CS 513 Software Systems(ESD)

Lab 1 – Environment Setup

Lab Objectives

We would be installing all the necessary packages and libraries that are needed for a simple 3-tier full stack development project up and running.
At the end of this lab, we will be ready with and environment that supports:

- Front-end:
  - HTML
  - CSS
  - ReactJS
  - JavaScript
- Middleware
  - Java
  - IntelliJ, Maven
  - Spring Boot Application, Postman
- Backend
  - MySQL

Lab Activities

Installing Java

- Ubuntu 22.04

https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-22-04

- Ubuntu 20.04

https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-20-04

- Ubuntu 18.04:

https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-18-04

For testing java installation:

$ java –version

$ javac –version

For changing java version:

$ sudo update–alternatives —config javac

$ sudo update–alternatives —config java

Installing MySQL

- For Ubuntu 22.04

https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04

- For Ubuntu 20.04

https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04

- For Ubuntu 18.04

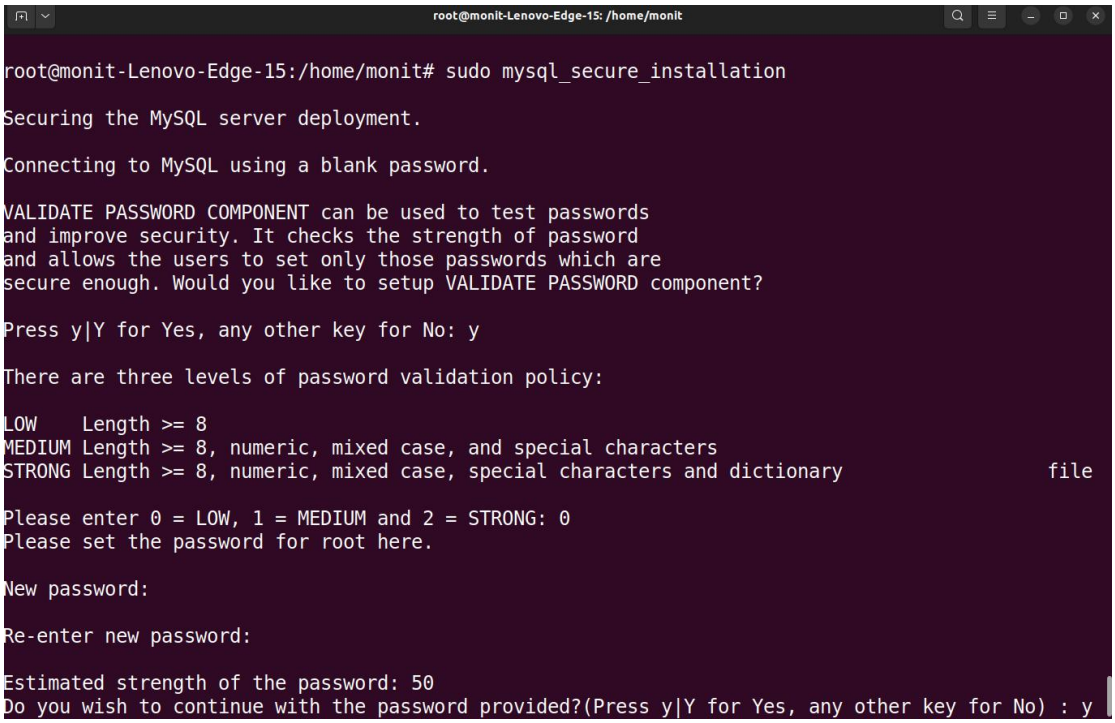
https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04

$ sudo apt update

$ sudo apt install mysql-server

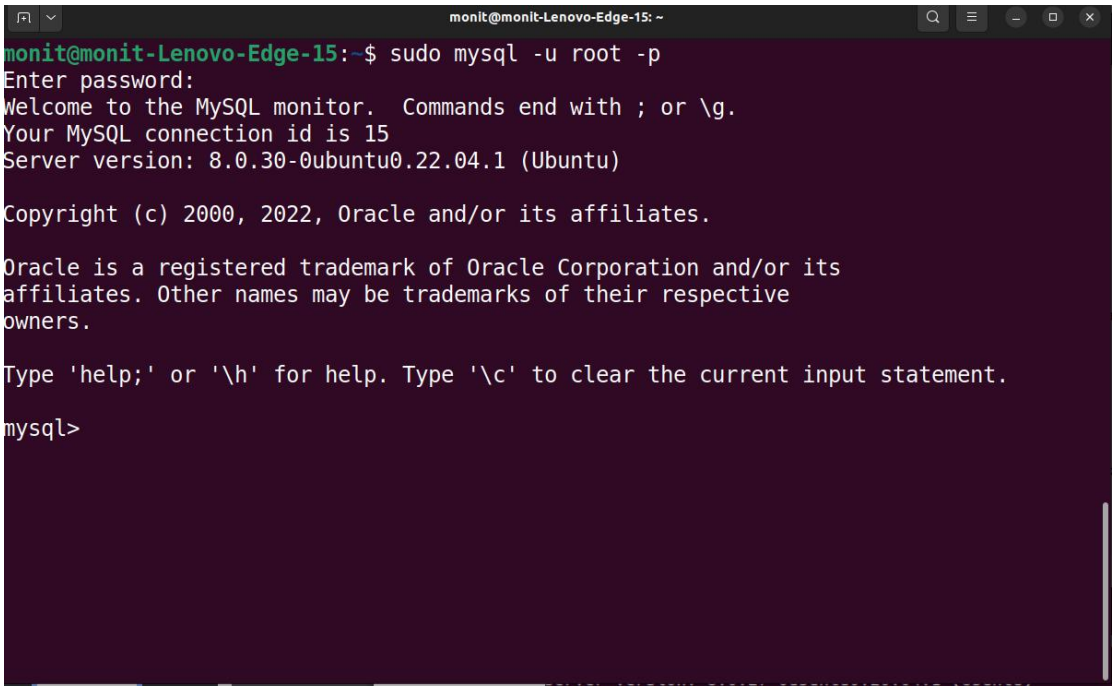
This will install mysql but will not prompt you to set a password or make any other configuration changes. because this leaves your installation of MySQL insecure, we will address this next.

