

Joget DX 8

Integrating With External System

Prerequisites

1. Understand Joget components in depth and able to make full use of its components.
2. Avid software developer with vast know-hows of web application technologies.

Content

1. Introduction
2. Single Sign On (SSO)
3. Embedding UI Page in an iFrame
4. SOAP Tool
5. JSON Tool
6. JSON API
7. JSON API Authentication
8. Using API Builder



Chapter 1

Introduction

Introduction

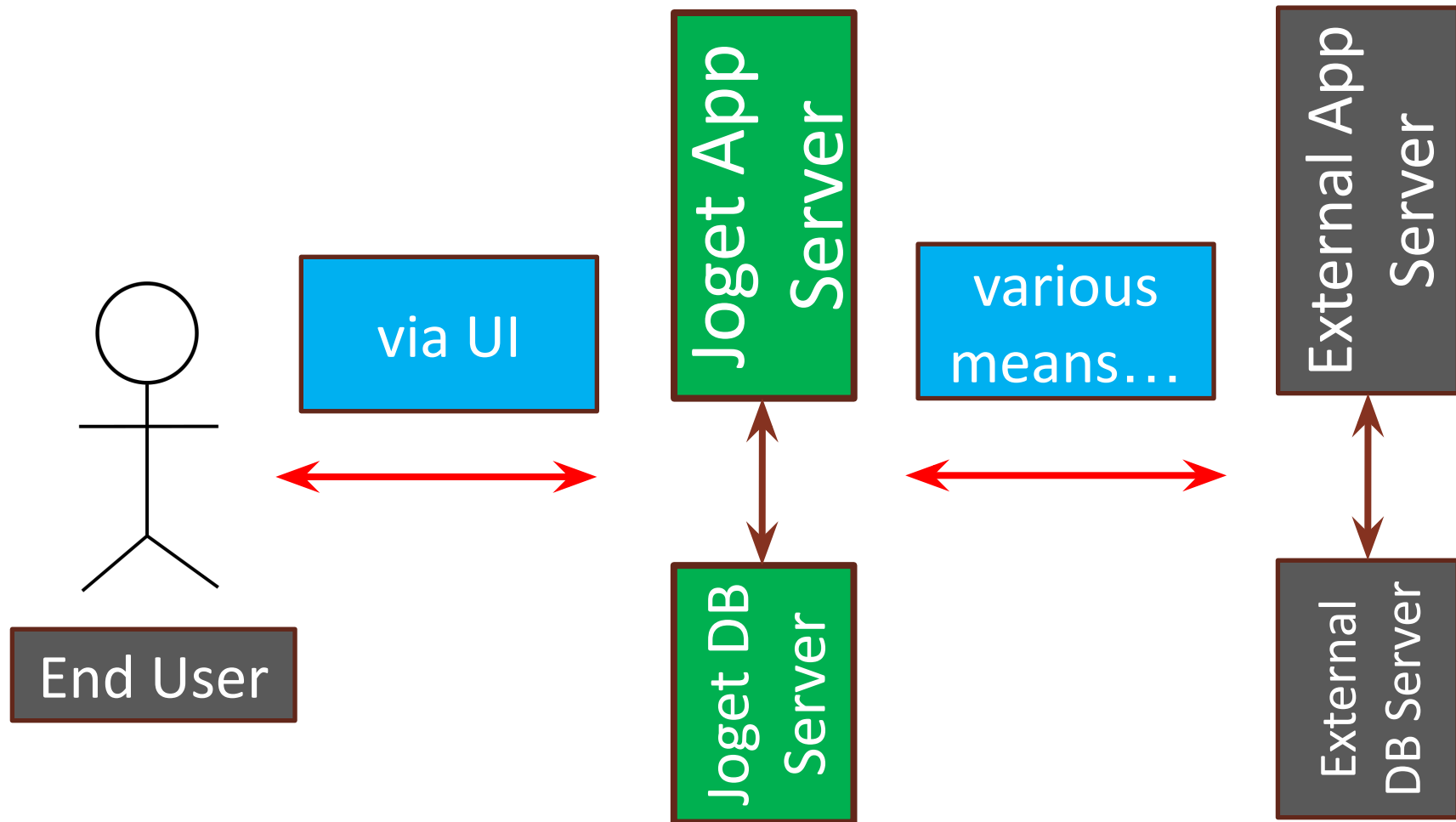
- There are various ways to integrate Joget with external applications.
- Typically, there are 4 main components and the client user to be considered about.
 1. Joget Application Server
 2. Joget Database Server
 3. External Application Server
 4. External Application Database Server
 5. Client user

Before We Start!

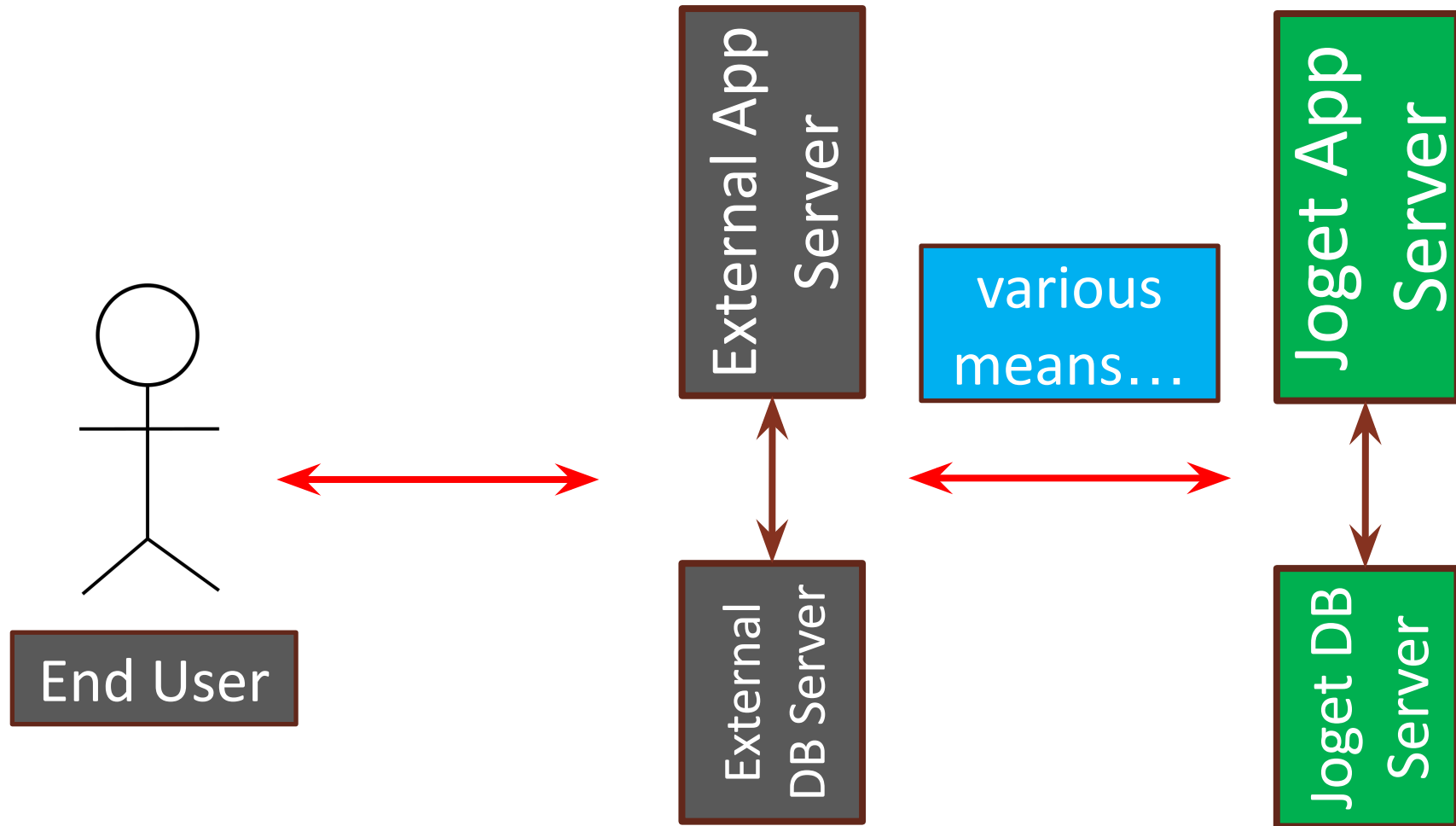
- Ensure you have configured the **API Domain Whitelist** in **General Settings** to allow JSON API requests.
- If a request is from a non-whitelisted domain, the response will be a HTTP 400 Bad Request

⚙️ System Settings	Performance Analyzer Threshold (ms)	<input type="text"/>
		<i>default: 100</i>
⚙️ General Settings	API Domain Whitelist (Separated by ';')	<input type="text" value="*"/>
📊 Datasource & Profile Settings		
👤 Directory Manager Settings		
🔌 Manage Plugins	API IP Whitelist (Separated by ';')	<input type="text" value="*"/>
🌐 Platform Translation		

