|  |  |  |
| --- | --- | --- |
| Liferay From Builder |  | **Liferay DXP** |

**Version:** 1.0

**Issue date: 18-02-2019**

**Author:** TechMahindra

**Document Version Control**

| Version | Date | Section | Nature of Amendment | Amendment Author |
| --- | --- | --- | --- | --- |
| 1.0 | 18/02/19 | All | Draft | Shubhalaxmi.S |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

of Contents

[Version: 1.0 1](#_Toc943309)

[**1.** **Introduction ………………………………………………………………………………….**](#_Toc943310)4

[**2.** **Form Creation**](#_Toc943311) 5

[**3.** **Form Properties**](#_Toc943312) 6

[**4.** **Form Element Set**](#_Toc943313) 7

[**5.** **Data Provider**](#_Toc943314) 9

**6. Form Rule…………………………………………………………………………………...**10

[**7.** **Storage Adapter**](#_Toc943315) 13

**8. Kaleo Work Flow…………………………………………………………………………...14**

**9. Auto Save……………………………………………………………………………………15**

## 

## **Introduction: Liferay Form**

Liferay built-in Forms application supports your most basic form building and the most advanced feature to build in Form.

**The Following feature are built-In Liferay Form**

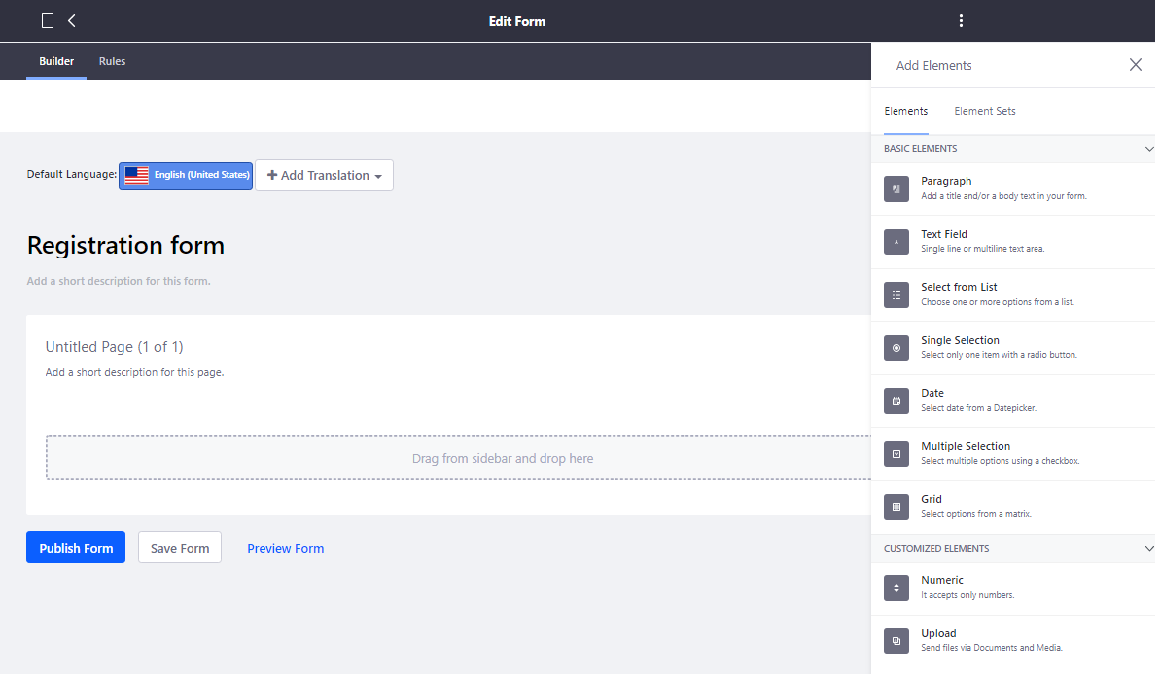
* Navigating to the Forms application
* Adding form fields, and supported field types. Configuring form fields
* Viewing and Exporting form entries , Deleting form entries
* Making a field appear based on the value entered into another field
* Configuring form permissions
* Providing a predefined value or a placeholder value

**Advanced features of the Liferay Form**

* Populating a Select field with a REST Data Provider
* Add extra pages to the form
* Validating fields using a number of different criteria
* Laying out the form
* Enabling CAPTCHA for a form
* Redirecting to a different URL after a successful form submission
* Sending an email notification whenever a form entry is submitted

**Liferay Form Builder: From Creation**

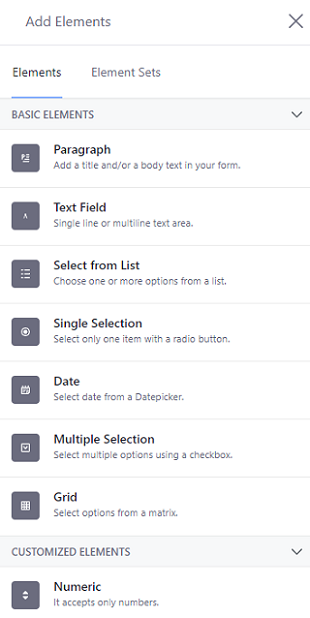
1. Create Form: Navigate to New Form Navigate to Menu > Content > Forms.
2. Click on  (New Form) to add the field to form and name the Form



**Liferay Form Builder: Form Properties**

Drag and drop the elements required to create the Form. Select the elements from Elements Tab.

