

# Spring Boot: Run from Command-Line



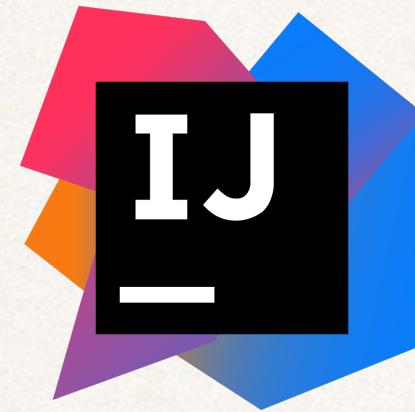
# Running from the Command-Line

# Running from the Command-Line

- During development we spend most of our time in the IDE

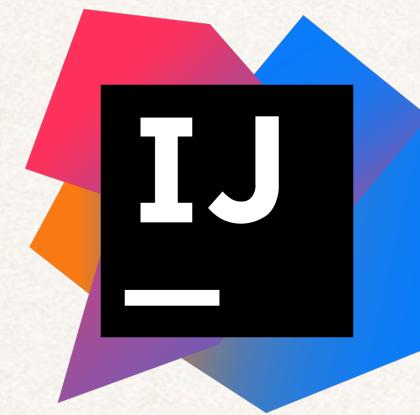
# Running from the Command-Line

- During development we spend most of our time in the IDE



# Running from the Command-Line

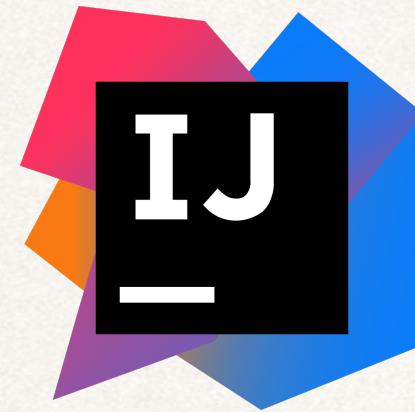
- During development we spend most of our time in the IDE



- However, we may want to run our Spring Boot app outside of the IDE

# Running from the Command-Line

- During development we spend most of our time in the IDE



- However, we may want to run our Spring Boot app outside of the IDE
- One approach is running from the command-line

# Running from the Command-Line

# Running from the Command-Line

- When running from the command-line

# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open/running

# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open/running



# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open/running

# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open/running
- Since we using Spring Boot, the server is embedded in our JAR file

# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open / running
- Since we using Spring Boot, the server is embedded in our JAR file
  - No need to have separate server installed / running

# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open / running
- Since we using Spring Boot, the server is embedded in our JAR file
  - No need to have separate server installed / running



# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open / running
- Since we using Spring Boot, the server is embedded in our JAR file
  - No need to have separate server installed / running

# Running from the Command-Line

- When running from the command-line
  - No need to have IDE open / running
- Since we using Spring Boot, the server is embedded in our JAR file
  - No need to have separate server installed / running
- Spring Boot apps are self-contained



