global Config"]&q=[Plugin:ifEmpty value="[Element
key="Location"]"][Element key="Location" context="selected" type="content" name="/WeatherAPI/Global
Config/Global Config"][/Plugin:ifEmpty]&days=[Element key="Number Of Forecast Days"
context="selected" type="content" name="/WeatherAPI/Global Config/Global Config"]&aqi=no&alerts=no
```

16. In this case, select the **If Not Empty plugin** and click **OK**.



Select the type of tag to create, and then define parameters for the selected tag. You can then either copy and paste the tag into your design, or click **OK** to insert the tag at the current cursor location.

Tag:

[Plugin:ifNotEmpty value=""][/Plugin:ifNotEmpty]

Select a tag type: Plugin Component

Select a plugin type: Logic

Select plugin to reference: If Not Empty Plugin (ifNotEmpty)

Type Description:

Rendering plugins related to programming logic

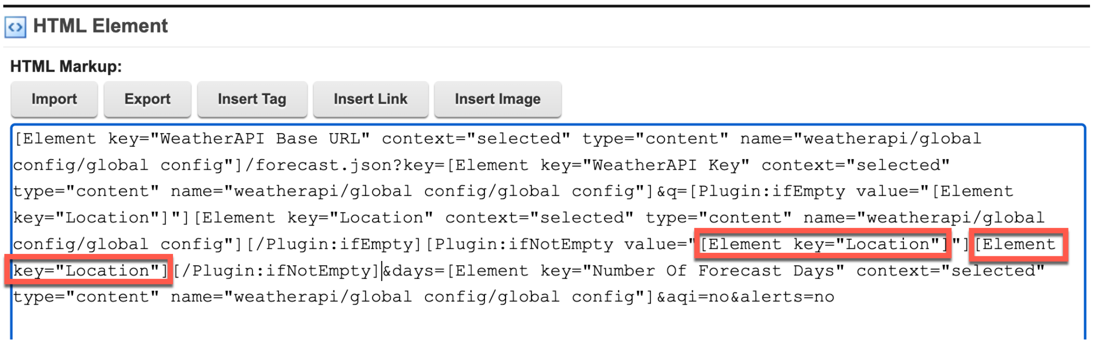
Tag Description:

Plugin that writes the enclosed body markup if the value parameter is not empty.
Format is [ifNotEmpty value='VALUE'].

☑ Include tag start and end.

☑ Include plugin tag identifier

17. Then insert in the value the **element tag** for **Location** again for the isNotEmpty value and for the result to show.

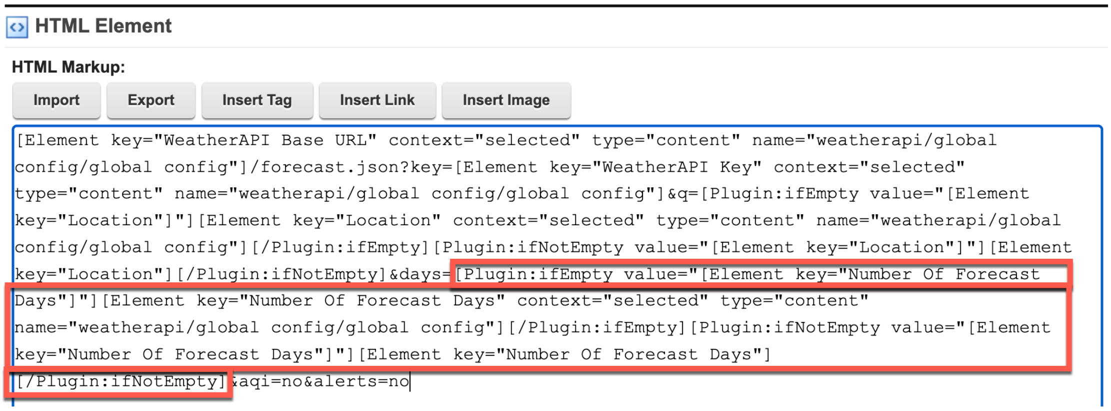


```
[Element key="WeatherAPI Base URL" context="selected" type="content" name="weatherapi/global
config/global config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected"
type="content" name="weatherapi/global config/global config"]&q=[Plugin:ifEmpty value="[Element
key="Location"]"][Element key="Location" context="selected" type="content" name="weatherapi/global
config/global config"][/Plugin:ifEmpty][Plugin:ifNotEmpty value="[Element key="Location"]"][Element
key="Location"][/Plugin:ifNotEmpty]&days=[Element key="Number Of Forecast Days" context="selected"
type="content" name="weatherapi/global config/global config"]&aqi=no&alerts=no
```

18. Then do the same for **Number of Forecast Days** for the days parameter. You may also use the downloaded resources.

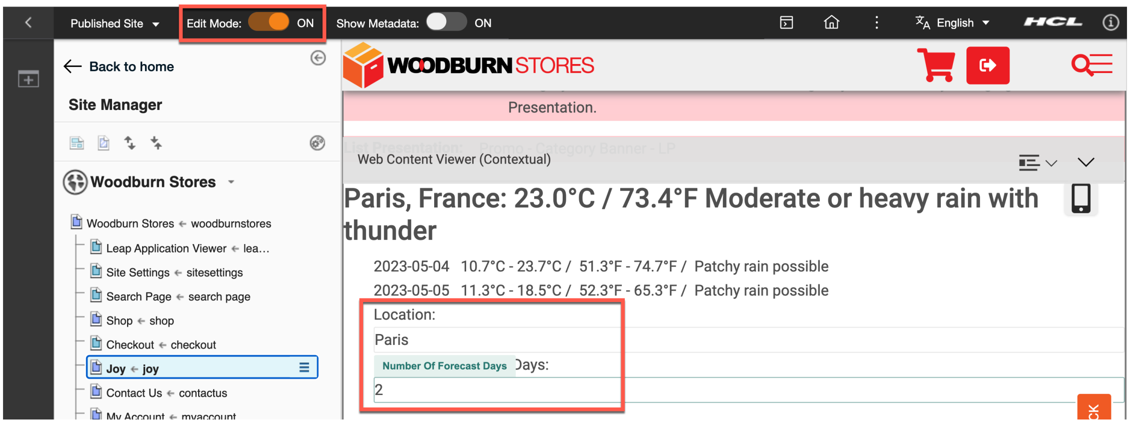
    [Element key="WeatherAPI Base URL" context="selected" type="content" name="weatherapi/global config/global config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected" type="content" name="weatherapi/global config/global config"]&q=**[Plugin:ifEmpty value="[Element key="Location"]"][Element key="Location" context="selected" type="content" name="weatherapi/global config/global config"][/Plugin:ifEmpty][Plugin:ifNotEmpty value="[Element key="Location"]"][Element key="Location"][/Plugin:ifNotEmpty]**&days=**[Plugin:ifEmpty value="[Element key="Number Of Forecast Days"]"][Element key="Number Of Forecast Days" context="selected" type="content" name="weatherapi/global config/global config"][/Plugin:ifEmpty][Plugin:ifNotEmpty value="[Element key="Number Of Forecast Days"]"][Element key="Number Of Forecast Days"][/Plugin:ifNotEmpty]**&aqi=no&alerts=no

    And you should get.



