Building for Multiple Environments

You will encounter the need to have different settings when building for dev, prod, test, etc.
This is easy to do, and only requires modifying your workspace's angular.json.

There's a couple of use cases for this:

- Using different environment.ts configuration when building for dev vs prod.
- When whitelabeling an app (same project for different customers).

This page is written as if you're wanting to swap in different environment.ts config files for dev and prod.
But, you can use the same technique to swap in other config, assets, and such, to build an app for different customers.

Environment.ts

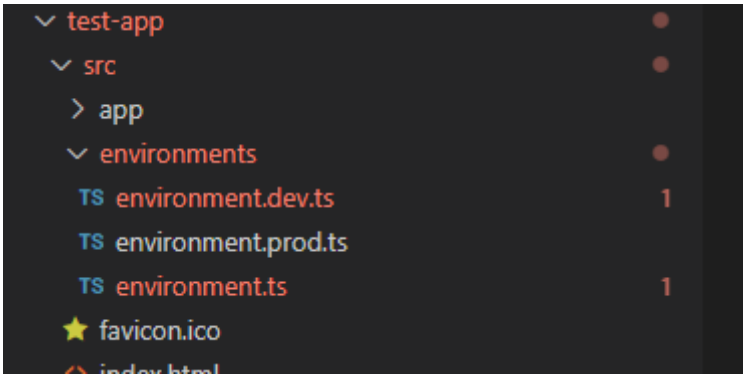
1. Make a folder where the environment.ts file lives in your app project.
This will let you keep the source tree organized.

2. Move the existing environment.ts into it.

3. Make a copy of the existing environment.ts and name it: environment.prod.ts

4. Rename environment.ts to environment.dev.ts.

Once done, your project folder will look like this:



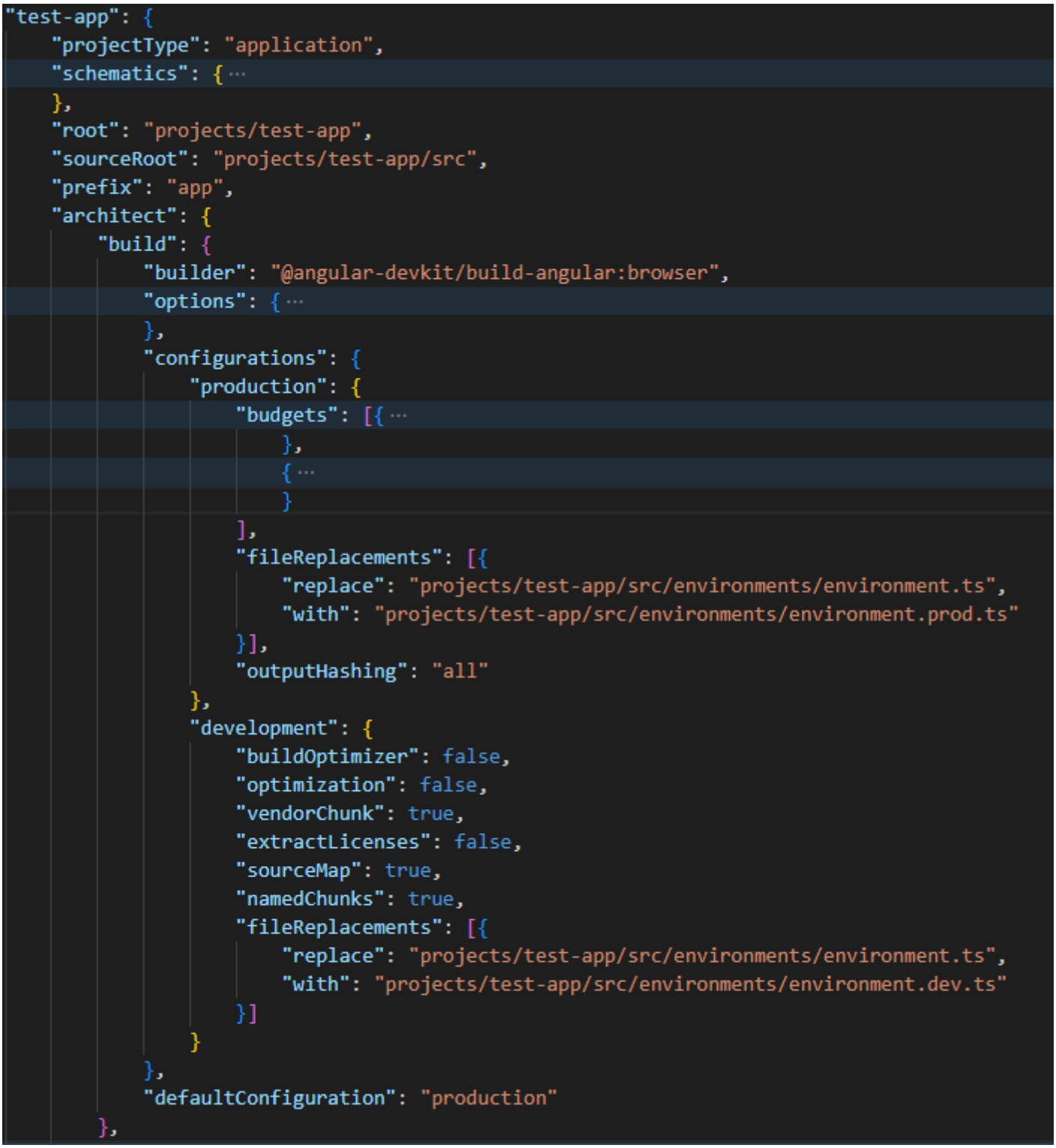
Angular.json

Now, open your angular.json file and locate the build configurations section for your app.
It will be in this path: projects -> appname -> architect -> build -> configurations

We will paste in this block into the development and production configuration nodes:

```
"fileReplacements": [{
    "replace": "projects/test-app/src/environments/environment.ts",
    "with": "projects/test-app/src/environments/environment.prod.ts"
}],
```

Making the configurations block look something like this:



Referenced Config

Do a quick search across your codebase for code that imports the environment class (from environment.ts), and update the path to include the environments folder that you created, earlier.

Below is an example, where it is imported in main.ts.

Found in main.ts

The below main.ts takes action on config from the environment.ts. We must update the import statement, to point to the path of the base environment class, in environments/environment:

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

It may also be imported in App.Module.ts.
Be sure to check, there.

Revision #1
Created 2 March 2025 20:44:00 by glwhite
Updated 2 March 2025 21:29:36 by glwhite