Joget as the Front-end

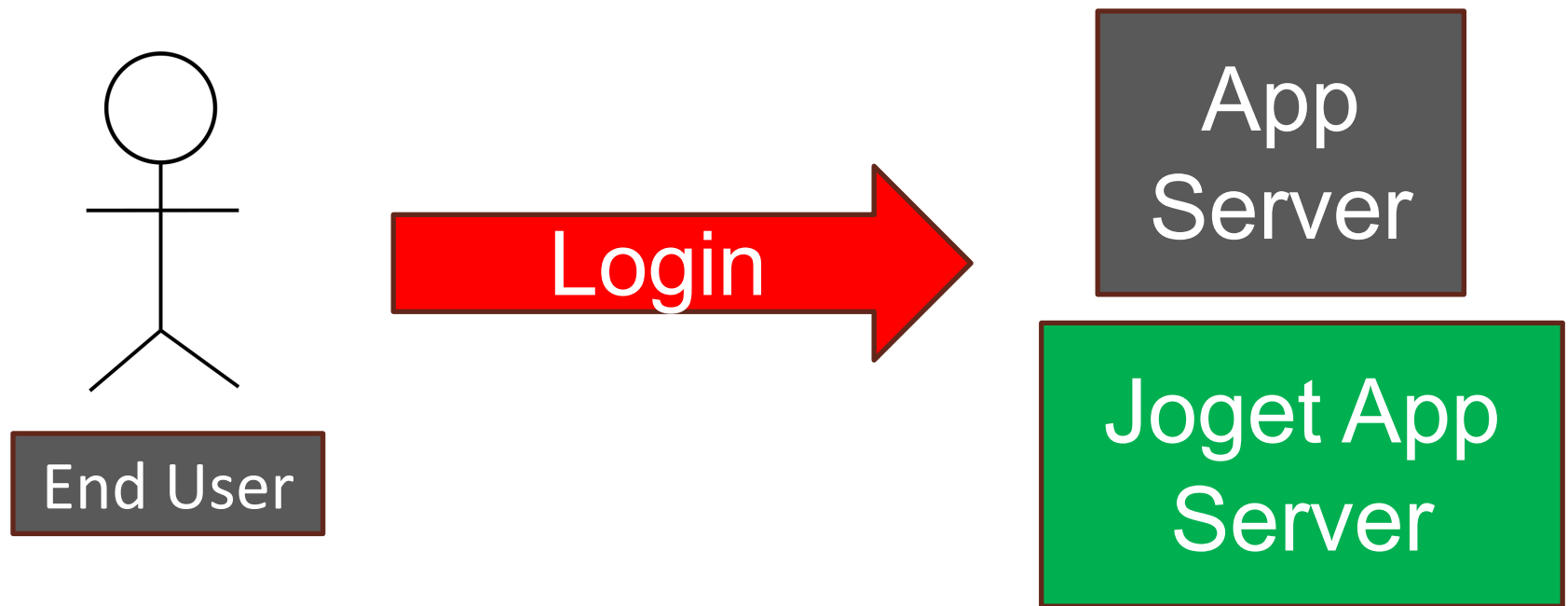


External App as the Front-end



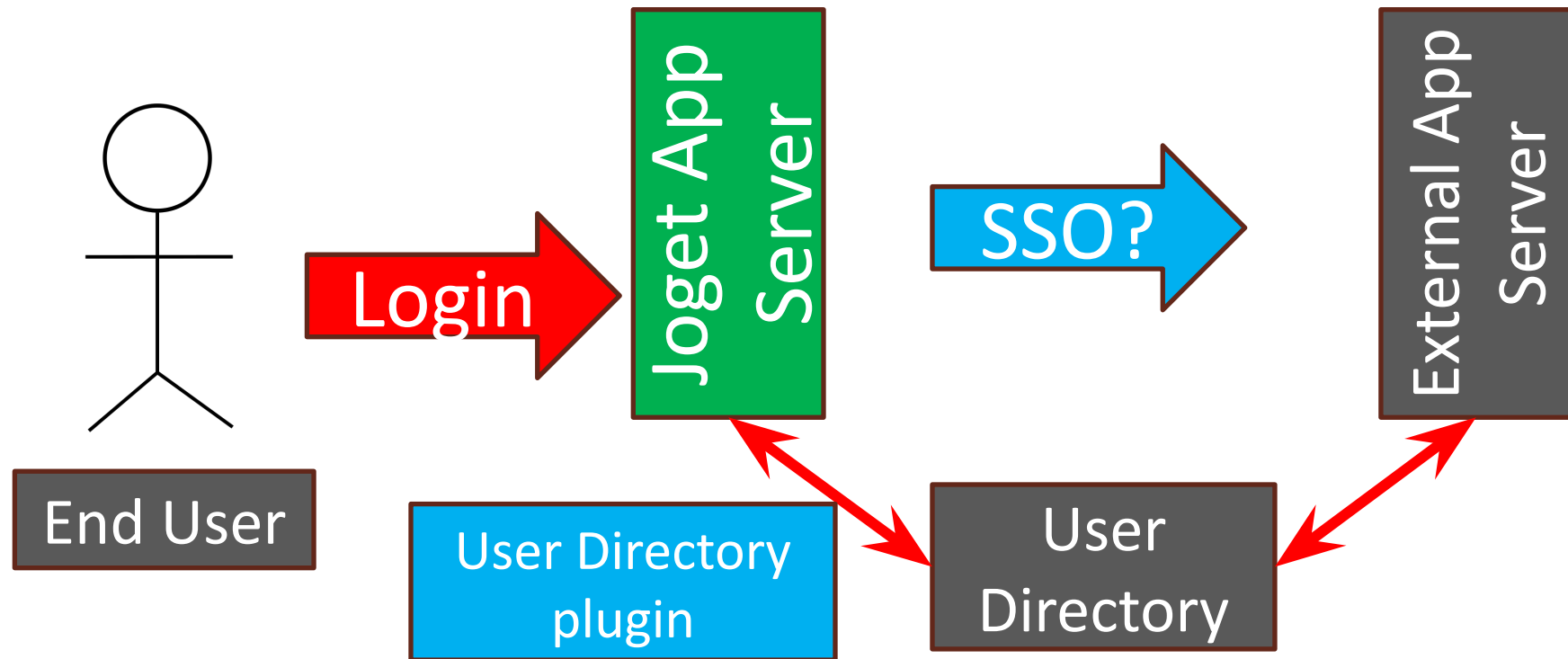
User Directory

- Is User session important?
- How do you want to handle the user session between the servers (Joget server and external application server)?



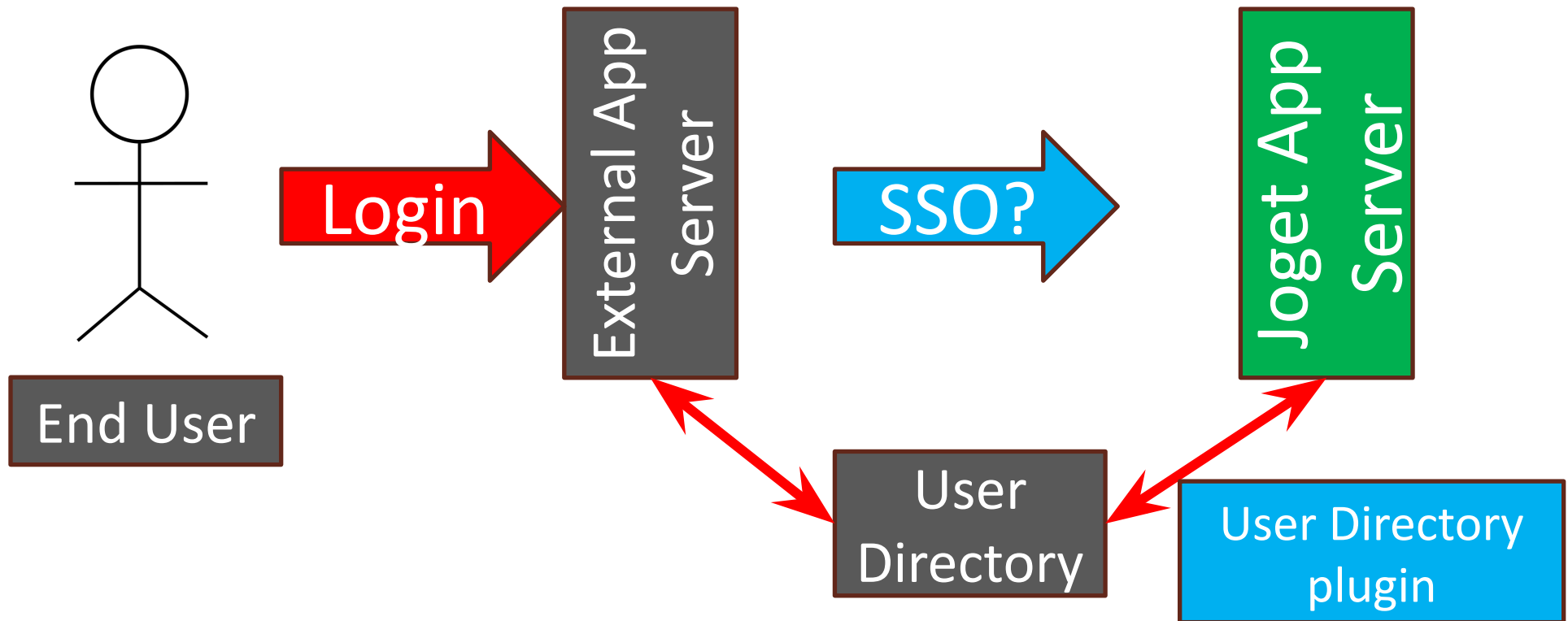
User Directory – Joget as Front-end

- Is the identity of the End User important to the External App Server?



User Directory – Ext App as Front-end

- Is the identity of the End User important to the Joget App Server?



Chapter Review

- Understand various ways of integration with Joget and external application server.



Chapter 2

Single Sign On (SSO)

Introduction

- User logs in to external system and implicitly gains access to Joget without being prompted to login again.
- Can be achieved via [Directory Manager plugins](#) and programmatically via Web Service plugin

Reference:

<https://dev.joget.org/community/display/DX8/Single+Sign+On+-+SSO>

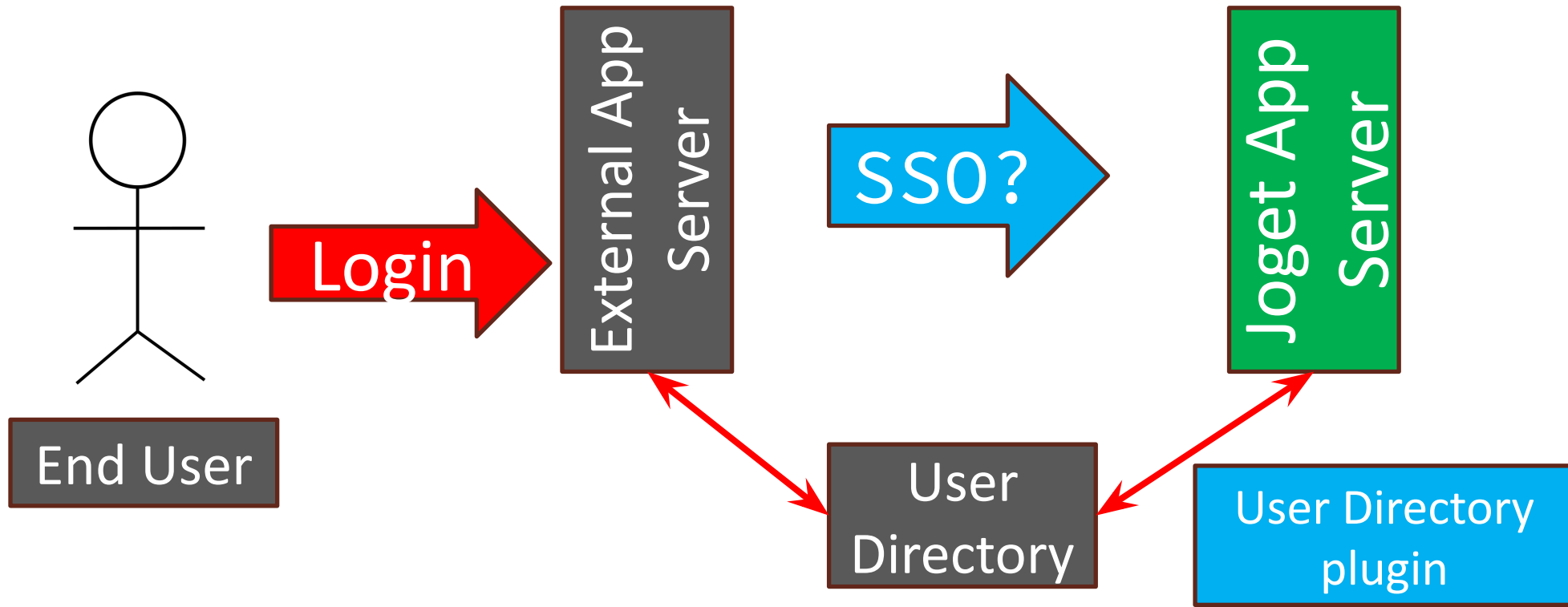
Various SSO Methods Available...

- G Suite
- Kerberos
- SAML (IDP initiated) (e.g.: Sharepoint, some ADs, etc.)
- OpenID Connect
- Programmatically via Web Service Plugin (Java)
- API Builder – SSO

See KB for detailed guides for all the above.

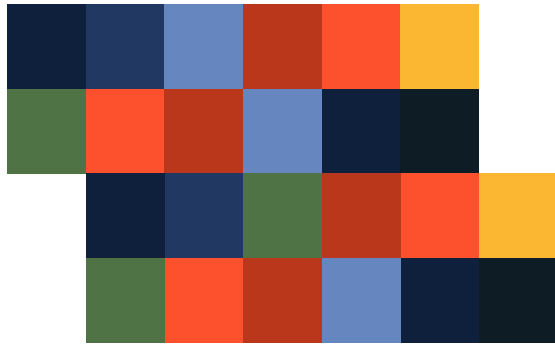
Setup Suggestion

- Can you now imagine the following setup?