# Running from the Command-Line

# Running from the Command-Line

- Spring Boot apps are self-contained

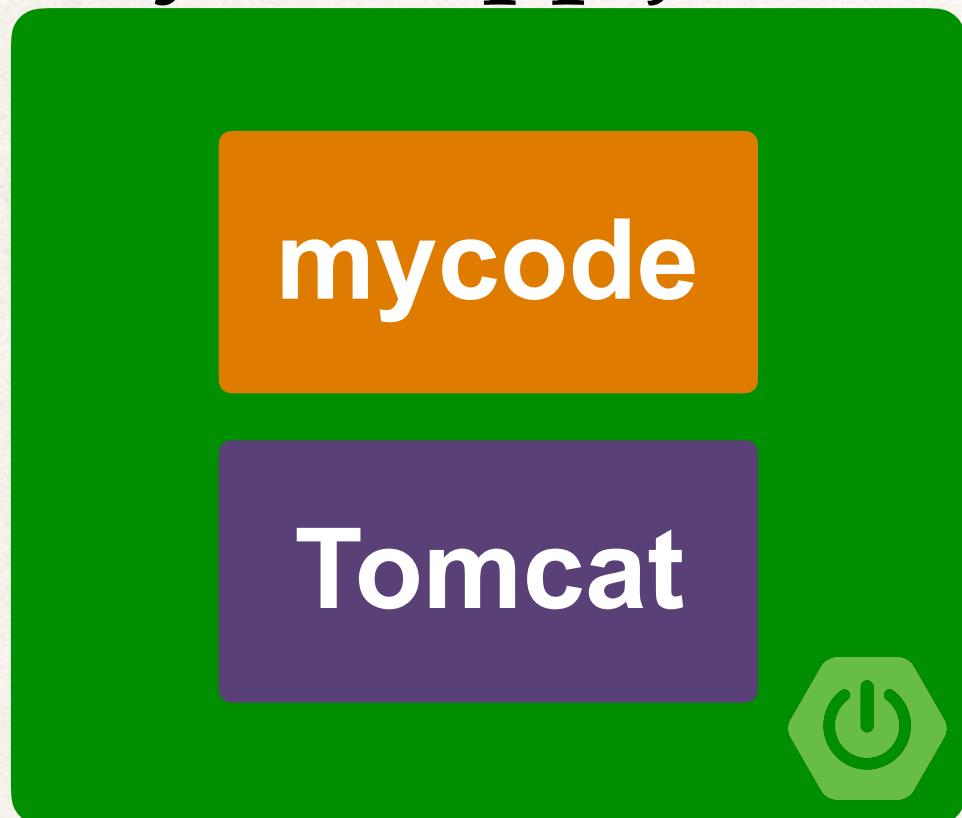


# Running from the Command-Line

- Spring Boot apps are self-contained



**mycoolapp.jar**

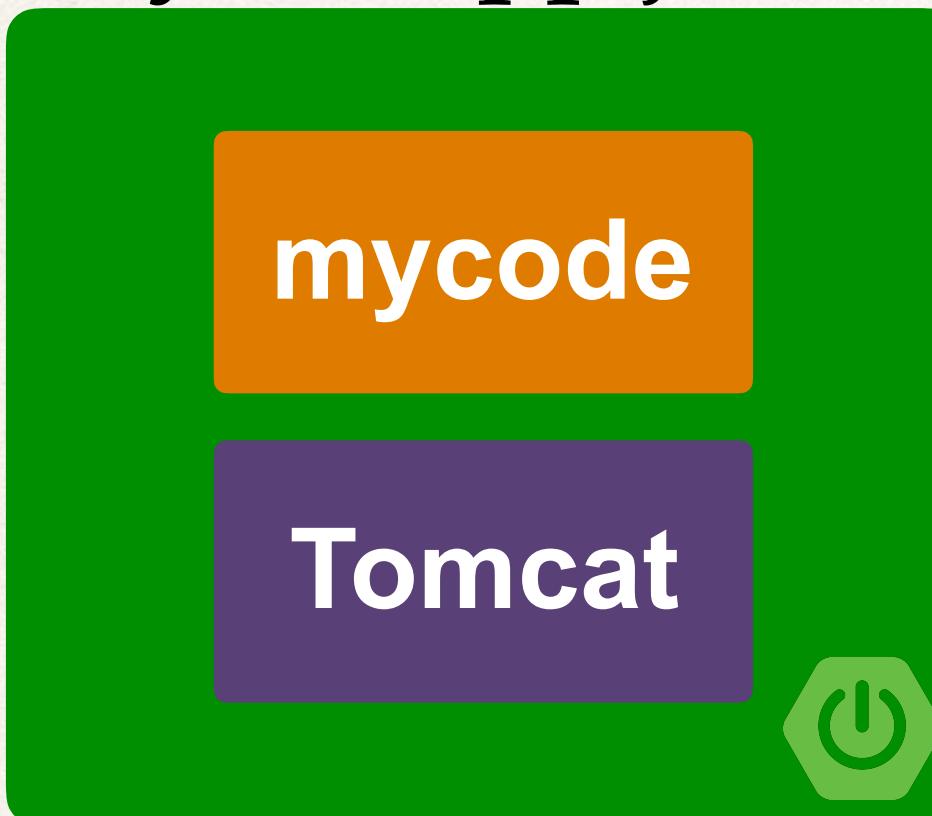


# Running from the Command-Line

- Spring Boot apps are self-contained



**mycoolapp.jar**



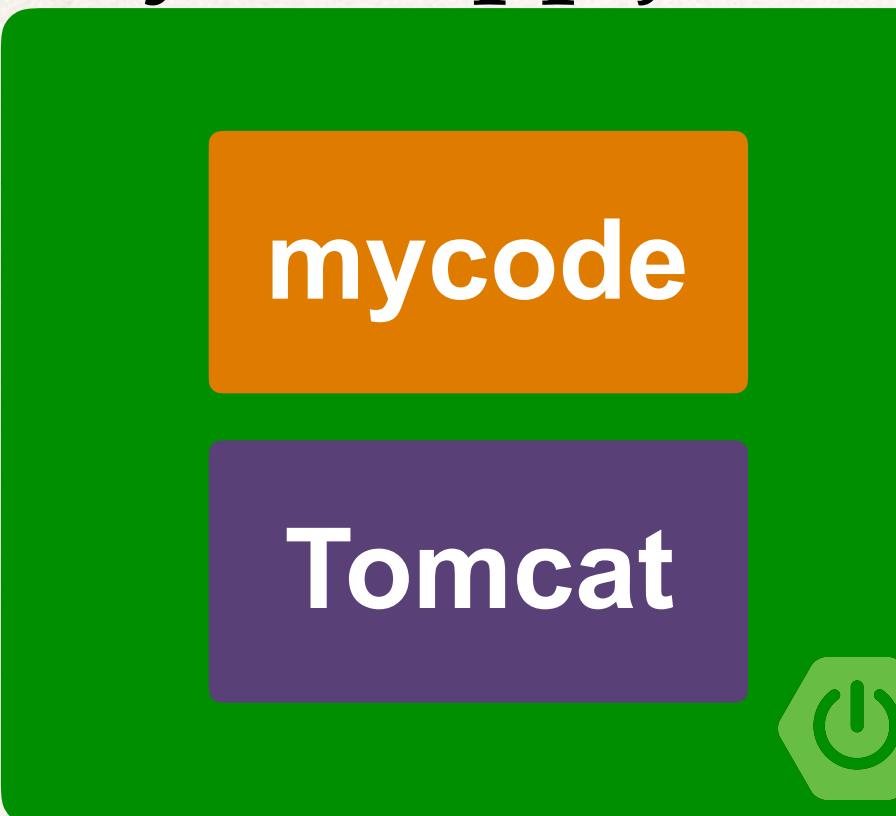
JAR file  
includes your application code  
AND  
includes the server

# Running from the Command-Line

- Spring Boot apps are self-contained



**mycoolapp.jar**



**Self-contained unit  
Nothing else to install**

**JAR file  
includes your application code  