```
[Element key="WeatherAPI Base URL" context="selected" type="content" name="weatherapi/global
config/global config"]/forecast.json?key=[Element key="WeatherAPI Key" context="selected"
type="content" name="weatherapi/global config/global config"]&q=[Plugin:ifEmpty value="[Element
key="Location"]"][Element key="Location" context="selected" type="content" name="weatherapi/global
config/global config"][/Plugin:ifEmpty][Plugin:ifNotEmpty value="[Element key="Location"]"][Element
key="Location"][/Plugin:ifNotEmpty]&days=[Plugin:ifEmpty value="[Element key="Number Of Forecast
Days"]"][Element key="Number Of Forecast Days" context="selected" type="content"
name="weatherapi/global config/global config"][/Plugin:ifEmpty][Plugin:ifNotEmpty value="[Element
key="Number Of Forecast Days"]"][Element key="Number Of Forecast Days"]
[/Plugin:ifNotEmpty]&aqi=no&alerts=no
```

19. Save it and then test it. Open your **Joy** page, put it in the **Edit Mod**e and change the **Location** and **Number of Forecast Days**. You may need to refresh the page to see your updates reflected.



Paris, France: 23.0°C / 73.4°F Moderate or heavy rain with thunder

2023-05-04 10.7°C - 23.7°C / 51.3°F - 74.7°F / Patchy rain possible
2023-05-05 11.3°C - 18.5°C / 52.3°F - 65.3°F / Patchy rain possible

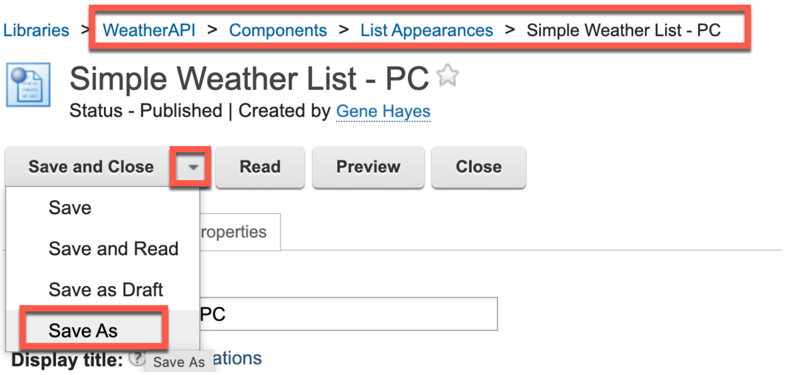
Location:
Paris
Number Of Forecast Days Days:
2

Congratulations! You successfully configured the user configuration to override the Location and Number Of Forecast Days default values easily using inline or WYSIWYG mode.

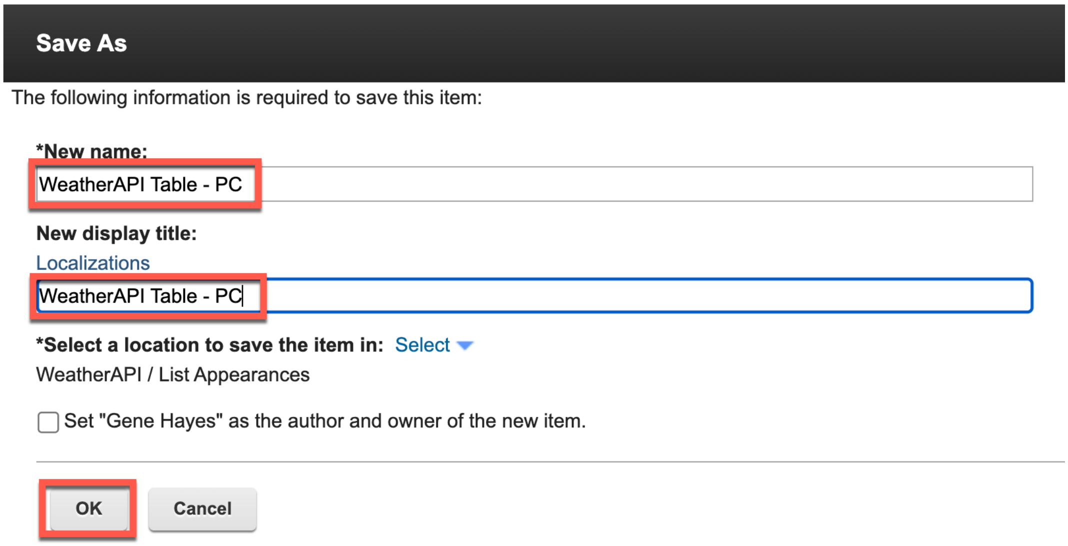
Part 5: Add List Appearance Options

And finally, you will add different ways to show the weather data and allow a business user to select from them. You will create a couple of variations of the list presentation, managed by Personalization Components, with a table and card UI, and then you will extend the List Definition content to allow the user to select between these variations.

1. In the Authoring Portlet, edit your existing your **WeatherAPI** library – **Components** – **List Appearance** folder - **Simple Weather List - PC** Personalization Component. Then make a copy, click **Save As**.



2. Name and give it the display title **WeatherAPI Table - PC**.



The following information is required to save this item:

*New name: WeatherAPI Table - PC

New display title: Localizations

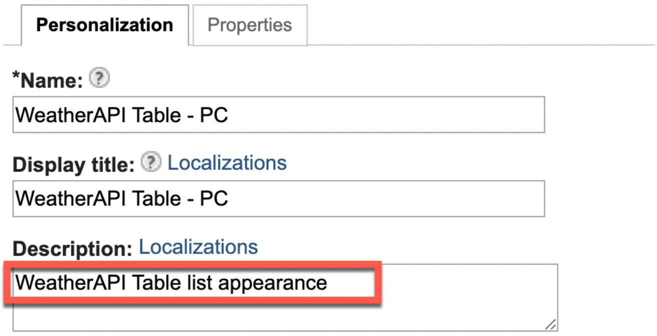
WeatherAPI Table - PC

*Select a location to save the item in: Select

WeatherAPI / List Appearances

□ Set "Gene Hayes" as the author and owner of the new item.

3. Then update the description.

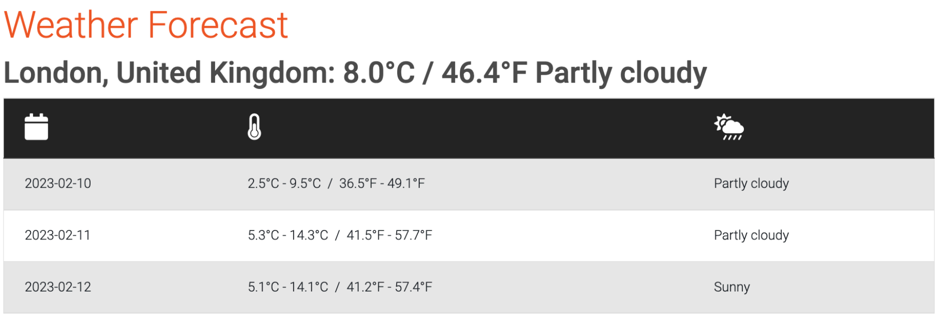


*Name: WeatherAPI Table - PC

Display title: Localizations WeatherAPI Table - PC

Description: Localizations WeatherAPI Table list appearance

4. And update the card **List Presentation Markup** to make it look like



Weather Forecast

London, United Kingdom: 8.0°C / 46.4°F Partly cloudy

2023-02-10	2.5°C - 9.5°C / 36.5°F - 49.1°F	Partly cloudy
2023-02-11	5.3°C - 14.3°C / 41.5°F - 57.7°F	Partly cloudy
2023-02-12	5.1°C - 14.1°C / 41.2°F - 57.4°F	Sunny

5. Manage the top of this presentation in the Header with this code. Use an editable profile tag to display the title of your content. Notice you may get multiple values with a single ListRenderingContext plugin tag.

```
<div class="container-fluid" style="font-size: 16px">
  <h1 class="display-6 text-primary ms-4 ms-md-0 mt-2">
    [EditableProperty field="title" type="content" context="current" format="div"]
          [Property type="content" context="current" field="title"]
     [/EditableProperty]
  </h1>
  <h2 class="ms-4 ms-md-0">
     [Plugin:ListRenderingContext action="getListProperty"
key="location.name"],&nbsp;[Plugin:ListRenderingContext action="getListProperty"
key="location.country"]:&nbsp;[Plugin:ListRenderingContext action="getListProperty"
key="current.temp_c"]&deg;C&nbsp;/&nbsp;[Plugin:ListRenderingContext
action="getListProperty" key="current.temp_f"]&deg;F&nbsp[Plugin:ListRenderingContext
action="getListProperty" key="current.condition.text"]
  </h2>
  <table class="table table-striped border">
    <thead class="table-dark">
      <tr>
        <th><i class="fas fa-2x fa-calendar"></i></th>
        <th><i class="fas fa-2x fa-thermometer-half"></i></th>
        <th><i class="fas fa-2x fa-cloud-sun-rain"></i></th>
      </tr>
    </thead>
    <tbody>
```

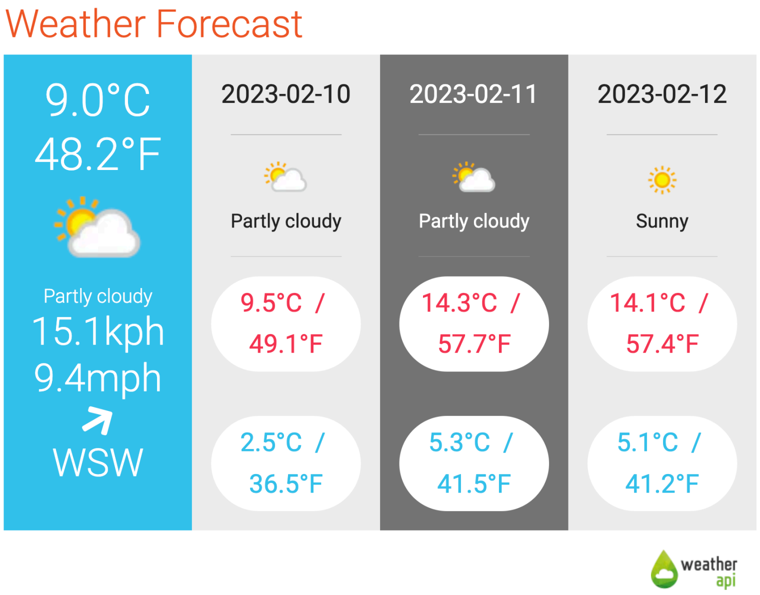
6. Change the result design, the rows in the table, to.

```
<tr>
  <td>[AttributeResource attributeName="date" separator=","]</td>
  <td>
    [AttributeResource attributeName="day.mintemp_c" separator=","]&deg;C - [AttributeResource attributeName="day.maxtemp_c" separator=","]&deg;C
    &nbsp;/&nbsp;
    [AttributeResource attributeName="day.mintemp_f" separator=","]&deg;F - [AttributeResource attributeName="day.maxtemp_f" separator=","]&deg;F
  </td>
  <td>[AttributeResource attributeName="day.condition.text" separator=","]</td>
</tr>
```

7. And update the Footer, to close the table, to.

```
    </tbody>
  </table>
</div>
```

8. Click **Save** and then **Save As** to create your Card Personalization Component. This will look like.

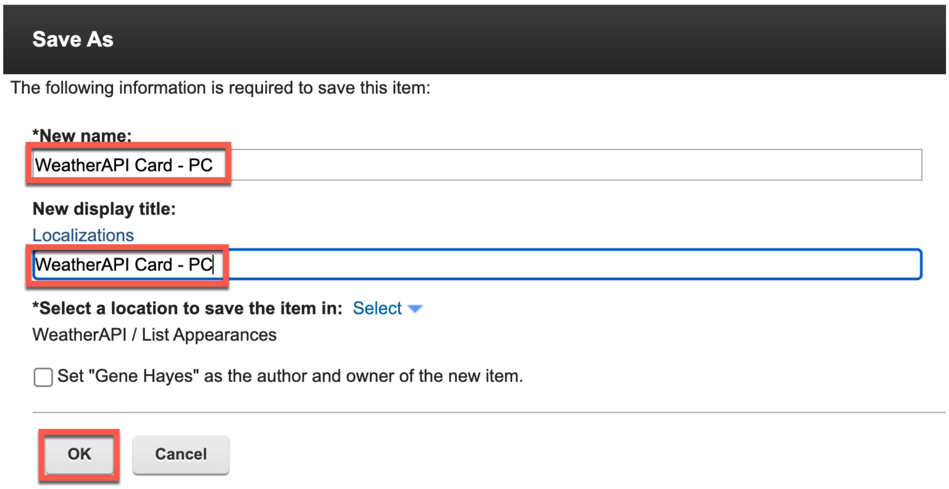


Weather Forecast

9.0°C
48.2°F
Partly cloudy
15.1kph
9.4mph
WSW

2023-02-10	2023-02-11	2023-02-12
Partly cloudy	Partly cloudy	Sunny
9.5°C / 49.1°F	14.3°C / 57.7°F	14.1°C / 57.4°F
2.5°C / 36.5°F	5.3°C / 41.5°F	5.1°C / 41.2°F

9. Name and give it the display title **WeatherAPI Card – PC** and click **OK**.



Save As

The following information is required to save this item:

*New name: WeatherAPI Card - PC

New display title: WeatherAPI Card - PC

*Select a location to save the item in: WeatherAPI / List Appearances

□ Set "Gene Hayes" as the author and owner of the new item.

10. Update the description again and update the Header again to manage the top and the left column of this card presentation.