$ sudo mysql_secure_installation



Now you can login into MySQL using following command

$ sudo mysql –u root –p



We have login into MySql with root privileges, follow the given process below to add the

user and login into MySql through non–root users.

mysql> show databases;

mysql> use mysql;

mysql> select Host, User from user;

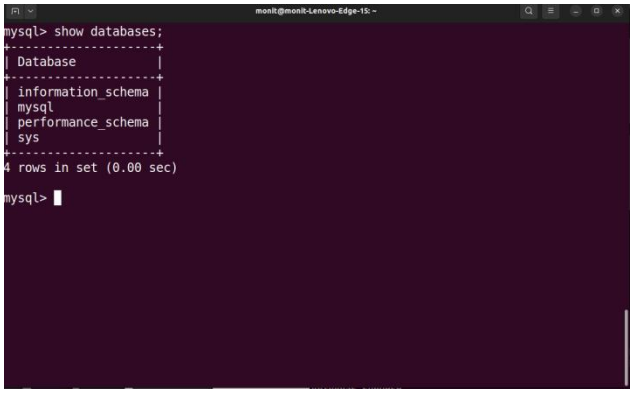
*mysql> alter user root@localhost identified with mysql_native_password by '**yourpassword**';*

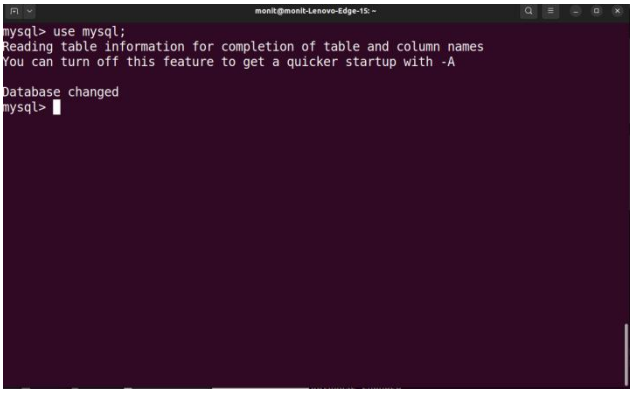
mysql> flush privileges;

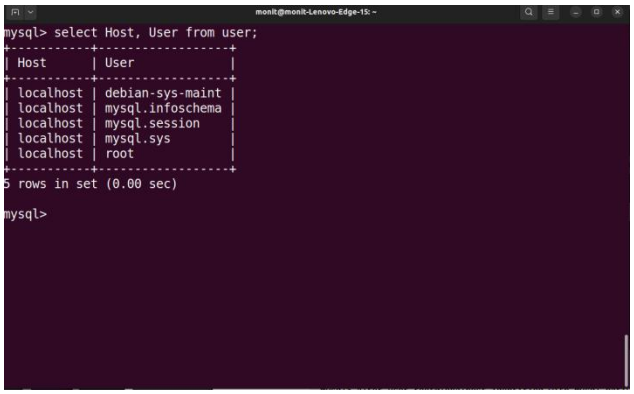
*mysql> create user **yourusername**@localhost identified by '**yourpassword**'*

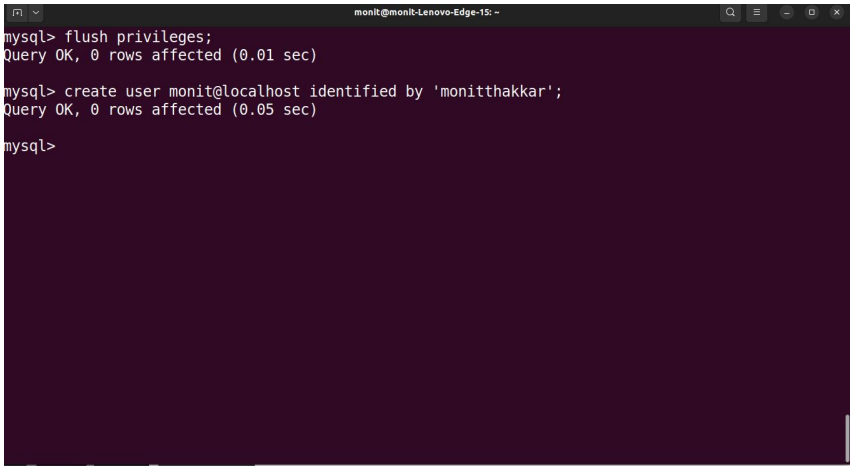
*mysql> GRANT ALL PRIVILEGES ON *.* to monit@localhost WITH GRANT OPTION;*