AND  
includes the server**

# Running from the Command-Line

# Running from the Command-Line

- Two options for running the app

# Running from the Command-Line

- Two options for running the app
- Option 1: Use **java -jar**

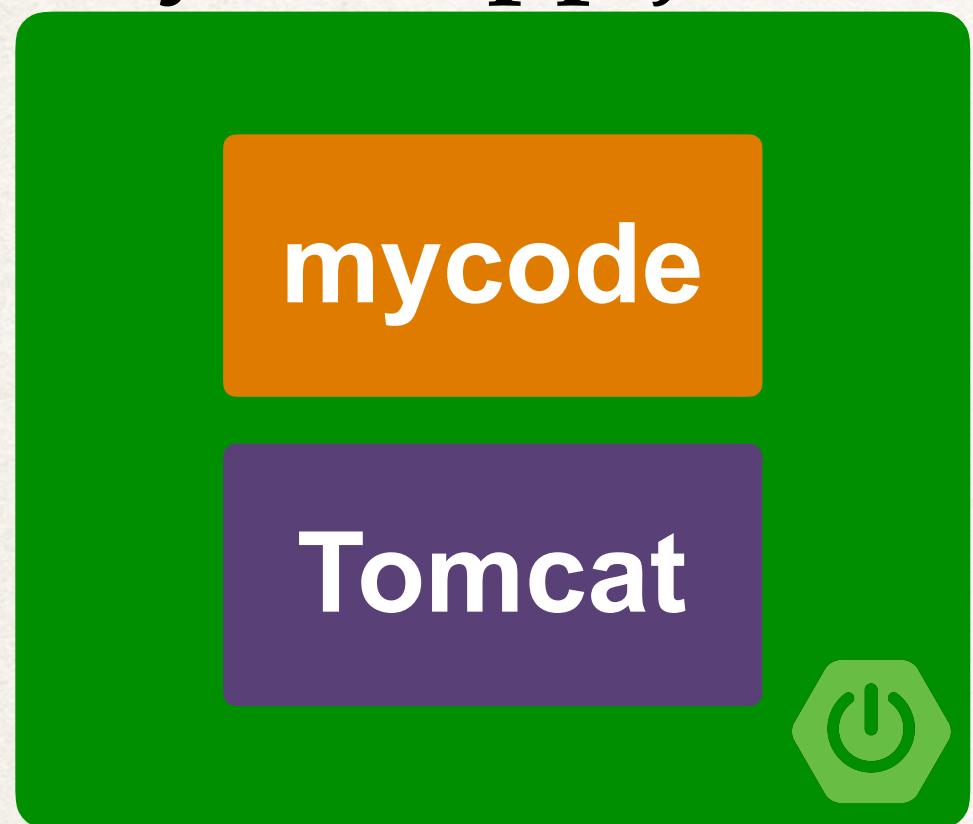
# Running from the Command-Line

- Two options for running the app
- Option 1: Use **java -jar**
- Option 2: Use Spring Boot Maven plugin
  - **mvnw spring-boot:run**

# Option 1: Use `java -jar`

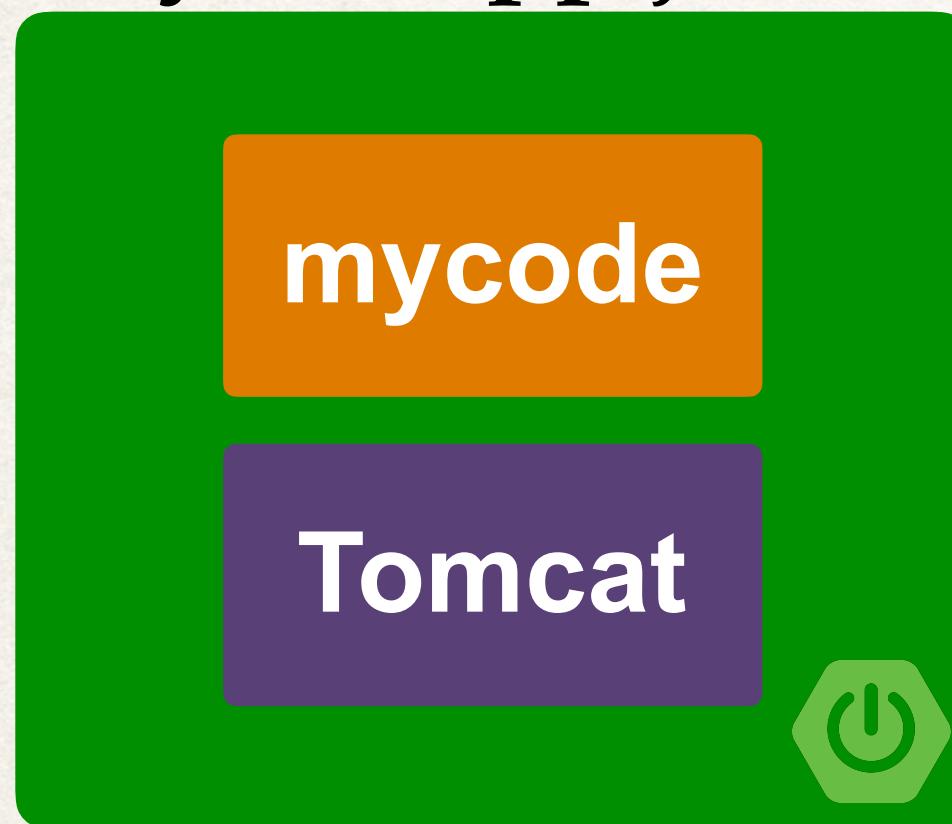
# Option 1: Use `java -jar`

`mycoolapp.jar`



# Option 1: Use `java -jar`

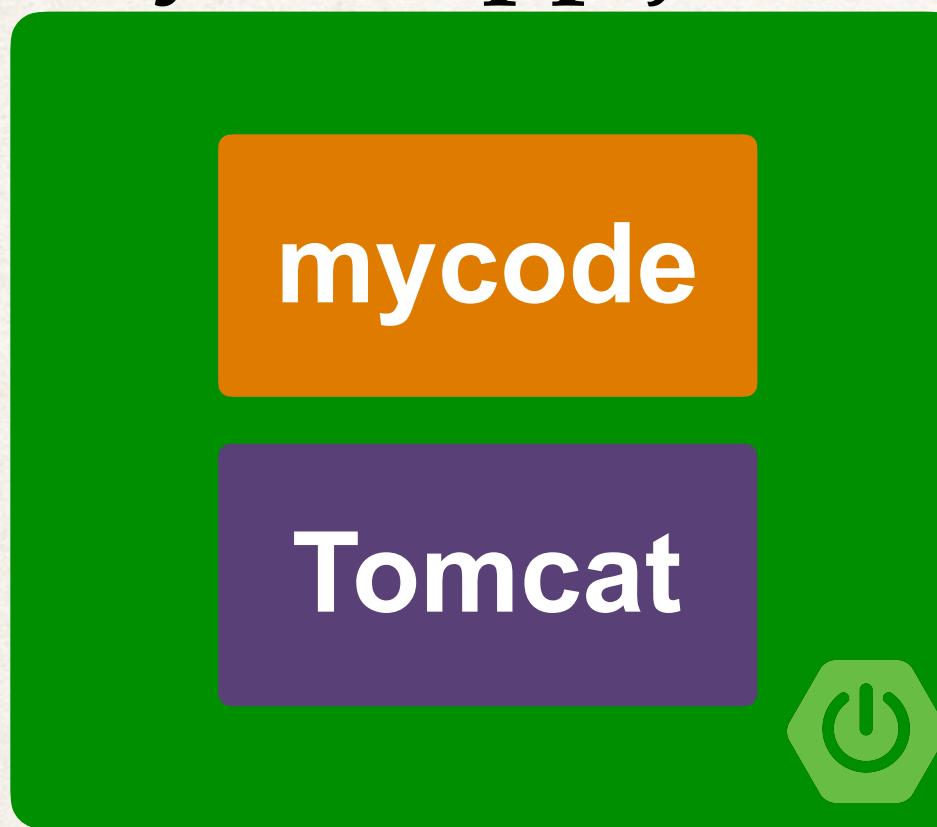
`mycoolapp.jar`



```
> java -jar mycoolapp.jar
```

