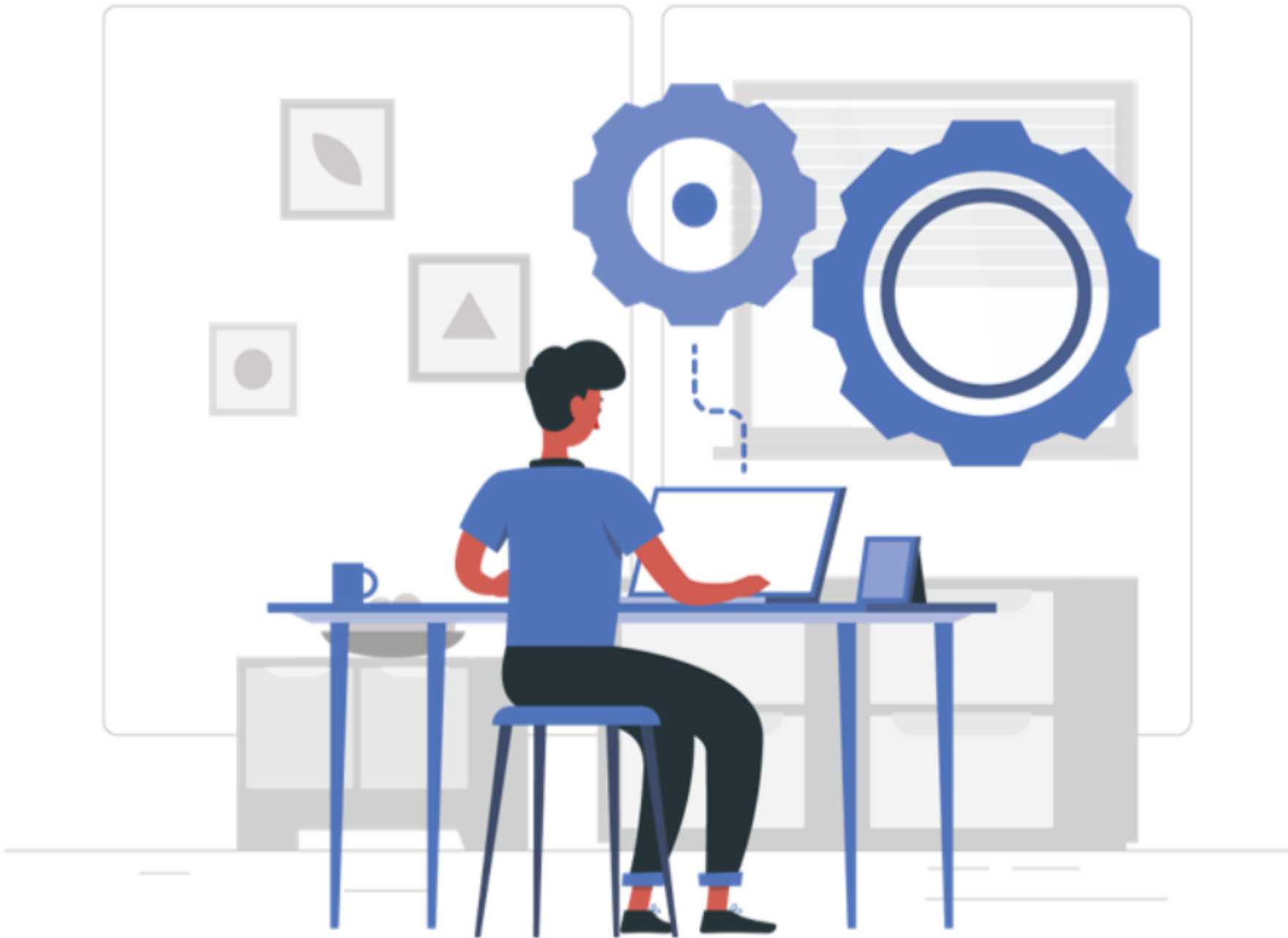


Challenge

Build the Skeleton of Spring Boot Application



Spring Boot With Service

ABC Corp wants a web application that can manage stock details; however, the developers' team faces deadline issues as a lot of its time is spent adding dependencies and managing configuration files instead of focusing on the end objective.

Developer Bob has been Developer Bob has been tasked with implementing the objectives above.

Help Bob select the correct technology and develop the back-end application.

Start with the first stage to help Jim create a simple Spring Boot application using Spring Initializr, and then run the application using the IntelliJ IDE.

CHALLENGE



Implementation Environment



The image shows the Spring Initializr web interface. It has a header with the 'spring initializr' logo. Below the header, there are three main sections: 'Project', 'Language', and 'Dependencies'. The 'Project' section has radio buttons for 'Maven Project' (selected) and 'Gradle Project'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Dependencies' section has a button 'ADD DEPENDENCIES... CTRL + B' and the text 'No dependency selected'. Below these sections is the 'Spring Boot' section with radio buttons for versions: '3.0.0 (SNAPSHOT)', '3.0.0 (M5)', '2.7.5 (SNAPSHOT)', '2.7.4' (selected), '2.6.13 (SNAPSHOT)', and '2.6.12'. At the bottom is the 'Project Metadata' section with three text input fields: 'Group' (containing 'com.example'), 'Artifact' (containing 'demo'), and 'Name' (containing 'demo'). At the very bottom are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'. The entire interface is enclosed in a light blue border.

- Create a Spring Boot application from the [Spring Initializr](#).
- Select Maven as the project and enter all project metadata details, such as group and artifact name.
- The packaging should be JAR and Java version 11.
- Click on the Generate button.
- Extract the project into your local machine.

Challenge: Tasks

- Export the project in your IntelliJ IDE.
- Within the src/main/java directory, create a package and name it com.jap.demo
- Create a package and name it service inside the root package.
- Within the service package, create a MessageService class.
- Annotate the class with the @Service annotation.
- Within the class, create a method that will return a string object, "Good day to all !!!"

Challenge: Tasks (cont'd)

- Within the main method create the object called ApplicationContext.
- Call the getBean() method of ApplicationContext and get the MessageService bean object.
- Call the method of the MessageService class.
- Run the boot application by using the Spring execution: mvn spring-boot:run
- Sample Output:

```
Hello World
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.558 s
[INFO] Finished at: 2022-09-05T13:46:29+05:30
[INFO] -----
```

Submission Instructions

- There is no boilerplate for the practice.
- Create a Git repository named **BEJ_C1_S4_Spring_Boot_MC_1**
- After completing the challenge, push the code back to git using the below commands.

```
git init
git remote add origin <url>
git add .
git commit -m "comments on the push"
git push -u origin master
```

- Submit it for review.