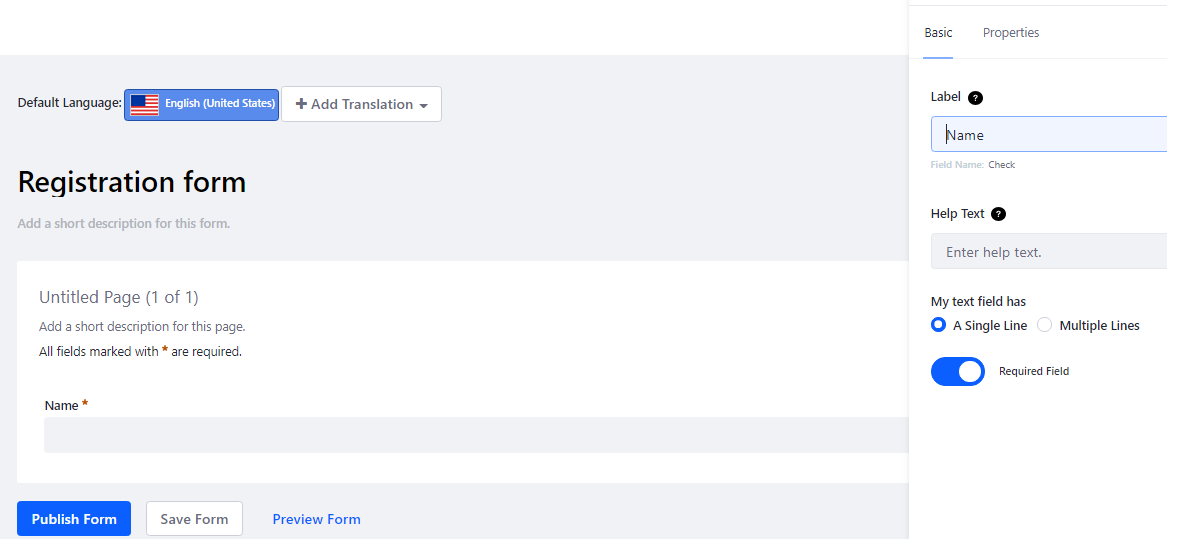
The Elements Type are:



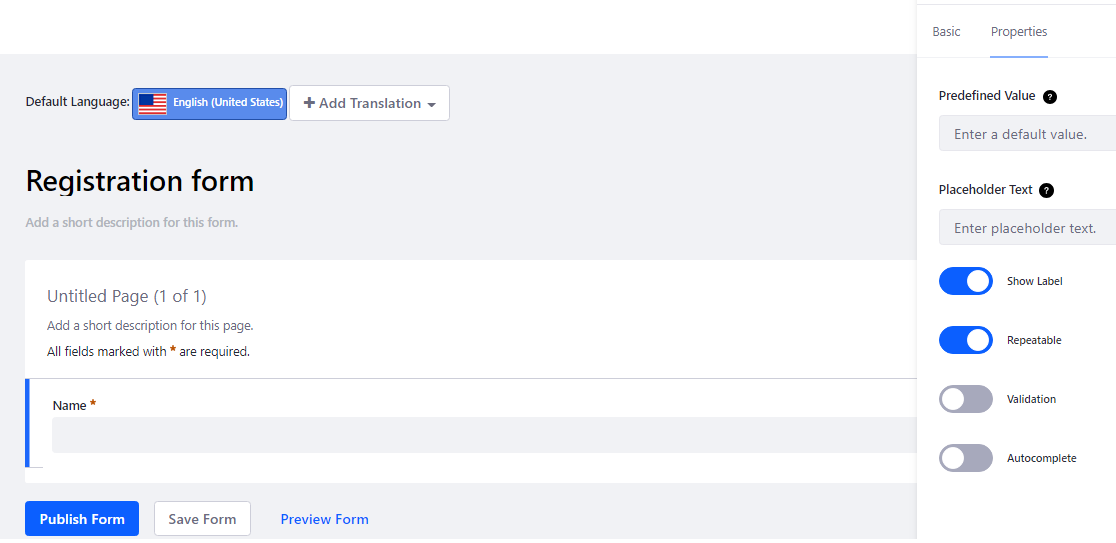
Each field consist of the configuration attributes properties and basic.