```
<div class="container-fluid">
  <h1 class="display-5 text-primary ms-4 ms-lg-0 mt-2">
   [EditableProperty field="title" type="content" context="current" format="div"]
         [Property type="content" context="current" field="title"]
    [/EditableProperty]
 </h1>
  <div class="row g-0 p-0 m-0 text-center">
    [Plugin:Equals text1="[Plugin:ListRenderingContext action="getListProperty" key="current.is_day"]"
text2="1"]
    <div class="col-lg col-sm-12 py-1 py-lg-4 p-4 text-white bg-info">
    [/Plugin:Equals]
    [Plugin:Otherwise]
    <div class="col-lg col-sm-12 py-1 py-lg-4 p-4 text-white bg-dark">
    [/Plugin:Otherwise]
      <div class="row g-0 p-0 m-0">
        <div class="col">
          <span class="display-4">[Plugin:ListRenderingContext action="getListProperty"
key="current.temp_c"]&deg;C</span>
        </div>
      </div>
      <div class="row g-0 p-0 m-0">
        <div class="col">
          <span class="display-4">[Plugin:ListRenderingContext action="getListProperty"
key="current.temp_f"]&deg;F</span>
        </div>
      </div>
      <div class="row g-0 p-0 m-0 d-flex align-items-center">
        <div class="col">
          <img style="width: 128px" src="[Plugin:ListRenderingContext action="getListProperty"
key="current.condition.icon"]">
        </div>
        <div class="col">
          <span class="display-6 fs-4">[Plugin:ListRenderingContext action="getListProperty"
key="current.condition.text"]</span>
        </div>
      </div>
      <div class="row g-0 p-0 m-0 text-white d-flex align-items-center">
        <div class="col">
          <div class="row g-0 p-0 m-0">
            <div class="col display-5">[Plugin:ListRenderingContext action="getListProperty"
key="current.wind_kph"]kph</div>
          </div>
          <div class="row g-0 p-0 m-0">
            <div class="col display-5">[Plugin:ListRenderingContext action="getListProperty"
key="current.wind_mph"]mph</div>
          </div>
        </div>
        <div class="col">
          <div class="row g-0 p-0 m-0">
            <i class="fas fa-3x fa-arrow-down" style="transform: rotate([Plugin:ListRenderingContext
action="getListProperty" key="current.wind_degree"]deg)"></i>
          </div>
          <div class="row g-0 p-0 m-0">
            <div class="col display-5">[Plugin:ListRenderingContext action="getListProperty"
key="current.wind_dir"]</div>
          </div>
        </div>
      </div>
    </div>
```

11. Change the result design, additional columns, to.

```
[Plugin:ifListIndex by='odd']
<div class="col-lg col-sm-12 py-1 py-lg-4 p-4 text-dark bg-light fs-2">
  <span>[AttributeResource attributeName="date" separator=","]</span>
  <hr class="m-1 m-lg-4 border-light">
  <div class="row g-0 p-0 m-0">
    <div class="col">
      <img style="width: 64px;" src="[AttributeResource attributeName="day.condition.icon" separator=","]">
    </div>
  </div>
  <div class="row g-0 p-0 m-0 text-center fs-4">
    <div class="col">
      [AttributeResource attributeName="day.condition.text" separator=","]
    </div>
  </div>
  <hr class="m-1 m-lg-4 border-light">
  <div class="row g-0 p-0 m-0">
    <span class="text-danger bg-white p-2 rounded-pill">
      [AttributeResource attributeName="day.maxtemp_c" separator=","]&deg;C
      &nbsp;/&nbsp;
      [AttributeResource attributeName="day.maxtemp_f" separator=","]&deg;F
    </span>
  </div>
  <br>
  <div class="row g-0 p-0 m-0 mt-1">
    <span class="text-info bg-white p-2 rounded-pill">
      [AttributeResource attributeName="day.mintemp_c" separator=","]&deg;C
      &nbsp;/&nbsp;
      [AttributeResource attributeName="day.mintemp_f" separator=","]&deg;F
    </span>
  </div>
</div>
[/Plugin:ifListIndex]
[Plugin:ifListIndex by='even']
<div class="col-lg col-sm-12 py-1 py-lg-4 p-4 text-white fs-2" style="background-color: var(--bs-gray);">
  <span>[AttributeResource attributeName="date" separator=","]</span>
  <hr class="m-1 m-lg-4">
  <div class="row g-0 p-0 m-0">
    <div class="col">
      <img style="width: 64px;" src="[AttributeResource attributeName="day.condition.icon" separator=","]">
    </div>
  </div>
  <div class="row g-0 p-0 m-0 text-center fs-4">
    <div class="col">
      [AttributeResource attributeName="day.condition.text" separator=","]
    </div>
  </div>
  <hr class="m-1 m-lg-4">
  <div class="row g-0 p-0 m-0">
    <span class="text-danger bg-white p-2 rounded-pill">
      [AttributeResource attributeName="day.maxtemp_c" separator=","]&deg;C
      &nbsp;/&nbsp;
      [AttributeResource attributeName="day.maxtemp_f" separator=","]&deg;F
    </span>
  </div>
  <br>
  <div class="row g-0 p-0 m-0 mt-1">
    <span class="text-info bg-white p-2 rounded-pill">
      [AttributeResource attributeName="day.mintemp_c" separator=","]&deg;C
      &nbsp;/&nbsp;
      [AttributeResource attributeName="day.mintemp_f" separator=","]&deg;F
    </span>
  </div>
</div>
[/Plugin:ifListIndex]
```

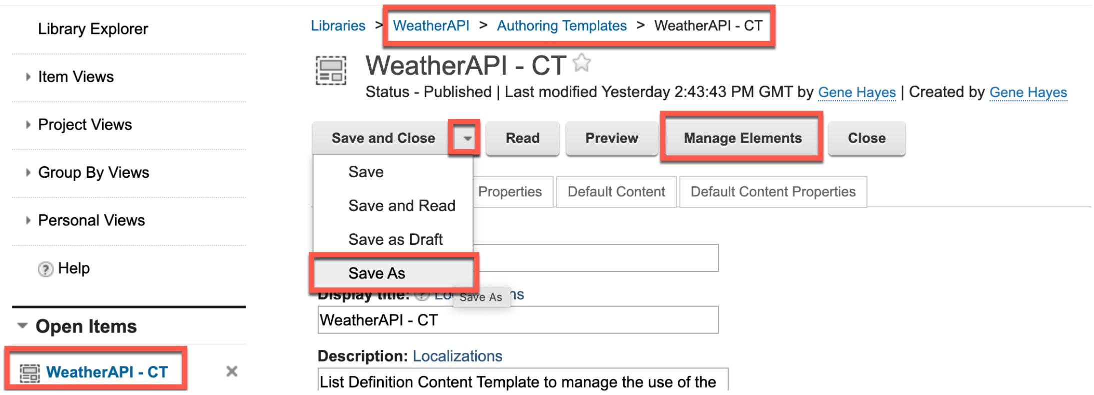
12. And update the Footer, to close the table, to.

