

Introduction



Installation

Installation steps

SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



5. Run command in terminal of your project `npm install`

Features



6. Install swfrontend package from [here](#)

7. Run command in terminal of your project `npm run eject` (Commit before running the command)

8. Copy the `config` folder, `.env` file to the root directory of the project

9. Copy the `ScreenRouter.js` to `src/App/` folder or simply create the file and copy the



[Introduction](#)

- > 8. Copy the config folder, .env file to the root directory of the project
- > 9. Copy the ScreenRouter.js to src/App/ folder or simply create the file and copy the below code and paste it into ScreenRouters.js .

[Installation steps](#)
[Setup env files](#)

[SWBackend](#)[SWFrontend](#)[Installation](#)[Before you start](#)[Configure Screen Routers](#)[UI_Meta Reference](#)[Video Tutorial](#)[Release Notes](#)[Data Science](#)[Features](#)

```
import React from "react";
import { Route, Switch } from "react-router-dom";

function getRoutes() {
  return {};
}

export default getRoutes;
```



- 10. Copy the given code in src/App/App.js

```
import React from 'react';
import {BrowserRouter} from "react-router-dom"
import MDMApp from 'swfrontend/MDMApp'
import {localStorageVariableName} from "swfrontend/AppConfigs";
import LoginPage from "swfrontend(LOGIN/LoginPage";
import getRoutes from "./ScreenRouters";
import "./CSS/App.css";
```

```
const App = () => {
```

Introduction



```
const App = () => {
  return (
    localStorage.getItem(localStorageVariableName.authToken) == null ?
      <LoginPage/> :
      <BrowserRouter><MDMApp routes={getRoutes} /></BrowserRouter>
    )
};

export default App;
```

Installation steps

SWBackend



Setup env files

SWFrontend



Installation



Before you start

Configure Screen Routers

11. Replace the following line into your public/index.html

```
<body style="background-color: #f3edeb; zoom: 0.9;">
```

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Setup env files

Features



- To handle framework level configuration for our react app we need to create .env files.
- We will be having two .env.development and .env.production files to configure our react app for development and production stages.
- Place both the .env.development and .env.production files outside src folder.

Introduction

- We will be having two `.env.development` and `.env.production` files to configure our react app for development and production stages.

Installation steps

SWBackend

- Place both the `.env.development` and `.env.production` files outside `src` folder.

Setup env files

SWFrontend

- Below is the example of `.env.development` file.

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science

Features

```
REACT_APP_SERVER_PROTOCOL=http
REACT_APP_SERVER_URL=localhost
REACT_APP_SERVER_PORT=8000
REACT_APP_API_PREFIX=testproject
REACT_APP_API_MASTER_ROUTE=master
REACT_APP_USER_NAME=guest
REACT_APP_USER_PASSWORD=tseug
REACT_APP_USER_TYPE=guest
REACT_APP_API_NAME=Master
REACT_APP_PROJECT_NAME=TEST PROJECT
REACT_APP_PROJECT_ROUTE=/app
REACT_APP_DEFAULT_PAGINATION_SIZE=20
REACT_APP_RBAC_USER_APP=users
REACT_APP_RBAC_USER_MODEL=userprofile
REACT_APP_RBAC_USERPROFILE_CHECK=false
REACT_APP_CUSTOM_LOGIN=false
REACT_APP_TOKEN_PREFIX=testproject
```

Introduction



SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



- Below is the example of `.env.production` file.

```
REACT_APP_SERVER_PROTOCOL=https
REACT_APP_SERVER_URL=172.24.138.114
REACT_APP_SERVER_PORT=8001
REACT_APP_API_PREFIX=testproject
REACT_APP_API_MASTER_ROUTE=master
REACT_APP_USER_NAME=guest
REACT_APP_USER_PASSWORD=tseug
REACT_APP_USER_TYPE=guest
REACT_APP_API_NAME=Master
REACT_APP_PROJECT_NAME=TEST PROJECT
REACT_APP_PROJECT_ROUTE=/app
REACT_APP_DEFAULT_PAGINATION_SIZE=20
REACT_APP_RBAC_USER_APP=users
REACT_APP_RBAC_USER_MODEL=userprofile
REACT_APP_RBAC_USERPROFILE_CHECK=false
REACT_APP_CUSTOM_LOGIN=false
REACT_APP_TOKEN_PREFIX=testproject
```

Installation steps

Setup env files

← 3 TIER ARCHITECTURE

BEFORE YOU START →

Introduction



Before you start

SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Project Instructions

Features



- Proper client side validations should be maintained for all forms.
- Integration testing like Cypress / Seleneum and reports of integration / unit testing should be generated before each code review and before deployments. Jest should be used for unit testing.