* **Label**: Field Label.
* **Help Text**: Help text to explain user, what this field about.
* **Single Line / Multiple Line:** Define if field value can contain single line or multiple line.
* **Required:** Mandatory fields.
* **Predefined value:** Default value of field.
* **Placeholder text:** Text to assist user, not submitted actually.
* **Field Visibility Expression:** Condition to display field.
* **Enable Validation:** Extra validation like contains/does not contain URL, email etc. for text and less than, greater than , equals etc for numbers . Default is false.
* **Show Label :** If you don’t want to show label to end user , by default its true .
* **Repeatable :** If field is repetitive . Default is false .
* **AutoCompelete :** Enter the value manually , data provider , AutoFill .

**Basic Element configuration**



**Properties Element Configuration**

****

**Liferay Form Element Set**

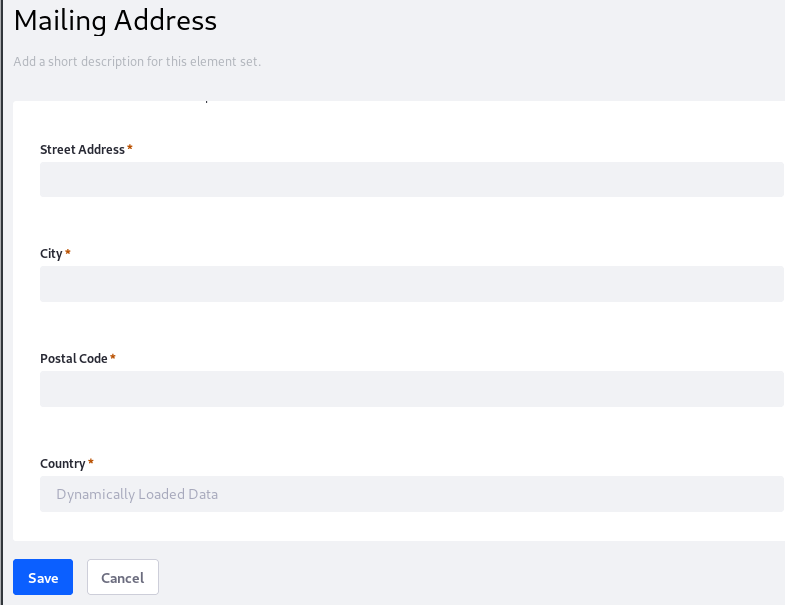
**Forms Element Set**

Element sets are more like composable Form fragments or reusable Form blocks.

**Create Element Set**

Navigate to Site Menu → Content → Forms.

Click the Element Sets tab, and any existing Element Sets appear, just like existing Forms are displayed in the Forms view. Click the Add button (icon-add.png). Save the Element set. So, It is available for the reuse in Form.

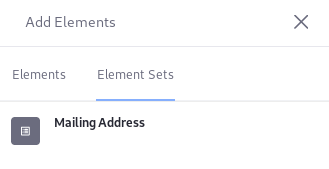


**Use Element Set**

1. Drag the Element Set onto the Form Builder, just like you would any single Form Element.

Note :

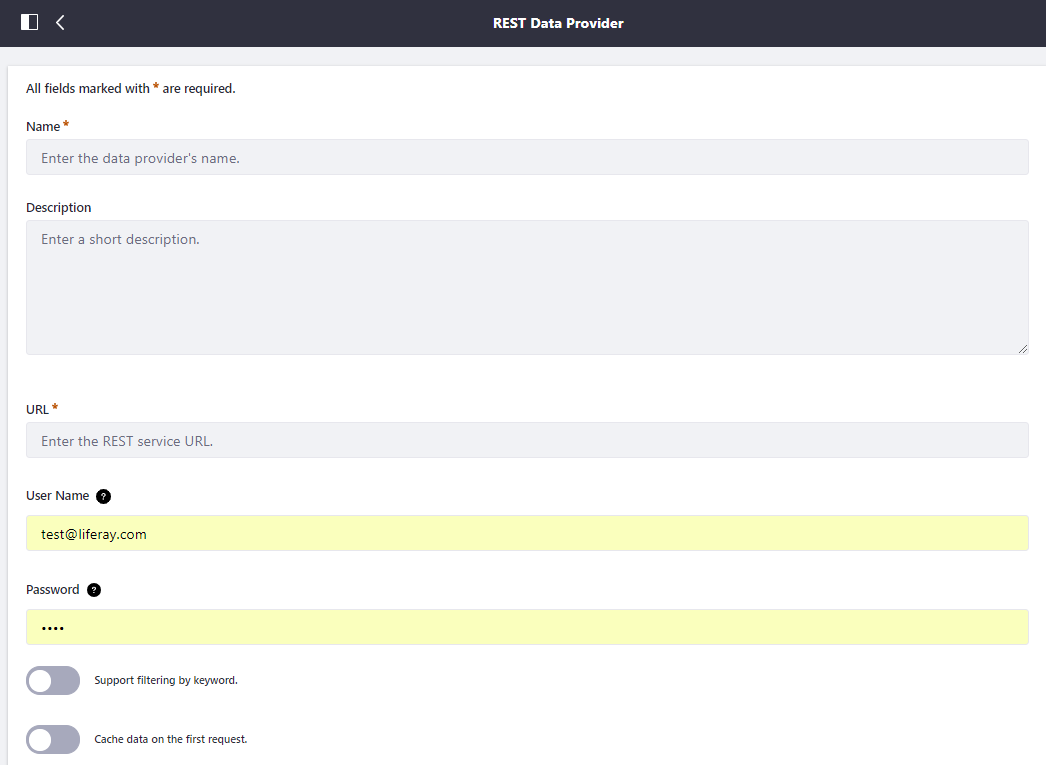
1. Element Set is added to a form , there’s no connection with the root Element Set . You’re free to move or configure the fields and elements