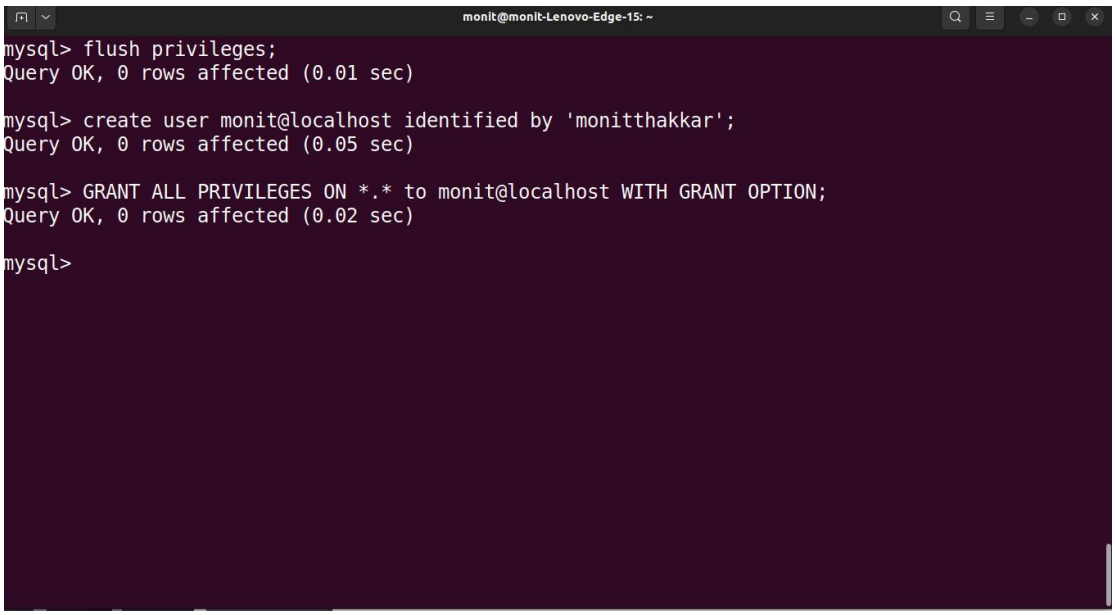
mysql>exit;

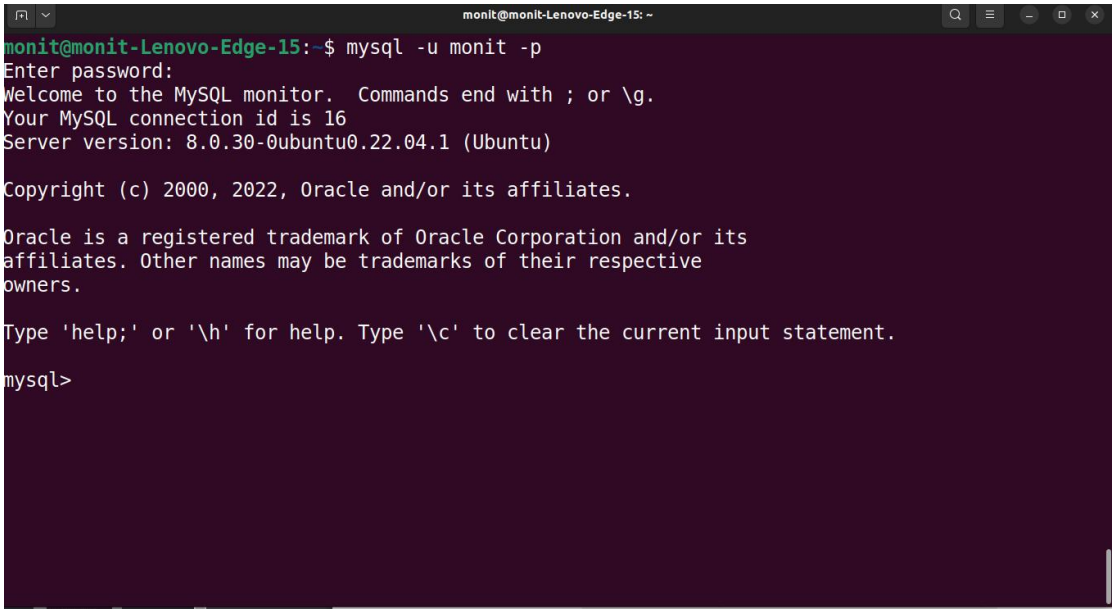









monit@monit-Lenovo-Edge-15: ~
mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
mysql> create user monit@localhost identified by 'monitthakkar';
Query OK, 0 rows affected (0.05 sec)
mysql> GRANT ALL PRIVILEGES ON *.* to monit@localhost WITH GRANT OPTION;
Query OK, 0 rows affected (0.02 sec)
mysql>


monit@monit-Lenovo-Edge-15: ~
monit@monit-Lenovo-Edge-15:~$ mysql -u monit -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.30-0ubuntu0.22.04.1 (Ubuntu)
Copyright (c) 2000, 2022, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>

Setting Up a React Development Environment:

Setting up a development environment for React is an essential first step on your journey to mastering React development. We'll guide you through the process to ensure you have all the necessary tools and configurations in place.

Step 1: Install Node.js and npm

We'll start by installing Node.js and npm, which are essential for developing React applications. Follow these steps:

1. Visit the [official Node.js website](https://nodejs.org/).

2. Download the LTS (Long Term Support) version, which is recommended for most users.

Install NodeJS in Ubuntu:
$ sudo apt update
$ sudo apt install nodejs
$ sudo apt install npm
If you face any issue to update node version to latest(18.x):
$ sudo npm install -g n
$ sudo n lts
(Open a new shell).

3. Run the installer and follow the installation instructions.

4. Verify that Node.js and npm are installed by opening your command line or terminal and running these commands:

$ node -v
$ npm -v

Ensure that you see the versions of Node.js and npm displayed.