# Option 1: Use `java -jar`

`mycoolapp.jar`

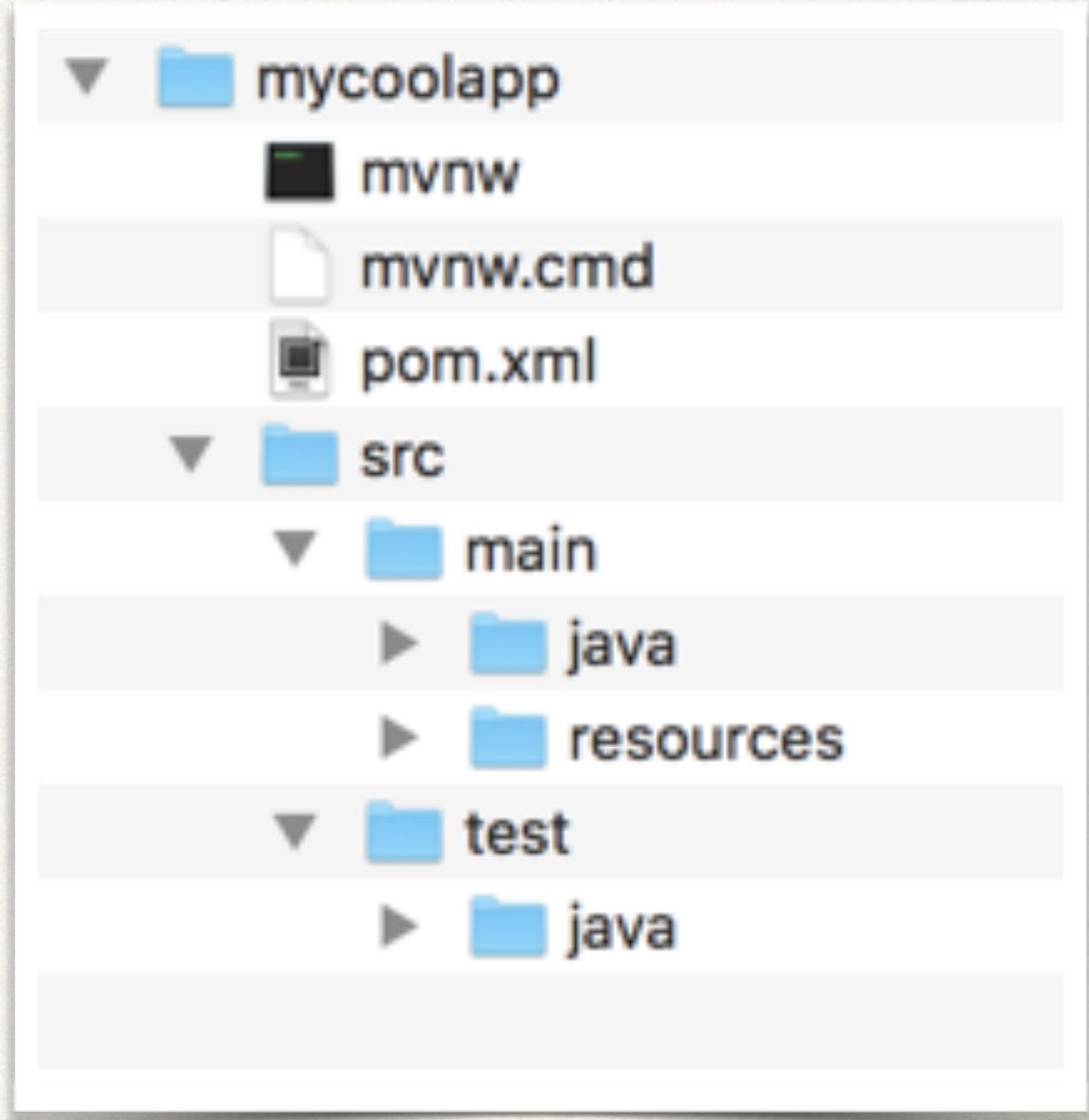


```
> java -jar mycoolapp.jar
```

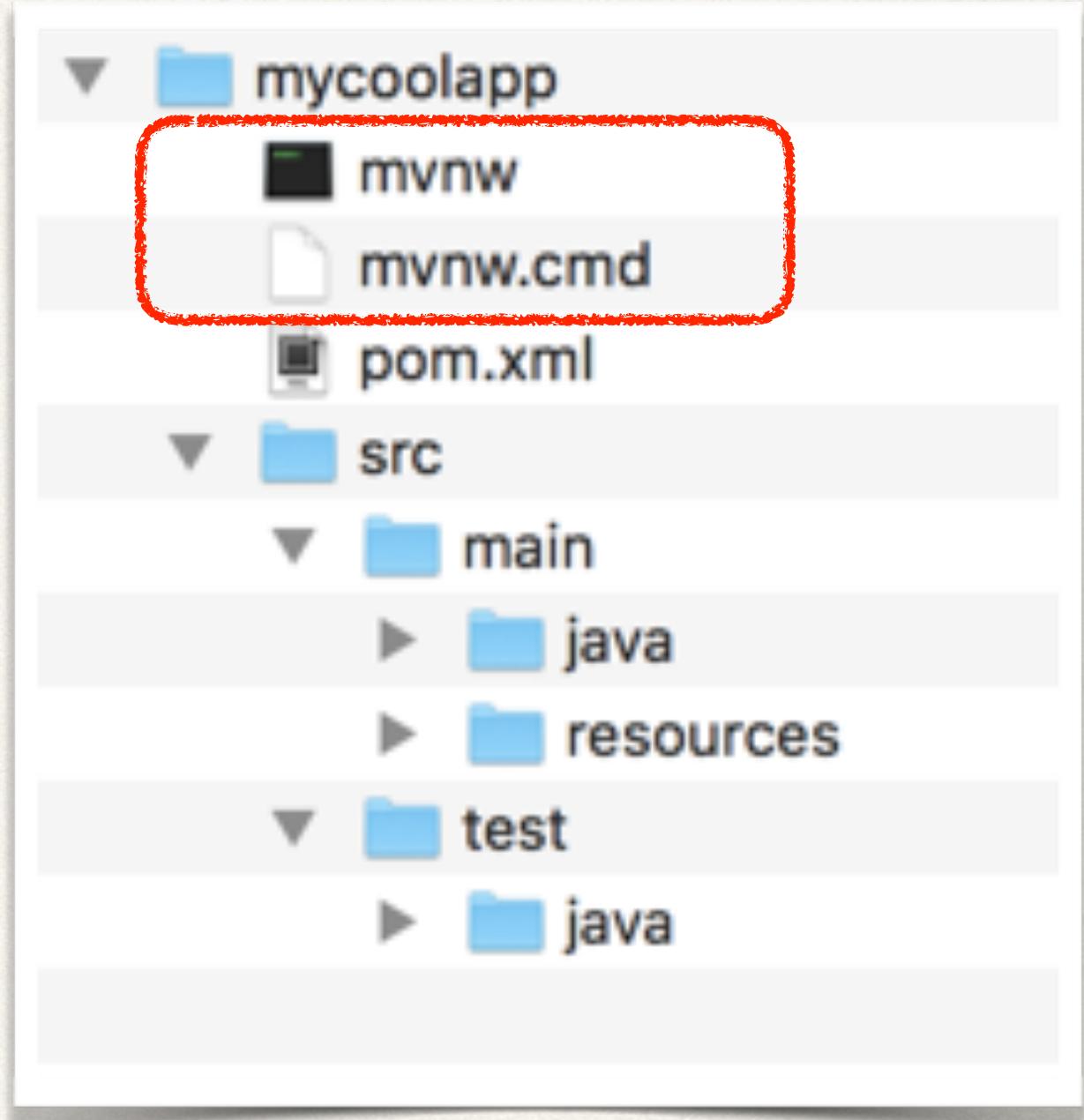
Name of our JAR file

# Option 2: Use Spring Boot Maven plugin

# Option 2: Use Spring Boot Maven plugin

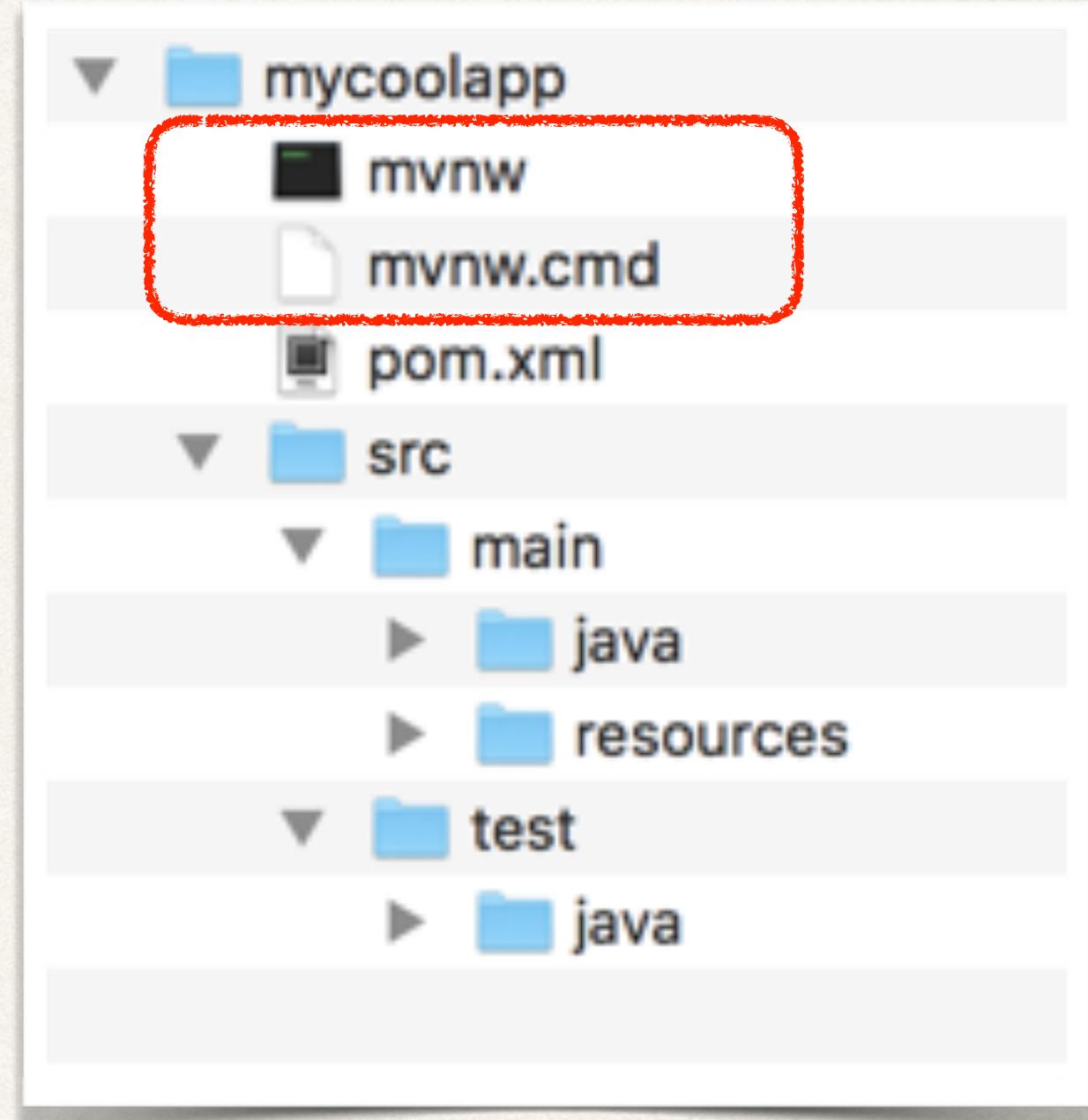


# Option 2: Use Spring Boot Maven plugin



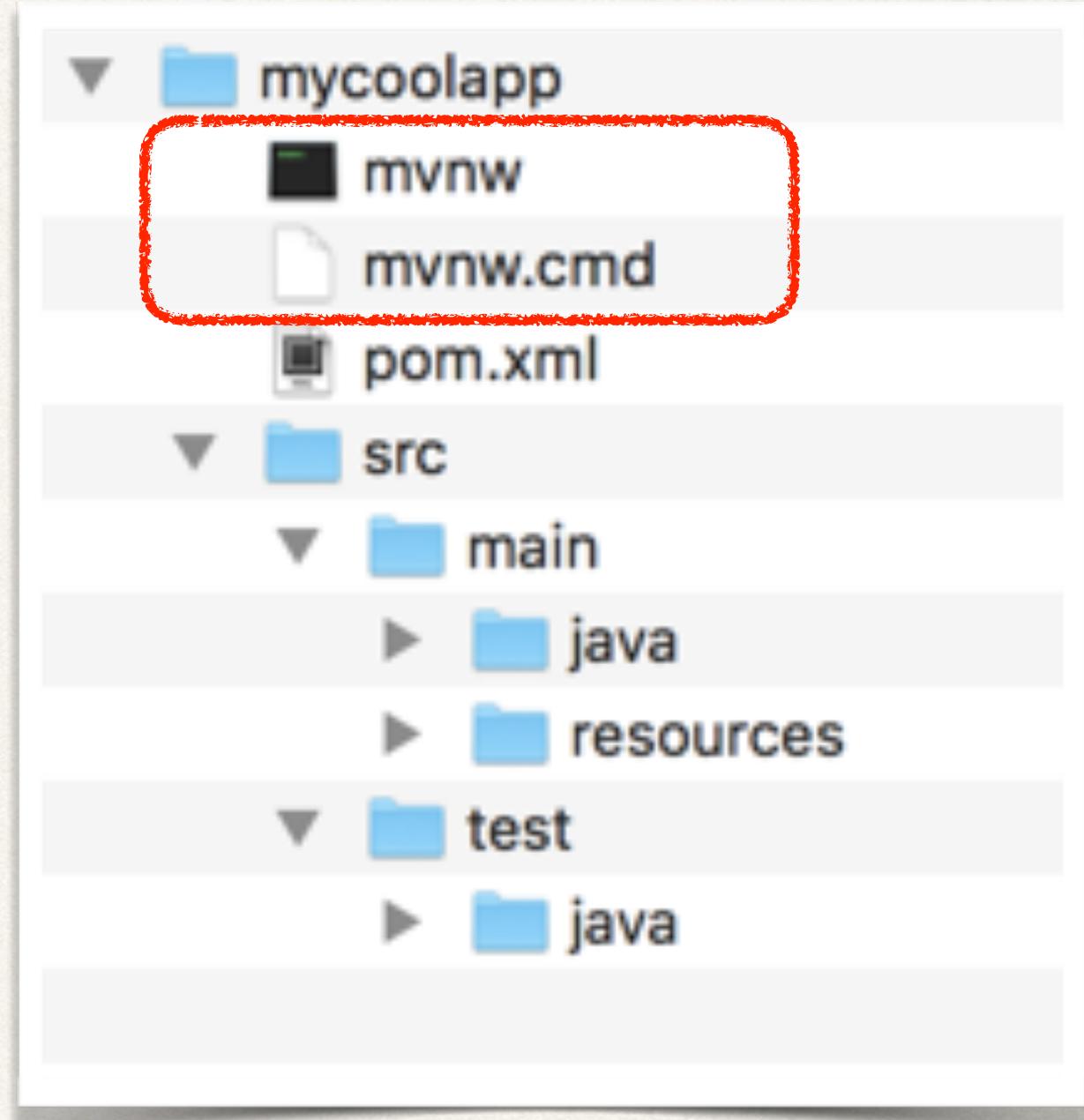
# Option 2: Use Spring Boot Maven plugin