```
    </div>
  <div class="row g-0 p-0 m-0">
    <div class="col-lg col-sm-12 py-1 py-lg-4 p-4 text-end">
      <a href="https://www.weatherapi.com/" title="Free Weather API" target=""><img
src='//cdn.weatherapi.com/v4/images/weatherapi_logo.png' alt="Weather data by WeatherAPI.com"
border="0"></a>
    </div>
  </div>
</div>
```

2. Again, save it. Then update your List Definition to allow your user to select from these presentations. First add a List Appearance element to your content template. Edit the **WeatherAPI – CT** again.

> You may also again make a backup of your WeatherAPI – CT Content Template, using **Save As** and then select your original again.

Then click **Manage Elements**.

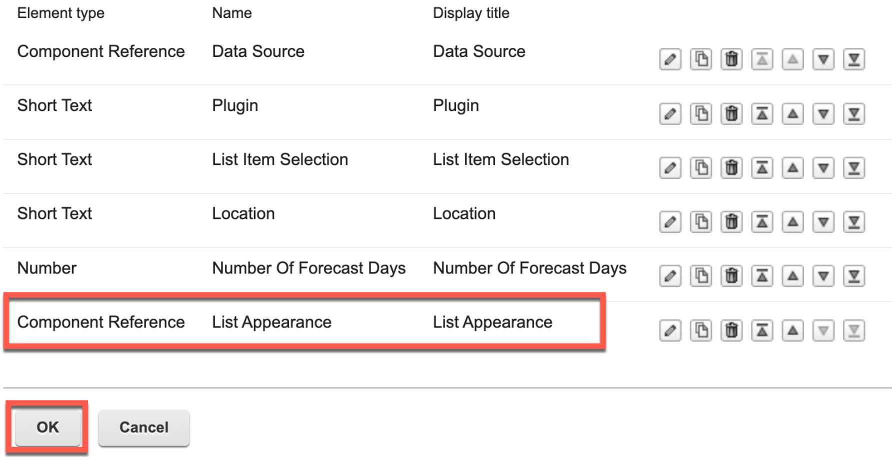


Status - Published | Last modified Yesterday 2:43:43 PM GMT by Gene Hayes | Created by Gene Hayes

Display title: WeatherAPI - CT

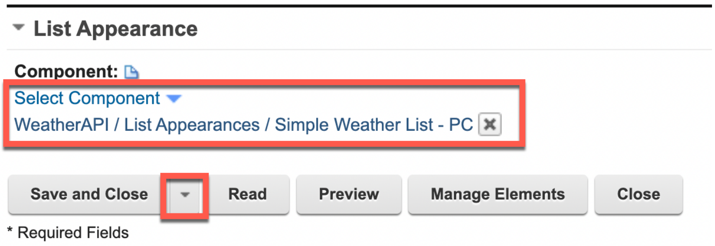
Description: List Definition Content Template to manage the use of the

13. Then, add the new Component Reference element **List Appearance** that will help to refer to any of these components. Click **OK**.

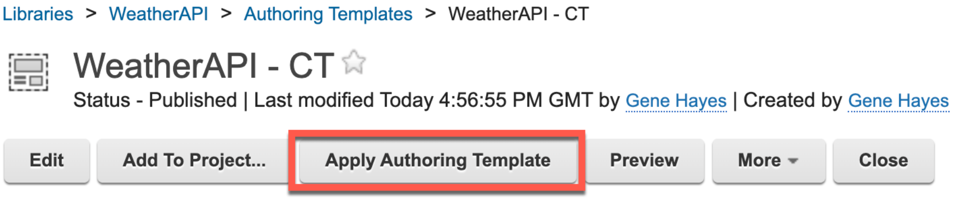


Element type	Name	Display title
Component Reference	Data Source	Data Source
Short Text	Plugin	Plugin
Short Text	List Item Selection	List Item Selection
Short Text	Location	Location
Number	Number Of Forecast Days	Number Of Forecast Days
Component Reference	List Appearance	List Appearance

14. Then set a default value, e.g. **Simple Weather List – PC** and click **Save and Read** to apply it to the existing content.

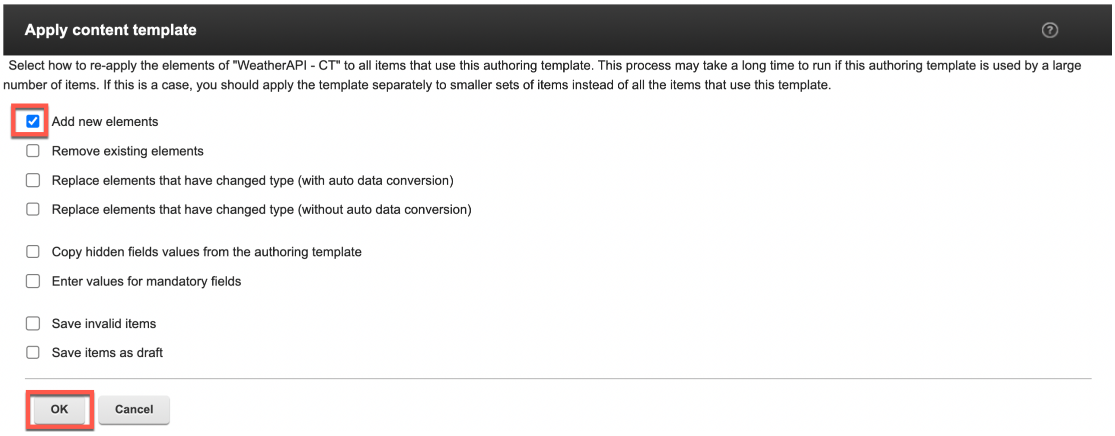


15. Then click **Apply Authoring Template**.



Status - Published | Last modified Today 4:56:55 PM GMT by Gene Hayes | Created by Gene Hayes

16. Select **Add new elements** and click **OK**.



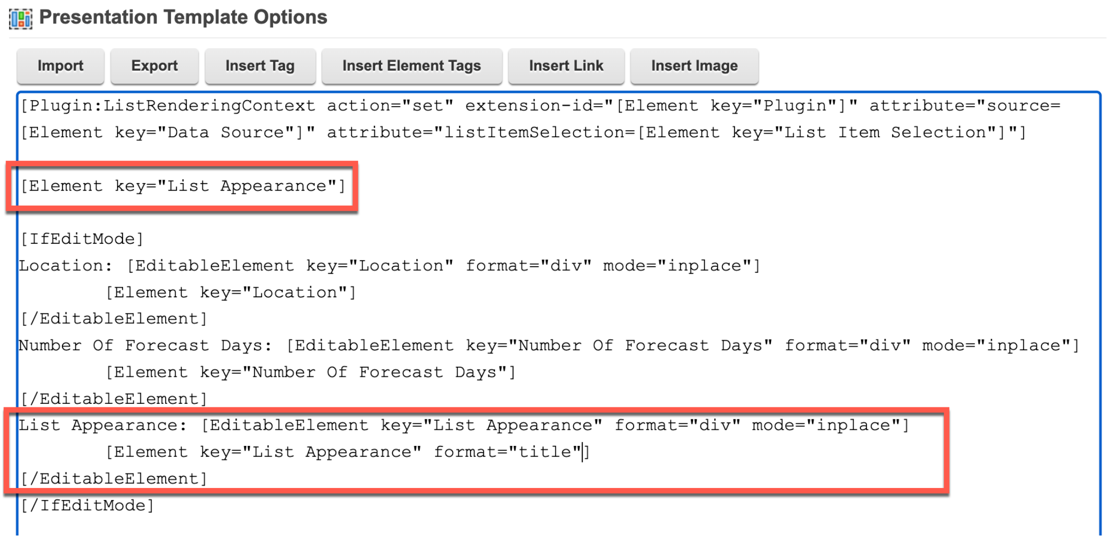
Select how to re-apply the elements of "WeatherAPI - CT" to all items that use this authoring template. This process may take a long time to run if this authoring template is used by a large number of items. If this is a case, you should apply the template separately to smaller sets of items instead of all the items that use this template.

☑ Add new elements
□ Remove existing elements
□ Replace elements that have changed type (with auto data conversion)
□ Replace elements that have changed type (without auto data conversion)

□ Copy hidden fields values from the authoring template
□ Enter values for mandatory fields

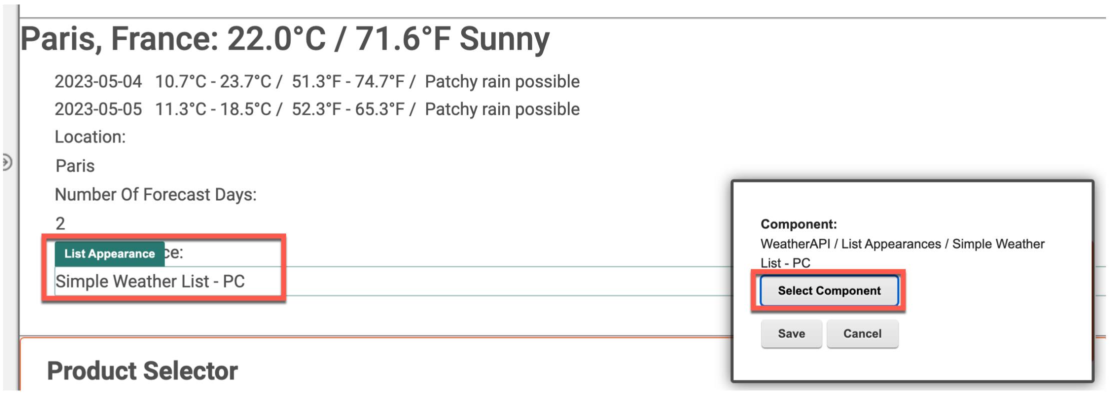
□ Save invalid items
□ Save items as draft

17. And then update the corresponding presentation template use this selected List Appearance. Edit the **WeatherAPI – PT** and replace the hardcoded **Simple Weather List – PC** component tag by an **element** tag for the new **List Appearance**. Then add the **List Presentation** as an **editable element** in the Edit Mode section, add a **format=”title”** to make it display the title of your selected List Appearance component and save it again.



```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]" attribute="source=
[Element key="Data Source"]" attribute="listItemSelection=[Element key="List Item Selection"]"]

[Element key="List Appearance"]