Chapter Review

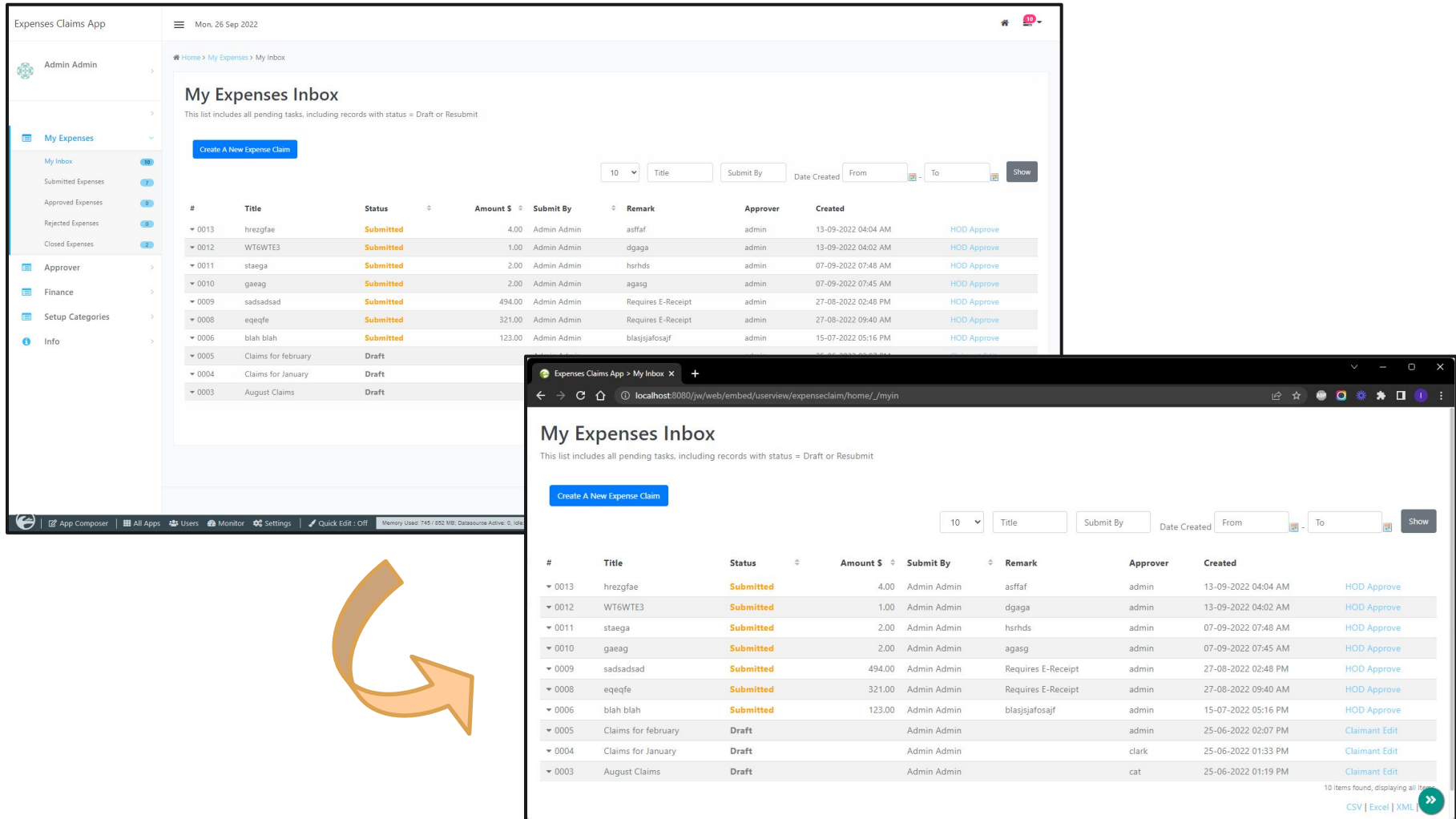
- Understand on how to use the SSO feature.



Chapter 3

Embedding UI Page in an iFrame

Embedding UI Page in an IFrame



The diagram illustrates the process of embedding a Joget UI page into an iFrame. On the left, the full Joget application is shown, including a sidebar with navigation options like 'My Expenses', 'Approvers', 'Finance', and 'Info'. The main content area displays 'My Expenses Inbox' with a table of expense claims. On the right, a browser window shows the same 'My Expenses Inbox' page embedded within an iFrame. The browser's address bar shows the URL: `localhost:8080/jv/web/embed/userview/expenseclaim/home/_myin`. An orange arrow points from the application window to the iFrame window, indicating the embedding process.

My Expenses Inbox

This list includes all pending tasks, including records with status = Draft or Resubmit

[Create A New Expense Claim](#)

10 | Title | Submit By | Date Created | From | To | Show

#	Title	Status	Amount \$	Submit By	Remark	Approver	Created
▼ 0013	hrezgfae	Submitted	4.00	Admin Admin	asffaf	admin	13-09-2022 04:04 AM HOD Approve
▼ 0012	WT6WTE3	Submitted	1.00	Admin Admin	dgaga	admin	13-09-2022 04:02 AM HOD Approve
▼ 0011	staega	Submitted	2.00	Admin Admin	hsrhds	admin	07-09-2022 07:48 AM HOD Approve
▼ 0010	gaeag	Submitted	2.00	Admin Admin	agasg	admin	07-09-2022 07:45 AM HOD Approve
▼ 0009	sadsadsad	Submitted	494.00	Admin Admin	Requires E-Receipt	admin	27-08-2022 02:48 PM HOD Approve
▼ 0008	eqeqfe	Submitted	321.00	Admin Admin	Requires E-Receipt	admin	27-08-2022 09:40 AM HOD Approve
▼ 0006	blah blah	Submitted	123.00	Admin Admin	blajsjsafosajf	admin	15-07-2022 05:16 PM HOD Approve
▼ 0005	Claims for february	Draft				admin	25-06-2022 02:07 PM Claimant Edit
▼ 0004	Claims for January	Draft				clark	25-06-2022 01:33 PM Claimant Edit
▼ 0003	August Claims	Draft				cat	25-06-2022 01:19 PM Claimant Edit

10 Items found, displaying all items

[CSV](#) | [Excel](#) | [XML](#)

Embedding UI Page in an IFrame

- The header, menu & footer of UI can be removed by following:
 - By adding parameter “**embed=true**” in the **URL** OR
 - By modifying the **URL**

From

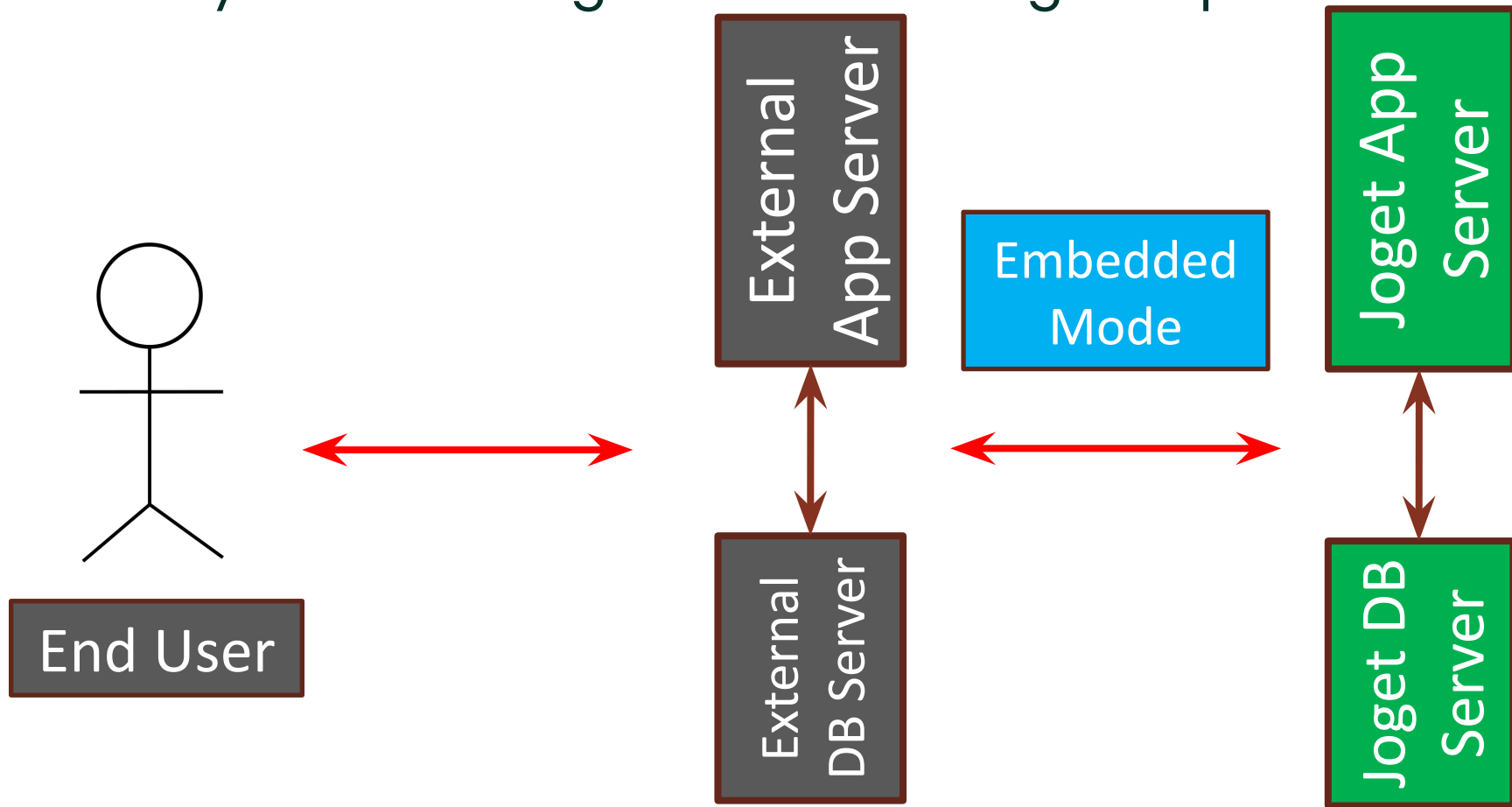
http://localhost:8081/jw/web/userview/crm/crm_userview_sales

To

[http://localhost:8081/jw/web/**embed**/userview/crm/crm_userview_sales](http://localhost:8081/jw/web/embed/userview/crm/crm_userview_sales)

Setup Suggestion

- Can you now imagine the following setup?



Exercise

- Choose a UI Page.
- Add in the necessary parameter to turn it into the Embedded mode.
- Use iFrame to display it in your custom app.

Chapter Review

- Understands the purpose of using Embedded mode of UI pages.



Chapter 4

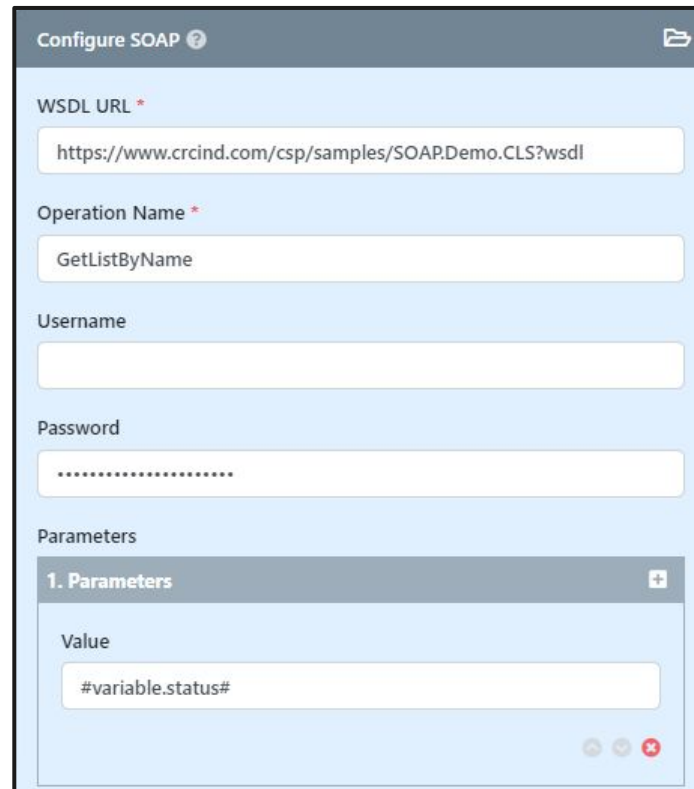
Using the SOAP Tool

Introduction

- The SOAP Tool allows one to invoke web service for integration purpose to return useful information from external sources into the process instance.
- Reference:
<https://dev.joget.org/community/display/DX8/SOAP+Tool>

How it works?

- The SOAP Tool will call the Web Service configured, passes the set of parameters set.



How it works?

- Data will be returned in JSON format.

```
INFO 30 Sep 2022 15:28:20 org.joget.plugin.enterprise.SoapTool -  
{"GetListByNameResult":{"PersonIdentification":{"DOB":"1942-08-15","ID":"132","Name"  
:"Xiang,Mark W.,"SSN":"332-29-2935"}}}
```

```
{  
  "GetListByNameResult":{  
    "PersonIdentification":{  
      "DOB":"1942-08-15",  
      "ID":"132",  
      "Name":"Xiang,Mark W.",  
      "SSN":"332-29-2935"  
    }  
  }  
}
```

How it works?

- Data can be then stored into Form table or Workflow Variable.

