

# FSM: UI Implementation - Guidelines & FAQs

## Enable FSM

Modules are enabled by MDMS config at <https://github.com/egovernments/egov-mdms-data/blob/DEV/data/pb/tenant/citymodule.json>

Create a new branch for the state if already doesn't exist from the master of repo [egovernments/digit-ui](#).

To add and enable any module in the new UI, `src/App.js` needs to be changed. We export only the `init` function of the module to take care of all the initializations.

```
1import { initFSMComponents } from "@egovernments/digit-ui-module-fsm";
2
3const enabledModules = ["FSM"];
4initFSMComponents();
5
```

## Changing CSS

CSS classes are published over CDN and can be seen at <https://unpkg.com/@egovernments/digit-ui-css/dist/index.css>

Any class can be overwritten. Please make changes in the `src/index.css` file.

## Customizing fields in a form

First, add any new component created in the registry,

```
1Digit.ComponentRegistryService.setupRegistry({
2  SelectName: SelectName
3});
```

Application config: The default config can be found at <https://github.com/egovernments/digit-ui-internals/blob/development/packages/modules/fsm/src/config/NewApplication/config.js>. The config has different items in the application form. The new config, if made, needs to be initialized at

```
Digit.Customizations.FSM.applicationFormConfig
```

```
1{
2  label: "ES_NEW_APPLICATION_PROPERTY_TYPE",
3  isMandatory: true,
4  type: "component",
5  route: "property-type",
6  key: "propertyType",
7  component: "SelectPropertyType",
8  texts: {
9    headerCaption: "",
10   header: "CS_FILE_APPLICATION_PROPERTY_LABEL",
11   cardText: "CS_FILE_APPLICATION_PROPERTY_TEXT",
```

```
12 submitBarLabel: "CS_COMMON_NEXT",
13 },
14 nextStep: "property-subtype",
15 }
```

- label: is the employee side label
- texts: is used for citizen side step form
- component: the component that is to be rendered here. Any component can be created to show here, examples can be seen at [pageComponents](#), These components are passed following params
  - t: translate function to be used to convert a key to localized text. e.g.,  
t("CS\_CREATECOMPLAINT\_MOHALLA")
  - config: current step config as shown above
  - onSelect: on clicking of next or input, this handled to be called to save the data in the form state
  - userType: employee or citizen
  - formData: the current form state.

### Customizing views

Use the followings to customizations are available:

Application details can be customized via defining

`Digit.Customizations.FSM.getApplicationDetailsTableRows`

The function will be passed following params

- id: application id
- service: application response object
- role: currently logged in user role CITIZEN or EMPLOYEE
- t: function used for translation

```
1 getApplicationDetailsTableRows: ({ id, service, role, t }) => {
2   if (role === "CITIZEN") {
3     return {
4       CS_FSM_APPLICATION_APPLICATION_NO: service.applicationNo,
5       CS_FSM_APPLICATION_DETAIL_STATUS: t("CS_COMMON_" + service.applicationStatus),
6       ...
7     };
8   }
9   return {};
10 }
```

### FAQ

In addition to any FAQs contained in the main implementation document.