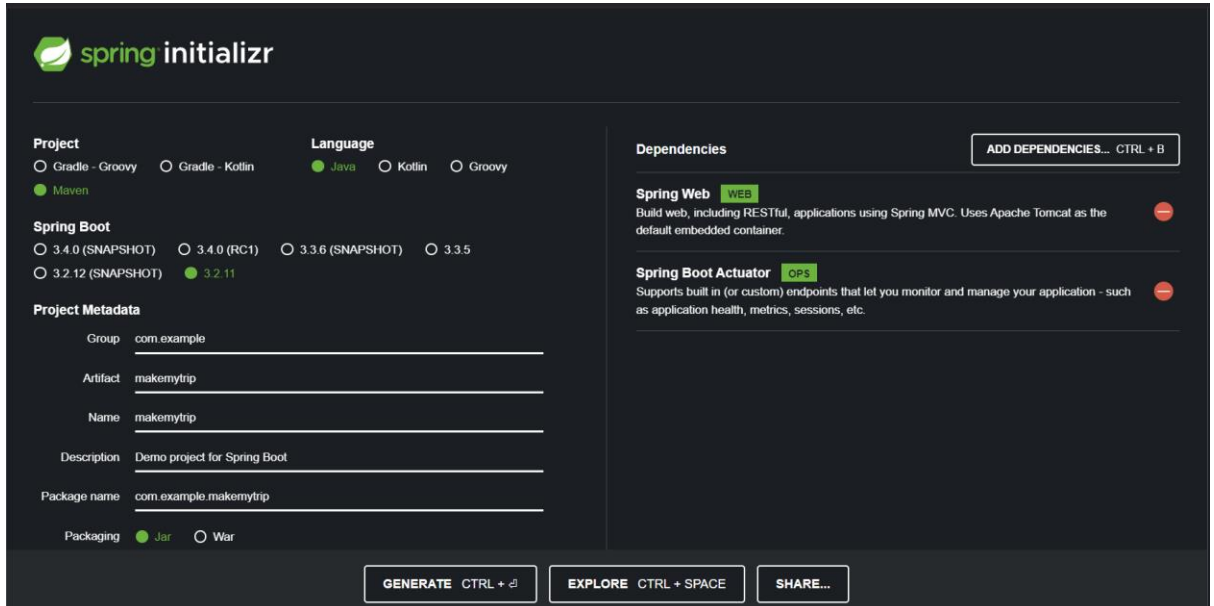


# 1<sup>st</sup> Assignment

## Spring Boot Project Setup and Version Control Workflow

### 1. Create a New Project with Spring Initializr



The screenshot shows the Spring Initializr web interface. It has a dark theme with green accents. The 'Project' section has radio buttons for 'Gradle - Groovy', 'Gradle - Kotlin', and 'Maven' (selected). The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for versions: '3.4.0 (SNAPSHOT)', '3.4.0 (RC1)', '3.3.6 (SNAPSHOT)', '3.3.5', '3.2.12 (SNAPSHOT)', and '3.2.11' (selected). The 'Project Metadata' section has input fields for 'Group' (com.example), 'Artifact' (makemytrip), 'Name' (makemytrip), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.makemytrip). The 'Packaging' section has radio buttons for 'Jar' (selected) and 'War'. The 'Dependencies' section on the right has a button 'ADD DEPENDENCIES... CTRL + B' and two listed dependencies: 'Spring Web' (WEB) and 'Spring Boot Actuator' (OPS), each with a red minus icon to its right. At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

- Go to [Spring Initializr](#).
- Set up your project with desired dependencies (e.g., Spring Web).
- Click **Generate** to download the project ZIP file.

### 2. Set Up Workspace

- Create a folder named myworkspace in the C drive.
- Extract the downloaded project ZIP file into C:\myworkspace.

### 3. Create a GitHub Repository

- Go to [GitHub](#) and create a new repository.
- Copy the repository URL for use later.

### 4. Build and Run the Spring Boot Application

- Open a terminal, navigate to your project folder in myworkspace, and run these commands:

bash

Copy code

mvn clean compile    # Compiles the project

`mvn clean package` # Packages the project as a JAR file

`mvn spring-boot:run` # Runs the Spring Boot application

- Verify the application is running at `http://localhost:8080`.

## 5. Connect to GitHub

- Open your IDE (IntelliJ IDEA or VS Code).
- Go to **VCS** (Version Control System) and select Git.
- Connect to the GitHub repository by pasting the repository URL.
- Push the code to GitHub:

bash

Copy code

`git push origin master`

## 6. Create Branches

- Create the following branches in your local repository:

bash

Copy code

`git branch master` # Creates master branch (if not already created)

`git branch prod` # Creates prod branch

`git branch preprod` # Creates preprod branch

`git branch dev` # Creates dev branch

## 7. Fetch and Work on the dev Branch

- Fetch branches from the remote repository:

bash

Copy code

`git fetch origin`

- Switch to the dev branch:

bash

Copy code

`git checkout dev`

- Stage, commit, and push changes to the dev branch:

bash

Copy code

git add .

git commit -m "Initial commit on dev branch"

git push origin dev

#### **8. Merge dev into preprod on GitHub**

- Go to your GitHub repository.
- Open a Pull Request (PR) to merge the dev branch into the preprod branch.
- Review and merge the PR to complete the process