Useful links

Project Instructions

Project Structure



Project Structure

Introduction



```
src
```

SWBackend



```
|- App
  |- Configs
    |- AppConfig.js
  |- CSS
    |- App.css
  |- Utils
    |- APIHelper.jsx
    |- HelperFunction.jsx
  |- App.js
  |- ScreenRouters.js
```

SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



```
|- [ModuleName]
  |- SubModuleName1 <--- Only If having submodules else skip this folder and continue
    |- Screens
```

Features



```
|- ScreenName1
  |- Screen1
    |- Components
      |- Component.css
      |- Component1.jsx
    |- Screen1.jsx
    |- Screen1DataProvider
```

Useful links

Project Instructions

Project Structure

[Introduction](#)

```
    |- App.css
    |- Utils
        |- APIHelper.jsx
        |- HelperFunction.jsx
    |- App.js
    |- ScreenRouters.jsx
```

[Useful links](#)[SWBackend](#)

```
    |- App.js
    |- ScreenRouters.jsx
```

[Project Instructions](#)[SWFrontend](#)

```
    |- cypress
        |-
    |- cypress.json
    |- [ModuleName]
        |- SubModuleName1 <--- Only If having submodules else skip this folder and continue
            |- Screens
                |- ScreenName1
                    |- Screen1
                        |- Components
                            |- Component.css
                            |- Component1.jsx
                        |- Screen1.jsx
                        |- Screen1DataProvider
                |- ScreenRouters.jsx
                |- Validators.jsx
```

[Project Structure](#)[Installation](#)[Before you start](#)[Configure Screen Routers](#)[UI_Meta Reference](#)[Video Tutorial](#)[Release Notes](#)[Data Science](#)

```
    |- Screen1.jsx
    |- Screen1DataProvider
```

[Features](#)

```
    |- Constants.jsx
    |- Helpers.jsx
    |- ModuleName.css
    |- ScreenRouters.jsx
```

- Example

Introduction



src

SWBackend



| - App
| - Configs
| - AppConfig.js

SWFrontend



| - CSS
| - App.css
| - Utils
| - APIHelper.jsx
| - HelperFunction.jsx
| - App.js
| - ScreenRouters.js

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



| - cypress
| - ..
| - cypress.json
| - BalanceConfirmation

Features



| - Screens
| - PartnerScreens
| - PartnerScreensFormView
| - Components
| - Component.css
| - Component1.jsx
| - PartnerScreensFormView.jsx
| - PartnerScreensFormViewDataProvider.jsx

Useful links

Project Instructions

Project Structure

[Introduction](#)

- | - PartnerScreensFormView.jsx
- | - PartnerScreensFormViewDataProvider.jsx

[SWBackend](#)

- | - PartnerScreensListView
- | - Components
- | - Component.css
- | - Component1.jsx
- | - PartnerScreensListView.jsx
- | - PartnerScreensListViewDataProvider.jsx

[SWFrontend](#)[Installation](#)[Before you start](#)[Configure Screen Routers](#)[UI_Meta Reference](#)[Video Tutorial](#)[Release Notes](#)[Data Science](#)

- | - ScreenRouters.jsx
- | - Validators.jsx
- | - SendBalanceConfirmation
- | - RequestBalanceConfirmation
- | - Constants.jsx
- | - Helpers.jsx
- | - ModuleName.css
- | - ScreenRouters.jsx

[Features](#)

- Download reference structure zip from [here](#)
- Download [ApiHelper.js](#) from [here](#).
- Download [AppConfigs.js](#) from [here](#).
- Download [DataProvider.js](#) (sample) from [here](#).

[Useful links](#)[Project Instructions](#)[Project Structure](#)

Screen Routers

Introduction >

SWBackend >

SWFrontend > ▼

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science >

Features >

What is ScreenRouters.js ?