- **mvnw** allows you to run a Maven project



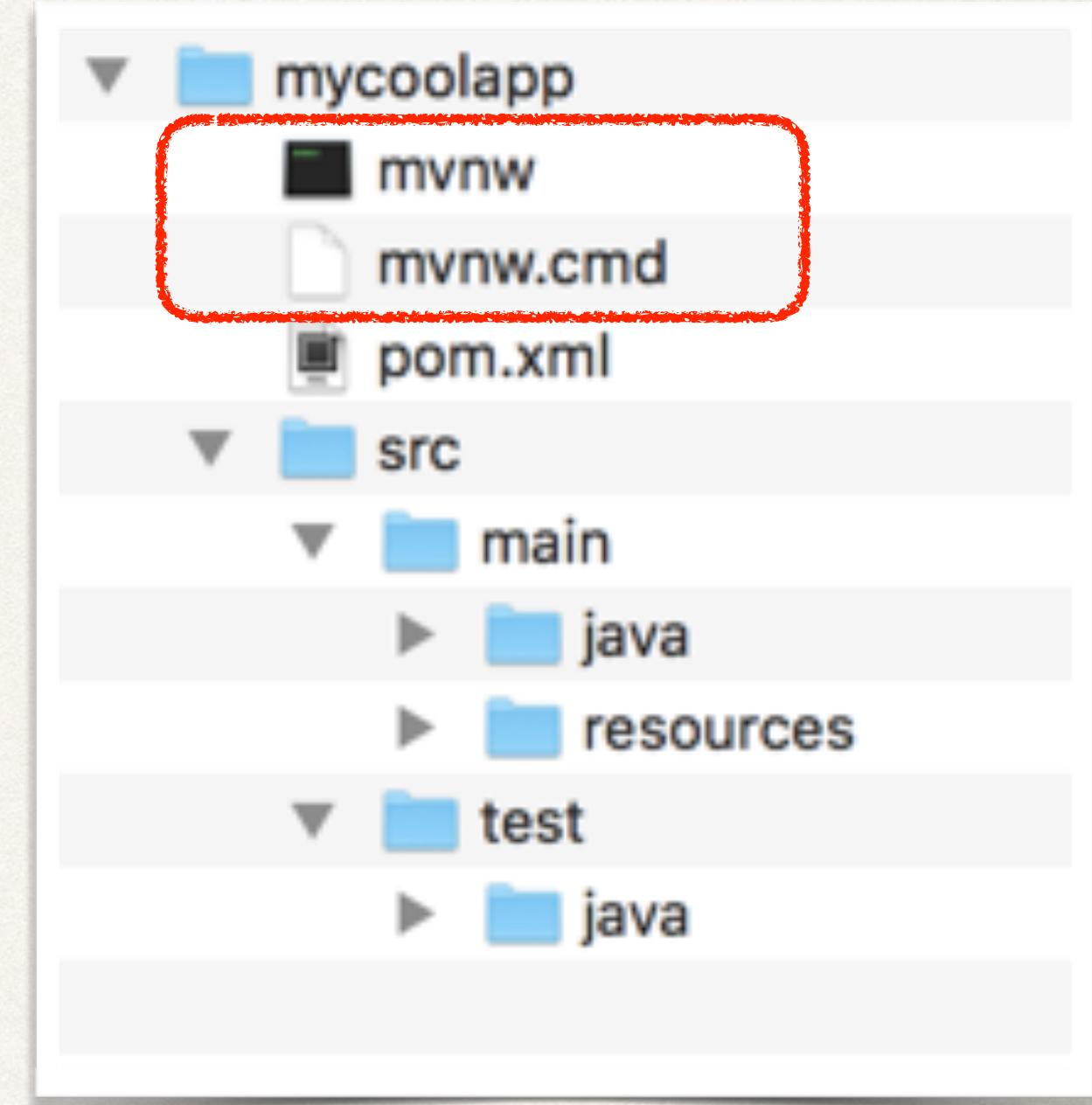
# Option 2: Use Spring Boot Maven plugin

- **mvnw** allows you to run a Maven project
  - No need to have Maven installed or present on your path



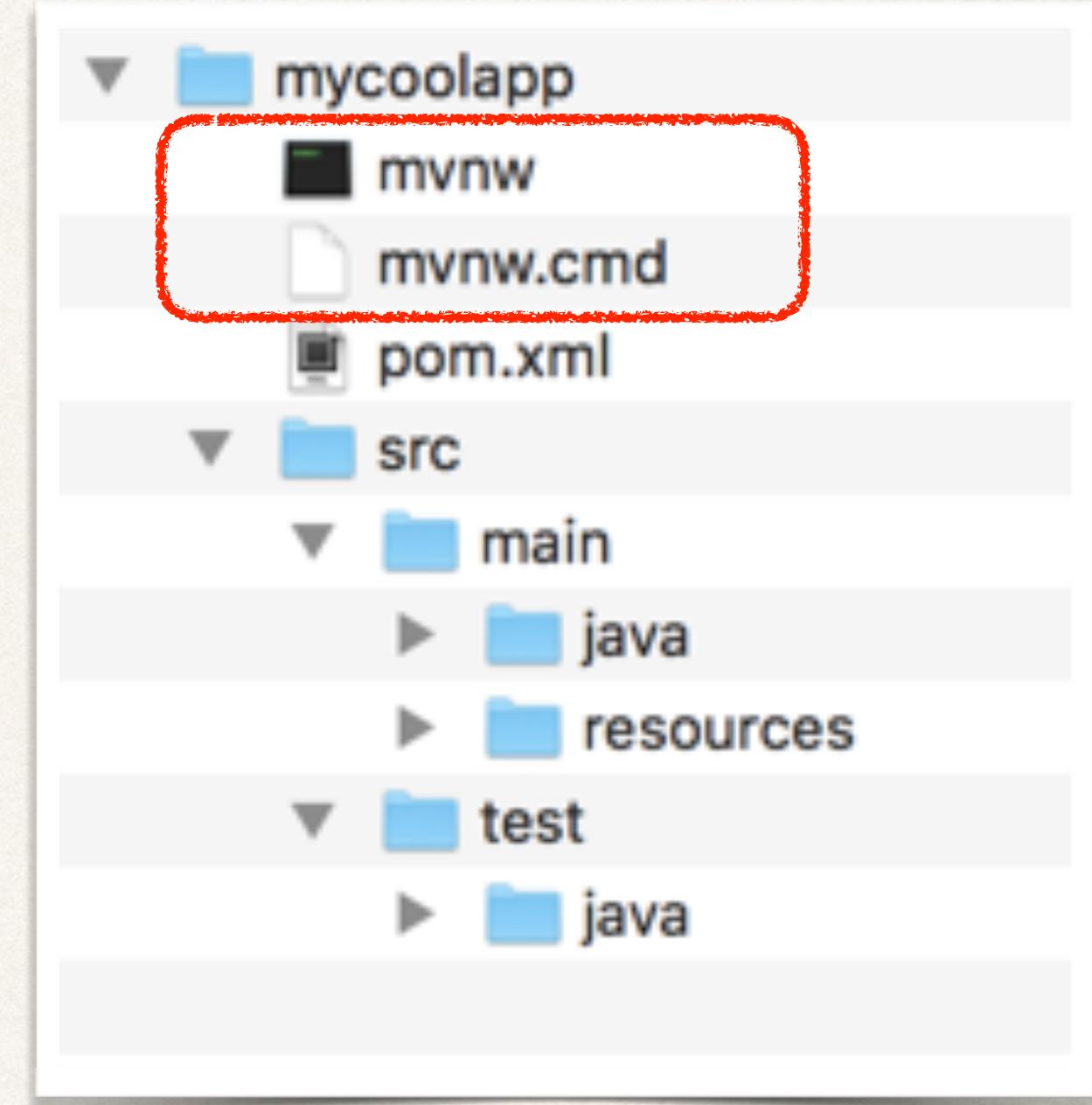
# Option 2: Use Spring Boot Maven plugin

- **mvnw** allows you to run a Maven project
  - No need to have Maven installed or present on your path
  - If correct version of Maven is NOT found on your computer