**Liferay Form : Data Provider**

Liferay Form data provider feature is to integrate the Rest API to the Form Builder .

* Navigate to Menu > Content > Forms . Click on the options button  and select **DataProvider**
* Click the Add button () . The Rest Data Provider form loads.



1. Fill out the Name and Description fields .

Name : Country

1. Enter the URL and authentication token for the REST Service Example

URL : http://localhost:8080/o/Javax-RS/user/get-data

User Name : [test@liferay.com](mailto:test@liferay.com)

Password : test

1. Output fields , specify the which field from REST SERVICE need to update the List

Label : Country Name

Path : $..nameCurrentValue

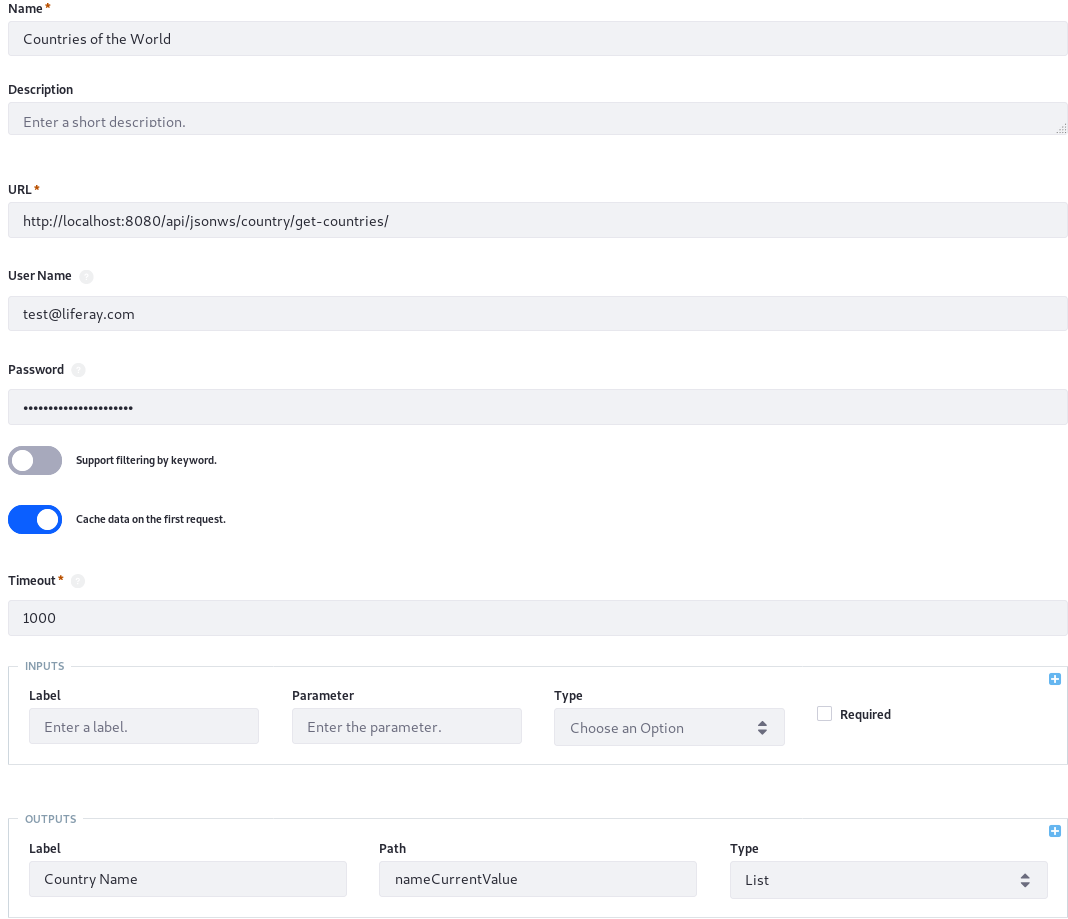
Type : List

1. Save the Data Provider

**Note :**

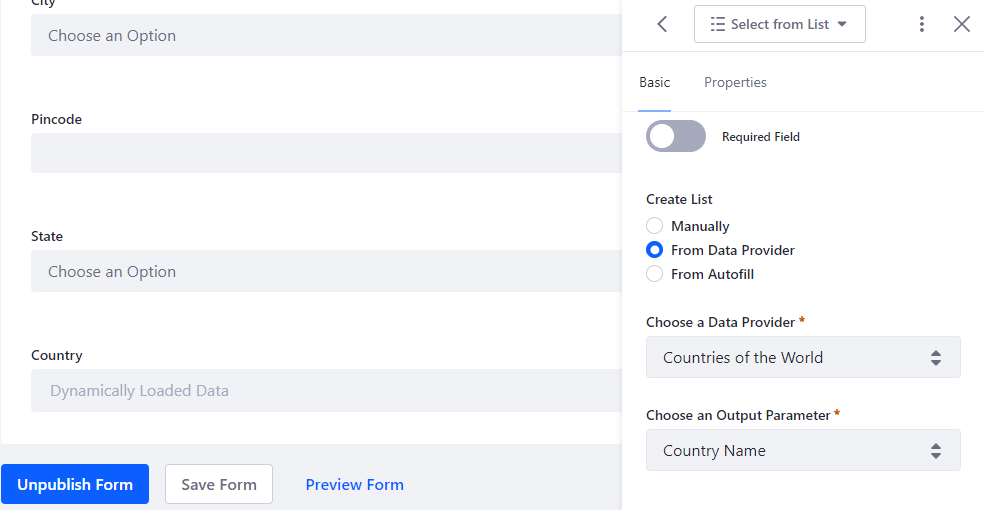
1. **Integrate the rest service increase the Timeout value**
