**Angular Application Architecture documentation**

*(for demo projects at Higgsup)*

1. **Background**

In this document, we will describe the Angular architecture we have chosen for demo projects at Higgsup in the upcoming period. This architecture based on [**ngx-admin**](https://github.com/akveo/ngx-admin) template (one of the most popular Angular template built on top of Angular 7+, Bootstrap4, and Nebular).

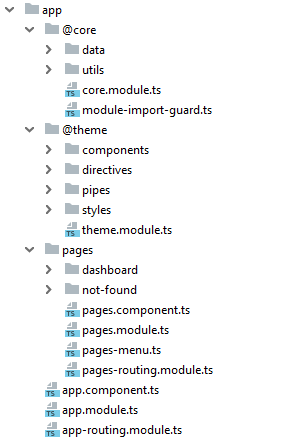
1. **Goals**

The architecture was built with these goals in mind:

* **Simplicity** (for small apps) and **Scalability** (for big apps)
* **Modular & Separation of concerns:** splitting application into different modules, each with a single purpose
* **Reusability**: so we can use this as a base architecture for some different projects
* **Optimization**: with routing and lazy-loading

1. **Architecture**

Below is the overall architecture with the root module - **AppModule** and 3 other sub-modules at the top level: **CoreModule**, **ThemeModule**, and **PagesModule**



* 1. **Root module**

Like any other Angular applications, we have a root module – **AppModule** – the first module that we bootstrap to launch the application.



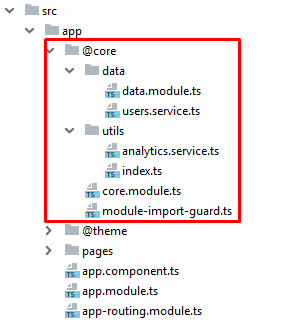
* **AppModule** imports 2 other sub-modules: ThemeModule and CoreModule, that means these 2 modules will be loaded eagerly, before the application start.
* **PagesModule** will be lazy-loaded by using ***loadChildren*** property in the route configuration of **AppRoutingModule**



* 1. **CoreModule**

**CoreModule** contains services and utils that will be used across the entire application:

* **/data**: services used to retrieve data through API
* **/utils**: other utilities shared across application

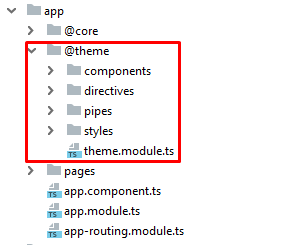


**CoreModule** must be imported only once by the **AppModule**, and then all services it provides can be injected and used in any other components in the application.

* 1. **ThemeModule**

**ThemeModule** defines the layout and styles of the application, it contains:

* **/components:** presentation components which define layout of the application, includes: header, sidebar, main content, and footer.
* **/directives** & **/pipes:** could be used in template of any components whose module import **ThemeModule**.
* **/styles:** .scss and .ts file, define the styles and variables for each theme.



* 1. **PagesModule**

**PagesModule** contains feature modules with routing, used to separate and organize the different features of the application. They are loaded only once, via lazy-loading.



1. **Conclusion**

I have explained the base architecture we will apply to new Angular projects in our company. Developers must follow strictly the project structure and always keep in mind those goals we have mentioned at the beginning of this document. Please read the README file for the coding guide & conventions.