Store to Form

Form

user

Base XML Object Name for Multirow Data

GetListByNameResult.PersonIdentification

Field Mapping

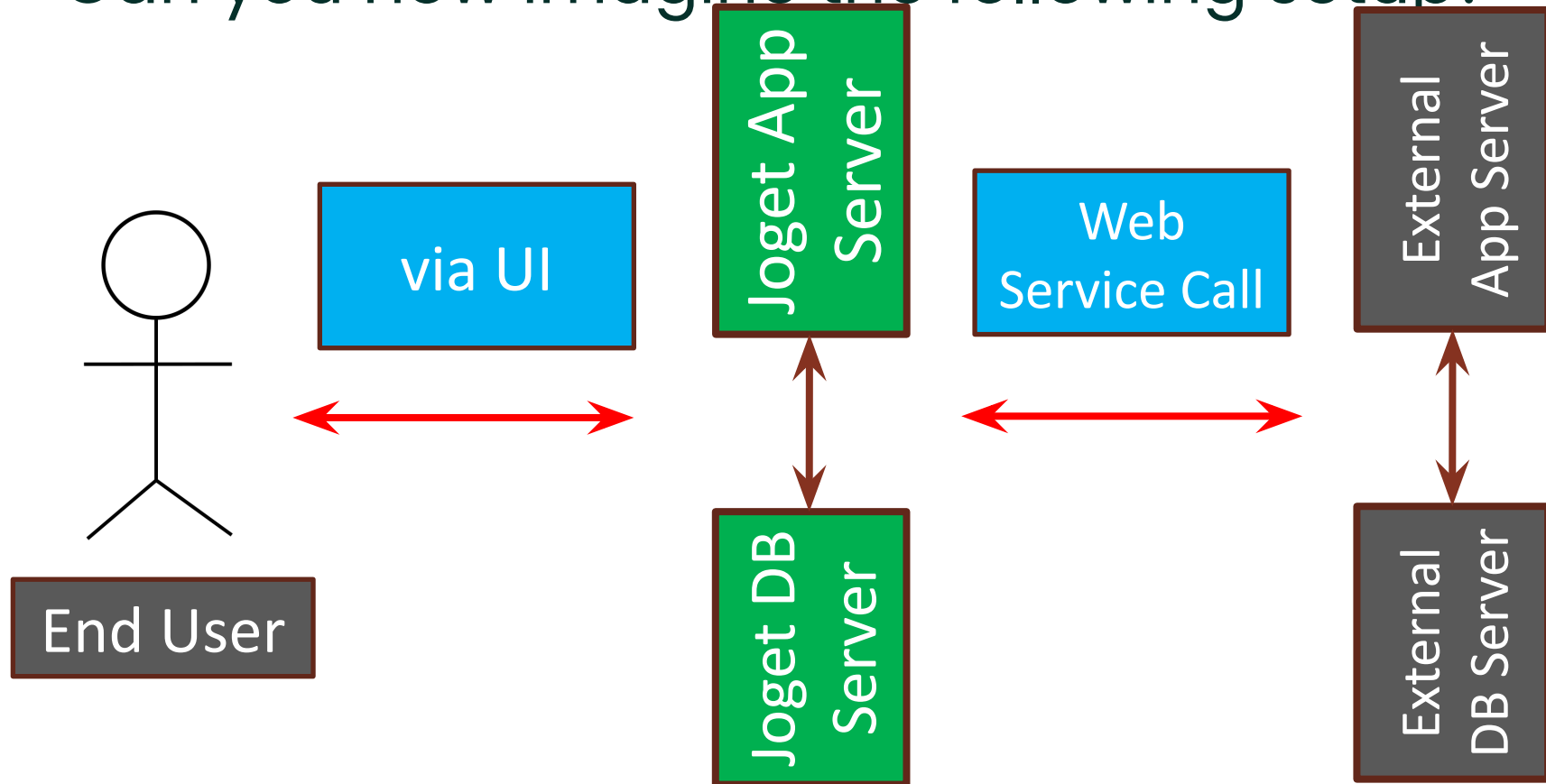
FIELD ID	XML OBJECT NAME
name	GetListByNameResult.PersonIdentification
user_id	GetListByNameResult.PersonIdentification
ssn	GetListByNameResult.PersonIdentification
dob	GetListByNameResult.PersonIdentification
fk	#process.recordId#

Exercise

- Go to <https://www.crcind.com/csp/samples/SOAP.Demo.CLS>
- Select a service, e.g.: "Get List By Name"
- Create a new Joget App with a **Activity** → **Tool** → **Activity** process.
- Map the Tool to a **SOAP Tool**, configure accordingly.

Setup Suggestion

- Can you now imagine the following setup?



Just In Case...

NOTE:

You can download this chapter's completed
Joget App. **18.4.jwa**

Chapter Review

- Be able to make calls from Joget to other web services by using the **SOAP Tool**.



Chapter 5

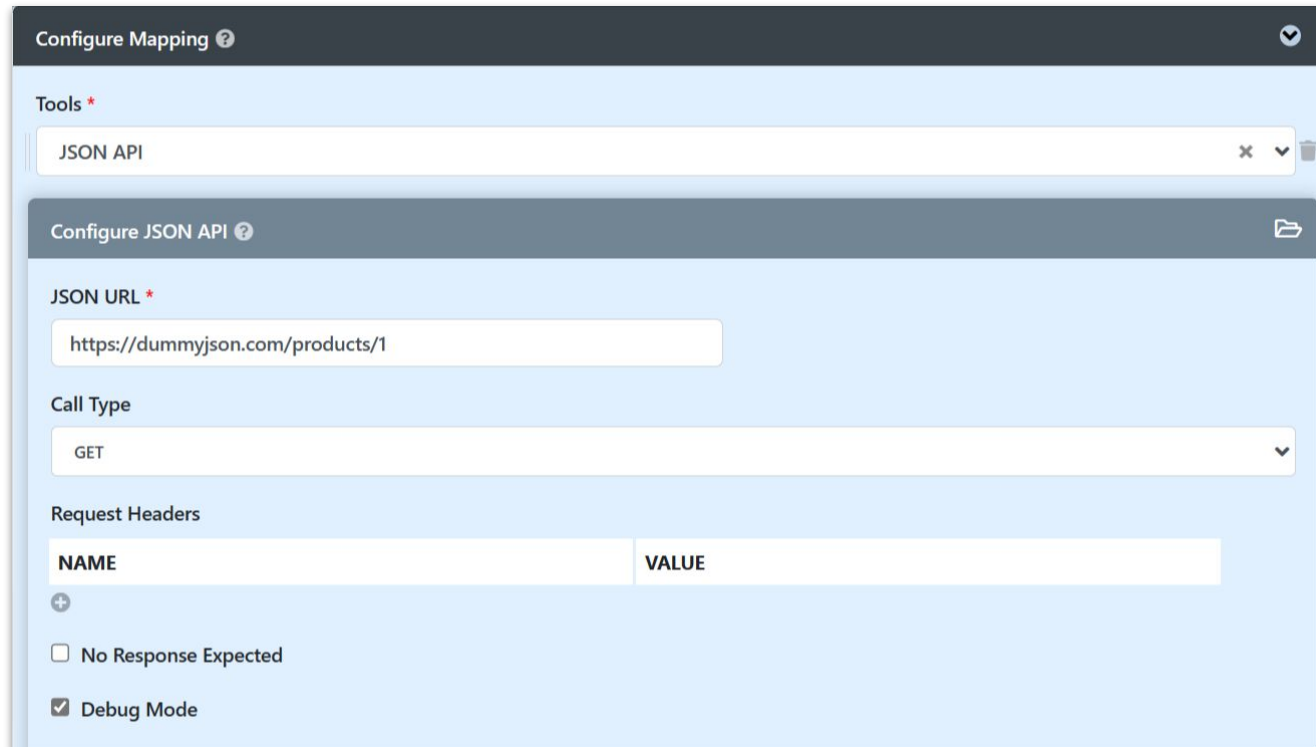
Using the JSON API Tool

Introduction

- The JSON API Tool enables one to issue a JSON web service call, and to save the returned data into Joget's form data and/or into the process's workflow variable.
- The JSON API Tool can be used as a process tool, a Form Data Store and as a Data Store Option for Lists.
- Reference:
<https://dev.joget.org/community/display/DX8/JSON+API+Tool>

JSON Tool For Processes

- The JSON Tool will call the JSON API endpoint, along with any required parameters.



The screenshot shows the 'Configure Mapping' window for the JSON Tool. It includes a 'Tools' section with 'JSON API' selected. Below is the 'Configure JSON API' section with the following fields:

- JSON URL ***:
- Call Type**:
- Request Headers**: A table with columns 'NAME' and 'VALUE'.
- ☐ No Response Expected
- ☒ Debug Mode

NAME	VALUE
------	-------

JSON Tool – How it works?

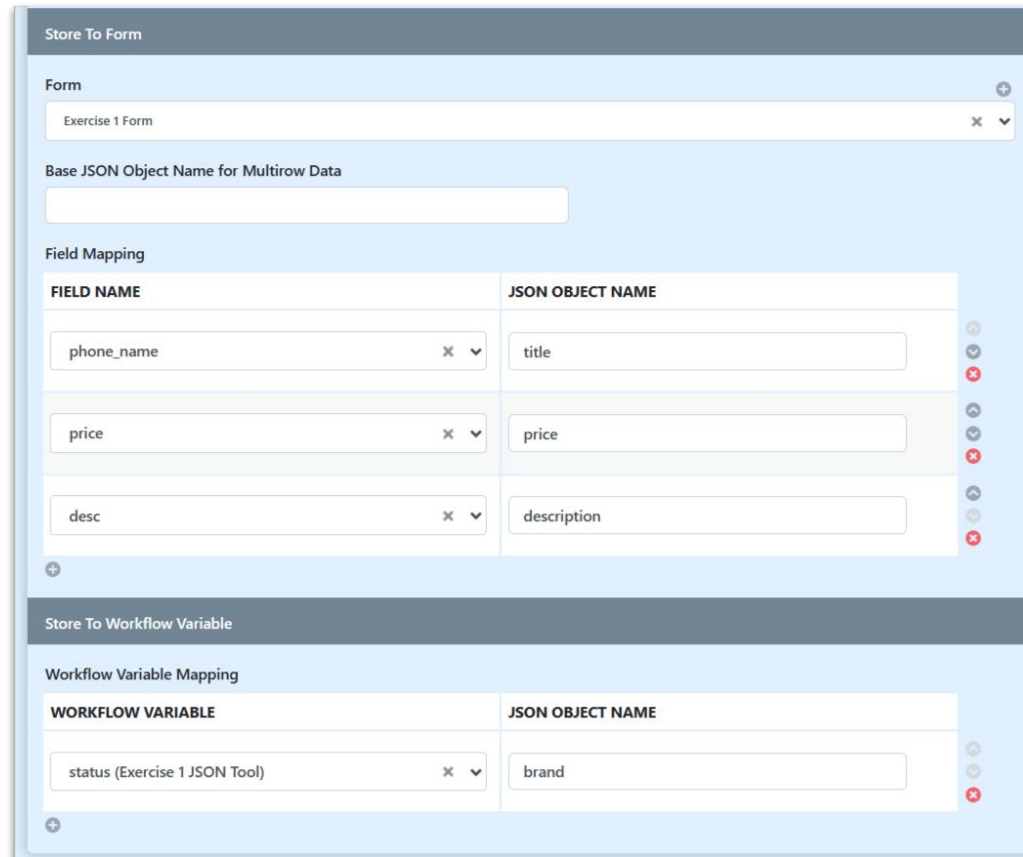
- Data will be returned in JSON format.

```
INFO 15 Feb 2023 15:06:40 org.joget.apps.app.service.JsonApiUtil - GET :  
https://dummyjson.com/products/1  
INFO 15 Feb 2023 15:06:41 org.joget.apps.app.service.JsonApiUtil -  
https://dummyjson.com/products/1 returned with status : 200  
INFO 15 Feb 2023 15:06:41 org.joget.apps.app.service.JsonApiUtil -  
{  
  "id":1,  
  "title":"iPhone 9",  
  "description":"An apple mobile which is nothing like  
apple",  
  "price":549,  
  "discountPercentage":12.96,  
  "rating":4.69,  
  "stock":94,  
  "brand":"Apple",  
  "category":"smartphones",  
  "thumbnail":"https://i.dummyjson.com/data/products/1/thu  
mbnail.jpg",  
  "images":["https://i.dummyjson.com/data/products/1/1.jpg",  
"https://i.dum  
myjson.com/data/products/1/2.jpg",  
"https://i.dummyjson.com/data/products/1/3.jpg",  
"h  
ttps://i.dummyjson.com/data/products/1/4.jpg",  
"https://i.dummyjson.com/data/products  
/1/thumbnail.jpg"]  
}
```

```
{  
  "title":"iPhone 9",  
  "description":"An apple mobile which is nothing like  
apple",  
  "price":549,  
  "brand":"Apple"  
}
```

JSON Tool – How it works?

- Data can be then stored into Form table or Workflow Variable.