Step 2: Install a Code Editor

Next, we need a code editor to write and manage our React code. While you can use any code editor or IDE of your choice, we recommend Visual Studio Code (VS Code) for its extensive support of JavaScript and React development:

1. Download and install [Visual Studio Code](https://code.visualstudio.com/).

2. Install some helpful VS Code extensions for React development:

- ESLint: For JavaScript linting and code quality.
- Prettier - Code formatter: For code formatting.
- Reactjs code snippets: Provides React code snippets for faster development.

You can install these extensions from the VS Code Extensions Marketplace.

Step 3: Create a React Application

Now, let's create your first React application using `create-react-app`, a tool that simplifies project setup:

1. Open your command line or terminal.

2. Run the following command to create a new React application (replace `my-react-app` with your preferred project name):

$ npx create-react-app my-react-app

3. This command generates a new directory with the specified project name (e.g., `my-react-app`) and sets up a basic React project structure for you.

Step 4: Navigate to the Project Directory

Change your current working directory to the newly created project directory:
$ cd my-react-app

Step 5: Start the Development Server

To view your React application locally during development, run the following command:

$ npm start

This starts a development server, and your React app will be available at `http://localhost:3000` by default. Open this URL in your web browser to see your React app in action.

Step 6: Edit Your React App

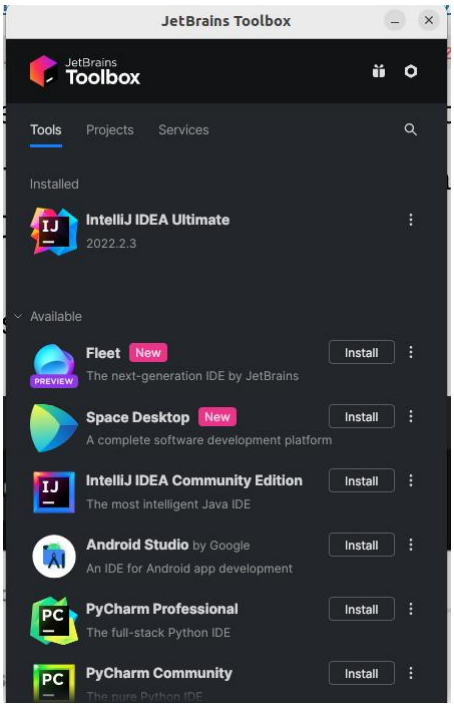
Now, open your chosen code editor and navigate to the project folder (e.g., `my-react-app`). You're ready to start editing React components in the `src` directory. Any changes you make will automatically trigger hot-reloading, allowing you to see instant results in your browser as you code.

We've successfully set up your development environment for React. You're now equipped to start building React applications, experimenting with components, and exploring the vast ecosystem of React libraries and tools.
Happy coding!

IntelliJ and Maven

Register using University Mail Address option:
Link: https://www.jetbrains.com/shop/eform/students
Confirm from your mail and set login credentials. (You should be able to see licence at: Link: https://account.jetbrains.com/licenses
Download the toolbox: https://www.jetbrains.com/toolbox-app/
Run the following command in your terminal:
$ sudo apt install libfuse2

Extract it and install it by double clicking the executable file to start the toolbox. (Create an account with your IIITB email address to avail 1 year free ultimate edition of IntelliJ. You can proceed with the community edition as well.)
Login with your account and install the desired version of intelliJ.



Open IntelliJ and create a new maven project. IntelliJ will automatically detect your java SDK when you create a new project. Proceed with default configuration. Create a new project to run the hello world program.

Right click on the source folder and create a new class called HelloWorld Paste in the following code in your newly created class HelloWorld and run it.

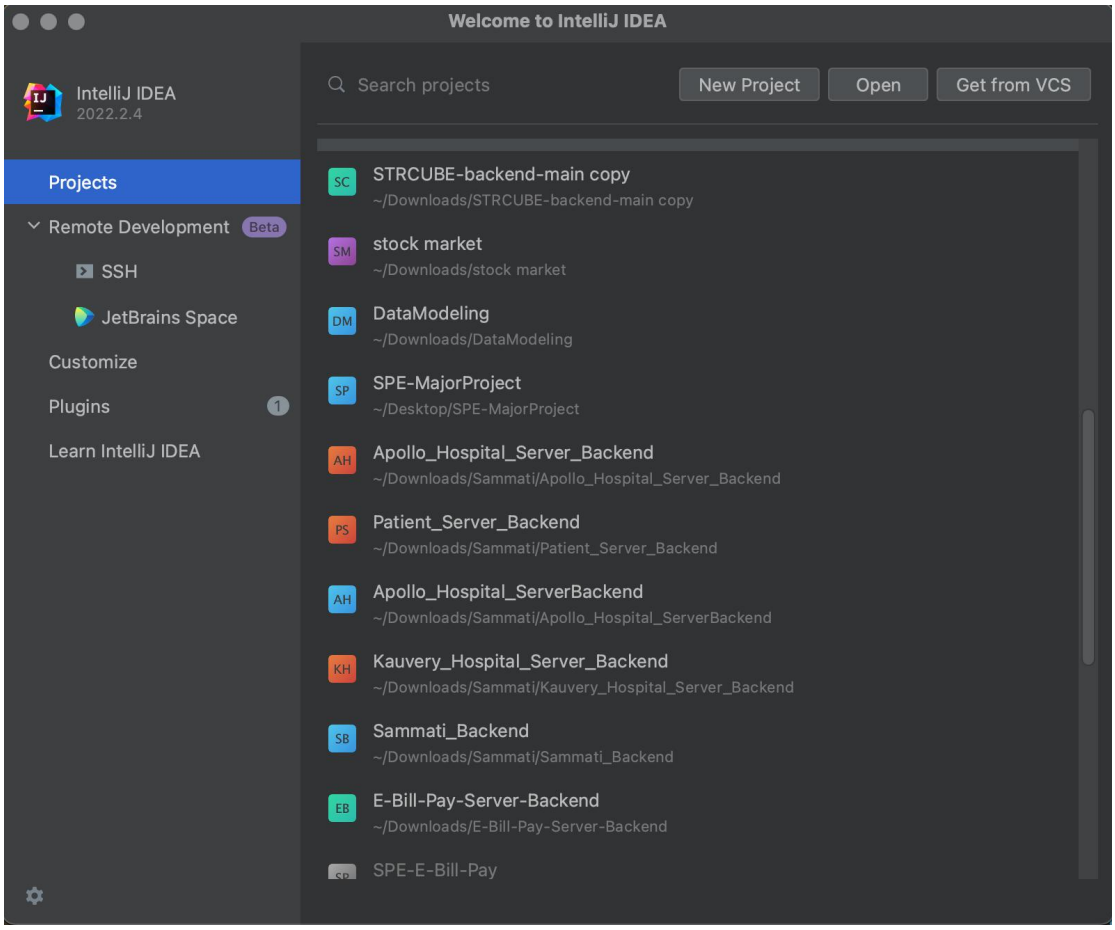
You should see “hello world!” on your output screen.

