***Angular 2+ UI Architecture Design & best practise***

|  |  |  |
| --- | --- | --- |
|  | | |
|  | Developed for: | *Boeing* |
|  | Project: | *IFDPRO* |
|  |  |
| Version no.: | *1.0* |
| Issue date: | *May 29, 2017* |
|  |  |
| Developed by: | *Simon Wang* |
|  |
|  |
|  |
|  |  |
| Reviewed by: |  |
|  |
|  |
|  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Owner(s): | *Boeing Vancouver, Jeppesen* |

Table of Contents

[1. Introduction 3](#_Toc483820722)

[2. Modules 3](#_Toc483820723)

[3. Routing 3](#_Toc483820724)

# **Introduction**

This document is not the guide of implementation detail. It describes the general good practice in Angular 2+ UI project. The initial version includes only the module and routing management.

# **Modules**

In angular, a module is a mechanism to group components, directives, pipes and services that are related, in such a way that can be combined with other modules to create an application. An angular application can thought of as a puzzle where each piece is a module. There are two types of modules, root module and feature module. The advantage of modules is to load the grouped piece on demand. Lazy loading modules helps us decrease the startup time. With lazy loading our application does not need to load everything at once. Modules that are lazily loaded will only be loaded when the user navigates to their routes.

Root module is AppModule which is implemented in file app.module.ts by convention. App/Root module is the main entry point. It does not belong to any feature modules and available application-wide.

# **Routing**

Routing is optional for Angular application. But routing can make application linkable and allow user to share link with others. It helps modularize the application and also security management.

Structure of Application



Structure in each module



**References**

Angular 4 style guide

<https://angular.io/docs/ts/latest/guide/style-guide.html>