JSON API List Data Store

New Feature
This is a new feature in Joget DX 8.

Introduction

JSON API List Data Store allows one to populate a List using JSON calls.

JSON API Properties

New feature

New feature in **Joget DX version 8.1 onwards.**

1. Added new property `Auto Handling Filters` for JSON API List Binder and perform filtering using **CQengine**
2. Support using object key to create row.
3. Added new property `Disable object keys as columns` to join object key as single column value and object value as another column value.

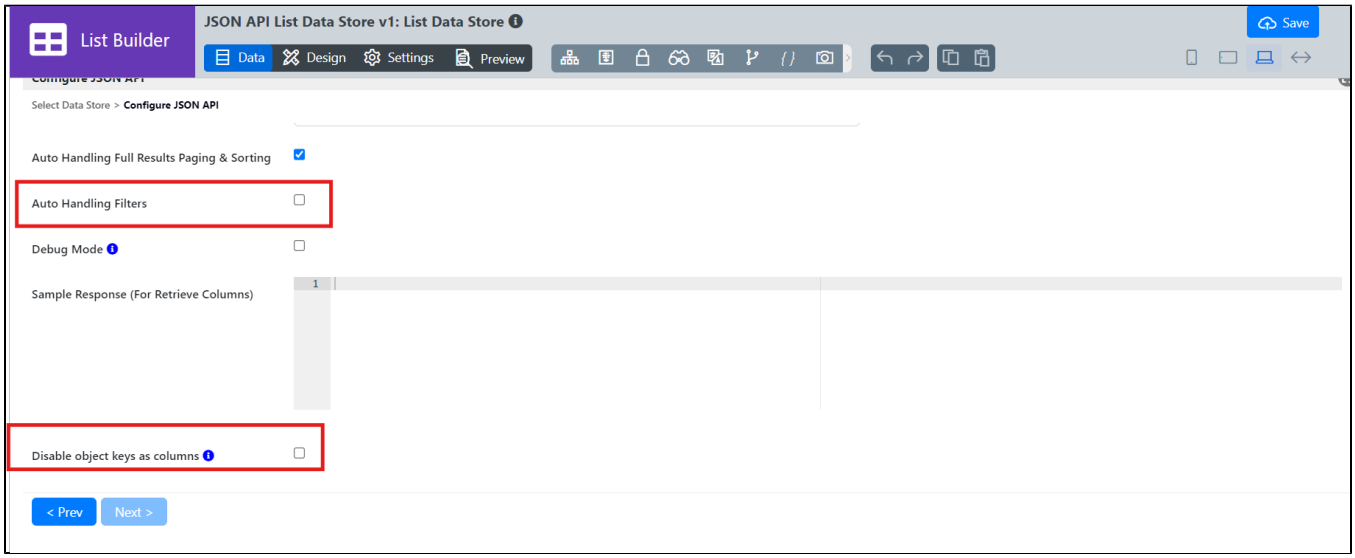
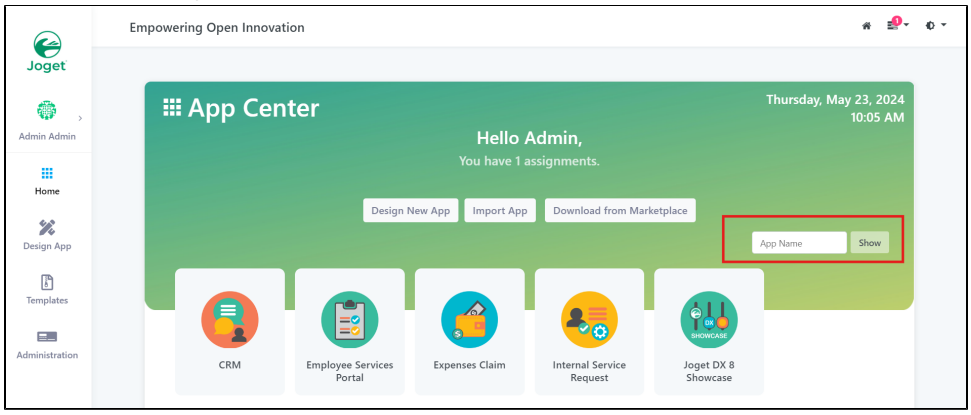


