

Step 1: Create Spring Boot Project using Spring Initializr

1. Go to Spring Initializr.
2. Fill in the project metadata:
 - Group: com.example
 - Artifact: student-management
3. Choose the following dependencies:
 - Spring Web
 - Spring Data JPA
 - MySQL Driver
4. Click Generate to download the project zip file.
5. Extract the downloaded file to a suitable location.

Step 2: Import Project into Eclipse IDE

1. Open Eclipse IDE.
2. Go to File > Import....
3. Select Existing Maven Projects and click Next.
4. Browse to the location where you extracted the project and select the root directory (where pom.xml file is available).
5. Click Finish to import the project.

Step 3: Copy Code into Project

1. Copy provided Java files to src/main/java/dataManagement.dm.
2. Copy stddata.html to src/main/resources/static.
3. Copy application.properties to src/main/resources.

Step 4: Configure Dependencies in pom.xml

1. Ensure pom.xml contains dependencies for:
 - Spring Boot Starter Web
 - Spring Boot Starter Data JPA
 - MySQL Connector Java
 - Spring Boot Starter Test
 - Dependencies for handling file uploads

Step 5: Configure Application Properties

1. Open application.properties and set MySQL database connection properties according to your local setup.

Step 6: Run the Application

1. Locate DmApplication.java in src/main/java/dataManagement.dm.
2. Right-click on DmApplication.java and choose Run As > Java Application.

Step 7: Test the Application

1. Access defined RESTful endpoints:
 - <http://localhost:8080/api/students/> to get all students.
 - Use other endpoints for CRUD operations and file uploads as described in StudentController.java.

Step 8: Verify Functionality

1. Verify CRUD operations and file upload/download functionality.

Run the Application:

1. Select the project folder (student-management).

Employee data management project notes

2. Right-click, choose Maven, then Update Project... to update dependencies.
3. Right-click DmApplication.java, located in src/main/java/dataManagement.dm.
4. Choose Run As > Java Application.

Access the Application:

1. Once the application is running, open a web browser.
2. Go to <http://localhost:8080/api/students/html> to access the application.

By following these corrected steps, you should be able to set up and run the Spring Boot project in Eclipse effectively

1. To upload a project to GitHub, follow these steps:

1. **Clone the Repository:** Open your command line interface (CLI) and navigate to the directory where you want to clone the repository. Then, run the following command to clone the "java-Genius" repository to your local machine:
`git clone <repo-link>`
2. **Create the EmpDataManagement Folder:** Navigate into the cloned repository's directory using the following command:
`cd <folder-name>`
3. **Then, create the "EmpDataManagement" folder using the command:**
`mkdir EmpDataManagement`
4. **Move Your Project Folder into EmpDataManagement:** Copy and paste your project folder inside the EmpDataManagement folder:
`cd EmpDataManagement`
5. **Initialize Git:** If the EmpDataManagement folder is not already a Git repository, initialize it:
`git init`
6. **Add Changes:** Add the changes to the staging area:
`git add .`
7. **Check Status:** Check the status of the repository to ensure all changes are staged:
`git status`
8. **Commit the Changes:** Commit the changes with a meaningful message:
`git commit -m "Added EmpDataManagement folder and project folder"`
9. **Push the Changes to the Remote Repository:** Push the committed changes to the remote repository: `git push origin main`

Verify on GitHub: Finally, go to your repository on GitHub and navigate to the "EmpDataManagement" folder to verify that your project folder has been successfully added.

Thank You!