- In the previous session we have learned about. How do we can define `sections` and `screens` inside our `EXTRA_USER_META` in `settings.py`. With the example of `Reports` section.
- But how do `NavBar` links know what to display when you click on particular links inside navbar. That's where `ScreenRouter.js` will help us.
- The `ScreenRouter.js` will perform the mapping between `NavBar` links and `React Components`.
- Currently our `EXTRA_USER_META` will look like as below :

```
EXTRA_USER_META = {  
    'MASTERS': [  
        {  
            'app': 'masters',  
            'title': 'MASTER'  
        }  
    ]  
}
```

What is ScreenRouters.js ?

Why Undefined Routers Error ?

Creating React Components

Mapping Components to ScreenRouters.js

Introduction



```
EXTRA_USER_META = {
    'MASTERS': [
        {
            'app': 'masters',
            'title': 'MASTER'
        }
    ],
    'Reports': [
        {
            'screen': 'SendBCReport',           <--- Use exact name in ScreenRouters.js
            'type': 'subsection',
            'title': 'Confirmations Sent'    <--- display name, for sidenavbar on UI
        },
        {
            'screen': 'RequestBCReport',
            'type': 'subsection',
            'title': 'Confirmations Requested'
        }
    ]
}
```

SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



What is ScreenRouters.js ?

Why Undefined Routers Error ?

Creating React Components

Mapping Components to ScreenRouters.js

- Now we will refresh our UI . As we can see our screen will be crashed and when we navigate to developer tools . We will see the following error.

Introduction



- Now we will refresh our UI . As we can see our screen will be crashed and when we navigate to developer tools . We will see the following error.

SWBackend



- ✖ > Uncaught (in promise) TypeError: Cannot read property 'routers' of undefined

```
at PageSet.js:59
at Array.map (<anonymous>)
at PageSet.js:37
at Array.map (<anonymous>)
at PageSet.render (PageSet.js:36)
at finishClassComponent (react-dom.development.js:17160)
at updateClassComponent (react-dom.development.js:17110)
at beginWork (react-dom.development.js:18620)
at HTMLUnknownElement.callCallback (react-dom.development.js:188)
at Object.invokeGuardedCallbackDev (react-dom.development.js:237)
at invokeGuardedCallback (react-dom.development.js:292)
at beginWork$1 (react-dom.development.js:23203)
at performUnitOfWork (react-dom.development.js:22154)
at workLoopSync (react-dom.development.js:22130)
at performSyncWorkOnRoot (react-dom.development.js:21756)
at react-dom.development.js:11089
at unstable_runWithPriority (scheduler.development.js:653)
at runWithPriority$1 (react-dom.development.js:11039)
at flushSyncCallbackQueueImpl (react-dom.development.js:11084)
at flushSyncCallbackQueue (react-dom.development.js:11072)
at scheduleUpdateOnFiber (react-dom.development.js:21199)
at Object.enqueueSetState (react-dom.development.js:12639)
at
ScreenRoot.push../node_modules/react/cjs/react.development.js.Component.setState (react.development.js:471)
at ScreenRoot.js:51
```

SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



- Now let's assume we want to display our component whenever user clicks onto the link inside NavBar . For that we need to configure our ScreenRouter.js as per below

What is ScreenRouters.js ?

Why Undefined Routers Error ?

Creating React Components

Mapping Components to ScreenRouters.js

Introduction



- Now let's assume we want to display our component whenever user clicks onto the link inside `NavBar`. For that we need to configure our `ScreenRouter.js` as per below guide.

SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Creating React Components

Features



- Let's first create `React Components` for our reports section. Here we will create two components as below :

- RequestBalanceReport (For Confirmations Requested Screen)
- SendBalanceReport (For Confirmations Send Screen)

//RequestBalanceReport.js File

What is ScreenRouters.js ?

Why Undefined Routers Error ?