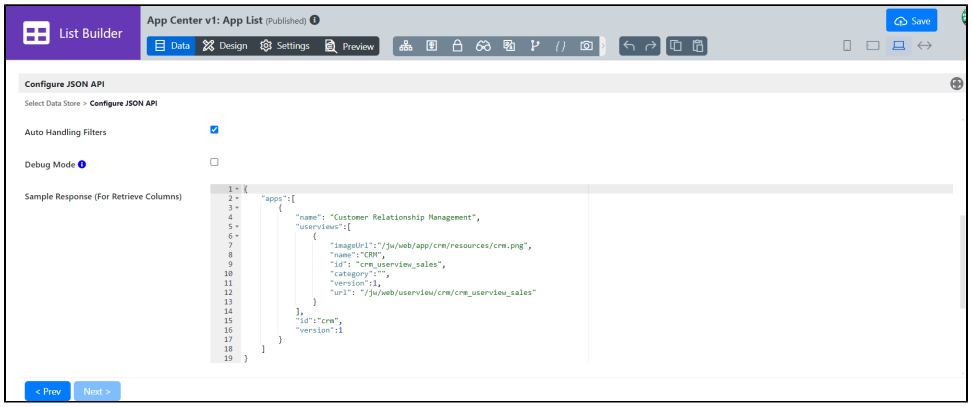
Figure 1 : Auto Handling Filters and Disable object keys as column in Joget DX version 8.1 onwards.

Change 1: New property `Auto Handling Filters` in JSON API List Data Store

- Once enabled, you can use the List Builder filter to filtering the JSON API response even the API end point does not support searching/filtering options.
- Currently, it support all the filter types in List Builder.
- Example usage of **Auto Handling Filters** in App center Homepage > Ability to have a search filter within JSON API List Data Store



- Configure JSON API for Auto Handling Filters
  - Open the List Builder in App center,
  - Click **Data > Configure JSON API**
  - Check the Auto Handling Filters box.
  - Add Sample Response to retrieve the specific column to be filtered



Change 2: New row creation mapping method in JSON API List Binder

- Before the change, rows only able to create through array in the JSON API response. Example, using `data` in `Base JSON Object Name for Multirow Data` field to create row

```
{
   data : [
      {
         key1 : "value1",
         key2 : "value2"
      }
   ]
}
```

- When the JSON API response does not having array but only containing object, it will not able to create row previously. Example:

```
{
  "data": {
    "key1": {
      "attr1": "Value 1",
      "attr2": "Value 2",
    },
    "key2": {
      "attr1": "Value 1",
      "attr2": "Value 2",
    }
  }
}
```

Let say, we want each key in `data` object to create as rows, we can use `data<>` in `Base JSON Object Name for Multirow Data` in this new change. It will tell the plugin to loop all the keys of `data` and create it as row.

- This changes also supporting inner object key as rows. Example:

```
{
  "key1" {
      "data": {
          "ckey1": {
              "attr1": "Value 1",
              "attr2": "Value 2",
          }
      }
   },
   "key2" {
      "data": {
          "ckey2": {
              "attr1": "Value 1",
              "attr2": "Value 2",
          }
      }
   }
}
```

We want to create row based on the `data` object keys. In this case, using `<>.data<>` in `Base JSON Object Name for Multirow Data`. The first `<>` is to tell it loop all the keys of root object, then `.data<>` is to tell it to create row based on object keys of `data` object.

Change 3: New property `Disable object keys as columns` in JSON API List Binder

- Before this changes, all the keys of object will be create as columns. Example:

```
{
   data : [
      {
         key1 : "value1",
         key2 : "value2"
         key3 : {
            ckey1 : "child value1",
            ckey2 : "child value2"
         }
      }
   ]
}
```

Will resulting columns `key1`, `key2`, `key3.ckey1`, `key3.ckey2`.

- In this new changes, once enabled `Disable object keys as columns`, for none `Base JSON Object Name for Multirow Data` object, it will join the object keys as single column and the value as another column. For the same response above, it will resulting column as `key1`, `key2`, `key3.KEY`, `key3.VALUE`. The value of `key3.KEY` will be `ckey1;ckey2` and value of `key3.VALUE` will be `child value1;child value2`