Spring Boot Application

Let's create the first spring application
(Refer: https://www.jetbrains.com/help/idea/your-first-spring-application.html)

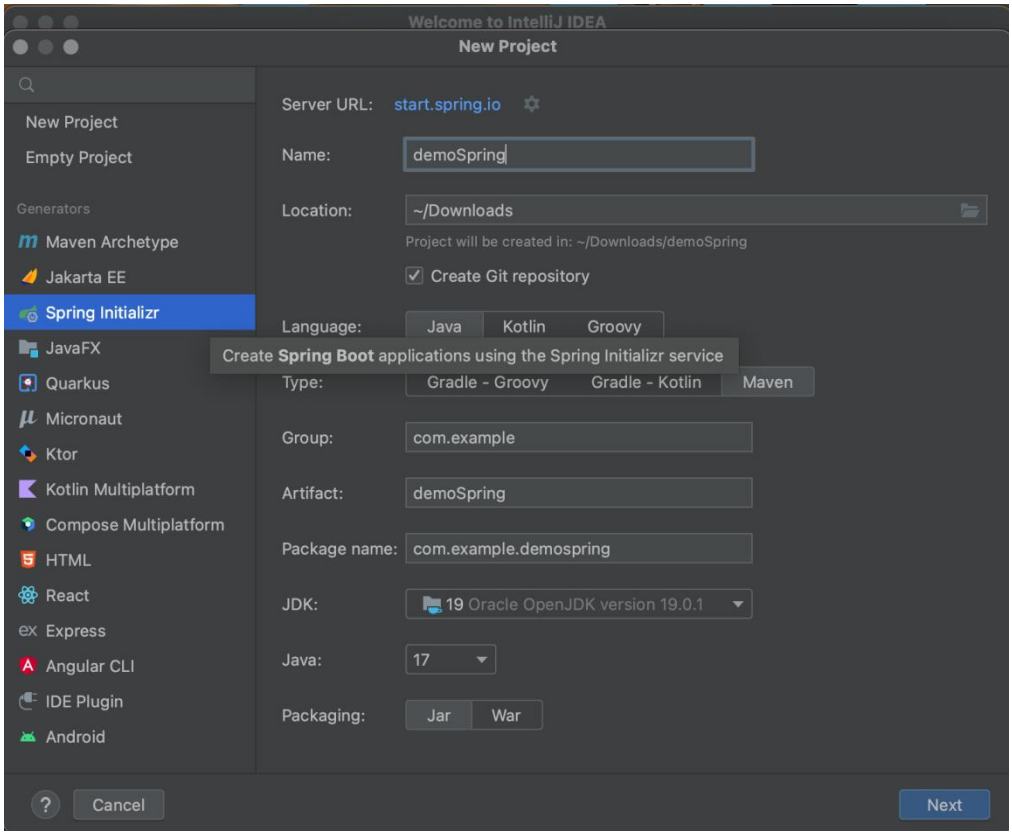
This tutorial describes how to create and run a Spring application in IntelliJ IDEA. It will be a Spring Boot Maven project generated by Spring Initializr. This is the quickest way to create a Spring application, and IntelliJ IDEA provides a dedicated project wizard for it. You will learn how to expose an HTTP endpoint and map it to a method that returns a greeting to the user when accessed through a web browser.

