DIGIT UI: Implementation - Development Guidelines & FAQs

Development Setup

We will use this doc for development setup guidelines and FAQs.

Repo: https://github.com/egovernments/digit-ui - Connect to preview

Use the dev branch as the base branch for the development.

Create branches as per following <bug/feature>/<JIRA ticket ID>, e.g.,

- bug/LIN-90
- feature/FSM-05

Once done with the change, create a pull request to the development branch. We will deploy and test this branch on QA and after the final QA signoff, it will be merged to the master branch with a release tag.

Clone the repo and following steps to start

```
## install

yarn install

## start dev server

yarn start
```

Development Guidelines

Component Storybook

Please access components here https://digit-ui-internals-react-components.surge.sh

Adding or modifying modules

To add and enable any module, src/App.js needs to be changed.

```
import { PGRModule, PGRLinks, PGRReducers } from "@egovernments/digit-ui-module-pgr";
import { FSMModule, FSMLinks } from "@egovernments/digit-ui-module-fsm";
```

```
const enabledModules = ["PGR", "FSM"];
const registry = new Registry({
  PGRLinks,
  PGRModule,
  FSMModule,
  FSMLinks,
});
```

Also, make sure, these modules are enabled by MDMS config at https://github.com/egovernments/egov-mdms-data/blob/DEV/data/pb/tenant/citymodule.json

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,

```
const registry = new Registry({
  SelectName: SelectName,
  PGRLinks,
  PGRModule,
  FSMModule,
  FSMLinks,
});
```

Use the followings to customizations are available:

PGR

1. complaint form - Digit.Customizations.PGR.complaintConfig

Default config is available at https://gist.github.com/abhinav-egov/15ae1d438a8fcacfb36164fbbb49d6a0 - Connect to preview

Create a config similar to the following

```
export const config = {
  routes: {
    "complaint-type": {
      nextStep: "pincode",
    },
    landmark: {
      nextStep: "apartment",
    },
    apartment: {
      component: "SelectName",
      texts: {
        header: "Apartment or Society",
        cardText: "CS_COMPLAINT_SUBTYPE_TEXT",
        submitBarLabel: "PT_COMMONS_NEXT",
      },
      inputs: [
        {
          label: "Apartment",
          type: "text",
          name: "additionalDetails.apartment",
          validation: {
            minLength: 6,
            maxLength: 7,
          },
          error: "CORE_COMMON_PINCODE_INVALID",
        },
      ],
      nextStep: "upload-photos",
    },
  },
};
```

Any new field should be part of `additionalDetails`

Customizing views

Use the followings to customizations are available:

PGR

1. Complaint details table on the details page of any complaint. - `Digit.Customizations.PGR.getComplaintDetailsTableRows`

The function has the following params available passed by callee:

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

```
getComplaintDetailsTableRows: ({ id, service, role, t }) => {
   if (role === "CITIZEN") {
     return {
       CS_COMPLAINT_DETAILS_COMPLAINT_NO: id,
       CS_COMPLAINT_DETAILS_APPLICATION_STATUS:
`CS_COMMON_${service.applicationStatus}`,
       CS_ADDCOMPLAINT_COMPLAINT_TYPE: `SERVICEDEFS.${complaintType}`,
       CS_ADDCOMPLAINT_COMPLAINT_SUB_TYPE:
`SERVICEDEFS.${service.serviceCode.toUpperCase()}`,
       CS_COMPLAINT_ADDTIONAL_DETAILS: service.description,
       CS_COMPLAINT_FILED_DATE:
Digit.DateUtils.ConvertTimestampToDate(service.auditDetails.createdTime),
       ES_CREATECOMPLAINT_ADDRESS: [
         service.address.landmark,
         Digit.Utils.locale.getLocalityCode(service.address.locality, service.tenantId),
         service.address.city,
         service.address.pincode,
       ],
     };
   }
   return {};
 }
```

FAQ

1. API calls are failing, what should I do?

   Check src/setupProxy.js for proxy url in dev mode

2. API calls are failing with no auth token

   Create a .env file with following

```
REACT_APP_USER_TYPE=

REACT_APP_CITIZEN_TOKEN=a3e5f0a2-4ff0-4680-855e-75051fb3e8f7

REACT_APP_EMPLOYEE_TOKEN=061bad07-c0d5-4200-a74f-ca5ed090cf30
REACT_APP_GRO_TOKEN=43af4c18-6418-4a35-8484-31f0700f465a
```

```
REACT_APP_LME_TOKEN=fa9f4184-dc64-495d-bded-31674c71b09e
```