Creating React Components

Mapping Components to ScreenRouters.js

[Introduction](#)

Creating React Components

[SWBackend](#)

- Let's first create `React Components` for our reports section. Here we will create two components as below :

[SWFrontend](#)

- `RequestBalanceReport` (For Confirmations Requested Screen)
- `SendBalanceReport` (For Confirmations Send Screen)

[Installation](#)[Before you start](#)[Configure Screen Routers](#)

```
//RequestBalanceReport.js File
```

[UI_Meta Reference](#)

```
import React from "react";
```

[Video Tutorial](#)

```
class RequestBalanceReport extends React.Component{
```

[Release Notes](#)

```
    render(){
        return(
            <div> Hello From RequestBalanceReport </div>
        );
    }
}
```

```
export default RequestBalanceReport;
```

```
//SendBalanceReport.js File
```

[What is ScreenRouters.js ?](#)[Why Undefined Router Error ?](#)[Creating React Components](#)[Mapping Components to ScreenRouters.js](#)

Introduction



```
//SendBalanceReport.js File
import React from "react";
class SendBalanceReport extends React.Component{
    render(){
        return(
            <div> Hello From SendBalanceReport </div>
        );
    }
}
export default SendBalanceReport;
```

SWBackend



```
class SendBalanceReport extends React.Component{
```

SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



Mapping Components to ScreenRouters.js

```
import React from "react";
import { Route, Switch } from "react-router-dom";

//Import your components here
import RequestBalanceReport from "<YOUR PATH>/RequestBalanceReport";
import SendBalanceReport from "<YOUR PATH>/SendBalanceReport";
```

What is ScreenRouters.js ?

Why Undefined Routers Error ?

Creating React Components

Mapping Components to ScreenRouters.js

Mapping Components to ScreenRouters.js

Introduction

>

```
import React from "react";
import { Route, Switch } from "react-router-dom";

//Import your components here
import RequestBalanceReport from "<YOUR PATH>/RequestBalanceReport";
import SendBalanceReport from "<YOUR PATH>/SendBalanceReport";

export const app = `${process.env.REACT_APP_PROJECT_ROUTE}`;
```

What is ScreenRouters.js ?

SWBackend

>

```
//Import your components here
import RequestBalanceReport from "<YOUR PATH>/RequestBalanceReport";
import SendBalanceReport from "<YOUR PATH>/SendBalanceReport";
```

Why Undefined Routers Error ?

SWFrontend

>

```
//Import your components here
import RequestBalanceReport from "<YOUR PATH>/RequestBalanceReport";
import SendBalanceReport from "<YOUR PATH>/SendBalanceReport";
```

Creating React Components

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science

>

```
/**
 * Custom component mapping
 * @param {component} : Component being mapped
 * @param {routers} : Creating route for a component specified with route link
 */

function getRoutes() {
  return {
```

Features

>

```
    RequestBCReport: {           <-- Screen name as defined inside EXTRA_USER_
      component: RequestBalanceReport,
      routers: [
        <Switch>
          <Route
            exact
            path={`${app}/RequestBCReport`} <-- Path same as defined inside EXTRA_US
            component={RequestBalanceReport}
          
```

Mapping Components to
ScreenRouters.js

Introduction



What is ScreenRouters.js ?

SWBackend



Why Undefined Routers Error ?

SWFrontend



Creating React Components

Installation

```
<Switch>
  <Route
    exact
    path={`${app}/RequestBCReport`}   <-- Path same as defined inside EXTRA_US
    component={RequestBalanceReport}
  />
</Switch>,
],
},
```

Before you start

```
SendBCReport: {                                <-- Screen name as defined inside EXTRA_USER_
  component: SendBalanceReport,
  routers: [
    <Switch>
```

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



```
      <Route
        exact
        path={`${app}/SendBCReport`}   <-- Path same as defined inside EXTRA_USER
        component={SendBalanceReport}
      />
```

Features



```
    </Switch>,
  ],
},
```

};

}

```
export default getRoutes;
```



UI_Meta Reference

Introduction >

SWBackend >

SWFrontend > ▼

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science >

Features >

What Is UI_Meta ?

- As discussed earlier the `UI_Meta` will help the framework to develop dynamic frontend screen. Along with `Create` , `Delete` , `Update` , `Display` operations.

List Of UI_Meta HTML Controls

TextBox

```
{  
    "label": "Auditor Name",  
    "decorator": "auditor_name",  
    "type": "textbox",  
    "required": "true",  
    "message": "Enter Auditor Name",  
    "id": "auditor_name",  
    "placeholder": "Please enter Auditor Name",  
}
```

[What Is UI_Meta ?](#)

[List Of UI_Meta HTML Controls](#)

[TextBox](#)

[TextArea](#)

[Date](#)

[Radio](#)

[Select \(Combobox\)](#)

[Select \(Combobox\) Dependent](#)

As discussed earlier the "ui_meta" will help the framework to develop dynamic frontend screen. Along with Create , Delete , Update , Display operations.