1. Go to File | New | Project.
OR we can see the following window after opening IntelliJ IDEA



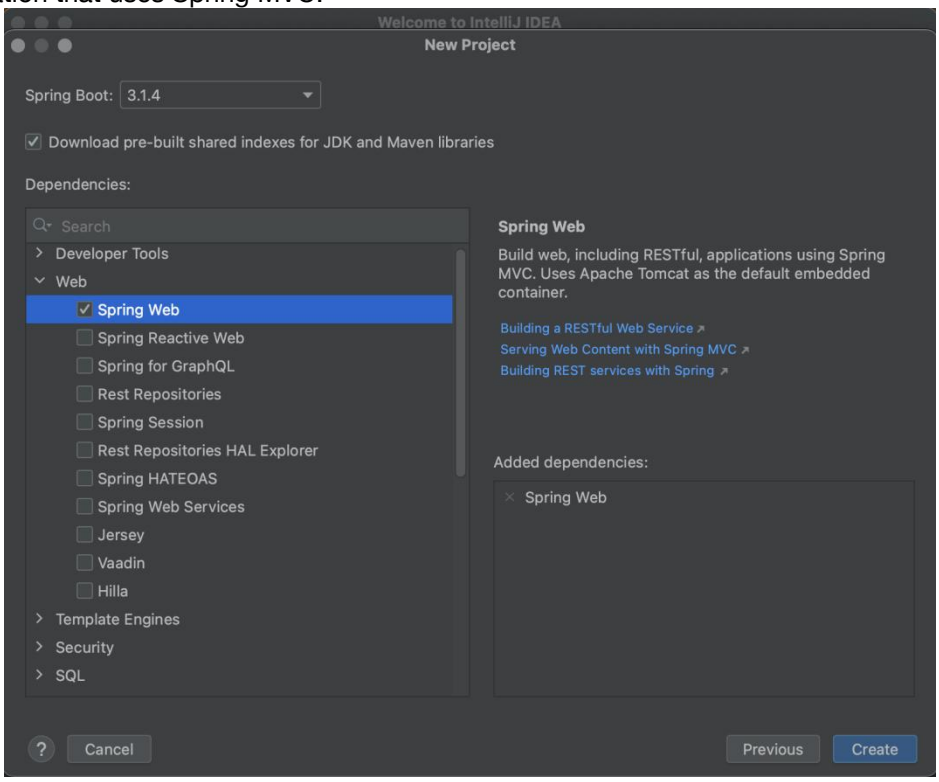
2. In the left pane of the New Project wizard, select Spring Initializr.

- Specify a name for the project: demoSpring.
- From the JDK list, select Download JDK and download the latest version of Oracle OpenJDK.
- Select the latest Java version.

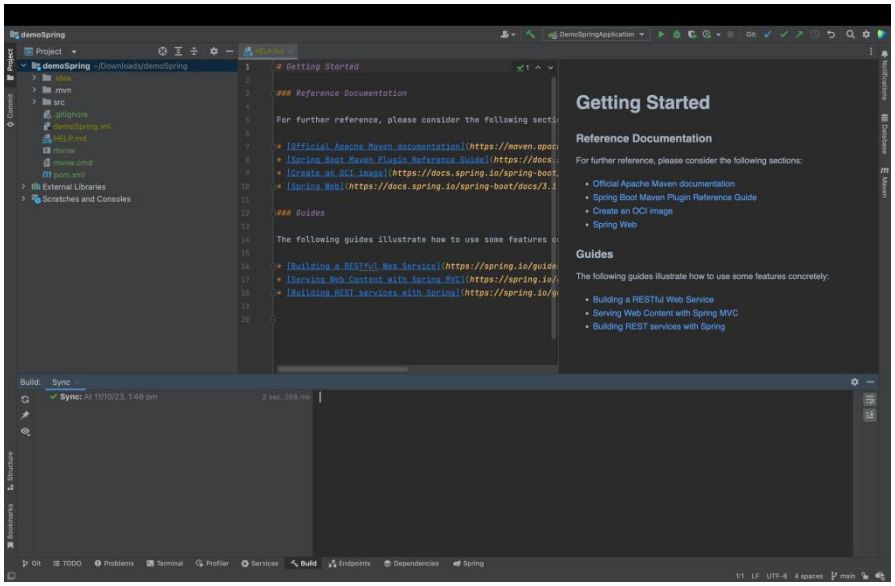


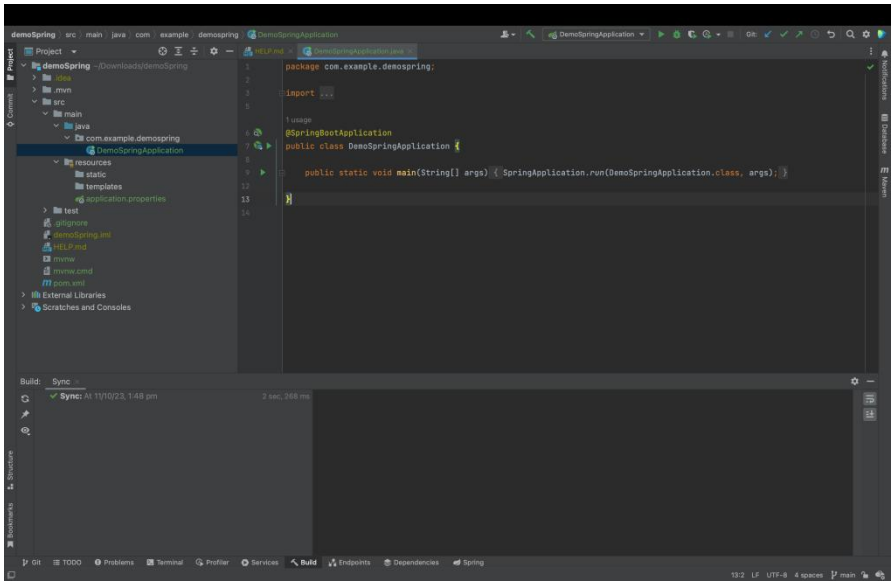
Click Next to continue.