The screenshot displays the 'JSON Tool' configuration interface, divided into two main sections: 'Store To Form' and 'Store To Workflow Variable'.

Store To Form Section:

- Form:** A dropdown menu is set to 'Exercise 1 Form'.
- Base JSON Object Name for Multirow Data:** An empty text input field.
- Field Mapping:** A table mapping JSON object fields to form fields.

FIELD NAME	JSON OBJECT NAME
phone_name	title
price	price
desc	description

Store To Workflow Variable Section:

- Workflow Variable Mapping:** A table mapping JSON object fields to workflow variables.

WORKFLOW VARIABLE	JSON OBJECT NAME
status (Exercise 1 JSON Tool)	brand

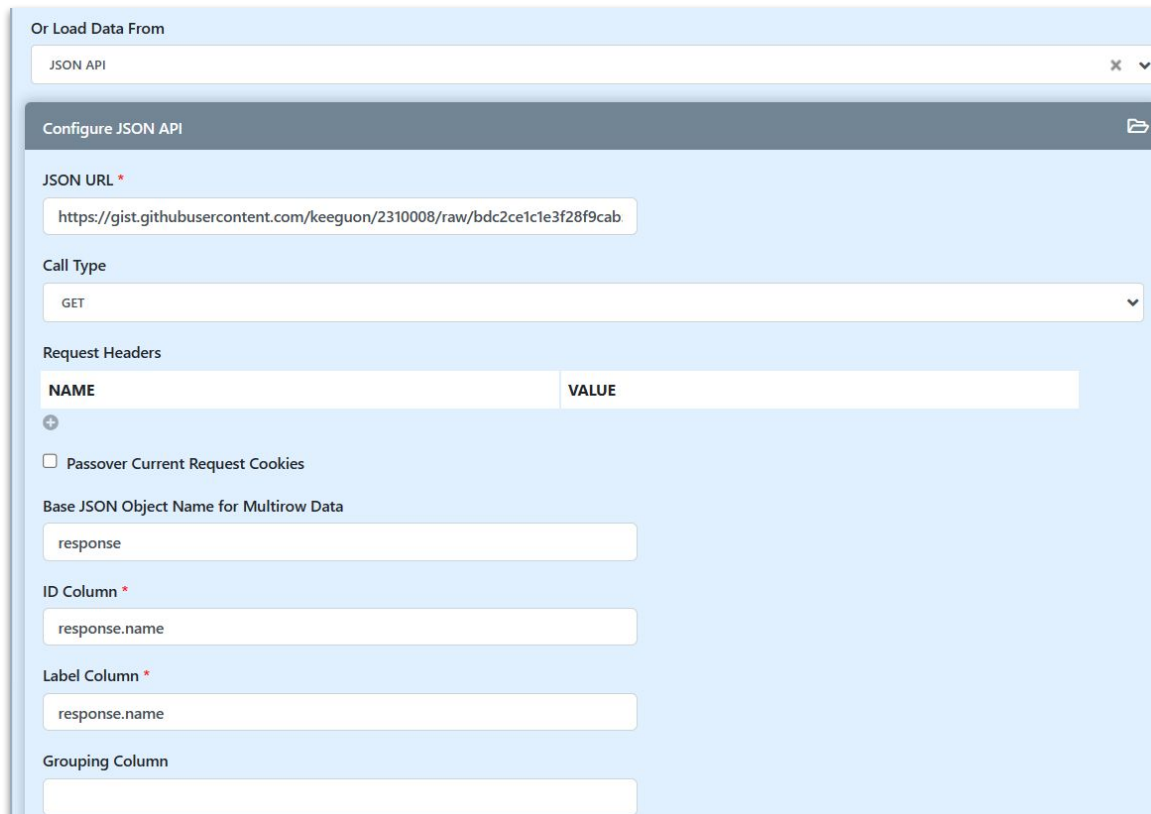
Exercise 1

- Try this JSON API endpoint:
<https://dummyjson.com/products/1>
- Create a new Joget App with a **Tool** → **Activity** process.
- Activity mapped to a form with following fields:
 - Title
 - Description
 - Price
 - Brand (from workflow variable)
- Map the Tool to a **JSON API**, configure accordingly.

Reference: <https://dummyjson.com/docs/products>

JSON API for Loading/Selecting Options

- The JSON API Form Options Data Store will call the JSON API endpoint, along with any required parameters.



The screenshot shows the 'Configure JSON API' form in Joget. The form is titled 'Or Load Data From' and has a dropdown menu set to 'JSON API'. Below this is a section titled 'Configure JSON API' with a folder icon. The form contains the following fields:

- JSON URL ***: A text input field containing the URL `https://gist.githubusercontent.com/keeguon/2310008/raw/bdc2ce1c1e3f28f9cab`.
- Call Type**: A dropdown menu set to 'GET'.
- Request Headers**: A table with two columns, 'NAME' and 'VALUE'. There is a '+' icon to add new headers.
- ☐ **Passover Current Request Cookies**: A checkbox.
- Base JSON Object Name for Multirow Data**: A text input field containing 'response'.
- ID Column ***: A text input field containing 'response.name'.
- Label Column ***: A text input field containing 'response.name'.
- Grouping Column**: A text input field.

JSON API Form Options Data Store

– How it works?

- Data will then be retrieved and displayed on the form element (Select Box / Check Box /Radio).

Exercise 2

Select Box Example

Argentina

Note: Depending on your internet speed, this JSON Form Option will take a few seconds to populate from <https://gist.githubusercontent.com/keeguon/2310008/raw/bdc2ce1c1e3f28f9cab5b4393c7549f38361be4e/countries.json>.

Afghanistan
Åland Islands
Albania
Algeria
American Samoa
Andorra
Angola
Anguilla
Antarctica
Antigua and Barbuda
Argentina
Armenia
Aruba
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados

Exercise 2

- Try this JSON API endpoint:
<https://gist.githubusercontent.com/keeguon/2310008/raw/bdc2ce1cle3f28f9cab5b4393c7549f38361be4e/countries.json>
- Create a new Form with a **Select Box**.
- Configure the select box with the **JSON API** Form options data store accordingly.

JSON API for Lists

- Using JSON API as a List Data Store enables one to load data into list retrieved via a web service call. The JSON List Data Store will call the JSON API endpoint, along with any required parameters.

Configure JSON API

Select Data Store > **Configure JSON API**

JSON URL *

https://gist.githubusercontent.com/keeguon/2310008/raw/bdc2ce1c1e3f28f9cab5b4393c7549f38361be4e/countries.json

Call Type

GET

Request Headers

NAME	VALUE
+	

Passover Current Request Cookies

☒

Base JSON Object Name for Multirow Data

response

Total Record Count JSON Object Name

Primary Key *

response.name

Auto Handling Full Results Paging & Sorting

☒

Debug Mode ?

☐

Sample Response (For Retrieve Columns)

1

JSON API List Data Store – How it works?

- Data will be then stored into the List.

<input type="checkbox"/>	name	code
<input type="checkbox"/>	Afghanistan	AF
<input type="checkbox"/>	Åland Islands	AX
<input type="checkbox"/>	Albania	AL
<input type="checkbox"/>	Algeria	DZ
<input type="checkbox"/>	American Samoa	AS
<input type="checkbox"/>	Andorra	AD
<input type="checkbox"/>	Angola	AO
<input type="checkbox"/>	Anguilla	AI
<input type="checkbox"/>	Antarctica	AQ
<input type="checkbox"/>	Antigua and Barbuda	AG

« < 1 2 3 4 5 6 7 8 > »

Exercise 3

- Try this JSON API endpoint:
<https://gist.githubusercontent.com/keeguon/2310008/raw/bdc2ce1c1e3f28f9cab5b4393c7549f38361be4e/countries.json>
- Create a new List.
- Configure the List with the **JSON API** List data store accordingly.

JSON API Form Data Store

- Using JSON API as a Form Data Store enables one to save data via a web service call or load data unto the form via a web service call. The JSON Form Data Store will call the JSON API endpoint, along with any required parameters.

Configure JSON API

Configure Form > Advanced > **Configure JSON API**

JSON URL *

Call Type

Request Headers

NAME	VALUE
+	

Passover Current Request Cookies
☐

Base JSON Object Name for Multirow Data

Primary Key *

Manual Field Mapping
☒

Field Mapping *

FIELD NAME		JSON OBJECT NAME	
<input type="text" value="name"/>	<input type="text" value="x"/>	<input type="text" value="title"/>	<input type="text" value="x"/>
<input type="text" value="price"/>	<input type="text" value="x"/>	<input type="text" value="price"/>	<input type="text" value="x"/>
<input type="text" value="desc"/>	<input type="text" value="x"/>	<input type="text" value="description"/>	<input type="text" value="x"/>
<input type="text" value="category"/>	<input type="text" value="x"/>	<input type="text" value="category"/>	<input type="text" value="x"/>
+			

Debug Mode

☒

JSON API Form Data Store – How it works?

- Data will be retrieved into the Form.

Title	iPhone 9
Description	An apple mobile which is nothing like apple
Price	549
Category	smartphones

Save

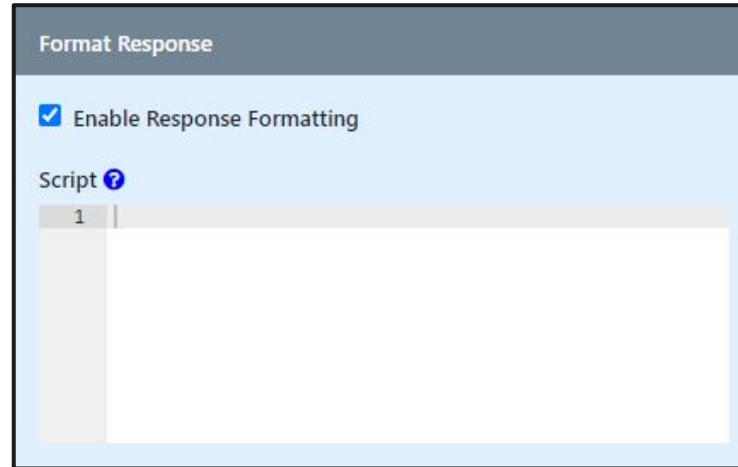
Exercise 4

- Try this JSON API endpoint:
<https://dummyjson.com/products/1>
- Create a new Form with **JSON API Form Data Store**(Load Data from).
 - Title
 - Description
 - Price
 - Brand (from workflow variable)
- Configure the Form Data Store accordingly.

Reference: <https://dummyjson.com/docs/products>

Enhanced JSON Tool

The Enhanced JSON API tool provides further configuration options for JSON response post-processing.