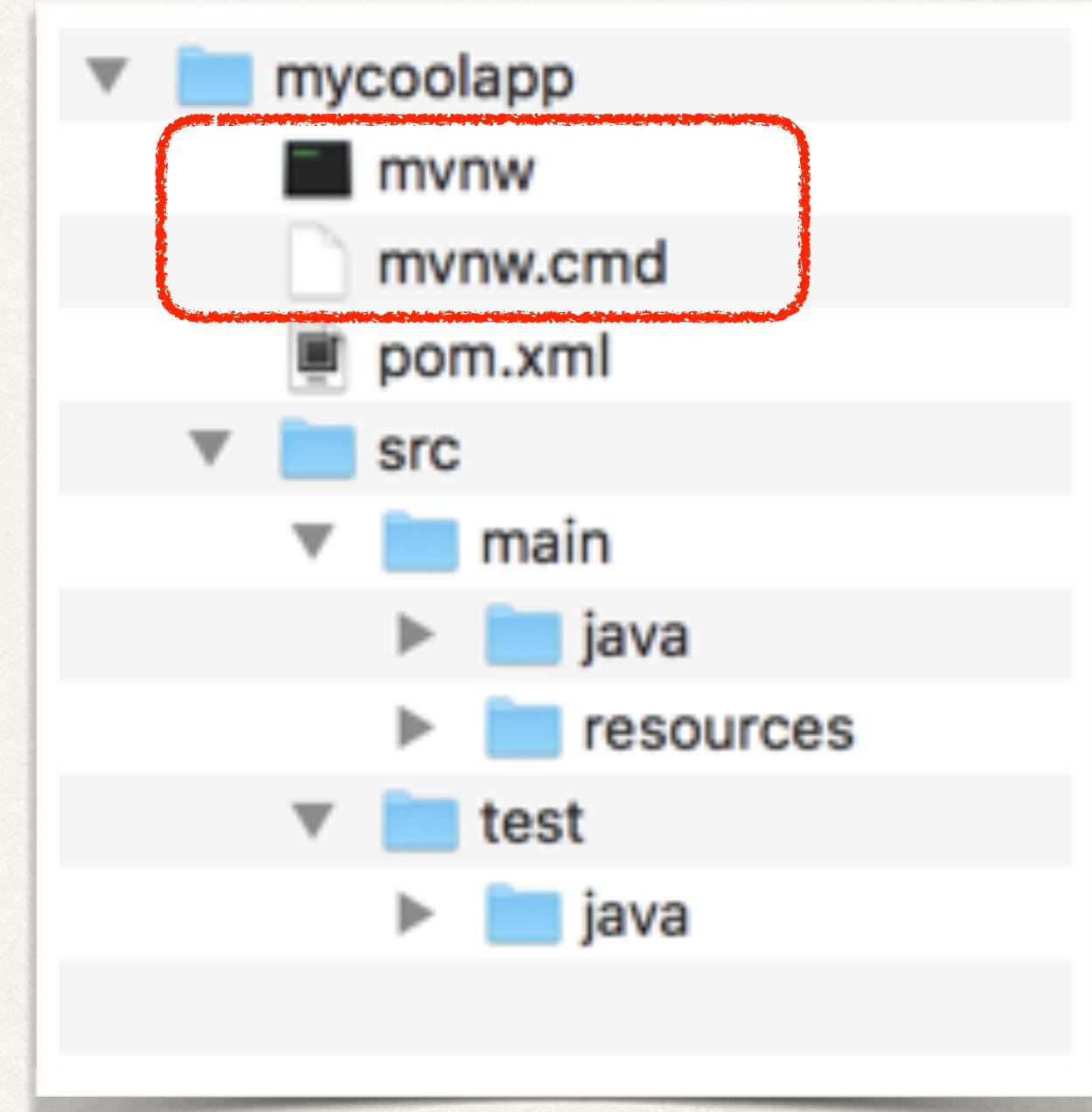
# Option 2: Use Spring Boot Maven plugin

- **mvnw** allows you to run a Maven project
  - No need to have Maven installed or present on your path
  - If correct version of Maven is NOT found on your computer
    - **Automatically downloads** correct version and runs Maven



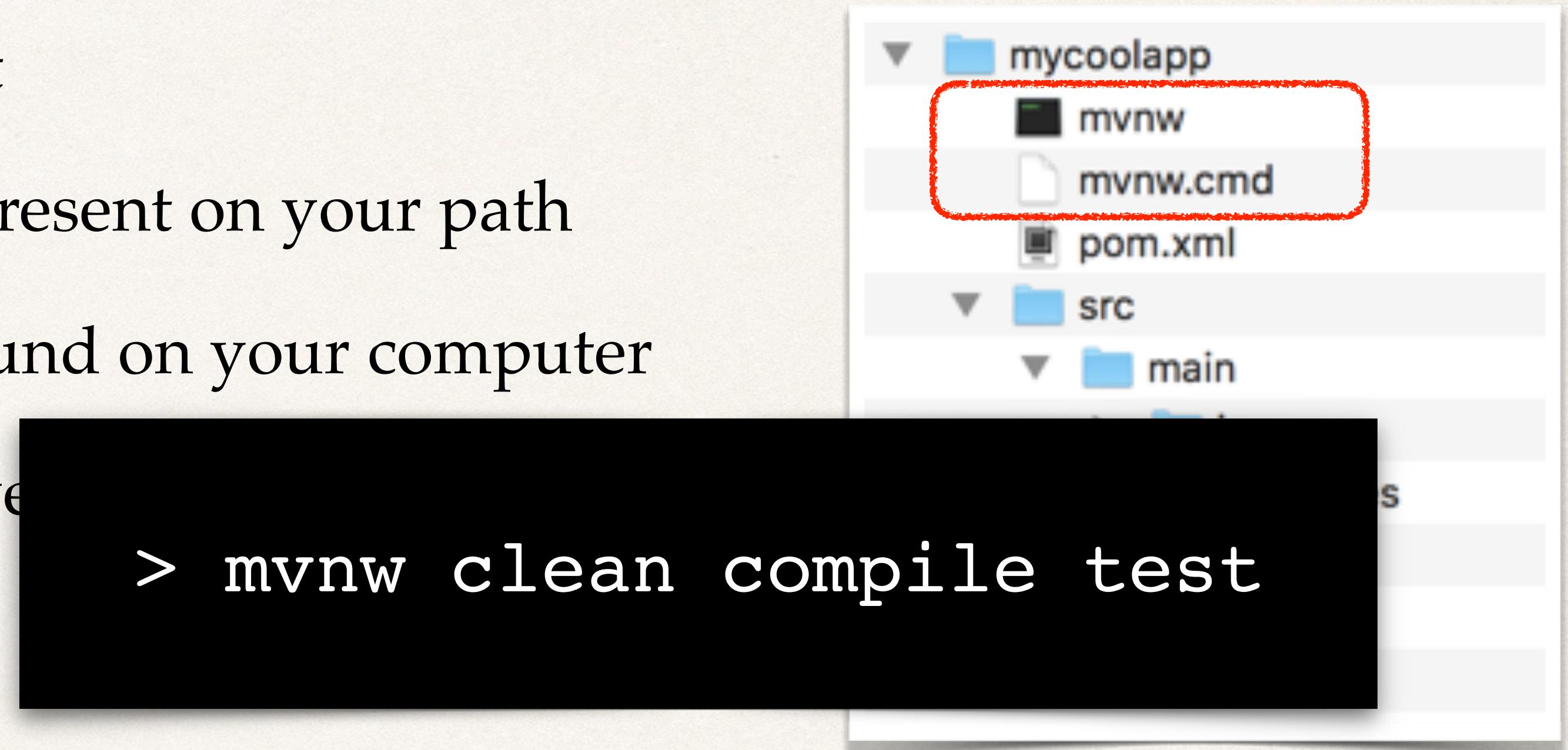
# Option 2: Use Spring Boot Maven plugin

- **mvnw** allows you to run a Maven project
  - No need to have Maven installed or present on your path
  - If correct version of Maven is NOT found on your computer
    - **Automatically downloads** correct version and runs Maven
- Two files are provided



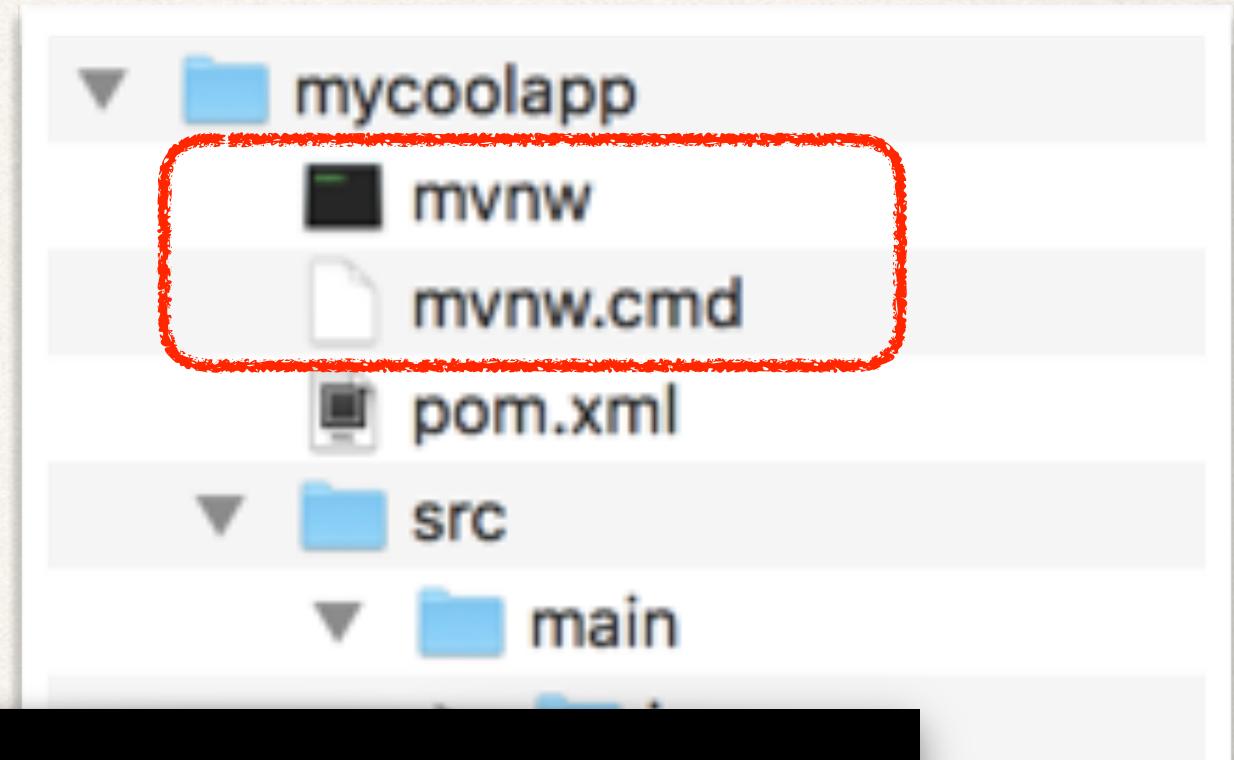
# Option 2: Use Spring Boot Maven plugin