[IfEditMode]
Location: [EditableElement key="Location" format="div" mode="inplace"]
        [Element key="Location"]
[/EditableElement]
Number Of Forecast Days: [EditableElement key="Number Of Forecast Days" format="div" mode="inplace"]
        [Element key="Number Of Forecast Days"]
[/EditableElement]
List Appearance: [EditableElement key="List Appearance" format="div" mode="inplace"]
        [Element key="List Appearance" format="title"]
[/EditableElement]
[/IfEditMode]
```

18. Now try it out again in your Joy page. Select the List Appearance and click **Select Component**.



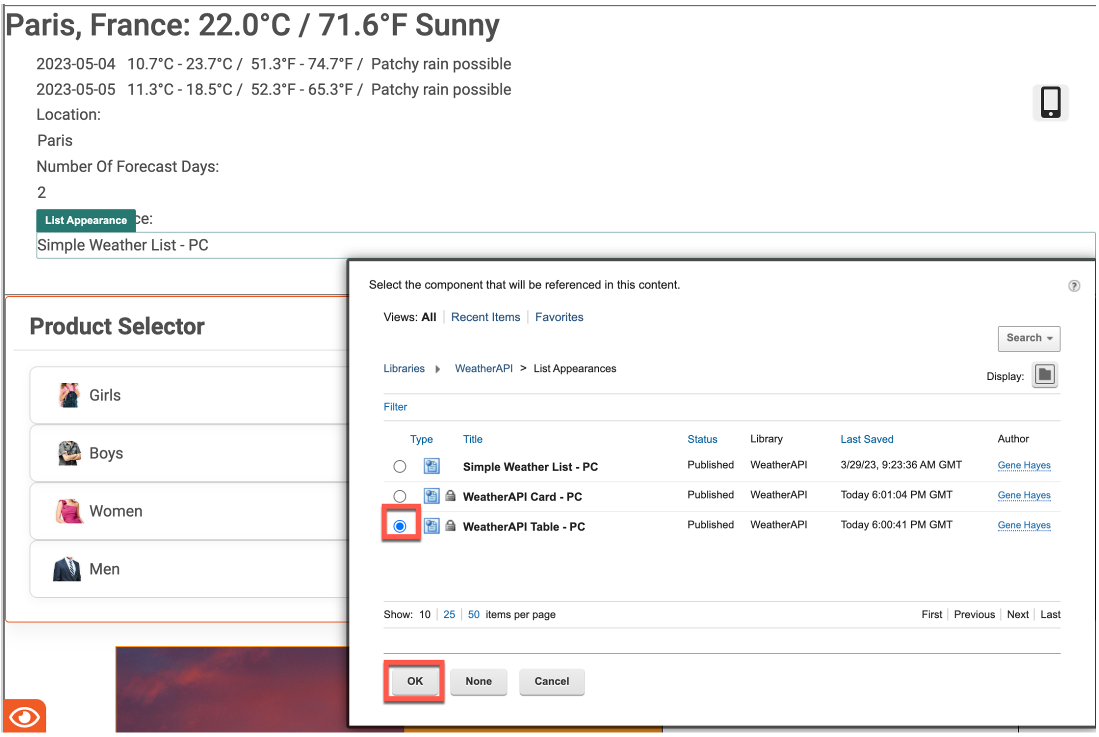
Paris, France: 22.0°C / 71.6°F Sunny

2023-05-04 10.7°C - 23.7°C / 51.3°F - 74.7°F / Patchy rain possible
2023-05-05 11.3°C - 18.5°C / 52.3°F - 65.3°F / Patchy rain possible

Location:
Paris
Number Of Forecast Days:
2
List Appearance ce:
Simple Weather List - PC

Component:
WeatherAPI / List Appearances / Simple Weather List - PC

19. Then select any of your other List Appearances, e.g. **WeatherAPI Table - PC** and click **OK**.



Paris, France: 22.0°C / 71.6°F Sunny

2023-05-04 10.7°C - 23.7°C / 51.3°F - 74.7°F / Patchy rain possible
2023-05-05 11.3°C - 18.5°C / 52.3°F - 65.3°F / Patchy rain possible
Location:
Paris
Number Of Forecast Days:
2
List Appearance
Simple Weather List - PC

Product Selector

Select the component that will be referenced in this content.

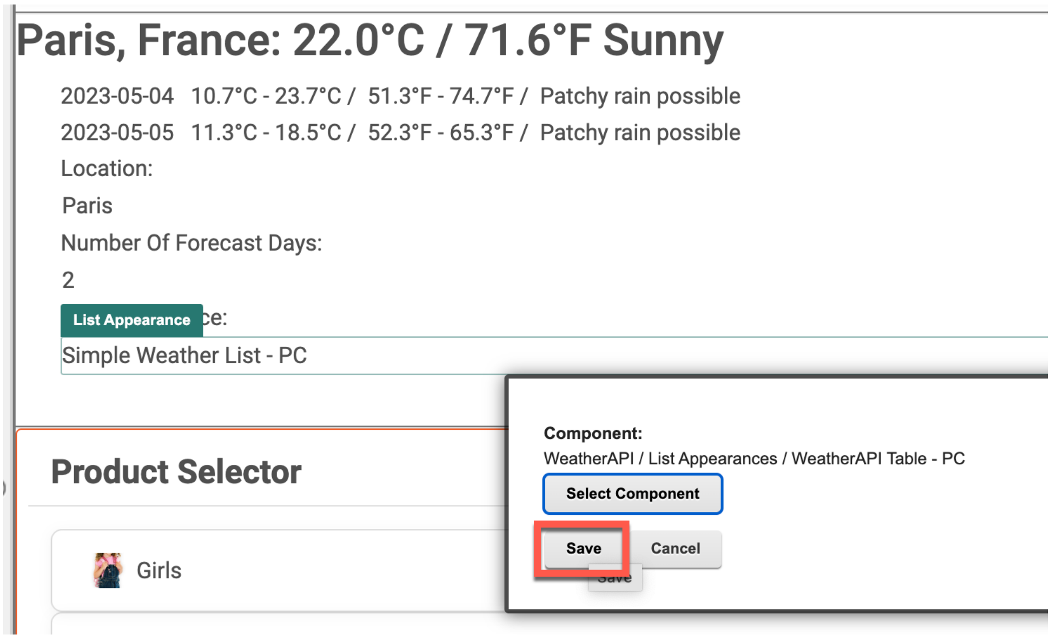
Views: All | Recent Items | Favorites

Libraries ▸ WeatherAPI > List Appearances

	Type	Title	Status	Library	Last Saved	Author
○		Simple Weather List - PC	Published	WeatherAPI	3/29/23, 9:23:36 AM GMT	Gene Hayes
○		WeatherAPI Card - PC	Published	WeatherAPI	Today 6:01:04 PM GMT	Gene Hayes
●		WeatherAPI Table - PC	Published	WeatherAPI	Today 6:00:41 PM GMT	Gene Hayes

Show: 10 | 25 | 50 items per page

20. And click **Save**.



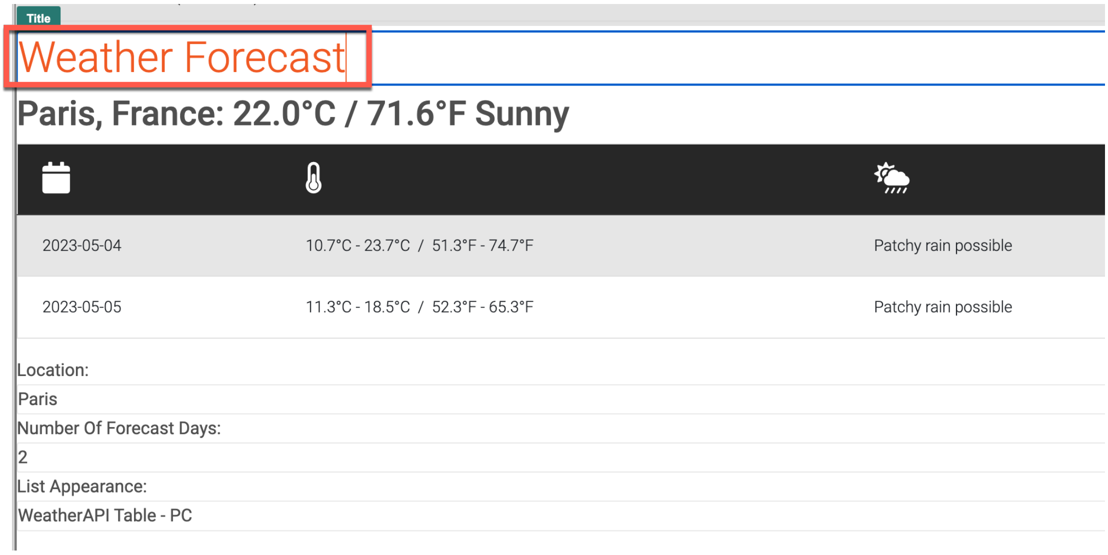
Paris, France: 22.0°C / 71.6°F Sunny

2023-05-04 10.7°C - 23.7°C / 51.3°F - 74.7°F / Patchy rain possible
2023-05-05 11.3°C - 18.5°C / 52.3°F - 65.3°F / Patchy rain possible
Location:
Paris
Number Of Forecast Days:
2
List Appearance
Simple Weather List - PC

Product Selector

Component:
WeatherAPI / List Appearances / WeatherAPI Table - PC

21. You may need to refresh your page and update your title.



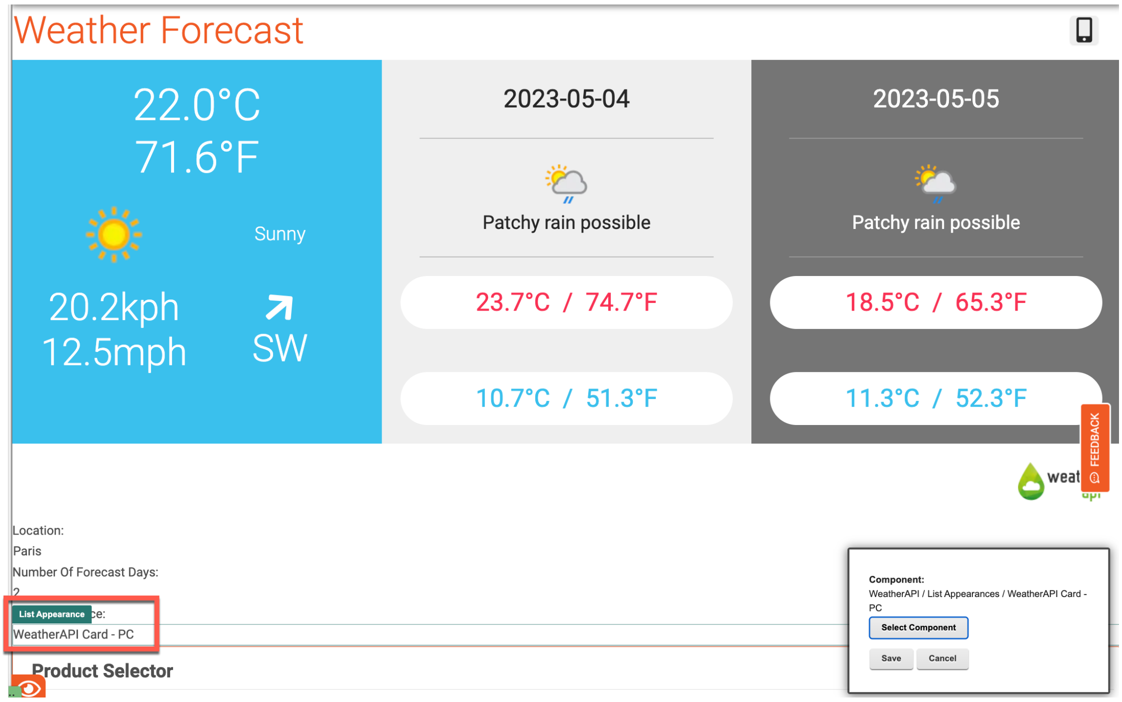
Weather Forecast

Paris, France: 22.0°C / 71.6°F Sunny

📅	🌡	🌦
2023-05-04	10.7°C - 23.7°C / 51.3°F - 74.7°F	Patchy rain possible
2023-05-05	11.3°C - 18.5°C / 52.3°F - 65.3°F	Patchy rain possible

Location:
Paris
Number Of Forecast Days:
2
List Appearance:
WeatherAPI Table - PC

22. You may also test your Card list appearance.



Weather Forecast

22.0°C
71.6°F
Sunny
20.2kph
12.5mph
SW

2023-05-04
Patchy rain possible
23.7°C / 74.7°F
10.7°C / 51.3°F

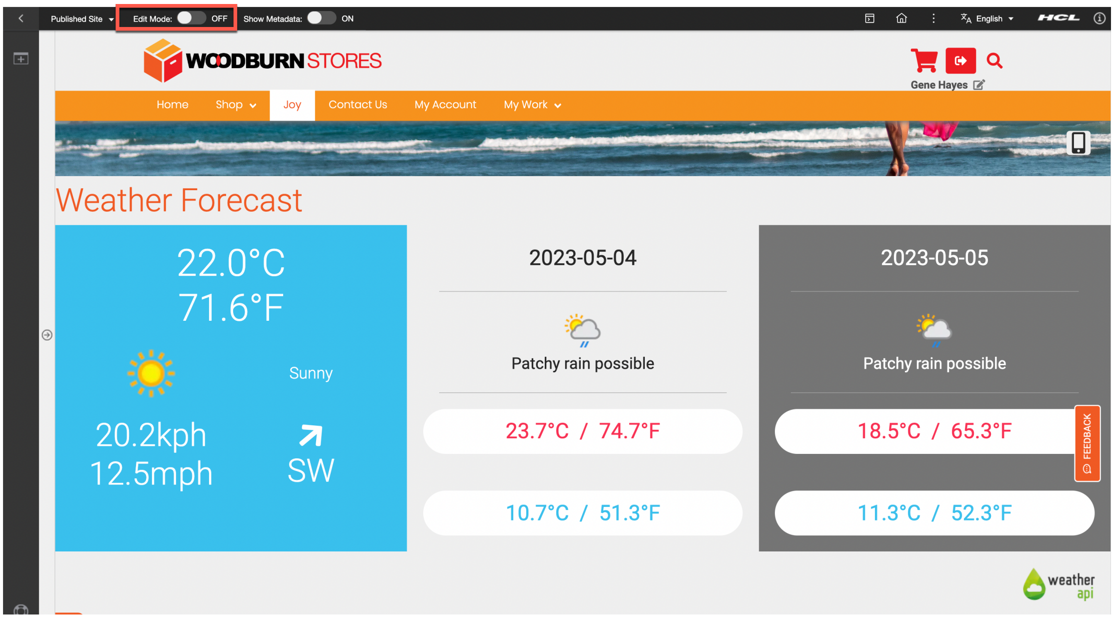
2023-05-05
Patchy rain possible
18.5°C / 65.3°F
11.3°C / 52.3°F

Location:
Paris
Number Of Forecast Days:
2
List Appearance:
WeatherAPI Card - PC

Product Selector

Component:
WeatherAPI / List Appearances / WeatherAPI Card - PC

23. And you may switch off the edit mode to see how it shows for your visitors.



Weather Forecast

22.0°C
71.6°F

Sunny

20.2kph
12.5mph

SW

2023-05-04

Patchy rain possible

23.7°C / 74.7°F

10.7°C / 51.3°F

2023-05-05

Patchy rain possible

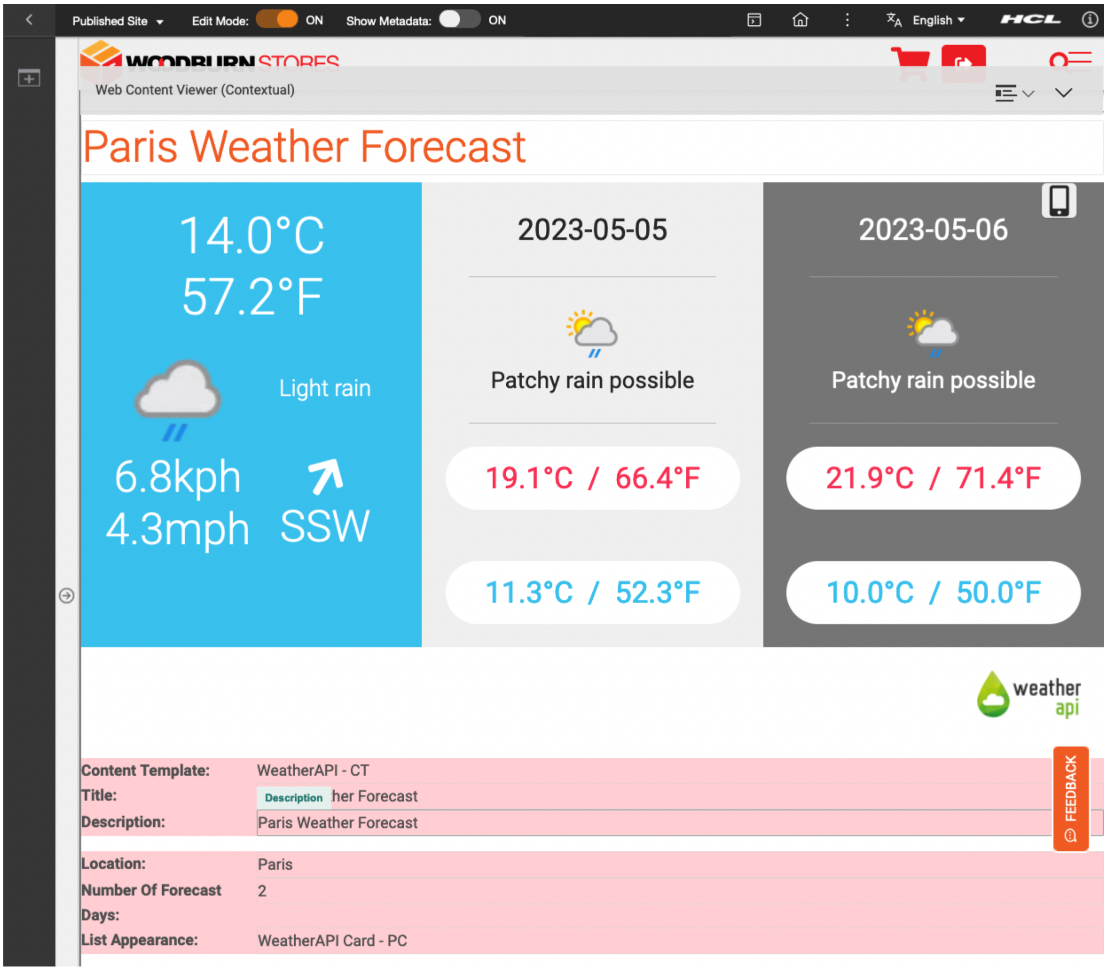
18.5°C / 65.3°F

11.3°C / 52.3°F

24. Optionally, you may change the look of the edit mode. If you are using the DX Solution Modules, you may use the Spotlight Metadata Components/Properties Updater to show the meta-data of the content, like its Content Template, Title and Description. You may also use a callback to update the page right after changing an element to show its impact.