Introduction



SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



What Is UI_Meta ?

List Of UI_Meta HTML Controls

TextBox

TextArea

Date

Radio

Select (Combobox)

Select (Combobox) Dependent

List Of UI_Meta HTML Controls

TextBox

```
{  
    "label": "Auditor Name",  
    "decorator": "auditor_name",  
    "type": "textbox",  
    "required": "true",  
    "message": "Enter Auditor Name",  
    "id": "auditor_name",  
    "placeholder": "Please enter Auditor Name",  
    "disabled": False,  
    "maxlength": 300  
}
```

TextArea

```
{  
    "label": "Auditor Address",  
    "decorator": "auditor_address",  
    "type": "textArea",  
    "required": "true",  
    "message": "Enter Auditor Address",  
    "id": "auditor_address",  
    "placeholder": "Please enter Auditor Address",  
    "disabled": False,  
    "maxlength": 500  
}
```

TextArea

Introduction >

```
{  
    "label": "Auditor Address",  
    "decorator": "auditor_address",  
    "type": "textarea",  
    "required": "true",  
    "message": "Enter Auditor Address",  
    "id": "auditor_name",  
    "placeholder": "Please enter Auditor Address",  
    "disabled": False,  
    "maxlength": 300  
}
```

What Is UI_Meta ?

SWBackend >

List Of UI_Meta HTML Controls

SWFrontend > ▾

Installation

TextBox

Before you start

TextArea

Configure Screen Routers

Date

UI_Meta Reference

Radio

Video Tutorial

Select (Combobox)

Release Notes

Select (Combobox) Dependent

Data Science >

Date



Features >

```
{  
    "label": "Last Updated : ",  
    "decorator": "last_updated_date",  
    "required": "true",  
    "message": "Last Updated Date",  
    "placeholder": "Select Date",  
    "type": "date",  
    "id": "last_updated_date",  
    "dateFormatList": "YYYY-MM-DDTHH:mm:ss",  
    "disabled": false  
}
```



Type here to search



Introduction



SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



Date

```
{  
    "label": "Last Updated : ",  
    "decorator": "last_updated_date",  
    "required": "true",  
    "message": "Last Updated Date",  
    "placeholder": "Select Date",  
    "type": "date",  
    "id": "last_updated_date",  
    "dateFormatList": "YYYY-MM-DDTHH:mm:ss",  
    "readonly": "true",  
    "disabled": True  
}
```

Radio

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "options": [{"label": "Active", "value": "active"}, {"label": "In Progress", "value": "in_progress"}, {"label": "Completed", "value": "completed"}]  
}
```

What Is UI_Meta ?

List Of UI_Meta HTML Controls

TextBox

TextArea

Date

Radio

Select (Combobox)

Select (Combobox) Dependent

Radio

Introduction



```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

What Is UI_Meta ?

SWBackend



```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

List Of UI_Meta HTML Controls

SWFrontend



```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

TextBox

Installation

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

TextArea

Before you start

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Date

Configure Screen Routers

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Radio

UI_Meta Reference

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Select (Combobox)

Video Tutorial

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Select (Combobox) Dependent

Release Notes

```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Data Science



```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Features



```
{  
    "label": "Is Active",  
    "decorator": "is_active",  
    "type": "radio",  
    "radioType": "group",  
    "id": "isActive",  
    "required": "true",  
    "message": "Please select the status",  
    "listed": "yes",  
    "list_data": [  
        {  
            "True": "True"  
        },  
        {  
            "False": "False"  
        },  
    ],  
    "link_api": "isactive",  
    "disabled": False,  
}
```

Select (Combobox)

Workflow Doc X | SWFramework X | SWFramework X | Workflow Doc X | WhatsApp X | MSN | Perso X | React App X | UI_Meta Ref X + - Chat

Not secure https://sw-doc.amns.in/docs/swfrontend/uimetreference/

AM/NS INDIA SWFramework Docs

Introduction



Select (Combobox)

SWBackend



{

SWFrontend



```
    "label": "Location",
    "decorator": "location",
    "type": "select",
    "required": "true",
    "message": "Select Location",
    "validator": None,
    "validateStatus": None,
    "id": "location",
    "placeholder": "Please Select Location",
    "option": "location",
    "linked": "yes",
    "link_api": "LocationMaster",
    "primary_key_field": "id",
    "foreign_table_attribute_name": "name",
    "disabled": False,
```

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



}

What Is UI_Meta ?

