□ Integration of Angular Application with Spring Boot (Same Port Setup)

□ Objective

To **merge an Angular frontend application** with a **Spring Boot backend**, so that:

- Spring Boot serves the Angular application.
- Both frontend and backend are accessible on the **same port** (e.g., `localhost:8080`).
- No need to separately host Angular on a different port like `4200`.
- Angular is bundled into the Spring Boot build and served from the `/static` folder inside the JAR/WAR.

□ Steps to Achieve Integration

□ Step 1: Copy Angular App Root Files to Spring Boot Root

Copy the following files from your Angular app (e.g., `EWDS-angular/`) to your Spring Boot root directory (`EWDS-springboot/`):

- `angular.json`
- `package.json`
- `package-lock.json`
- `tsconfig.json`
- `tslint.json` (if applicable)

These files are necessary to build the Angular project from within the Spring Boot project structure.

□ Step 2: Merge Angular's `src/` into Spring Boot's `src/`

Copy all contents from Angular's `src/` folder (like `app/`, `assets/`, `index.html`, `main.ts`, `styles.css`, etc.) into the Spring Boot project's `src/` directory:

```
EWDS-angular/src/ => EWDS-springboot/src/
```

Ensure **file conflicts** are handled carefully (e.g., no overwriting Spring Boot Java code inside `src/main/java`).

□ Step 3: Modify Angular Build Command

Update the `scripts` section of `package.json`:

```
"scripts": {
  "build": "ng build --watch"
}
```

This allows automatic Angular rebuilds when files change.

□ Step 4: Adjust `angular.json` Build Settings

1. Under `"architect" -> "build"`, update the `builder`:

   ```
   "builder": "@angular-devkit/build-angular:browser"
   ```

2. If `@angular-devkit/build-angular:browser` is missing, install it:

   ```
   npm install -g @angular/cli
   npm install
   npm install --save-dev @angular-devkit/build-angular
   ```

3. Change the `options` section under `"build"`:

   ```
   "options": {
     "outputPath": "target/classes/static",
     "index": "src/index.html",
     "main": "src/main.ts",
     "polyfills": ["zone.js"],
     ...
   }
   ```

Key changes:

- `outputPath` points to Spring Boot's `static` folder.
- Rename `browser` to `main`.

□ Final Build & Run Instructions

□ Angular Build

From your Spring Boot root directory:

```
npm run build
```

 This builds Angular files and places them inside `target/classes/static`, which Spring Boot uses to serve static content.

Run Spring Boot App

In IntelliJ or terminal:

```
mvn spring-boot:run
```

Now:

- `http://localhost:8080/` → serves **Angular application**.
- `http://localhost:8080/api/**` → serves **Spring Boot REST APIs**.

Package and Deploy

To generate the Spring Boot JAR:

```
mvn clean install
```

Deploy the generated JAR file to your cloud or server platform. It will serve both the Angular frontend and Spring Boot backend from a single JAR and port.

Summary

Path	Description
`/`	Angular Frontend
`/api/**`	Spring Boot REST APIs
`target/classes/static`	Angular static assets after build

Notes

- You can automate Angular build before packaging Spring Boot with a Maven plugin if needed.
- `--watch` is useful during local development, but for production, use regular `ng build`.
- Use Angular environments (prod/dev) appropriately when building.

Would you like me to generate a ready-to-use `angular.json` snippet or Maven plugin configuration to include Angular build in `mvn install`?