2. **Read the Value to Output Parameter from JSON STRUCTURE $..namecurrentvalue**

[**https://github.com/json-path/JsonPath/blob/master/README.md**](https://github.com/json-path/JsonPath/blob/master/README.md)



**ADD Data-Provider to FORM**

1. Navigate to FROM . Drag and drop the Selected List to FORM .
2. Basic Properties of the Selected List . In the Create List choose Form Data- provider
3. Choose the Data provider required in the drop down
4. Choose the out put parameter required in the drop down .



**Liferay Form: Form Rule**

Form rules, you can train your form fields to behave as you wish. There are several things you can make them do:

**Show/hide :**Based on a predefined condition, set the visibility of a form field.

**Enable/disable:**Use a predefined condition to enable or disable a field.

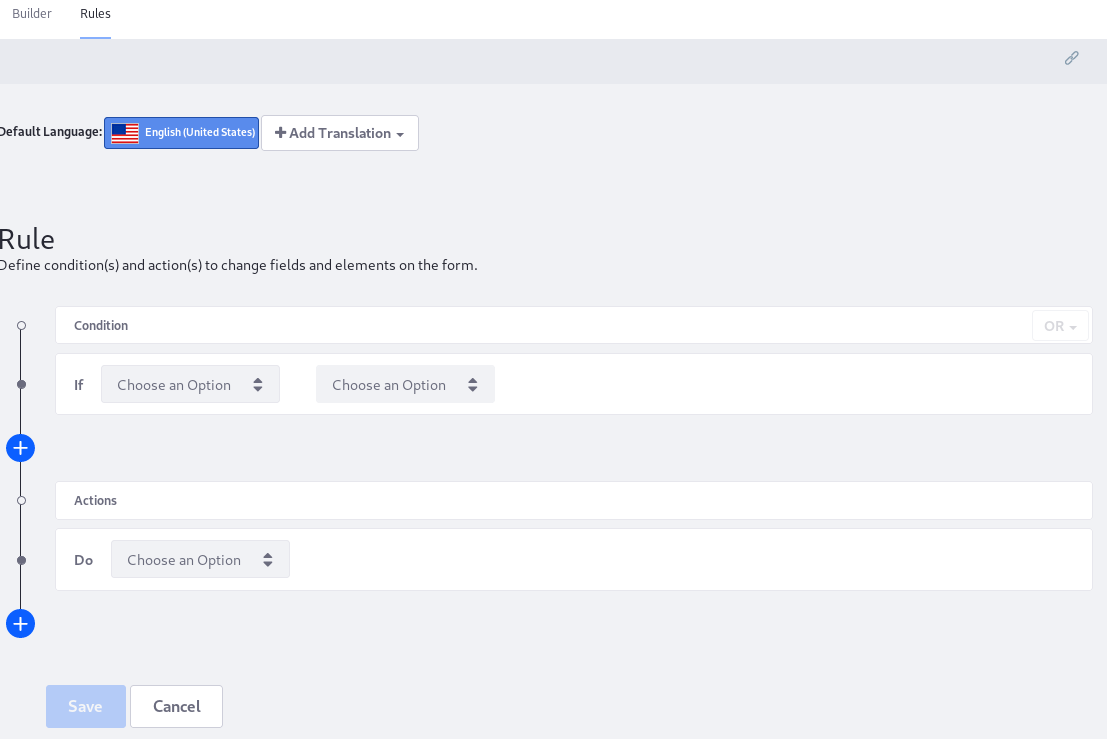
**Require:**Use a predefined condition to make a field required.