Configure JSON API

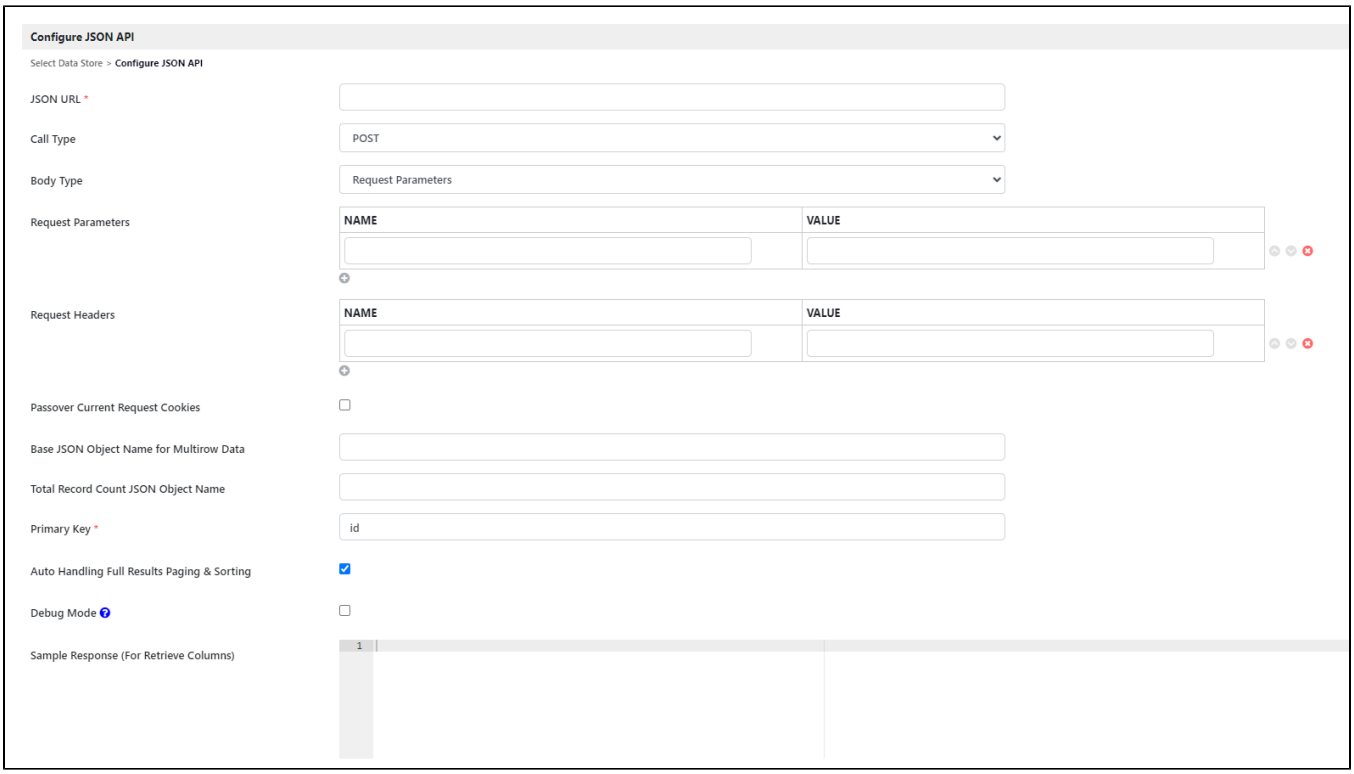


Figure 1: Configure JSON API

Name	Description

JSON URL	URL to be called. Mandatory Field.
	`//You may also insert the following URL parameters to enable pagination`
	`"https://example.com/api/data?pageSize={size}&startOffset={start}"`
	• `pageSize={size}`: This parameter is intended to specify the number of items (data entries) to be returned in a single page or response. The actual value for `{size}` would be replaced with a numerical value indicating the desired page size.
	• `startOffset={start}`: This parameter is intended to specify the starting offset or index from where the data should be retrieved. The actual value for `{start}` would be replaced with a numerical value indicating the starting point for fetching data.

Call Type	• GET • POST • PUT • DELETE

<table>
<tr><td>Body Type

(Only Applicable to Call type = POST or PUT)</td><td>Select the POST or PUT method:

• Request Parameters
  ○ Add name(s) and value(s) to the request header

<table><tr><th>Field</th><th>Description</th></tr><tr><td>Name</td><td>Name (or Type) of the Request Header</td></tr><tr><td>Value</td><td>Link or Value of the Request Header</td></tr></table>
• Request Parameters as JSON Payload
  ○ When POST Method is set to "POST Parameters", these parameters will be sent as a UrlEncodedFormEntity.

  When POST Method is set to "POST Parameters as JSON Payload", these parameters will be sent as a StringEntity in a form of an escaped JSON string.

• Custom JSON Payload
  ○ Write your own JSON to be the payload. It will be sent as a StringEntity.