- **mvnw** allows you to run a Maven project
  - No need to have Maven installed or present on your path
  - If correct version of Maven is NOT found on your computer
    - **Automatically downloads** correct version and runs Maven
- Two files are provided
  - **mvnw.cmd** for MS Windows



# Option 2: Use Spring Boot Maven plugin

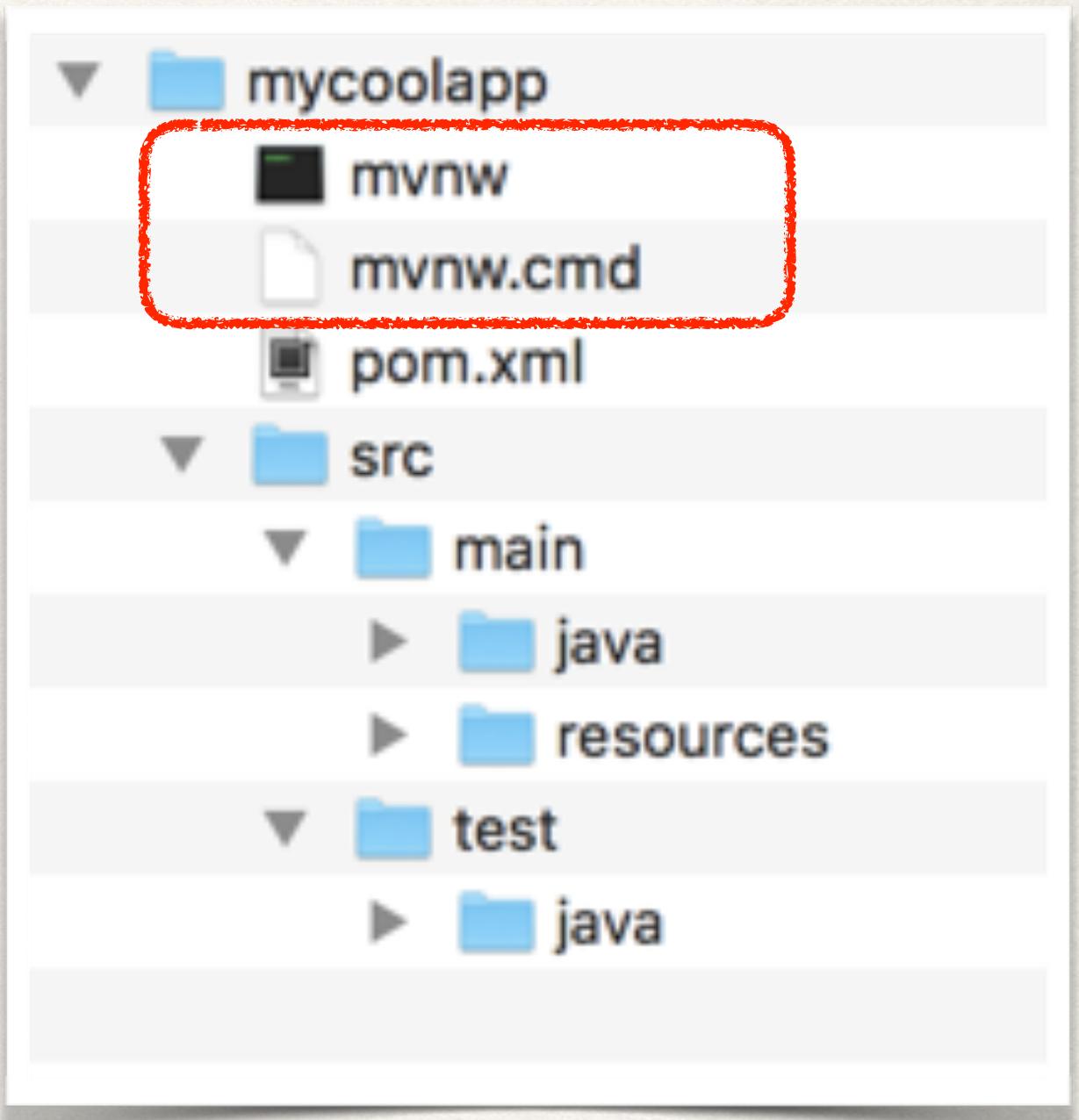
- **mvnw** allows you to run a Maven project
  - No need to have Maven installed or present on your path
  - If correct version of Maven is NOT found on your computer
    - **Automatically downloads** correct version of Maven and runs Maven
- Two files are provided
  - **mvnw.cmd** for MS Windows
  - **mvnw.sh** for Linux/Mac



```
> mvnw clean compile test
```

