



Ministry of Housing and Urban Affairs
Government of India



Ministry of Housing and Urban Affairs
Government of India



ET Panache

National Urban Digital Mission

Building cities that work for people

The background of the slide is a collage of three photographs. The leftmost photo shows a man with a mustache, wearing a light-colored striped shirt, smiling and counting a stack of Indian Rupee banknotes. The middle photo shows a family of three: a woman in a pink sari, a woman in a purple sari, and a man in a light blue shirt, all smiling. The rightmost photo shows a man in a blue and red jacket standing in a crowd, looking towards the camera.

Customizing Frontend

Outline

1. Overview
2. Local setup
3. Adding/Removing fields
4. Making Mandatory/Non mandatory
5. Writing new forms
6. Fields changes (Text to Dropdown)

Overview

UI - frontend/micro-ui/web/micro-ui-internals

- Meant for eGov development team to build components and default modules.
- Contains following modules:
 - CSS Library
 - UI Components (react-components)
 - Utils Library: Contains Services, Localization handling and React Hooks.
 - UI Modules
 - Core - containing login, routing and global state.
 - PGR
 - FSM
 - PT
 - Payment,

Local Setup

Cloning, Transpiling, Bundling, and Running apps locally:-

1. Clone the App from the remote repo:
 - a. `git clone <Remote URL>`
2. Switch path to `/frontend/micro-ui/web`
 - a. Run `"yarn install"` → to install all required dependencies in UPYOG-UI app
 - b. Run `"yarn dev:start"` → what it does?
 - i. It first builds CSS and sass files for all the modules.
 - ii. Transpiles all the packages present inside the dev-package folder to ES5 version and links dependencies.
 - iii. A separate new window will be opened, we can check the latest changes in that browser.

Adding/Removing fields

Remove a field:

In the MDMS configuration, we can just remove an object and update the mdms file.
So it automatically reflects in UI.

Add a new Field:

Addition of a new field can be done by first updating a mdms file, then add a new component in customisation folder and register in same component name.

The same process to be followed for the overriding any component or screen

Making Mandatory/Non mandatory

Making field mandatory and non mandatory can be controlled by MDMS change
Refer

[FSM Config](#)

Example

In every config we have a attribute called isMandatory , it can be set as true or false

```
isMandatory": true,
```

Writing new forms and Fields changes (Text to Dropdown)

Modifying Any Application details, Hooks, Api's can be done through overriding the existing file.

Example

[Overriding Component](#)



Thank You

“UPYOGal Transformation is more about humans than UPYOGal”