**Jump to Page:**Based on user input, skip over some form pages directly to a relevant page.

**Autofill with Data Provider** : Use a [data provider](https://dev.liferay.com/discover/portal/-/knowledge_base/7-1/data-providers) to populate fields when a condition is met in another field.

**Create the Form Rule**

* Save the form.
* Open the Rule Builder by clicking the Rules tab at the top of the Edit Form screen.
* In the rule builder view, you can now begin developing your form rule. Click the Add button (icon-add.png) to get started.



**Condition in the Form Rule**

* *Is equal to* a specific value or another field’s value.
* *Is not equal to* a specific value or another field’s value.
* *Contains* a specific value or another field’s value.
* *Does not contain* a specific value or another field’s value.
* *Is empty*. This assumes you want to do something if a field *is* empty.
* *Is not empty*. This assumes you want to do something as long as a field is *not* empty.

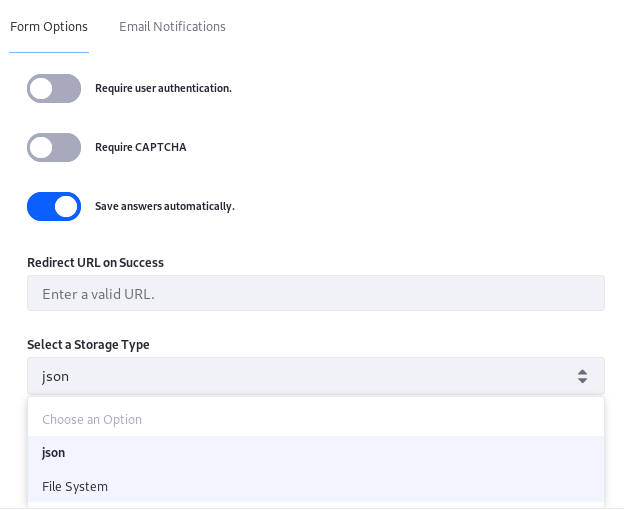
**Liferay Form: Storage Adapter**

The record is added to Form , the Form API routes the processing request through the storage adapter API . The same is true CRUD Operation . The default adapter is JSONStorageAdapter , it implements the Storage Adapter interface to provide JSON storage of form entry data.

**Enable Storage Adapter**

The storage adapter is enabled at the individual form level. Create a new form, and select the Storage Adapter before saving or publishing the form. If you wait until first Saving the Form, the default Storage Adapter is already assigned to the Form, and this setting is no longer editable.

1. Go to the Site Menu → Content → Forms, and click the Add button ().
2. In the Form Builder view, click the Options button (icon-options.png) and open the Settings window.
3. From the select list field called Select a Storage Type, choose the desired type and click Done.

****

**Implement the Storage Adapter**

First declare the class a Component that provides a StorageAdapter implementation. To implement a storage adapter, extend the abstract BaseStorageAdapter class.

@Component(service = StorageAdapter.class)

public class FileSystemStorageAdapter extends BaseStorageAdapter {

**Step1 : Name the Storage Type**

The only method without a base implementation in the abstract class is getStorageType. For the file system storage example, just make it return "Custom Table".

@Override

public String getStorageType() {

return "Custom Table";

}

Return a human readable String, as getStorageType determines what appears in the UI when the form creator is selecting a storage type for their form. The String value you return here is added to the StorageAdapterRegistry’s Map of storage adapters.

**Step2: Override the CRUD Methods**

To add the Custom table override of the CRUD method of BaseStorage .Storage Adapter has built in CRUD operation method.

The method to override is save File .

private void saveFile(

long structureVersionId, long fileId, DDMFormValues formValues)

throws IOException {

String serializedDDMFormValues = \_ddmFormValuesJSONSerializer.serialize(

formValues);

File formEntryFile = getFile(structureVersionId, fileId);

FileUtil.write(formEntryFile, serializedDDMFormValues);

}

**Integrate the Custom Table**

1)Create the Service Builder using the Liferay Module Project .

Ex : EmployeeFormService .

