

Upgrade the Github project to Angular RC5. This exercise requires a lot of focus and mostly likely 1 hour of your time. Eliminate all distractions and try to upgrade the project in one session.

Use the following tips as a roadmap. Finish one step before moving on to the next. Run the application only after you finish all the steps. If the application is not working, look in the console tab. Most of the errors you'll encounter are to do with forgetting to register a provider, or importing a module dependency.

## New Seed Project

- Copy the new seed project files into the existing project. Overwrite existing files.
- Run **npm install** to ensure you get all the packages.
- Remember: once you replace **package.json**, you'll lose the reference to underscore library. The following tutorial shows you a systematic approach to adding underscore or any other libraries in Angular 2:

[https://medium.com/@s\\_eschweiler/using-external-libraries-with-angular-2-87e06db8e5d1#.3tqv772vk](https://medium.com/@s_eschweiler/using-external-libraries-with-angular-2-87e06db8e5d1#.3tqv772vk)

## Modules

- Start by defining a module for the shared area.
- In **PaginationComponent** we use **\*ngIf** and **\*ngFor**. As you learned earlier, these directives are defined in **CommonModule**, so you should import **it** into **SharedModule**.

# Exercise

By: Mosh Hamedani

- Next, define the **PostsModule**. This module is dependent on `RouterModule.forRoot` (for using pagination and spinner) and **HttpModule** (for using the Http service). So, make sure to import them into this module.
- We use **PostService** and **UserService** in the classes in **PostsModule**. You should register these two classes as providers so they're available for dependency injection everywhere.
- Clean up the components in **PostsModule** by removing the **providers** and **directives** properties in their metadata.
- You're done with **PostsModule**. Now, move on to **UsersModule**.
- In **UsersModule** you should import both **FormsModule** and **ReactiveFormsModule** because we're using model-driven forms here.
- Remember to clean up the components in this module by removing the **providers** and **directives** properties in their metadata.
- You've created 3 new modules: **Shared**, **Posts**, and **Users**. Now, import all these modules into **AppModule**.
- As you're revisiting all the existing classes, use the new recommended style guide to make your code cleaner and more maintainable. So, change:

```
import {Component} from '@angular/core';
```

to

```
import { Component } from '@angular/core'.
```

Note the space between curly braces and Component. Also, align "from ..." parts.

## Forms

- Modify the template for the user form to use the new directives.

## Routes

- Currently, all routes are defined on top of **AppComponent**. Extract users and posts routes and move them to their corresponding modules.
- In the new router module, routes cannot start with a slash. So, remove all the pretending slashes.
- Import all the routes into **AppModule**.
- The new router module gives us the ability to highlight the currently selected menu in the navigation bar. Previously, we implemented this by using **isRouteActive** method in **NavBarComponent**. We don't need this method anymore. Instead, you can use **routerLinkActive** attribute as follows:

```
<a routerLink="users" routerLinkActive="active">Users</a>
```

The **routerLinkActive** attribute applies the “**active**” class to the link that represents the active route.