List Of UI_Meta HTML Controls

TextBox

TextArea

Date

Radio

Select (Combobox)

Select (Combobox) Dependent

Select (Combobox) Dependent

```
{  
    "label": "Country",
```

Select (Combobox) Dependent

Introduction



SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science



Features



```
{
    "label": "Country",
    "decorator": "country_id",
    "type": "select",
    "required": "true",
    "message": "Select Country",
    "validator": None,
    "validateStatus": None,
    "id": "country_id",
    "placeholder": "Please Select Country",
    "option": "country_id",
    "linked": "yes",
    "link_api": "Country",
    "primary_key_field": "country_id",
    "foreign_table_attribute_name": "name",
    "disabled": False,
},
{
    "label": "State",
    "decorator": "sid",
    "type": "select",
    "required": "true",
    "message": "Select State",
    "validator": None,
}
```

What Is UI_Meta ?

List Of UI_Meta HTML Controls

TextBox

TextArea

Date

Radio

Select (Combobox)

Select (Combobox) Dependent

[Introduction](#)

```
        "primary_key_field": "country_id",
        "foreign_table_attribute_name": "name",
        "disabled": False,
    },
    {
        "label": "State",
        "decorator": "sid",
        "type": "select",
        "required": "true",
        "message": "Select State",
        "validator": None,
        "validateStatus": None,
        "id": "sid",
        "placeholder": "Please Select State",
        "option": "country_id",
        "linked": "yes",
        "dropdown_dependent_field": [
            {
                "apiCall": "Country",
                "decorator": "country_id"
            },
            ],
        "link_api": "state",
        "primary_key_field": "sid",
        "foreign_table_attribute_name": "name",
        "disabled": False,
    }
```

[SWBackend](#)[What Is UI_Meta ?](#)[SWFrontend](#)[Installation](#)[List Of UI_Meta HTML Controls](#)[Before you start](#)[TextBox](#)[Configure Screen Routers](#)[TextArea](#)[UI_Meta Reference](#)[Date](#)[Video Tutorial](#)[Radio](#)[Release Notes](#)[Select \(Combobox\)](#)[Data Science](#)[Select \(Combobox\) Dependent](#)[Features](#)[Video](#)[API](#)[Database](#)[Machine Learning](#)[Deployment](#)

Release Notes

Introduction >

Release Notes v0.1.0 :-

SWBackend >

Release Notes v0.1.2 :-

SWFrontend >▼

Release Notes v0.1.3 :-

Installation

Release Notes v0.1.4 :-

Before you start

Release Notes v0.1.5 beta :-

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science >

Release Notes v0.1.2 :-

- Added file upload wrapper
- Added login module component
- Bug fixes

Features >

Release Notes v0.1.3 :-

- Fixed "ApplyReadOnlyPermission", which handles consecutive restrict field in formview
- New submit/clear section in formheader
- Removal of submit/clear from formfooter
- New fonts applied



Type here to search



Release Notes v0.1.3 :-

Introduction



- Fixed "ApplyReadOnlyPermission", which handles consecutive restrict field in formview
- New submit/clear section in formheader
- Removal of submit/clear from formfooter
- New fonts applied
- Loading progress to be shown when applied/removed filters
- Clear option in search
- Removed bug of "applyRemovePermission" in FormView
- Removed bug of SearchBoxWrapper
- New FileUploadWrapper, Loginscreen added

[Release Notes v0.1.0](#) :-

[Release Notes v0.1.2](#) :-

[Release Notes v0.1.3](#) :-

[Release Notes v0.1.4](#) :-

[Release Notes v0.1.5 beta](#) :-

SWBackend



SWFrontend



Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

[Release Notes](#)

Data Science



Features



Release Notes v0.1.4 :-

- Added Dashlets
- Added Breadcrumbs to Contentpane.js
- Added props: "boolean: shouldResizeColumns" to GridViewWrapper for resizing columns
- Added Improvements in CSS
- Added Fonts compatible with all browsers added
- Added FileUploadWrapper updated for preview on MOSSDMS
- Added RadioButtonWrapper setting defaultvalues
- Added SelectBoxWrapper autosetting selectbox if there is only 1 option in the list
- Added ListHeader column filter
- Added ListView pagination