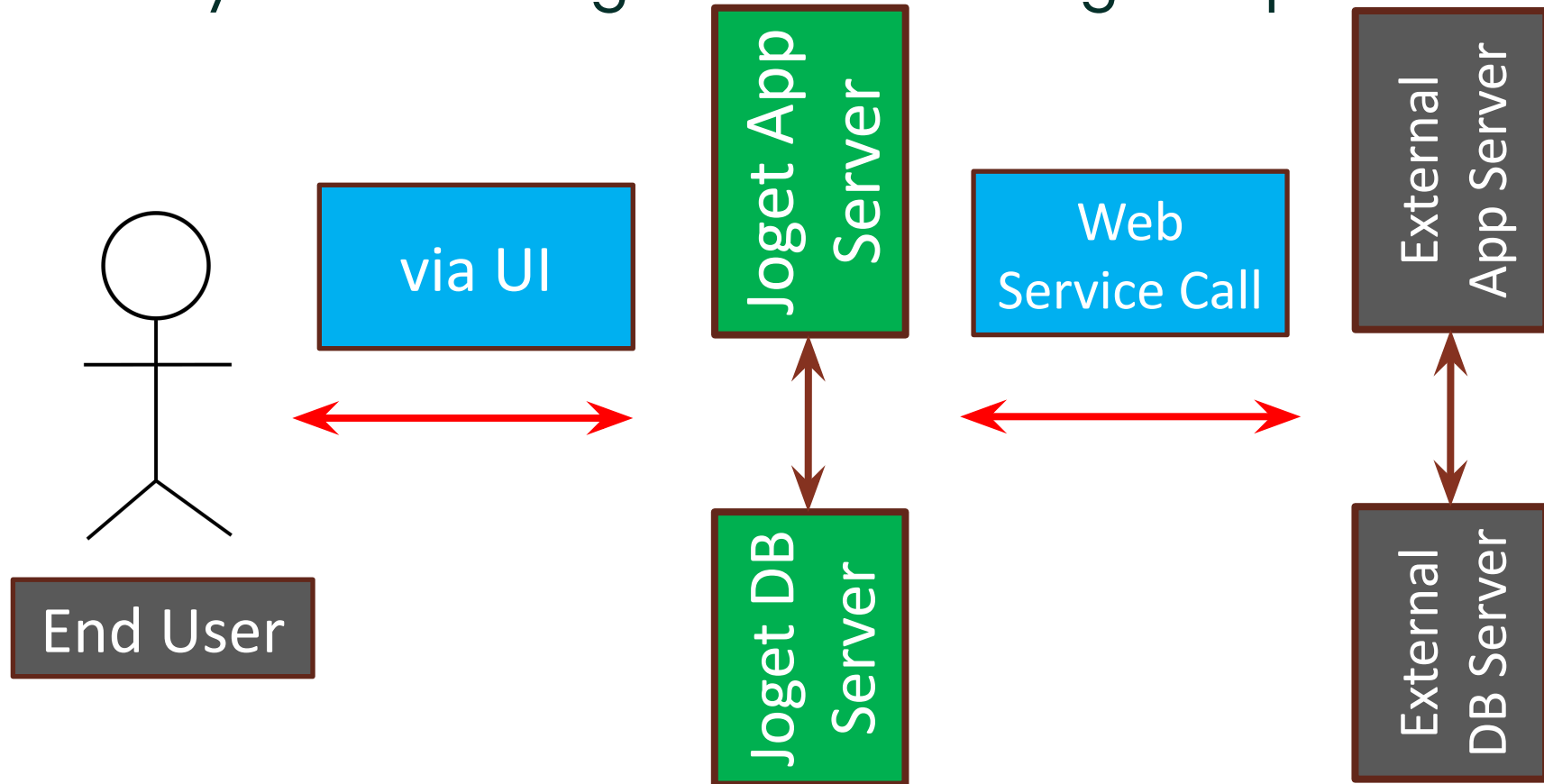
The screenshot shows a configuration window titled "Format Response". It contains a checkbox labeled "Enable Response Formatting" which is checked. Below this is a section labeled "Script" with a help icon. Under the "Script" label, there is a text area with a line number "1" and a vertical cursor, indicating a script editor.

The Enhanced JSON tool is available for download at the [JogetOSS GitHub](https://github.com/JogetOSS/JogetOSS).

Reference: <https://dev.joget.org/community/display/marketplace/Enhanced+JSON+Tool+Plugin>

Setup Suggestion

- Can you now imagine the following setup?



Just In Case...

NOTE:

You can download this chapter's completed
Joget App. **18.5.jwa**

Chapter Review

- Be able to make calls from Joget to other JSON web services by using the **JSON Tool**.



Chapter 6

JSON API

JSON API

- Joget provides a comprehensive list of APIs for Process related tasks.
- Full list of the APIs can be obtained from <https://dev.joget.org/community/display/DX8/JSON+API>

Sample List of APIs

- [web/json/monitoring/running/process/list](#)
- [web/json/workflow/currentUsername](#)
- [web/json/workflow/assignment/completeWithVariable/\(:activityId\)](#)
- [web/json/workflow/assignment/complete/\(:activityId\)](#)

web/json/monitoring/running/process/list

- **URL:** /web/json/monitoring/running/process/list
- **Method:** HTTP GET/POST
- **Description:** Retrieve running process list
- **Parameters**
 - packageId - (Optional) package id
 - processId - (Optional) process definition id without version
 - processName - (Optional) process name
 - version - (Optional) process version
 - sort - (Optional) column name to be sorted
 - desc - (Optional) Boolean value to determine whether to sort by ascending or descending order (true equals to descending)
 - start - (Optional) where rows start from
 - rows - (Optional) number of rows per page

Sample Result

```
{
  "total":2,
  "desc":false,
  "sort":"name",
  "start":0,
  "data":
  [
    {"id":"3724_mdec_v1002_mdec_wp1",
      "serviceLevelMonitor":"<span
class=\"dot_red\"></span>",
      "name":"mdec_wp1","state":"open.running",
      "due":"Fri Mar 20 14:01:27 SGT 2009",
      "startedTime":"Fri Mar 20 13:51:27 SGT
2009","version":"2"},
    {"id":"3725_mdec_v1002_mdec_wp1",
      "serviceLevelMonitor":"<span
class=\"dot_red\"></span>",
      "name":"mdec_wp1",
      "state":"open.running",
      "due":"Fri Mar 20 14:03:16 SGT 2009",
      "startedTime":"Fri Mar 20 13:53:16 SGT
2009",
      "version":"2"}
  ]
}
```

web/json/workflow/currentUsername

Sample Result

```
{ "username": "admin" }
```

- **URL**
/web/json/workflow/currentUsername
- **Method**
HTTP GET/POST
- **Description**
Get current logged in user's username
- **Parameters**
 - callback – (Optional) a function (in JavaScript) to call back after invoking this method

/web/json/workflow/assignment/completeWithVariable/(:activityId)

- **URL**
/web/json/workflow/assignment/completeWithVariable/(:activityId)
- **Method** HTTP POST
- **Description**
Set activity variable
Variables can be passed as parameters with the **var_** prefix
- **Parameters**
 - callback – (Optional) a function (in JavaScript) to call back after invoking this method
 - activityId – activity id
 - var_(workflow variable id) – (Optional) set workflow variable value

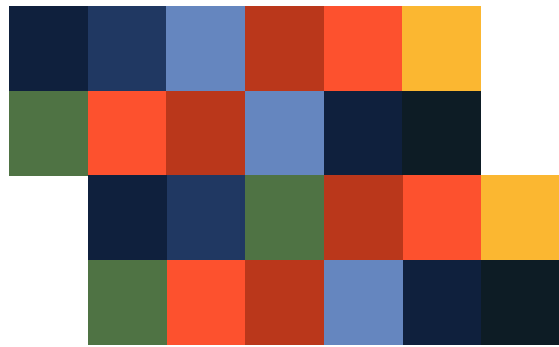
/web/json/workflow/assignment/completeWithVariable/(:activityId)

Sample Result

```
{  
  "activityId": "1079_563_crm_process1_approve_proposal",  
  "assignment": "admin",  
  "nextActivityId": "1093_563_crm_process1_send_proposal",  
  "processId": "563_crm_process1",  
  "status": "completed"  
}
```

Chapter Review

- Understand and able to retrieve the list of available APIs.



Chapter 7

JSON API

Authentication

Using JSON API

- Can be called using AJAX call (front-end/web) or through server-backend to retrieve data from or to perform action to Joget System.
- Return response in JSON format.
- Example:

web/json/workflow/assignment/list/count

{"total":11}

Passing Parameters

{Host}/jw/web/json/workflow/assignment/list/count

This JSON API is accessed as
anonymous

{Host}/jw/web/json/workflow/assignment/list/count?**j_username=admin**
&**j_password=admin**

{Host}/jw/web/json/workflow/assignment/list/count?**j_username=admin**
&**hash=14ACD782DCFEB2BCDE2B271CCD559477**

By using **j_username** and **j_password** or **hash** parameter.

This JSON API is accessed as **admin**

Password Hashing

- Used in JSON API authentication and JavaScript Single Sign ON (SSO).
- Prevents a user's password from being directly exposed during authentication.
- **Formula**
 - `md5(username + "::" + md5Base16(password))`
 - E.g. Assuming that the username is "admin" and the password is "admin", the resulting hash should be **"14ACD782DCFEB2BCDE2B271CCD559477"**.
- Online reference:
<http://dev.joget.org/community/display/DX8/Hashed+Password>

Password Hashing Sample Java Code

```
public static String md5(String content) {
    try {
        MessageDigest m = MessageDigest.getInstance("MD5");
        byte[] data = content.getBytes();
        m.update(data, 0, data.length);
        BigInteger i = new BigInteger(1, m.digest());
        return String.format("%1$032X", i);
    } catch (Exception ex) {}
    return "";
}

public static String md5Base16(String content) {
    try {
        MessageDigest md = MessageDigest.getInstance("MD5");
        byte[] bytes = md.digest(content.getBytes());
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < bytes.length; i++) {
            byte b = bytes[i];
            String hex = Integer.toHexString((int) 0x00FF & b);
            if (hex.length() == 1) {
                sb.append("0");
            }
            sb.append(hex);
        }
        return sb.toString();
    } catch (Exception e) {}
    return "";
}
```


Master Login

- Master Login Username and Password is a set of credential that can be used to login to Joget system as different user.
- This is particularly catered to integration needs:-
 - Server-backend to Joget server where individual's credential is not required to perform calls on behalf.
- Reference:
<http://dev.joget.org/community/display/DX8/JSON+API+Authentication>

JSON API Authentication – Master Login

System Administration Settings

Master Login Username

master

Master Login Password

.....

master login hash:0b244463ce522dc21bc58b42d13abcf0

In System Settings >
General Setting

{Host}/jw/web/json/workflow/assignment/list/count?j_username={master-login-username}&j_password={master-login-password}&loginAs={username}

{Host}/jw/web/json/workflow/assignment/list/count?j_username={master-login-username}&hash={master-login-hash}&loginAs={username}

By using **Master Login** feature, user's password is not exposed in JSON API authentication.

Basic Http Authentication

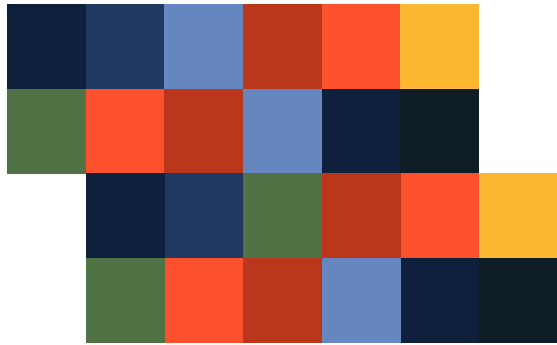
- A set of user credential encoded to login to Joget system.
- Credentials are passed in the request header.
- Username and password not exposed in clear text
- **Formula:**
 - "Basic" + base64(username + ":" + password)
 - E.g. Assuming that the username is "admin" and the password is "admin", the result should be "**Basic YWRtaW46YWRtaW4=**".
- Reference:
<https://dev.joget.org/community/display/DX8/JSON+API+Authentication#JSONAPIAuthentication-BasicHttpAuthentication>

Caution

- Do **NOT** expose clear text password in the DOM
- Ensure that "API Domain Whitelist" & "API IP Whitelist" is configured according to principle of least privilege

Chapter Review

- Understand how to authenticate with Joget's JSON APIs.



Chapter 8

Using API Builder

API Builder

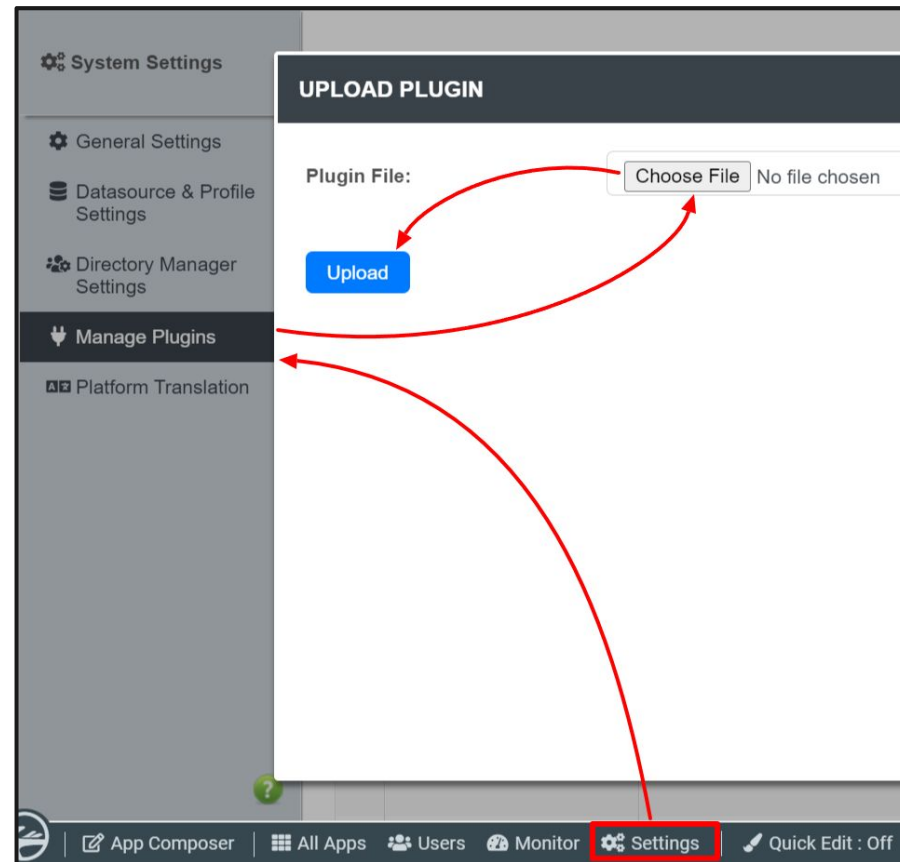
- API Builder is a new type of plugin called Custom Builder.
- Allows one to create your own customized JSON APIs for Joget Apps.

