# Angular Refactoring Workshop

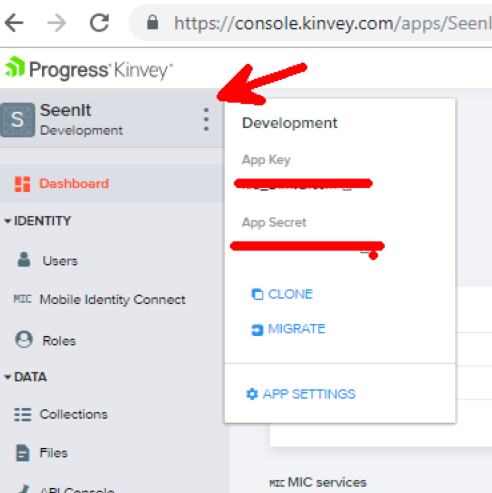
Exercise and homework for the ["Angular Fundamentals" course @ "SoftUni"](https://softuni.bg/courses/angular-2-fundamentals)

## Initial Setup

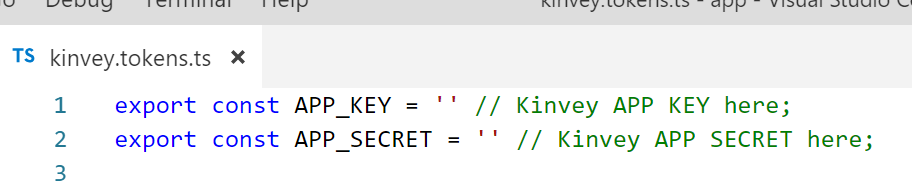
Explore the provided **skeleton**. Your job will be to **refactor** the application using all the knowledge you've gained so far in this course.

Create a registration at [Kinvey](https://console.kinvey.com). If a two-factor token after successful login is needed download the [google authenticator](https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=en) app on google play.

Create an application at Kinvey give it a name (Forum, SeentIt). Get the **API KEY** and **API SECRET** from the menu

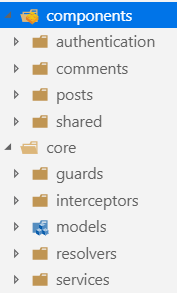


Copy paste the tokens inside the **kinvey.tokens.ts** file. The credentials could be placed inside **environment.ts** as well.



## Project Structure

Your project structure should look like this:



### Components

The **authentication** folder should have **login** and **register** components.

The **comments** folder should have two components (**comment-create** and **comment-info**)

The **shared** folder should have **header**, **footer** and **content (navigation and router outlet)** components

### Core

Here you should store all **injectable's** in the application.

## Refactor

### Generate Components

Generate a header, footer and content component and refactor the application logic from app.component

Generate a post-info child component to reuse some of the logic (sent event emitters for post deletion)

Generate a comment-info and a comment-create component

### Add Children Routes

Fix the routing to work with **children routes** for the **/posts** path.

### Add Models

Two models like **PostInfo** and **CommentInfo** and use them inside services and components.

### Add Interceptors

One interceptor to **clone requests** and add authorization headers (could be Basic or Kinvey) and also **save the token** inside local storage.

One interceptor for **handling errors**.

### Add Resolvers/Use the Async Pipe

Create one resolver to fetch a **single post** for both pages (**edit** and **details**). Practice the **async pipe** as well.

### Add Modules

Create a module for **posts**, **comments**, **shared** and **core** folders and import them inside **app.module.ts**

## Lazy Loading

Lazy Load the **Post Module** and create a **canLoad** authentication guard to protect it.