Release Notes v0.1.4 :-

Introduction

- >
 - Added Dashlets
 - Added Breadcrumbs to Contentpane.js

[Release Notes v0.1.0 :-](#)

SWBackend

- >
 - Added props: "boolean: shouldResizeColumns" to GridViewWrapper for resizing columns

[Release Notes v0.1.2 :-](#)

SWFrontend

- >
 - Added Improvements in CSS

[Release Notes v0.1.3 :-](#)

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science

- >
 - Redesigned RBAC's Users, Groups section as per new MDM's design
 - Redesigned FormViewContent to have Workflow Section **below** Save section
 - Redesigned Timeline: Tail color changed

Features

- >
 - Removed all the obsolete code
 - Removed all the warnings generated during react compilation
 - Removed all console logs
 - Bug resolved: Timeline.js crashes, added a new condition on line:14
 - Bug resolved: Mobile Responsive sidebar
 - Bug resolved: Remarks were not getting saved in workflow
 - Bug resolved: MultipleSelectBox Wrapper edit not persistent
 - Bug resolved: File type check before upload in FileUploadWrapper
 - Bug resolved: Cascaderdropdown edit data not setting

[Release Notes v0.1.4 :-](#)

[Release Notes v0.1.5 beta :-](#)



Not secure

<https://sw-doc.amns.in/docs/swfrontend/release/>

Docs

AMNS
INDIA

SWFramework

[Introduction](#)

- Removed all console logs
- Bug resolved: Timeline.js crashes, added a new condition on line:14
- Bug resolved: Mobile Responsive sidebar
- Bug resolved: Remarks were not getting saved in workflow
- Bug resolved: MultipleSelectBox Wrapper edit not persistent
- Bug resolved: File type check before upload in FileUploadWrapper
- Bug resolved: CascaderWrapper edit data not setting
- Bug resolved: Validation trigger onBlur event in all wrappers
- Bug resolved: MultipleEmailWrapper validation,edit value and disabling
- Bug resolved: MultipleMobileWrapper numeric validation and disabling
- Bug resolved: MultipleSelectBox removing old values
- Bug resolved: SelectBoxWrapper invalid values removal
- Bug resolved: TextBoxWrapper capital case characters resetting cursor
- Beta: CacheMechanism, RBAC's workflow UI

[Release Notes v0.1.0 :-](#)[Release Notes v0.1.2 :-](#)[Release Notes v0.1.3 :-](#)[Release Notes v0.1.4 :-](#)[Release Notes v0.1.5 beta :-](#)[SWBackend](#)[SWFrontend](#)[Installation](#)[Before you start](#)[Configure Screen Routers](#)[UI_Meta Reference](#)[Video Tutorial](#)[Release Notes](#)[Data Science](#)[Features](#)

Release Notes v0.1.5 beta :-

- Added DateFilter Support
- Added Excel Import Functionality
- Added Protocol env variable for https
- Added nivo, ckeditor library
- Added Wrappers for Line, Bar, Pie and Map charts for using in Dashboard
- Added ckbackend library package
- Added RichTextWrapper
- Added File Upload/Download functionality for Multiple/Single file upload

Introduction

>

- Added DateFilter Support
- Added Excel Import Functionality
- Added Protocol env variable for https

SWBackend

>

- Added nivo, ckeditor library
- Added Wrappers for Line, Bar, Pie and Map charts for using in Dashboard
- Added ckbackend library package
- Added RichTextWrapper

Installation

Before you start

Configure Screen Routers

UI_Meta Reference

Video Tutorial

Release Notes

Data Science

>

- Autoresize column size if columns are <=3

Features

>

- Bug resolved: TextArea Wrapper not marking as required
- Bug resolved: logout button redirecting to "/"
- Bug resolved: LoginPage.js fixes for custom_login
- Bug resolved: xls to xlsx for excel export
- Bug resolved: readOnlyPermission function in FormView splicing other fields
- Bug resolved: DateFormat issue in the formview
- Bug resolved: Selectbox not accepting static options from meta

Release Notes v0.1.0 :-

Release Notes v0.1.2 :-

Release Notes v0.1.3 :-

Release Notes v0.1.4 :-

Release Notes v0.1.5 beta :-