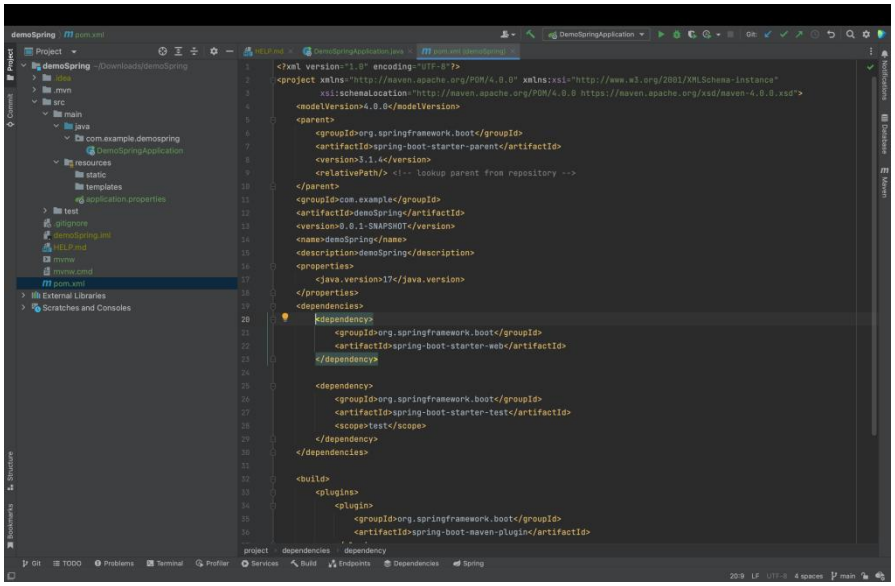
3. Select the Spring Web dependency under Web. This dependency is required for any web application that uses Spring MVC.



Click Create to generate and set up the project.


Getting Started
Reference Documentation
For further reference, please consider the following sections:
Official Apache Maven documentation
Spring Boot Maven Plugin Reference Guide
Create an OCI image
Spring Web
Guides
The following guides illustrate how to use some features concretely:
Building a RESTful Web Service
Serving Web Content with Spring MVC
Building REST services with Spring


package com.example.demospring;
import ...
@SpringBootApplication
public class DemoSpringApplication {
public static void main(String[] args) { SpringApplication.run(DemoSpringApplication.class, args); }
}


<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.1.4</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.example</groupId>
<artifactId>demoSpring</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>demoSpring</name>
<description>demoSpring</description>
<properties>
<java.version>17</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>

pom.xml

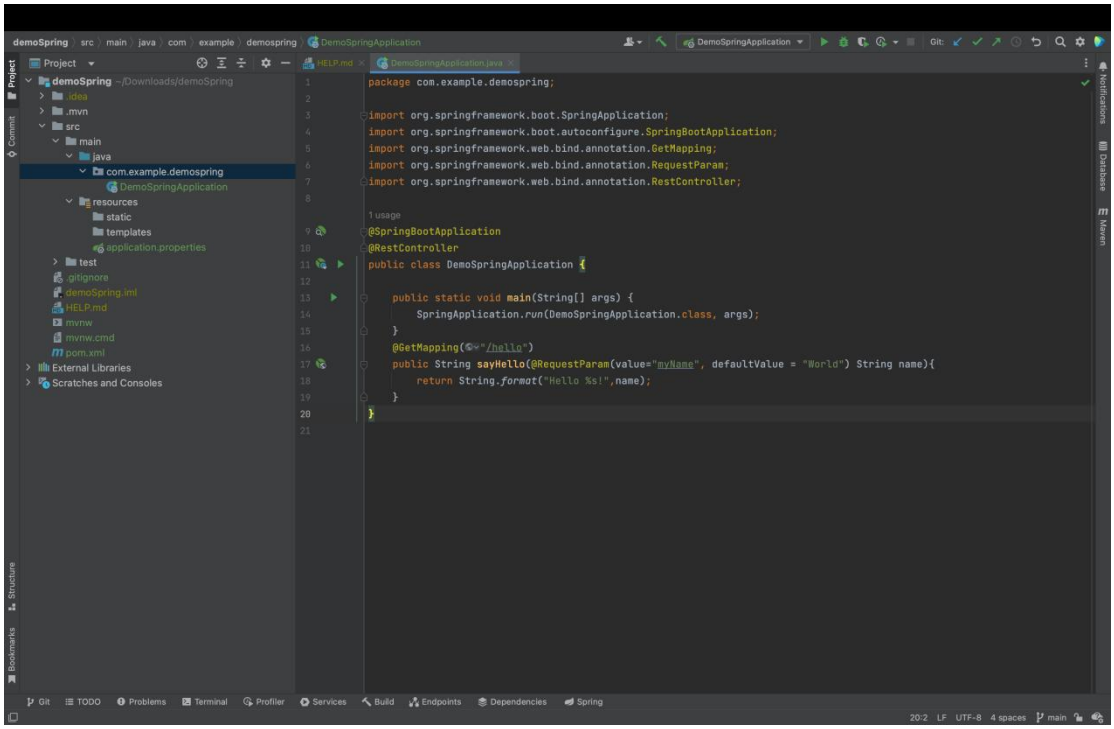
Add a method that sends a greeting

Spring Initializr creates a class with the main() method to bootstrap your Spring application. In this tutorial, we'll add the sayHello() method directly to this class.

1. Open the SpringBootTutorialApplication.java file under
   src/main/java/com/example/demospring.

IntelliJ IDEA provides the Go to File action to quickly find and open files. Go to Navigate | File, start typing the name of the file and select it from the list.