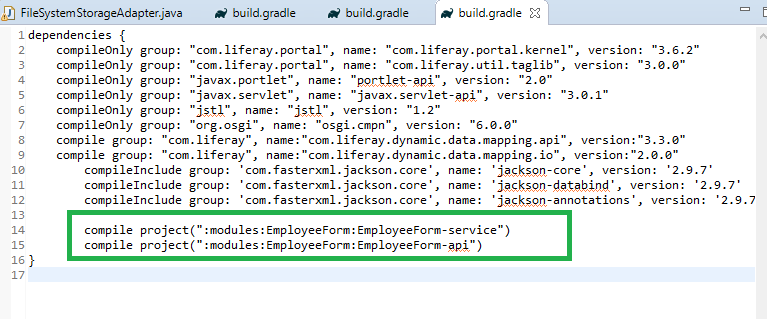
2)Edit the Service.xml. To required columns to be created in custom Table

3) We can access the API’s EmployeeForm Service in FileStorageAdapter component

To save the data in the data base .

**Note:**

**Add the dependencies of the service-builder component** EmployeeFormService **to the Storage Adapter component FileSystemStorageAdapter through build gradle.**

****

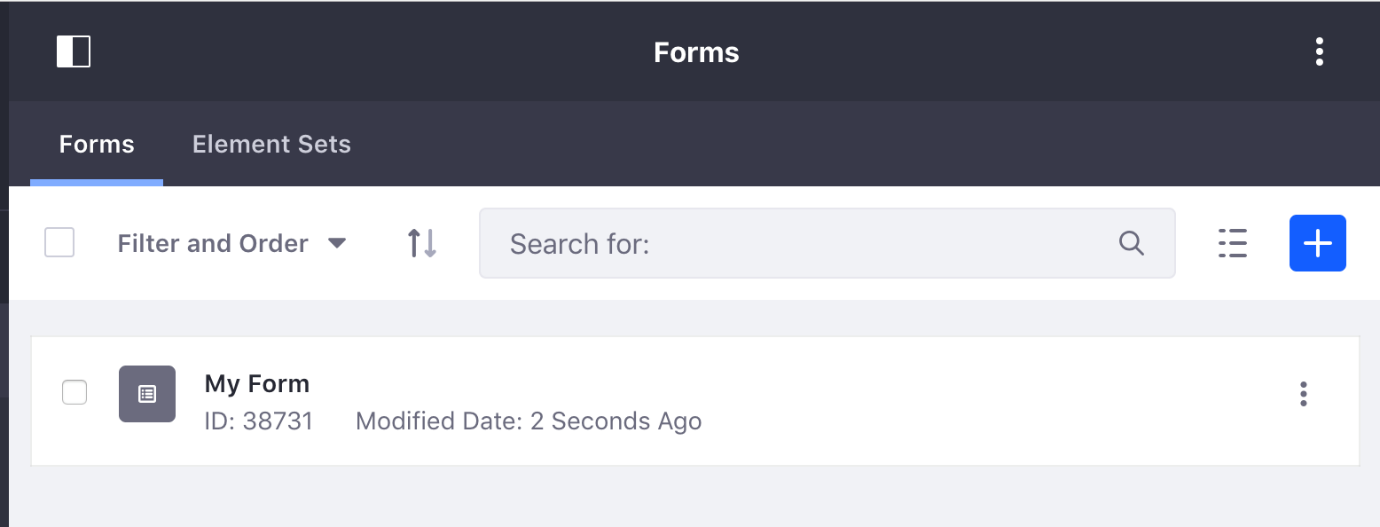
**Liferay Form : Kaleo Work Flow**

**WorkFlow and Forms**

Kaleo is a workflow engine for sending a submitted asset through a workflow process .