```
[Plugin:ListRenderingContext action="set" extension-id="[Element key="Plugin"]" attribute="source=[Element key="Data Source"]" attribute="listItemSelection=[Element key="List Item Selection"]"]

[Element key="List Appearance"]

[IfEditMode]
[Component name="spotlight/metadata components/properties updater"]
<style>
 .meta-data {
  background-color: #FFCDD2 ;
 }
  .meta-data-size {
    width: 200px;
  }
</style>
<div class="collapse2 mt-3 meta-data" id="dx-ta-1">
  <div class="properties-section d-flex flex-column w-100">
    <div class="d-flex flex-row ">
      <div class="meta-data-size">
        <strong>Location:</strong>
      </div>
      <div class="ms-3 form-outline w-100">
        [EditableElement key="Location" type="auto" mode="inplace" context="current" callback="var myCallBack=new i$.Promise(); myCallBack.then(function(){window.location.reload(false)});return myCallBack;"]
          [Element key="Location"]
        [/EditableElement]
      </div>
    </div>
    <div class="d-flex flex-row ">
      <div class="meta-data-size">
        <strong>Number Of Forecast Days:</strong>
      </div>
      <div class="ms-3 form-outline w-100">
        [EditableElement key="Number Of Forecast Days" type="auto" mode="inplace" context="current" callback="var myCallBack=new i$.Promise(); myCallBack.then(function(){window.location.reload(false)});return myCallBack;"]
          [Element key="Number Of Forecast Days"]
        [/EditableElement]
      </div>
    </div>
    <div class="d-flex flex-row ">
      <div class="meta-data-size">
        <strong>List Appearance:</strong>
      </div>
      <div class="ms-3 form-outline w-100">
        [EditableElement key="List Appearance" type="auto" mode="inplace" context="current" callback="var myCallBack=new i$.Promise(); myCallBack.then(function(){window.location.reload(false)});return myCallBack;"]
          [Element type="auto" context="current" format="title" key="List Appearance"]
        [/EditableElement]
      </div>
    </div>
  </div>
</div>
[/IfEditMode]
```

25. Again save and try it out. You may now also easily update the description as well.



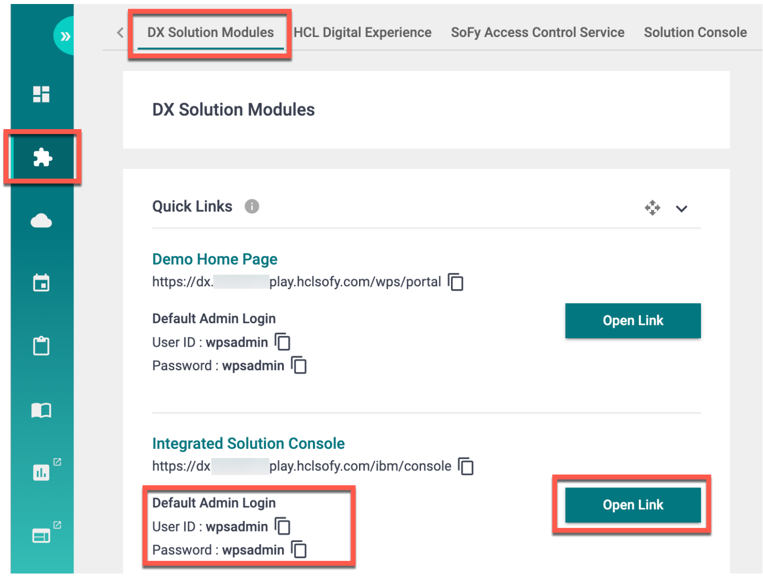
Content Template: WeatherAPI - CT
Title: her Forecast
Description: Paris Weather Forecast
Location: Paris
Number Of Forecast Days: 2
List Appearance: WeatherAPI Card - PC

You have successfully added more list appearances and allowed a business user to easily select any of them.

Optional Part 6: Authorize External Access

While this is typically an administrator task, you may optionally learn how to setup the signer certificate to allow use https and the HTTP Outbound Connection (AJAX Proxy) for the Weather API to overcome the CORS specification to authorize access of the external APIs.

1. If you want to use https, you need to configure the signer certificate. Log into the Integrated Solution Console. If you use HCL SoFy, go to the **Solution Content** and **DX Solution Modules** and then click **Open Link** under the Integrated Solution Console. Note the default admin login User ID and Password you will need to log in. For other deployments, open a new browser tab or window and open <host>/ibm/console.



DX Solution Modules

Quick Links

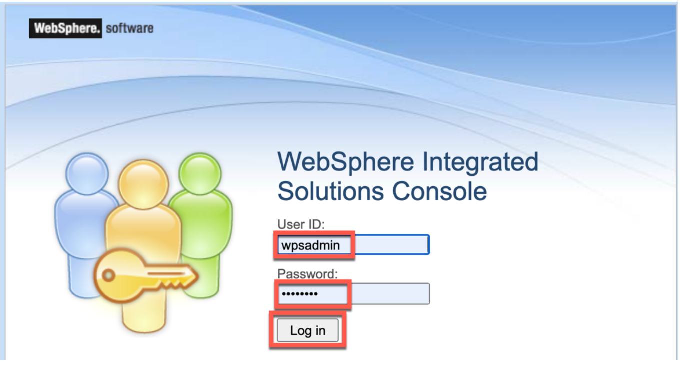
Demo Home Page
https://dx.   play.hclsofy.com/wps/portal

Default Admin Login
User ID : wpsadmin
Password : wpsadmin

Integrated Solution Console
https://dx   play.hclsofy.com/ibm/console

Default Admin Login
User ID : wpsadmin
Password : wpsadmin

2. Then enter your admin credentials from the Solution Console: **wpsadmin/wpsadmin** and click **Log in**.

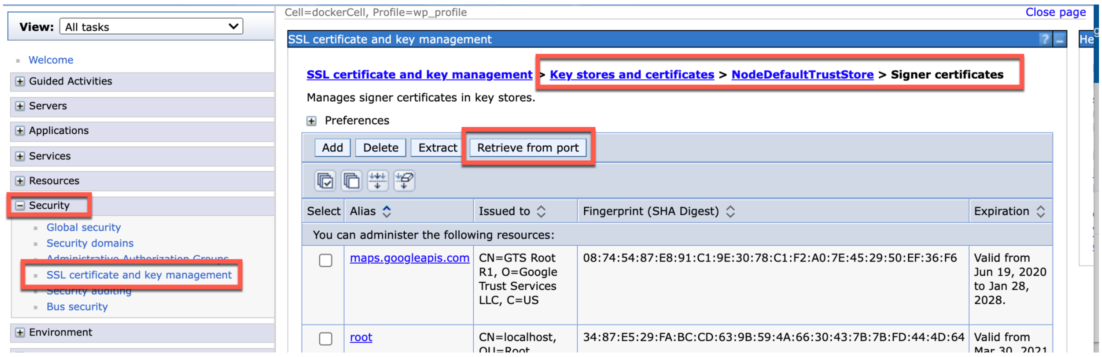


WebSphere Integrated Solutions Console

User ID: wpsadmin

Password: ••••••••

3. Expand **Security** and click **SSL certificate and key management**, under Related Items **Key stores and certificates**, **NodeDefaultTrustStore**, under Additional Properties **Signer certificates** and **Retrieve from port**.



Cell=dockerCell, Profile=wp_profile

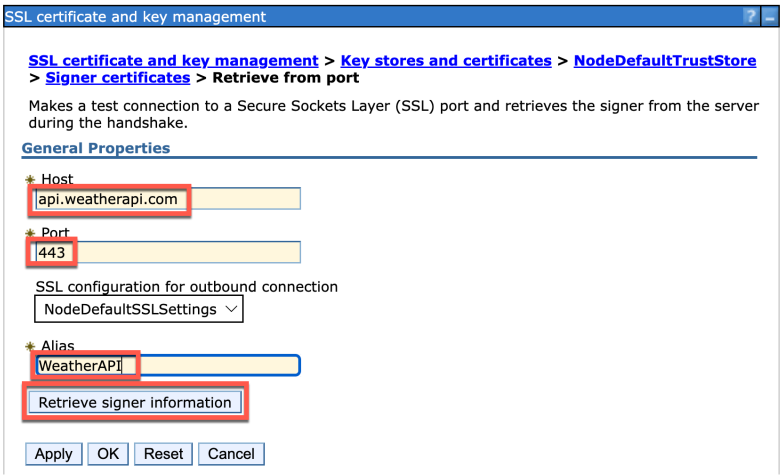
SSL certificate and key management

SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates

Manages signer certificates in key stores.

Select	Alias	Issued to	Fingerprint (SHA Digest)	Expiration
You can administer the following resources:				
□	maps.googleapis.com	CN=GTS Root R1, O=Google Trust Services LLC, C=US	08:74:54:87:E8:91:C1:9E:30:78:C1:F2:A0:7E:45:29:50:EF:36:F6	Valid from Jun 19, 2020 to Jan 28, 2028.