2. Add the sayHello() method with all of the necessary annotations and imports so that the file looks like this:



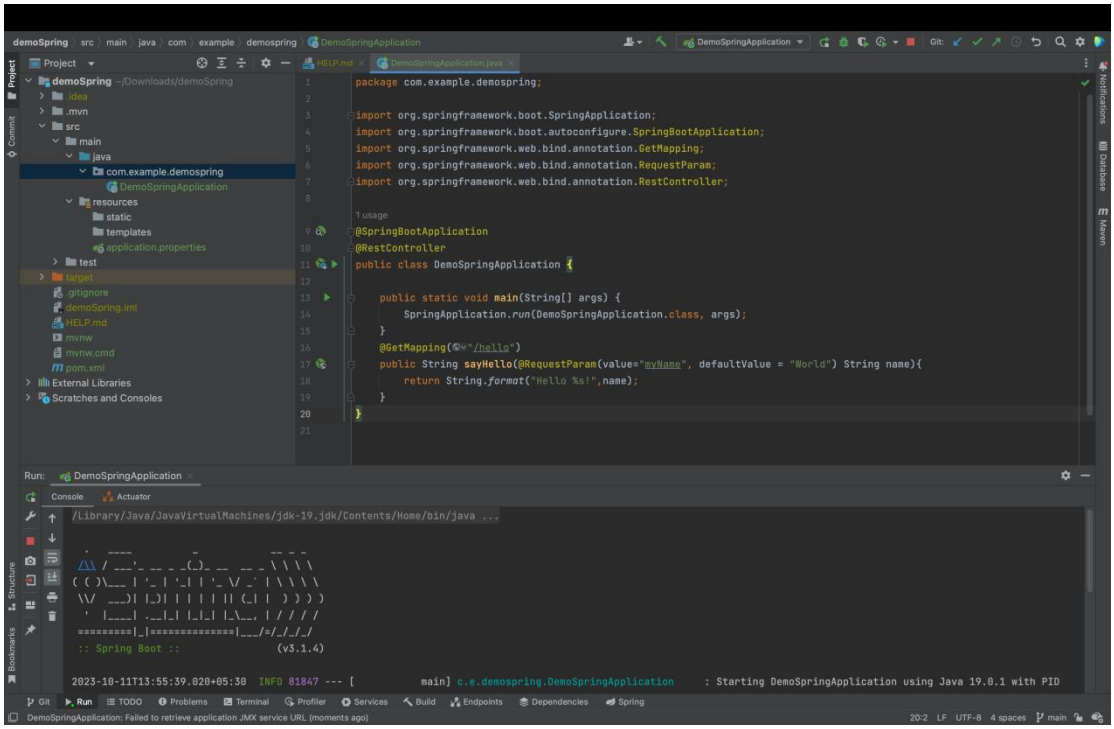
- The sayHello() method takes the name parameter and returns the word Hello combined with the parameter value. Everything else is handled by adding Spring annotations:
- The @RestController annotation marks the SpringBootTutorialApplication class as a request handler (a REST controller).
- The @GetMapping("/hello") annotation maps the sayHello() method to GET requests for /hello.
- The @RequestParam annotation maps the name method parameter to the myName web request parameter. If you don't provide the myName parameter in your web request, it will default to World.

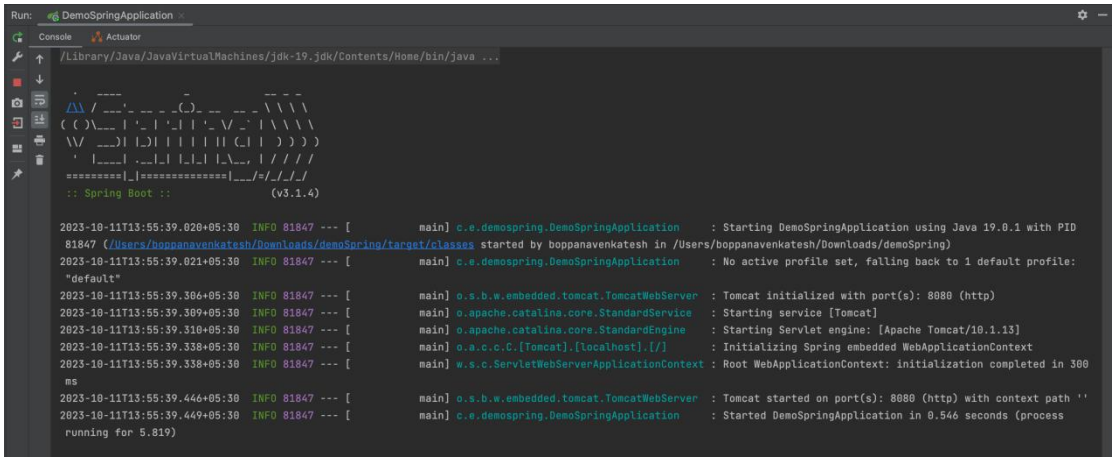
Run your Spring application

IntelliJ IDEA creates a Spring Boot run configuration that you can use to run your new Spring application.

You can use the shortcut or Run icon in the gutter of the SpringBootTutorialApplication.java file next to the class declaration or the main() method declaration.

By default, IntelliJ IDEA shows your running Spring Boot application in the Run tool window.





The Console tab shows the output of Spring log messages. By default, the built-in Apache Tomcat server is listening on port 8080. Open your web browser and go to http://localhost:8080/hello. If you did everything right, you should see your application respond with Hello World!.

This is the default generic response. You can provide a parameter in your web request to let the application know how to greet you properly.
For example, try http://localhost:8080/hello?myName=ESD TA 2023.

Using Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

Download and Extract postman on your machine.
https://www.postman.com/downloads/

Youtube Video: https://youtu.be/cp6qBCg5Y_I?feature=shared
or follow the Installation guidelines:
https://learning.postman.com/docs/getting-started/installation/installation-and-updates/#installing-postman-on-linux