Reference:

<https://dev.joget.org/community/display/marketplace/API+Builder>

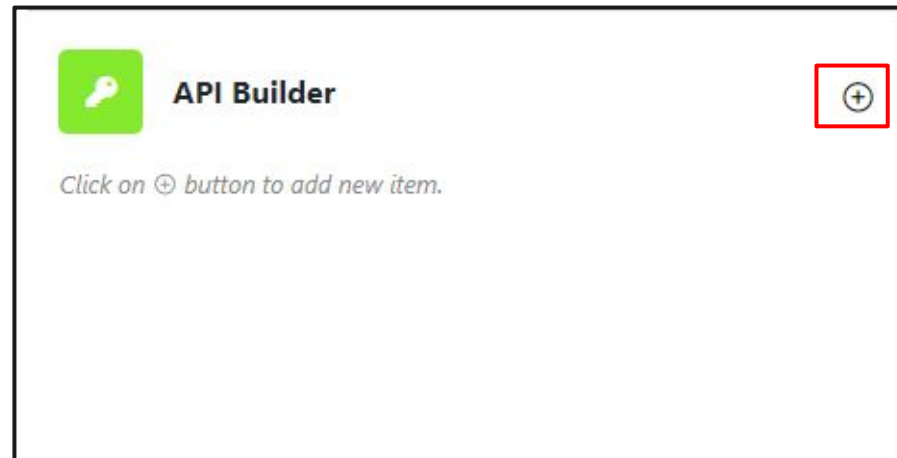
Installing API Builder

- Download and install the [plugin](#) from Joget Marketplace.



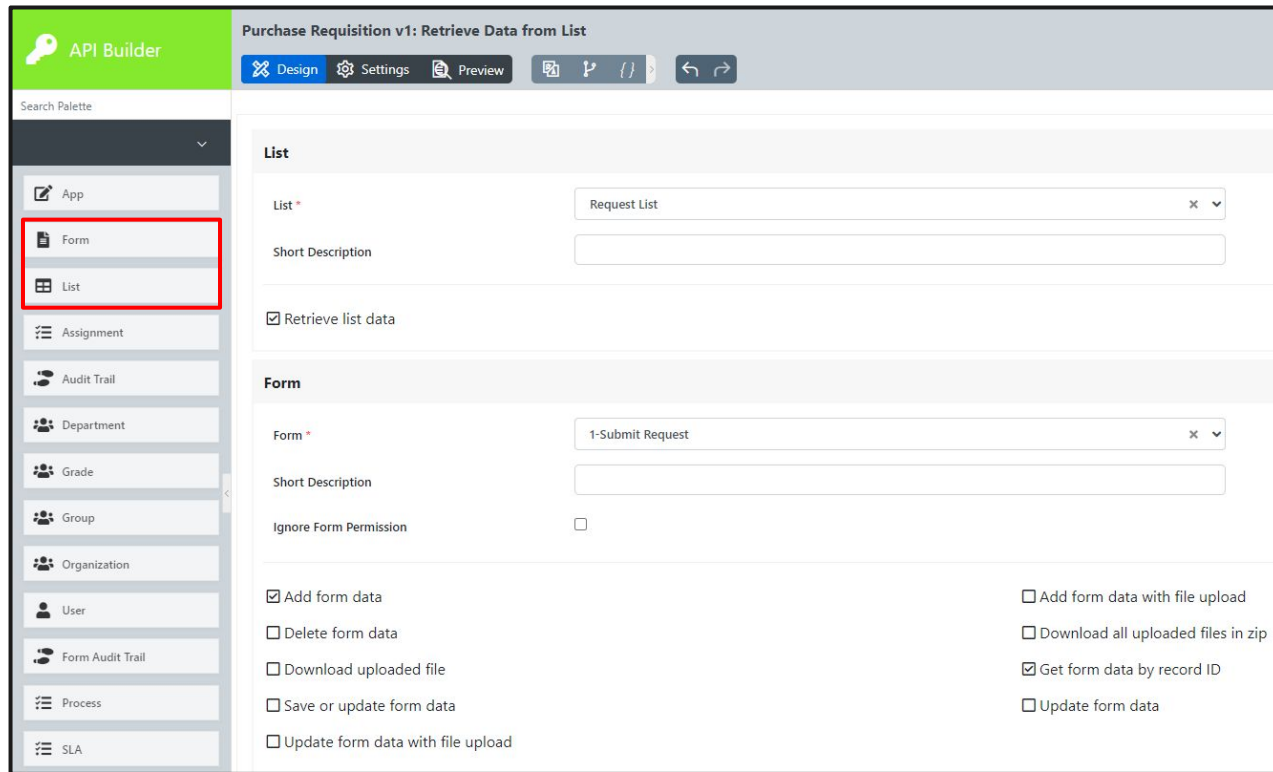
Exercise

- Continue to use the app from the previous chapters. (14.5.jwa)
- Click on the new menu Addon Builders and click on Create new API Key.



Exercise

- Try retrieving
 - all Request data from the list,
 - add form data to Submit Request and,
 - retrieve form data by record ID.



The screenshot shows the Joget API Builder interface for a workflow titled "Purchase Requisition v1: Retrieve Data from List". The interface is divided into a left sidebar and a main workspace.

Left Sidebar (Search Palette):

- App
- Form (highlighted with a red box)
- List (highlighted with a red box)
- Assignment
- Audit Trail
- Department
- Grade
- Group
- Organization
- User
- Form Audit Trail
- Process
- SLA

Main Workspace:

The workspace is titled "Purchase Requisition v1: Retrieve Data from List" and has tabs for Design, Settings, and Preview. It is divided into two main sections: "List" and "Form".

List Section:

- List ***: A dropdown menu with "Request List" selected.
- Short Description**: A text input field.
- Retrieve list data**: A checkbox that is checked.

Form Section:

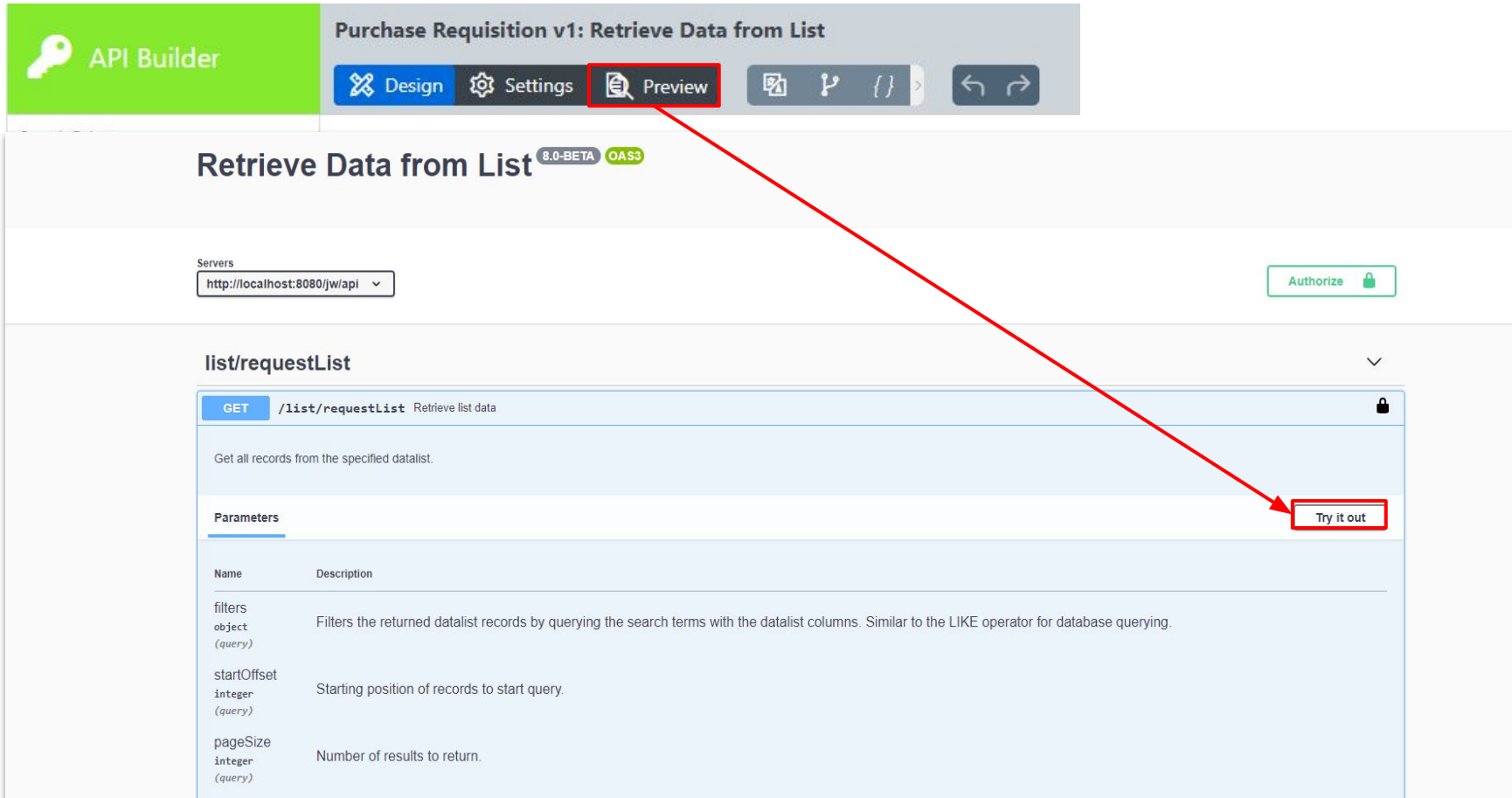
- Form ***: A dropdown menu with "1-Submit Request" selected.
- Short Description**: A text input field.
- Ignore Form Permission**: A checkbox that is unchecked.

Bottom Section:

- Add form data**: A checkbox that is checked.
- Delete form data**: A checkbox that is unchecked.
- Download uploaded file**: A checkbox that is unchecked.
- Save or update form data**: A checkbox that is unchecked.
- Update form data with file upload**: A checkbox that is unchecked.
- Add form data with file upload**: A checkbox that is unchecked.
- Download all uploaded files in zip**: A checkbox that is unchecked.
- Get form data by record ID**: A checkbox that is checked.
- Update form data**: A checkbox that is unchecked.

Generate OAS3 Document

- Click on Preview to generate an OAS3 document .
- Click on an API method and test it out.



The screenshot shows the Joget API Builder interface for the 'Purchase Requisition v1: Retrieve Data from List' API. The 'Preview' button in the top navigation bar is highlighted with a red box. A red arrow points from this button to the 'Try it out' button in the 'list/requestList' API method section, which is also highlighted with a red box.