□	root	CN=localhost, OU=Root	34:87:E5:29:FA:BC:CD:63:9B:59:4A:66:30:43:7B:7B:FD:44:4D:64	Valid from Mar 30, 2021

4. As Host, enter the host of WeatherAPI **api.weatherapi.com**, enter **443** (for https) in the **Port** field, and an Alias name to distinguish this certification, e.g. **WeatherAPI**. Then click **Retrieve signer information**.



SSL certificate and key management

SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > Signer certificates > Retrieve from port

Makes a test connection to a Secure Sockets Layer (SSL) port and retrieves the signer from the server during the handshake.

General Properties

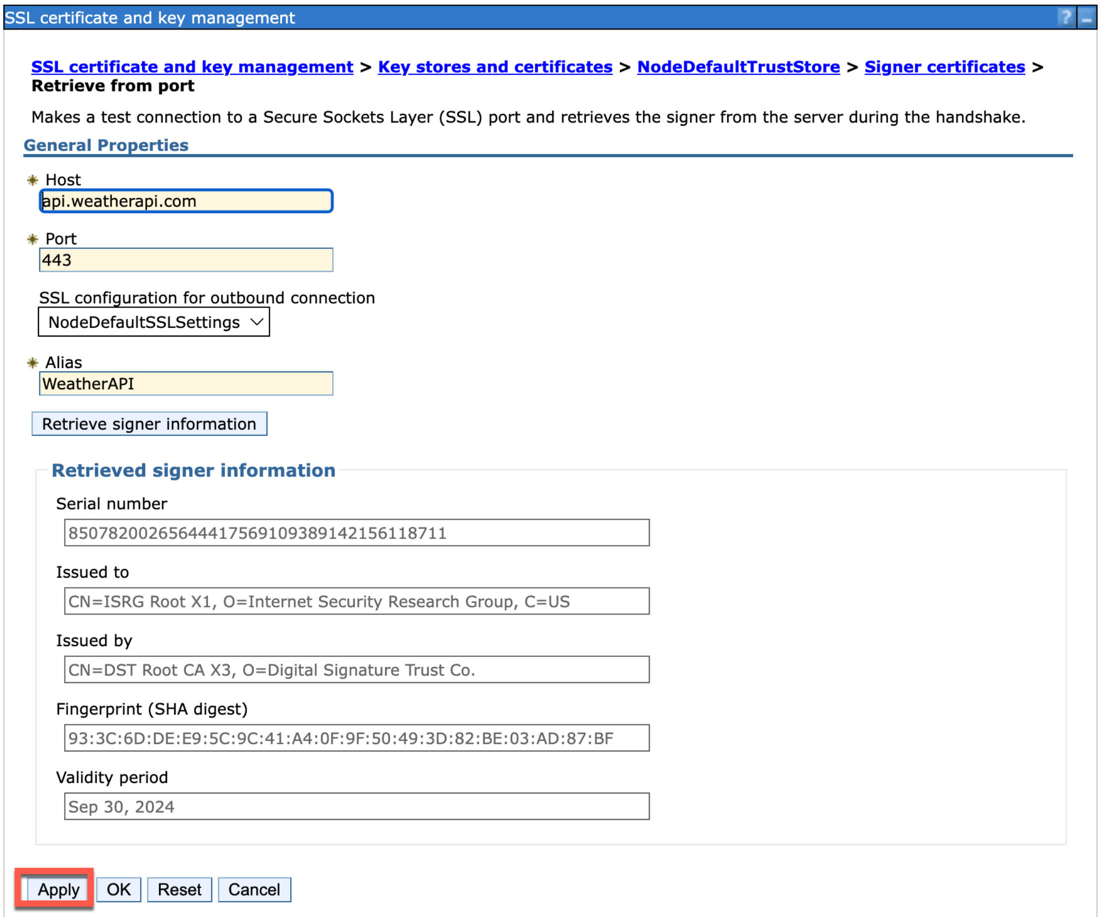
* Host: api.weatherapi.com

* Port: 443

SSL configuration for outbound connection: NodeDefaultSSLSettings

* Alias: WeatherAPI

5. Verify that the certificate information is for a certificate that you can trust and click **Apply**.



SSL certificate and key management

SSL certificate and key management > **Key stores and certificates** > **NodeDefaultTrustStore** > **Signer certificates** >
Retrieve from port

Makes a test connection to a Secure Sockets Layer (SSL) port and retrieves the signer from the server during the handshake.

General Properties

* Host
api.weatherapi.com

* Port
443

SSL configuration for outbound connection
NodeDefaultSSLSettings

* Alias
WeatherAPI

Retrieve signer information

Retrieved signer information

Serial number
85078200265644417569109389142156118711

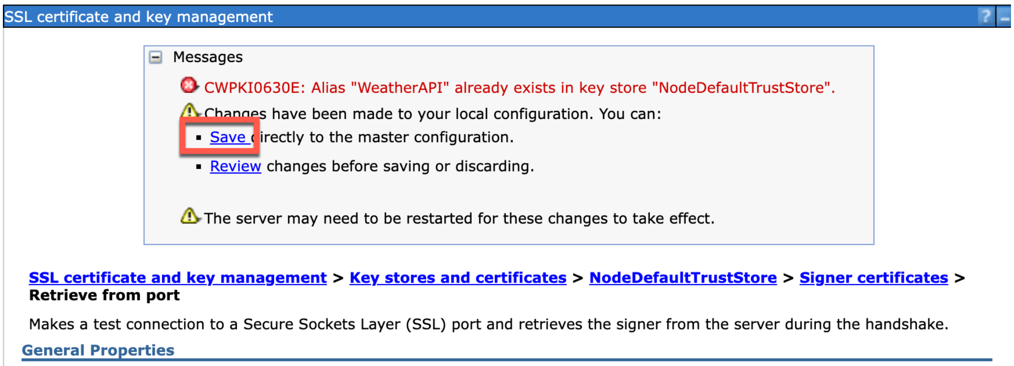
Issued to
CN=ISRG Root X1, O=Internet Security Research Group, C=US

Issued by
CN=DST Root CA X3, O=Digital Signature Trust Co.

Fingerprint (SHA digest)
93:3C:6D:DE:E9:5C:9C:41:A4:0F:9F:50:49:3D:82:BE:03:AD:87:BF

Validity period
Sep 30, 2024

6. Then click **Save** to save it directly to the master configuration.



SSL certificate and key management

Messages

CWPKI0630E: Alias "WeatherAPI" already exists in key store "NodeDefaultTrustStore".

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

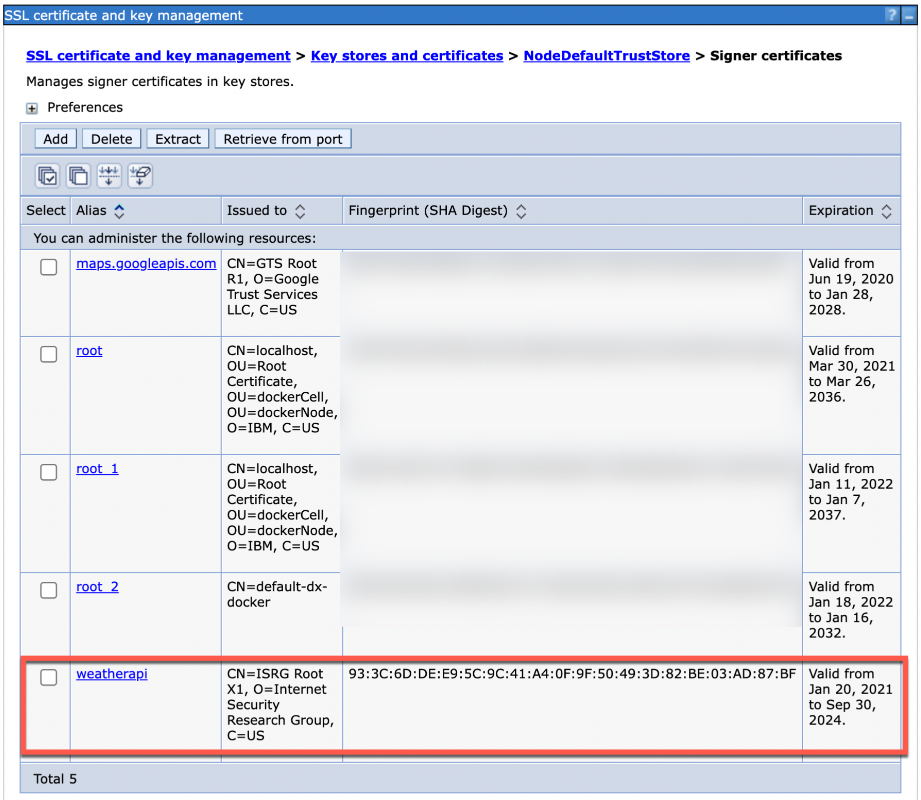
The server may need to be restarted for these changes to take effect.

SSL certificate and key management > **Key stores and certificates** > **NodeDefaultTrustStore** > **Signer certificates** >
Retrieve from port

Makes a test connection to a Secure Sockets Layer (SSL) port and retrieves the signer from the server during the handshake.

General Properties

7. You should see it added to your signer certificates.



SSL certificate and key management

SSL certificate and key management > Key stores and certificates > NodeDefaultTrustStore > **Signer certificates**

Manages signer certificates in key stores.

⊞ Preferences

Add | Delete | Extract | Retrieve from port

Select	Alias	Issued to	Fingerprint (SHA Digest)	Expiration
You can administer the following resources:				
□	maps.googleapis.com	CN=GTS Root R1, O=Google Trust Services LLC, C=US	[redacted]	Valid from Jun 19, 2020 to Jan 28, 2028.
□	root	CN=localhost, OU=Root Certificate, OU=dockerCell, OU=dockerNode, O=IBM, C=US	[redacted]	Valid from Mar 30, 2021 to Mar 26, 2036.
□	root_1	CN=localhost, OU=Root Certificate, OU=dockerCell, OU=dockerNode, O=IBM, C=US	[redacted]	Valid from Jan 11, 2022 to Jan 7, 2037.
□	root_2	CN=default-dx-docker	[redacted]	Valid from Jan 18, 2022 to Jan 16, 2032.
□	weatherapi	CN=ISRG Root X1, O=Internet Security Research Group, C=US	93:3C:6D:DE:E9:5C:9C:41:A4:0F:9F:50:49:3D:82:BE:03:AD:87:BF	Valid from Jan 20, 2021 to Sep 30, 2024.

Total 5

8. Then, to allow to use the REST APIs from WeatherAPI with http, or in some cases even with http, you need to set up the outbound HTTP connection (proxy). You may open a shell on a DX Core pod, as documented in https://opensource.hcltechsw.com/digital-experience/latest/deployment/manage/container_configuration/run_core_config_engine/ and then read and update the proxy-xml, using https://opensource.hcltechsw.com/digital-experience/latest/extend_dx/portlets_development/web2_ui/outbound_http_connection/cfg_outbound_http_connections/adm_tools_for_cfg_outbound_http_conn/cfg_outbound_http_using_cfgtsk/outbhttp_cfg_tsk_update/. You would need to create a temporary XML file to describe how the HTTP Outbound Connection must be configured into your DX environment. You may create the file using the touch command, e.g.:

```
touch /tmp/weatherapi-policy.xml
```

9. Then, enter the following policy rules configuration for the weatherapi policy for the Get method for both http and https and save the file. You may also use this from the downloaded resources.

```
<?xml version="1.0" encoding="UTF-8"?>

<proxy-rules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.ibm.com/xmlns/prod/sw/http/outbound/proxy-config/2.0">

  <policy active="true" basic-auth-support="true" name="http_weatherapi_policy"
url="http://api.weatherapi.com/*">

    <actions>

      <method>GET</method>

    </actions>

  </policy>

  <policy active="true" basic-auth-support="true" name="https_weatherapi_policy"
url="https://api.weatherapi.com/*">

    <actions>

      <method>GET</method>

    </actions>

  </policy>
</proxy-rules>
```

10. Finally, run the command to launch the **ConfigEngine** task update-outbound-http-connection-config and import the new Outbound HTTP Connection configurations. This command may look like:

```
/opt/HCL/WebSphere/Profiles/wp_profile/ConfigEngine/ConfigEngine.sh update-outbound-http-connection-config -DConfigFileName=/tmp/weatherapi-policy.xml -DPortalAdminId=<portal-admin-id> -DPortalAdminPwd=<portal-admin-password> -DWasUserId=<was-admin-id> -DWasPassword=<was-admin-password>
```

11. And now you may update your data source URL, now stored in the Global Config WeatherAPI Base URL, to use https instead of http for your REST API calls.

You have successfully configured the **Outbound HTTP Connection** to work with **Weather APIs**.

Conclusion

Using this tutorial, you learned how to use HCL Digital Data Connector to manage the integration of external data sources in an HCL Digital Experience site and manage a global and user configuration to make it easy to define what and how to show.

Resources

Refer to the following resources to learn more:

HCL Digital Experience Home - https://hclsw.co/dx

HCL Digital Experience on HCL Solutions Factory (SoFy) - https://hclsofy.com/

HCL Software - https://hclsw.co/software

HCL Product Support - https://hclsw.co/product-support

HCL DX Product Documentation - https://hclsw.co/dx-product-documentation

HCL DX Support Q&A Forum - https://hclsw.co/dx-support-forum

HCL DX Video Playlist on YouTube - https://hclsw.co/dx-video-playlist

HCL DX Product Ideas - https://hclsw.co/dx-ideas

HCL DX Product Demos - https://hclsw.co/dx-product-demo

HCL DX Did You Know? Videos - https://hclsw.co/dx-dyk-videos

HCL DX GitHub - https://hclsw.co/dx-github

HCL DX Web Developer Toolkit - https://github.com/HCL-TECH-SOFTWARE/WebDevToolkitForDx

Legal statements

This edition applies to version 9.5, release 207 of HCL Digital Experience and to all subsequent releases and modifications until otherwise indicated in new editions. It uses the DX Solution Modules with additional resources in HCL SoFy.

When you send information to HCL Technologies Ltd., you grant HCL Technologies Ltd. a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with HCL Technologies Ltd.

Disclaimers

This report is subject to the HCL Terms of Use (https://www.hcl.com/terms-of-use) and the following disclaimers:

The information contained in this report is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied, including but not limited to the implied warranties of merchantability, non-infringement, and fitness for a particular purpose. In addition, this information is based on HCL's current product plans and strategy, which are subject to change by HCL without notice. HCL shall not be responsible for any direct, indirect, incidental, consequential, special or other damages arising out of the use of, or otherwise related to, this report or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from HCL or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of HCL software.

References in this report to HCL products, programs, or services do not imply that they will be available in all countries in which HCL operates. Product release dates and/or capabilities referenced in this presentation may change at any time at HCL's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. The underlying database used to support these reports is refreshed on a weekly basis. Discrepancies found between reports generated using this web tool and other HCL documentation sources may or may not be attributed to different publish and refresh cycles for this tool and other sources. Nothing contained in this report is intended to, nor shall have the effect of, stating.

or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results. You assume sole responsibility for any results you obtain or decisions you make as a result of this report. Notwithstanding the HCL Terms of Use (https://www.hcl.com/terms-of-use), users of this site are permitted to copy and save the reports generated from this tool for such users own internal business purpose. No other use shall be permitted.