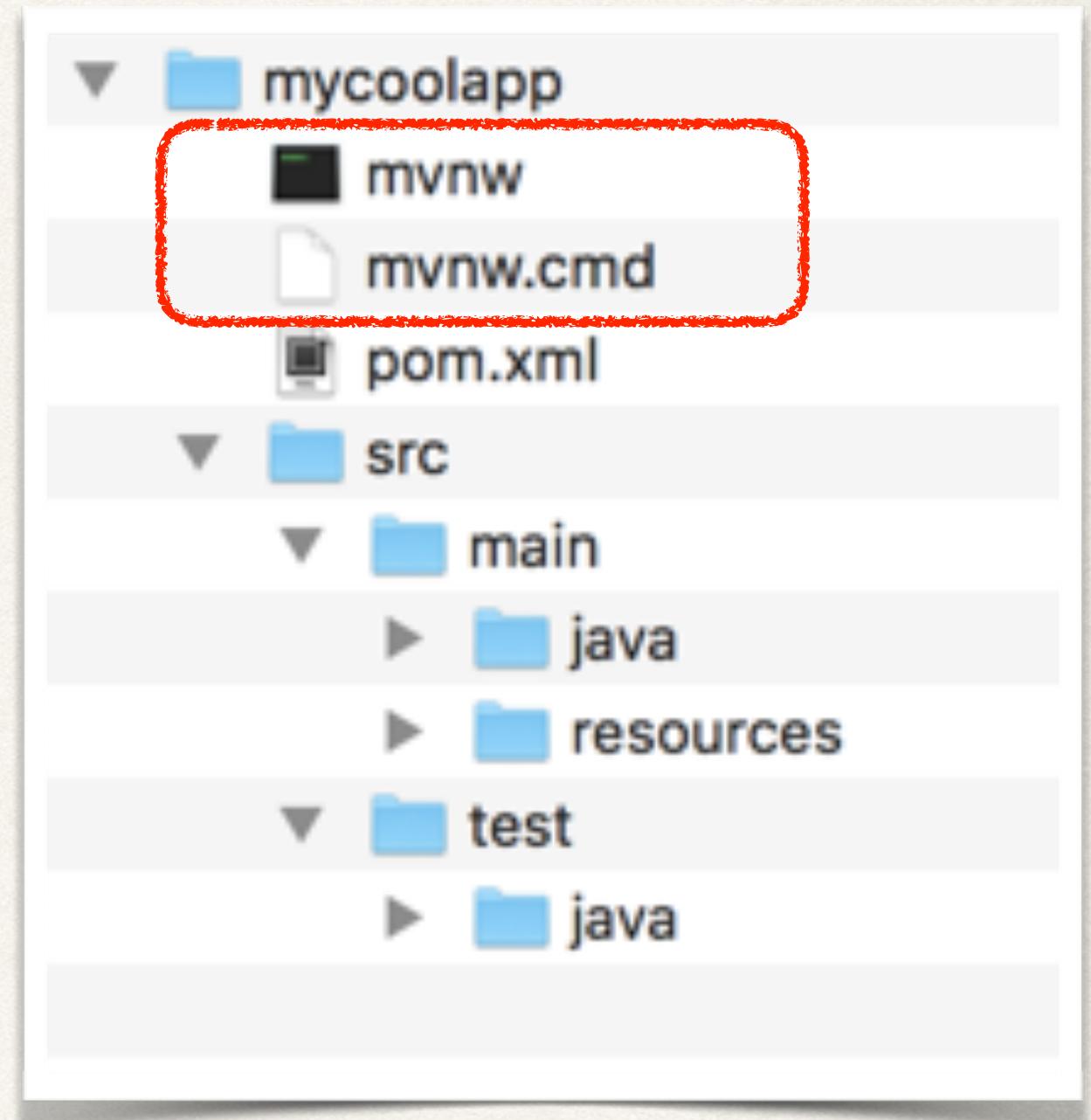


# Maven Wrapper files



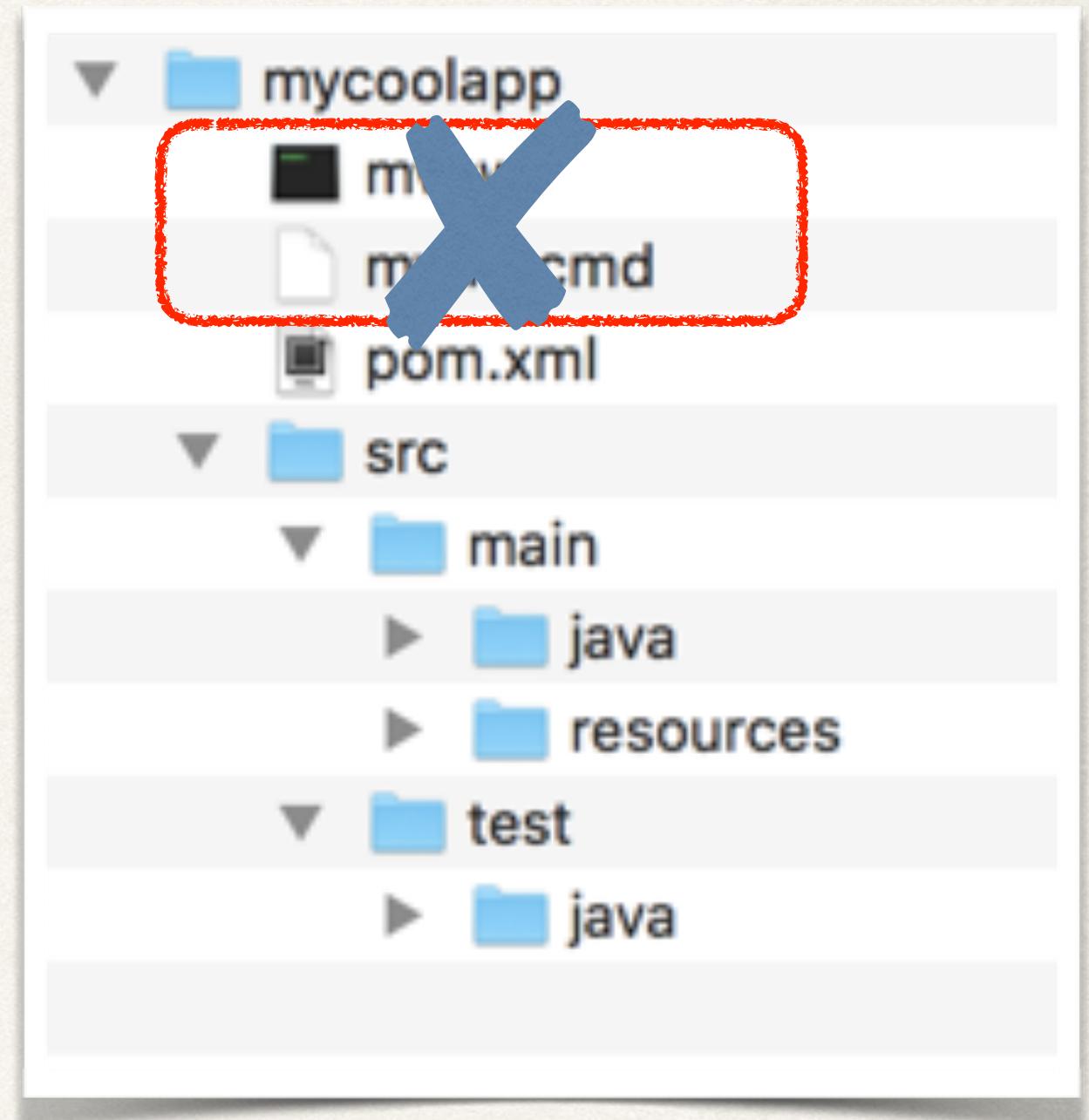
# Maven Wrapper files

- If you already have Maven installed previously



# Maven Wrapper files

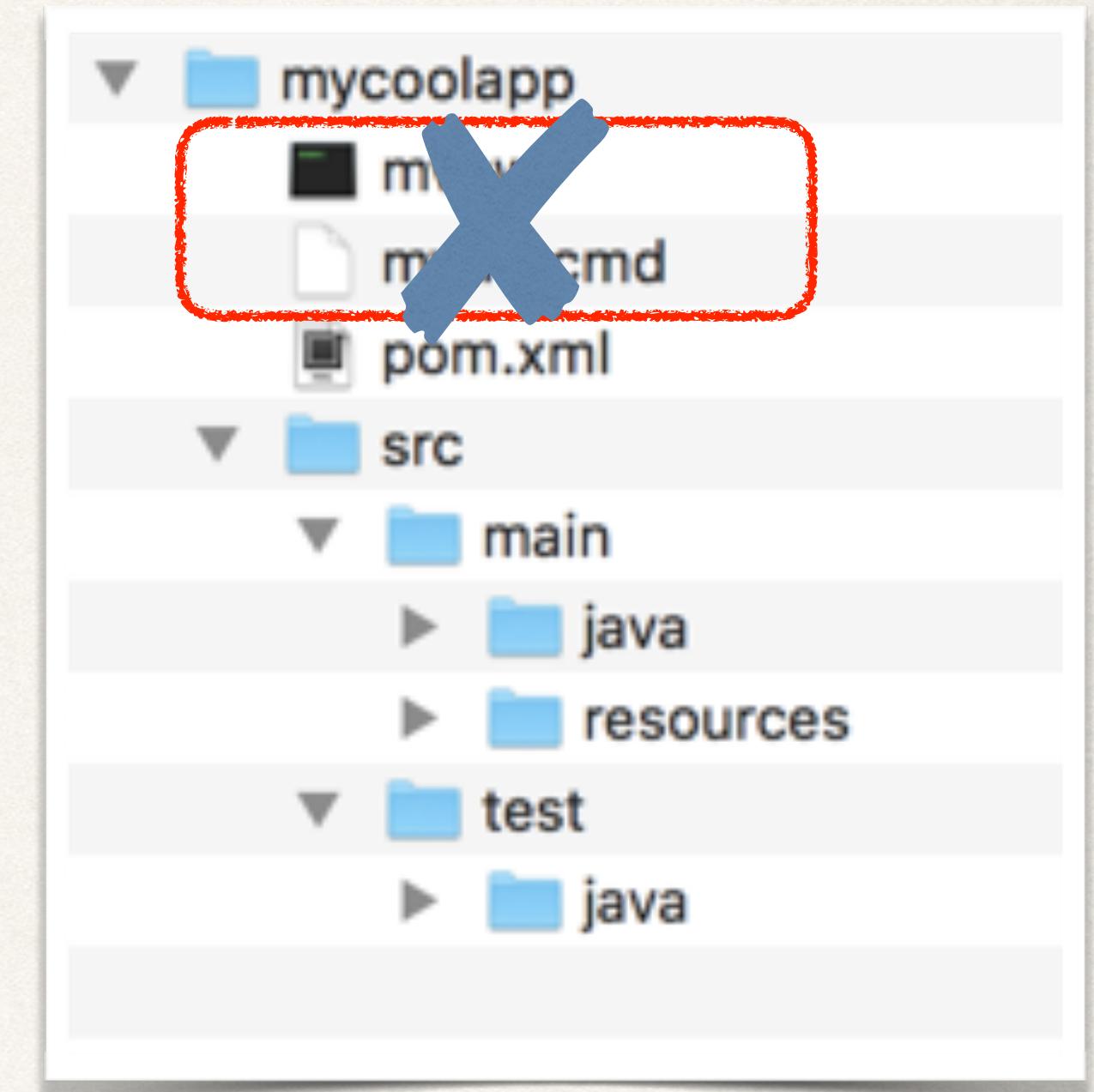
- If you already have Maven installed previously
  - Then you can ignore/delete the **mvnw** files



# Maven Wrapper files

- If you already have Maven installed previously
  - Then you can ignore/delete the **mvnw** files
- Just use Maven as you normally would

```
$ mvn clean compile test
```



# Option 2: Use Spring Boot Maven plugin

# Option 2: Use Spring Boot Maven plugin

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

# Option 2: Use Spring Boot Maven plugin

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

To package executable jar  
or war archive

Can also easily run the app

# Option 2: Use Spring Boot Maven plugin

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

To package executable jar  
or war archive

Can also easily run the app

```
$ ./mvnw package
```

```
$ ./mvnw spring-boot:run
```

# Option 2: Use Spring Boot Maven plugin

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

To package executable jar  
or war archive

Can also easily run the app

```
$ ./mvnw package
```

```
$ ./mvnw spring-boot:run
```

Can also just use:

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

# Development Process

Step-By-Step

# Development Process

Step-By-Step

1. Exit the IDE

# Development Process

Step-By-Step

1. Exit the IDE
2. Package the app using **mvnw package**

# Development Process

Step-By-Step

1. Exit the IDE
2. Package the app using **mvnw package**
3. Run app using **java -jar**

# Development Process

Step-By-Step

1. Exit the IDE
2. Package the app using **mvnw package**
3. Run app using **java -jar**
4. Run app using Spring Boot Maven plugin, **mvnw spring-boot:run**