**PAF UI Architecture :**

1. PAF UI follows an Angular MVC structure which supports the separation of concerns.
2. UI MVC Architecture Diagram :

Model (Scope data)

View (.html files)

Controllers (.js files)

Contollers – It controls how the view and the model will communicate. It’s the presentation logic.

View - They include the UI view files

Modal - They have the binding code through Javascript functions.

1. Folder structure :

Views

PafFramework

index.html,

nav.html,

main.html,

appMetaData.html

index.js

Scripts

Services

Controllers

MainController.js,

NavController.js,

AppMetaDataController.js

MainService.js,

gridStructure.js,

Please find below screenshot for the UI project folder structure :



* Index.html -> This page has the code to import the libraries and define the script files like controllers and services files. The Angular bootstraps in this file. The view is divided into subviews like header view , main content view and footer view.
* nav.html - > This page is the header view which has the navigation bar with links to different pages and also a side bar with different services provided.
* NavController.js -> It has the functionality to show/hide the side bar with search box functionality to search menu name in the dropdown.

**Navbar** :



**Sidebar:**



* main.html - > This page is the main landing page with data populated in grid form with filter and sort facility using Ag-Grid.
* MainController.js - > It has the functionality to fetch the grid data using the response object received from the mainService file. GridOptions rowData is filled with this response object for which below services are added as a dependency in the controller. Scope variables are used to access this data on the html UI.
* MainService.js -> It returns the response of the Rest API urls received from backend team whose response is stored in an object using http service of AngularJs. It is providing the row data of Grid.
* gridStructure.js -> This is a reusable services which is returning the column definition of the grid structure. It can be used in any controller which requires Grid Header structure.



* appMetaData.html -> This is a Application Metadata page which we access from the navbar. It displays information regarding the application and its components.
* AppMetaDataController.js -> It has the functionality to toggle between different tabs in between the page.



**Routing :**

Index.js - > This file specifies the routing between main page and other pages through Angular state provider which has option to specify states i.e views with its html and controller file.