Springboot-angular integration

Tuesday, January 7, 2020 4:41 PM

Directory of D:\devel\java\spring-boot-angular-integration\templateapp

```
01/07/2020  05:12 PM    <DIR>          .
01/07/2020  05:12 PM    <DIR>          ..
01/07/2020  02:08 PM    <DIR>          app-backend
01/07/2020  03:05 PM    <DIR>          app-frontend
01/06/2020  06:42 PM               680 pom.xml
               1 File(s)            680 bytes
               4 Dir(s)  348,082,884,608 bytes free
```

/templateapp/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project  … >
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.comp.codename</groupId>
    <artifactId>demo-parent</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>pom</packaging>
    <name>springboot-angular-integration</name>
    <description>Demo project for Spring Boot and
    angular</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <modules>
        <module>app-backend</module>
        <module>app-frontend</module>
    </modules>

</project>
```

Interesting Note:

- The Client application is packaged as webjar. The artifactId should always be lowercased. https://www.webjars.org/contributing
- Spring-boot is autoconfigured to handle webjars.
- Content/Resources within `META-INF/resources` will be served by spring-boot.
- Change the angular build output in `angular.json -> outputPath` By default app name appears in output path `dist/appname` Exa: `dist/client-app`. Just keep it `dist` so that `appname` does not leak into backend pom.xml
- Reference: https://shekhargulati.com/2017/11/08/a-minimalist-guide-to-building-spring-boot-angular-5-applications/

Alternative 1:

/templateapp/app-frontend/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project  … >
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.comp.codename</groupId>
    <artifactId>demo-frontend</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <name>demo-frontend</name>
    <description>Client app</description>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
            <plugin>
                <groupId>com.github.eirslett</groupId>
                <artifactId>frontend-maven-plugin</artifactId>
                <version>1.8.0</version>
                <configuration>
                    <nodeVersion>v12.14.0</nodeVersion>
                </configuration>
                <executions>
                    <execution>
                        <id>install-npm</id>
                        <goals>
                            <goal>install-node-and-npm</goal>
                        </goals>
                    </execution>
                    <execution>
                        <id>npm-install</id>
                        <goals>
                            <goal>npm</goal>
                        </goals>
                    </execution>
                    <execution>
                        <id>npm-build</id>
                        <goals>
                            <goal>npm</goal>
                        </goals>
                        <configuration>
                            <arguments>run-script build</arguments>
                        </configuration>
                    </execution>
                    <execution>
                        <id>npm-test</id>
                        <goals>
                            <goal>npm</goal>
                        </goals>
                        <configuration>
                            <arguments>run-script e2e</arguments>
                        </configuration>
                        <phase>test</phase>
                    </execution>
                </executions>
            </plugin>
            <plugin>
                <artifactId>maven-resources-plugin</artifactId>
                <executions>
                    <execution>
                        <id>copy-build</id>
                        <phase>prepare-package</phase>
                        <goals>
                            <goal>copy-resources</goal>
                        </goals>
```

so that appName does not leak into backend pom.xml

- Reference: https://shekhargulati.com/2017/11/08/a-minimalist-guide-to-building-spring-boot-angular-5-applications/

Alternative 1:

- Use maven modules to organize frontend and backend as is. Instead of packaging client as webjar, copy them to server's `src/main/resources` or `static` folder. It will be served by spring-boot automatically.
- Use the `angular.json -> outputPath` to `"outDir": "../AppName-Service/src/main/resources/static/",`

Alternative 2:

- Create spring-boot and angular app in same folder
- The src folder will have angular as well main/java directory.
- Send angular output to `src/main/resources` or `static` folder using `angular.json -> outputPath`

Note:
Ensure angular is using the node and node_modules managed by maven frontend plugin.

How?
At the root of frontend application.

```
$ cat > npm
#!/bin/sh
cd $(dirname $0)
PATH="$PWD/node/":$PATH
node "node/node_modules/npm/bin/npm-cli.js" "$@"
$ chmod +x npm

./npm install @angular/cli

$ cat > ng
#!/bin/sh
cd $(dirname $0)
PATH="$PWD/node/":"$PWD":$PATH
node_modules/@angular/cli/bin/ng "$@"
$ chmod +x ng

$ ./ng --version
$ ./ng e2e
```

From <https://github.com/dsyer/spring-boot-angular>

```
                        <id>copy-build</id>
                        <phase>prepare-package</phase>
                        <goals>
                            <goal>copy-resources</goal>
                        </goals>
                        <configuration>
                            <!-- Webjars expect resources in /META-INF/resources
 folder in a jar file. -->
                            <outputDirectory>${basedir}/target/classes/META-
INF/resources</outputDirectory>
                            <resources>
                                <resource>
                                    <directory>dist/client-app</directory>
                                </resource>
                            </resources>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
</project>
```

/templateapp/next-backend/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project …>
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.2.2.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.comp.codename</groupId>
    <artifactId>app-backend</artifactId>
    <version>1.0.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <name>demo-server</name>
    <description>Demo server project</description>

    <properties>
        <java.version>1.8</java.version>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>com.comp.codename</groupId>
            <artifactId>demo-frontend</artifactId>
            <version>1.0.0-SNAPSHOT</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.springframework.security</groupId>
            <artifactId>spring-security-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```