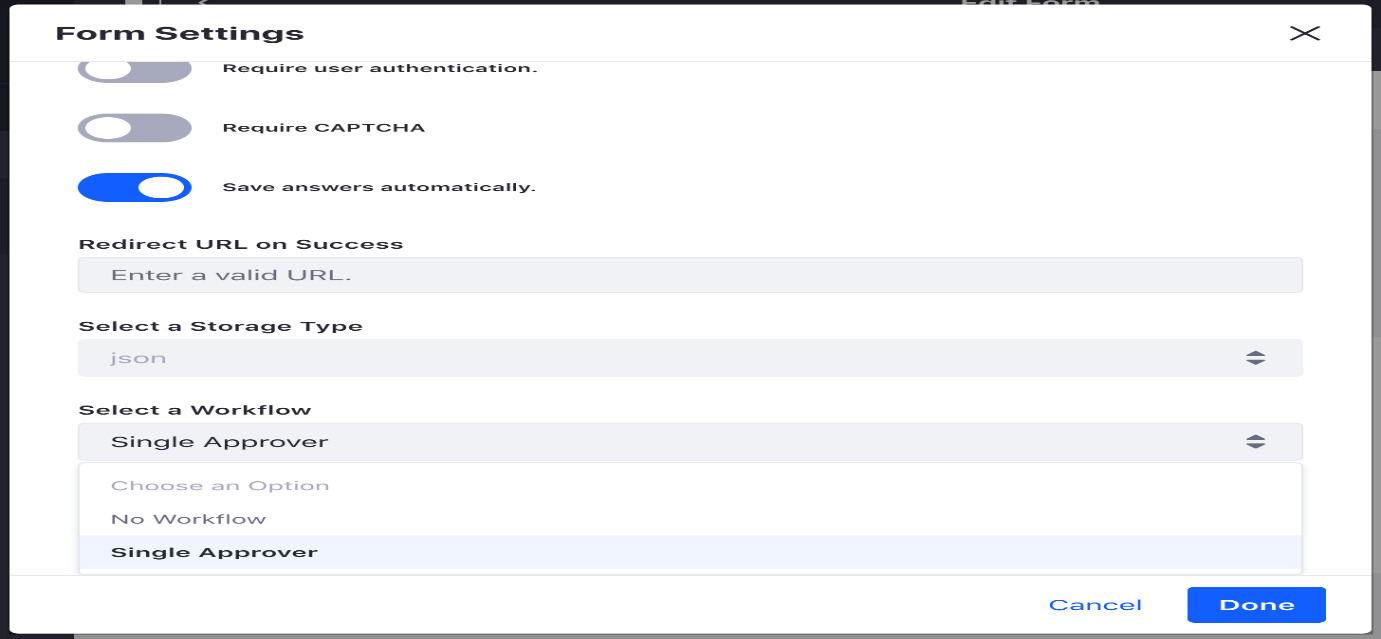
**Enabling Workflow In a Form**

1. Navigate to Content -> Forms and click on the form to open.



**Navigate Form to Enable WorkFlow**

1. Click the Options button (icon-options.png) and choose Settings.
2. The Settings window has a Select a Workflow drop-down. Find the workflow you want, select it, and then click Done.

****

**Testing WorkFlow**

Test the workflow process:

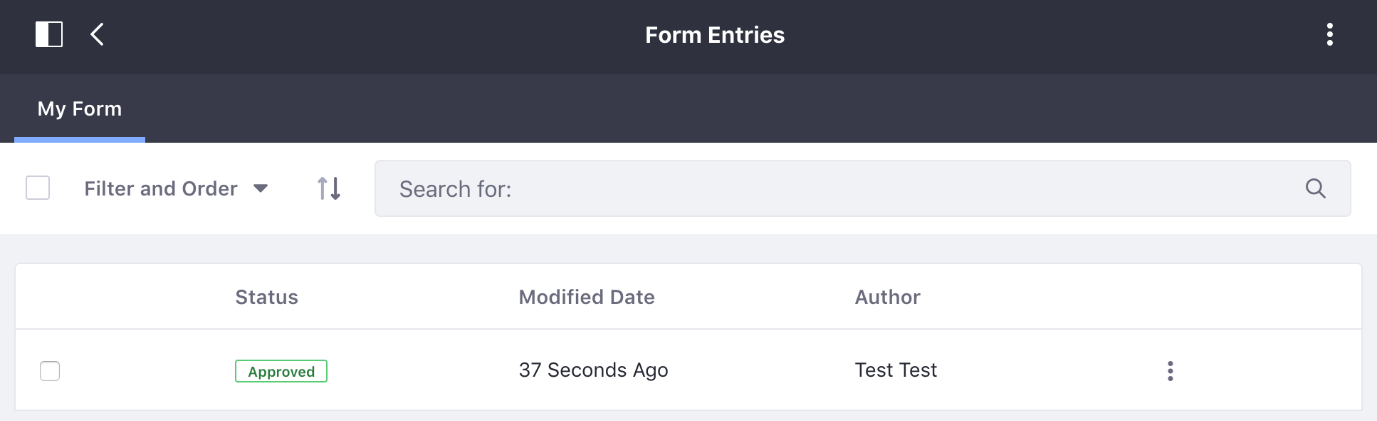
1. Add the form to a page.
2. Click Submit for Publication to submit the form entry.

Next go find the form entry in the Forms widget:

1. Go back to the Forms application in the Menu in your Site’s Content section.
2. Click the Form’s Actions button (icon-actions.png) and select View Entries.

Now approve the form record:

1. Navigate to My Account → My Workflow tasks.
2. Click the Assigned to My Roles tab.
3. Click on the form entry.
4. Click the Actions button (icon-actions.png) and choose Assign to Me.
5. Click Done.
6. Click the Actions button (icon-actions.png) again, then click Approve.
7. Click Done again.
8. Navigate back to the View Entries screen for the form, and now the entry is marked as Approved.

****

**Liferay Form : Auto Save**

Default, a form is auto-saved every minute. You won’t notice anything in the form builder while the back-end auto-saves the form. You can change the auto-save duration in Control Panel → Configuration → System Settings → Forms (in the Content section). To disable auto-save, set the interval to 0.

**Note :** **unpublished forms, an auto-save works just like a manual save. For a published form, however, auto-saved data isn’t automatically propagated to the form. You must click the Save button in the form builder to publish the changes.**

****