  This option is available only when "Custom JSON Payload" is selected.</td></tr>
<tr><td>Request Headers</td><td>Add name(s) and value(s) to the request header

<table><tr><th>Field</th><th>Description</th></tr><tr><td>Name</td><td>Name (or Type) of the Request Header</td></tr><tr><td>Value</td><td>Link or Value of the Request Header</td></tr></table></td></tr>
<tr><td>Passover Current Request Cookies</td><td>When checked, Current Request Cookies will be passed to the server with the request.</td></tr>
<tr><td>Base JSON Object Name for Multirow Data</td><td>Name of the base JSON object (e.g. data).</td></tr>
<tr><td>Total Record Count JSON Object Name</td><td>Total number of records in the JSON object (e.g. data.total)</td></tr>
<tr><td>Primary Key</td><td>Unique Identifier of the JSON object (e.g. data.id). Mandatory Field.</td></tr>
<tr><td>Auto Handling Full Results Paging & Sorting</td><td>When checked, This option enables automatic sorting for the results returned through the JSON URL.

Uncheck this, if you are performing pagination.</td></tr>
<tr><td>Debug Mode</td><td>Show relevant debug entries in the server log for debugging purposes.</td></tr>
</table>

Sample Response (for Retrieve Columns)	Enter a Sample Response of how the JSON object is formatted.
	
	Must insert a sample response into the field if you are performing pagination.
	
	e.g.
	
	JSON Sample Response
	
	{
	"data":{
	"id": "G-001",
	"description": "",
	"name": "Managers",
	"organization.name": ""
	}
	}

Download Demo App

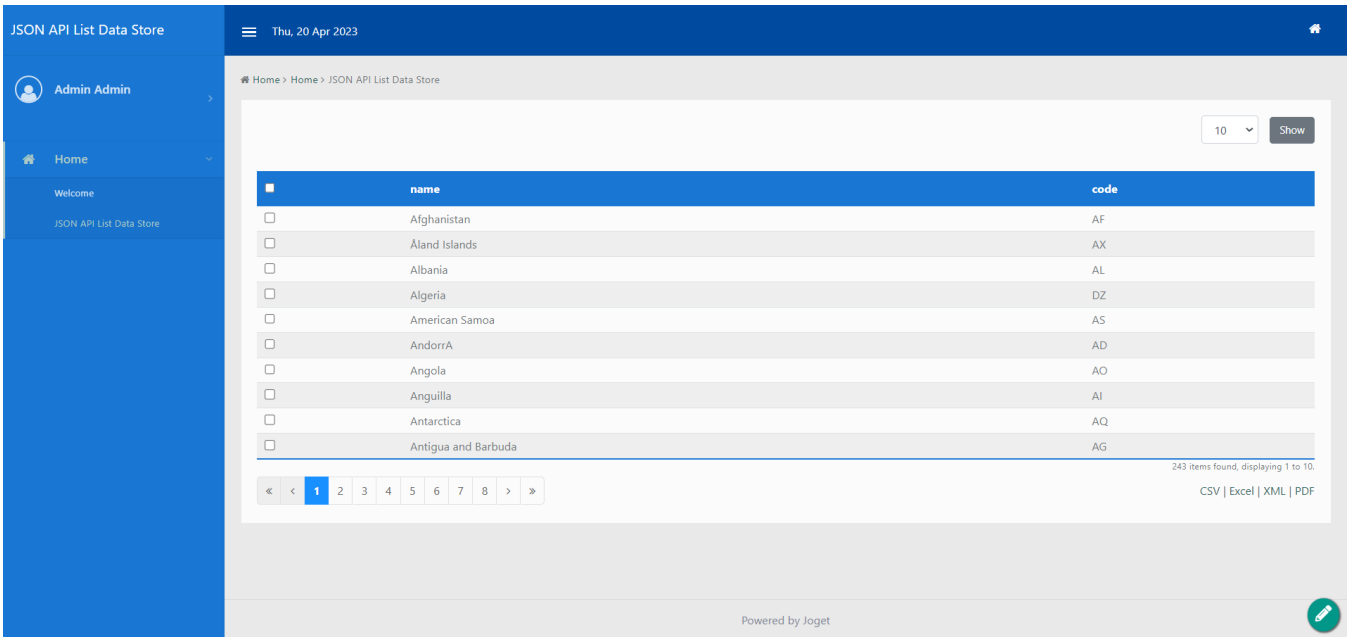


Figure 2: Download the demo app below to view how JSON API is used to populate a list.

- app_kb_dx8_json_api_list_data_store.jwa