API Builder

Purchase Requisition v1: Retrieve Data from List

Design Settings **Preview**

Retrieve Data from List 8.0-BETA OAS3

Servers
http://localhost:8080/jw/api

Authorize

list/requestList

GET /list/requestList Retrieve list data

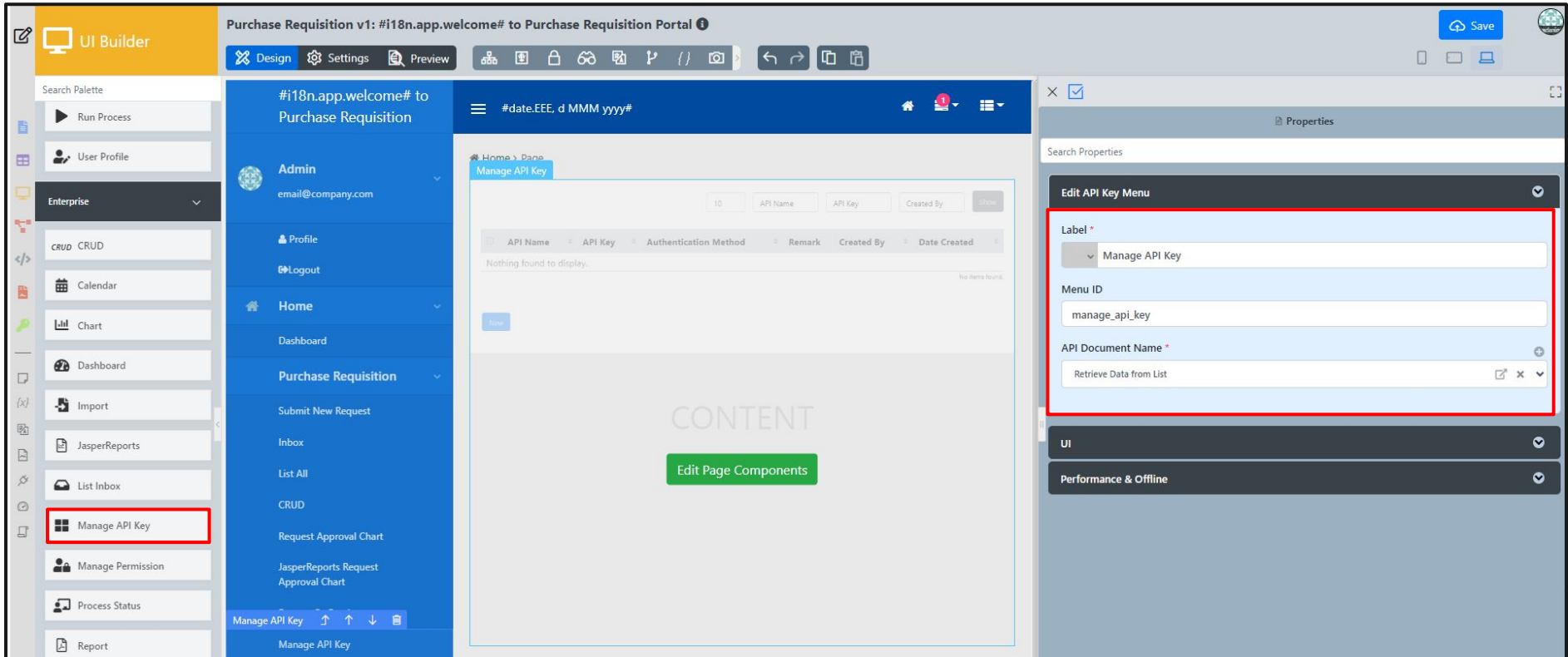
Get all records from the specified datalist.

Parameters

Name	Description
filters object (query)	Filters the returned datalist records by querying the search terms with the datalist columns. Similar to the LIKE operator for database querying.
startOffset integer (query)	Starting position of records to start query.
pageSize integer (query)	Number of results to return.

Manage API Key

- UI element to control access to API methods created with API Builder.



The screenshot displays the Joget UI Builder interface for a 'Purchase Requisition v1' application. The left sidebar shows the 'Enterprise' menu with 'Manage API Key' highlighted. The main canvas shows a 'Manage API Key' menu configuration. The right sidebar shows the 'Properties' panel with the 'Edit API Key Menu' section expanded, showing the following configuration:

- Label ***: Manage API Key
- Menu ID**: manage_api_key
- API Document Name ***: Retrieve Data from List

The main canvas also shows a table with columns: API Name, API Key, Authentication Method, Remark, Created By, and Date Created. The table is currently empty, displaying 'Nothing found to display.' and 'No items found.'

Exercise

- Create an API key.
- Try calling the API methods with and without the API key.

API Name *	Exercise API
API Key	01022cbef2824ab9bfbf465c568e998d
Authentication Method *	API Key
Domain Whitelist *	<div>*</div> <p>One domain in one line. Wildcard (*) is allowed.</p>
IP Address Whitelist *	<div>*</div> <p>One IP Address in one line. Wildcard (*) is allowed.</p>
Remark	API Key Test
Log request query string, headers, payload & response	<input type="checkbox"/>

Manage API Key Log

- There's also a log with details on each attempt.

Admin Admin

Dashboard

Purchase Requisition

Submit New Request

Inbox

List All

CRUD

Request Approval Chart

JasperReports Request Approval Chart

Reports By Barchart

Manage API Key

10

Show

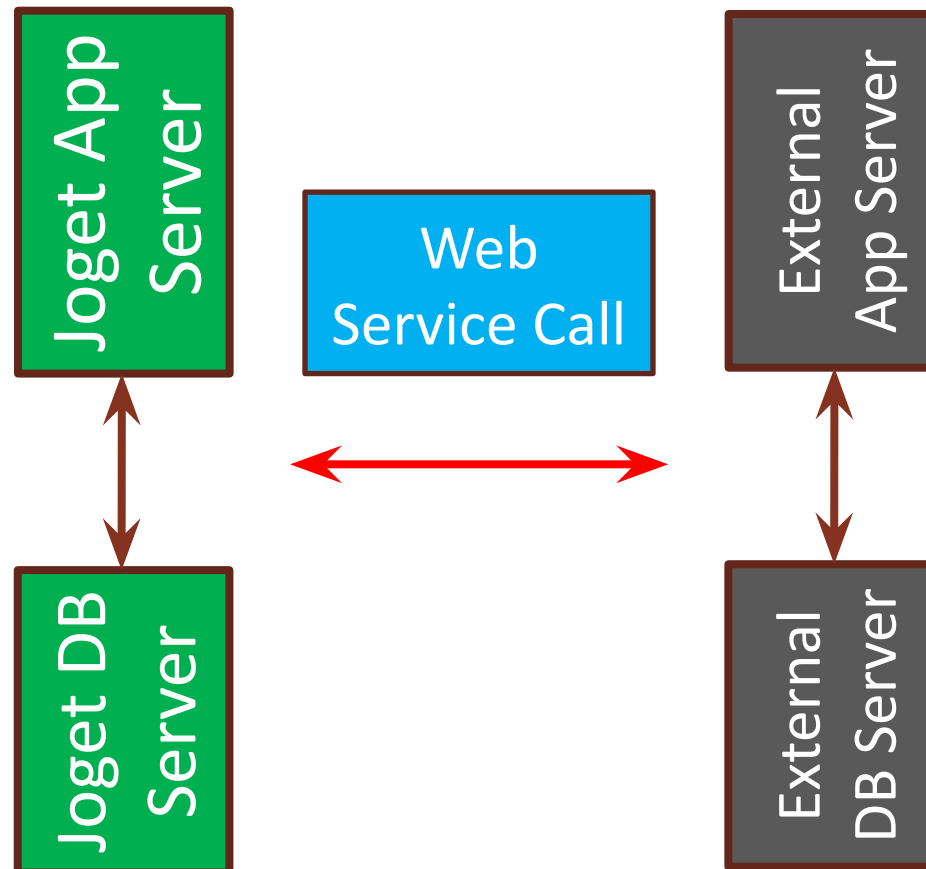
Date	API ID	API Key	Status	Method	User Agent	Source IP	Message	Execute Time (ms)
16-11-2022 01:39 PM	API-6bbd95bb-4f67-4c8a-ab78-f2795b7aa4be	2c822f1b016a4dcea700d5577f7c0caa	200	list/requestList	PostmanRuntime/7.29.2	0:0:0:0:0:0:1		4
16-11-2022 01:39 PM	API-6bbd95bb-4f67-4c8a-ab78-f2795b7aa4be	2c822f1b016a4dcea700d5577f7c0caa	200	list/requestList	PostmanRuntime/7.29.2	0:0:0:0:0:0:1		5
16-11-2022 01:38 PM	API-6bbd95bb-4f67-4c8a-ab78-f2795b7aa4be	2c822f1b016a4dcea700d5577f7c0caa	200	list/requestList	PostmanRuntime/7.29.2	0:0:0:0:0:0:1		5
16-11-2022 01:38 PM	API-6bbd95bb-4f67-4c8a-ab78-f2795b7aa4be	2c822f1b016a4dcea700d5577f7c0caa	200	list/requestList	PostmanRuntime/7.29.2	0:0:0:0:0:0:1		6
16-11-2022 01:38 PM	API-6bbd95bb-4f67-4c8a-ab78-f2795b7aa4be	2c822f1b016a4dcea700d5577f7c0caa	200	list/requestList	PostmanRuntime/7.29.2	0:0:0:0:0:0:1		781

5 items found, displaying all items.

CSV | Excel | XML | PDF

Setup Suggestion

- Can you now imagine the following setup?



Chapter Review

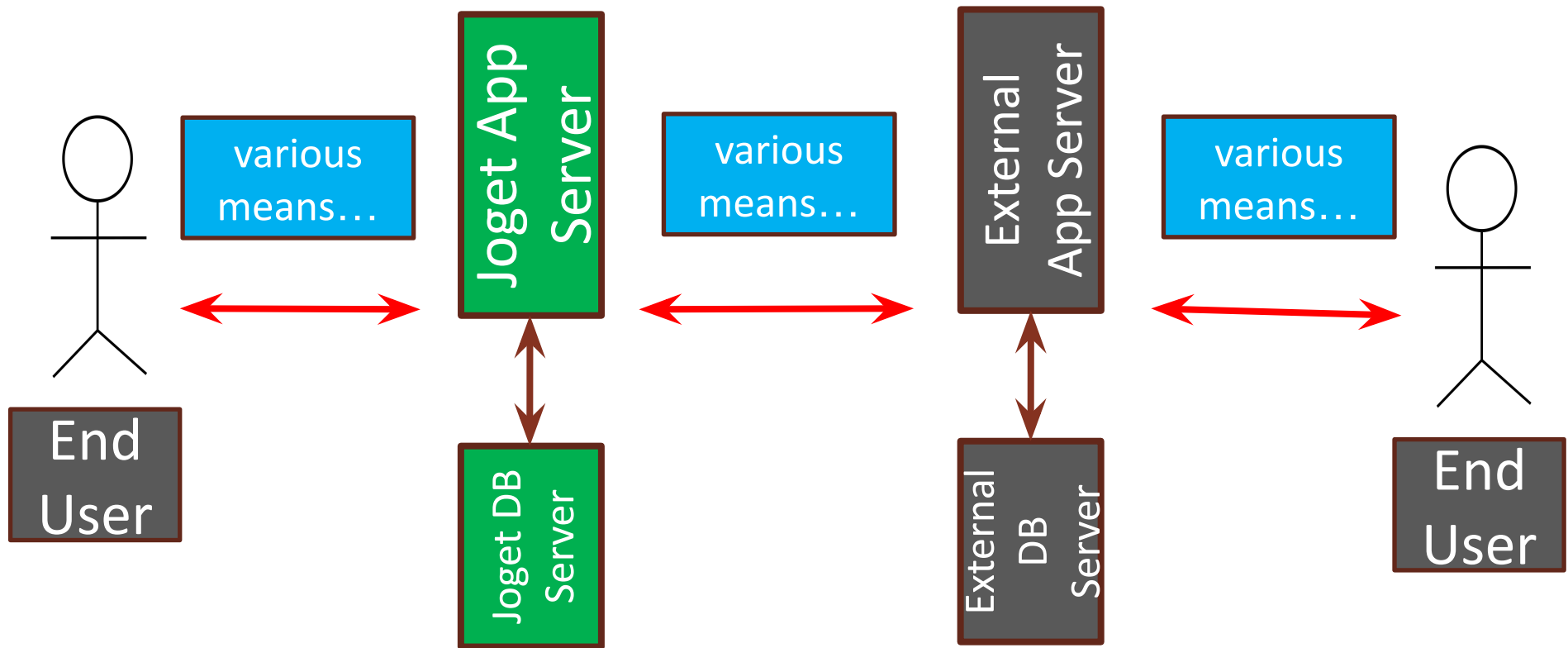
- Understand and being able to use API Builder effectively.

Module Review

1. Introduction
2. Single Sign On (SSO)
3. Embedding UI Page in an iFrame
4. SOAP Tool
5. JSON Tool
6. JSON API
7. JSON API Authentication
8. Using API Builder

Module Review

- Can you now picture various setup scenarios to integrate Joget with your external app?



Stay Connected with Joget

- <http://www.joget.org>
- <http://community.joget.org>
- <http://twitter.com/jogetworkflow>
- <http://facebook.com/jogetworkflow>
- <http://youtube.com/jogetworkflow>
- <http://slideshare.net/joget>