SOFTWARE DEVELOPER MANUAL

Student Management Application with Spring Boot and Angular

Philipp Bomers

July 1, 2022

Contents

1. Introduction

This training project is about a Spring Boot (Java) and Angular (TypeScript) application, which should replace an existing Excel spreadsheet that assigns students to projects in terms of time. This work focuses on clean programming, extensive testing, and detailed documentation. Therefore, the software is suitable for beginners as a learning project.

2. Project Setup

To start the project, follow the instructions for Microsoft Visual Studio Code on Microsoft Windows 10. Other operating systems and IDEs may require a customized configuration. First, it is necessary to fulfill the following requirements:

- Install node.js: https://nodejs.org/en/
- Install Git: https://git-scm.com/download/win
- Install Java 17 (JDK): https://www.oracle.com/java/technologies/downloads/#jdk17-windows
- Download Maven: https://maven.apache.org/download.cgi
- Visual Studio Code (Optional): https://code.visualstudio.com/

Then, to use the Angular functionality, open the Windows PowerShell (possibly it is necessary to run it as an administrator) and insert:

```
npm install -g @angular/cli
```

Confirm the installation with the return key. The installation process can take a while, even if not recognizable, because it downloads the necessary files for Angular, including a recognizable size. If the installation process fails because of missing user rights, execute the following statement with administrator rights and try the first step again.

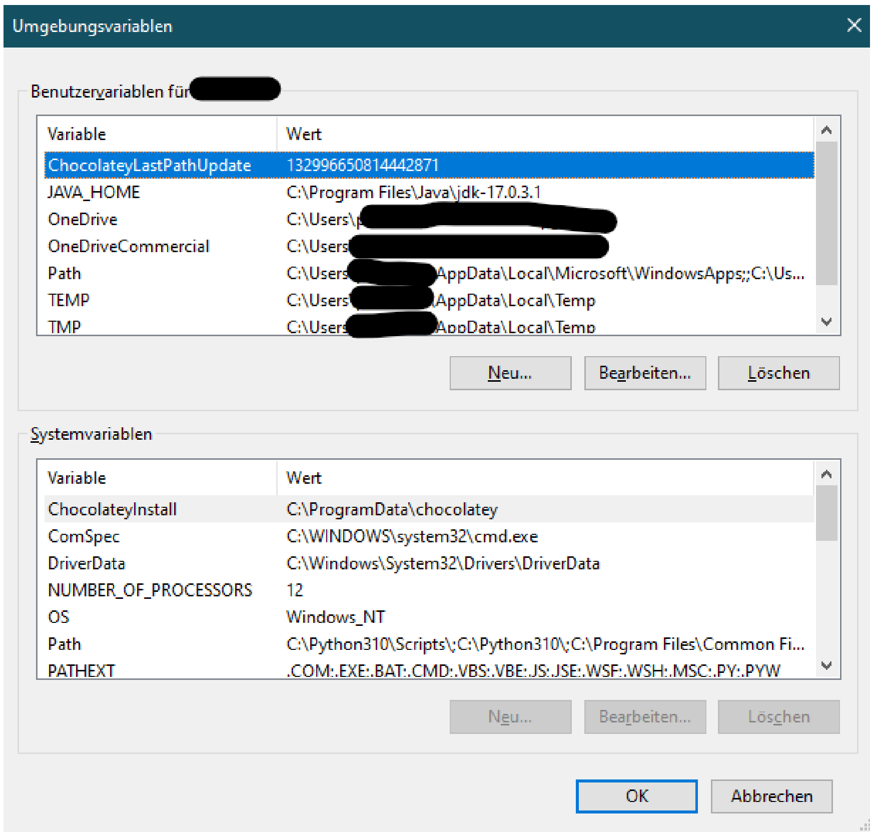
```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
```

After the installation, it is necessary to define the JAVA_HOME path variable to run the JUnit tests.

- Open the Windows start menu and write "variable"
- Click on "System Variables"

- Insert "JAVA_HOME" including the path to the Java folder.

In the end, it should look similar to the following picture.



Benutzervariablen für [redacted]

Variable	Wert
ChocolateyLastPathUpdate	132996650814442871
JAVA_HOME	C:\Program Files\Java\jdk-17.0.3.1
OneDrive	C:\Users\[redacted]
OneDriveCommercial	C:\Users[redacted]
Path	C:\Users[redacted]AppData\Local\Microsoft\WindowsApps;;C:\Us...
TEMP	C:\Users[redacted]AppData\Local\Temp
TMP	C:\Users[redacted]AppData\Local\Temp

Systemvariablen

Variable	Wert
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\Python310\Scripts\;C:\Python310\;C:\Program Files\Common Fi...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW

Additionally, add MAVEN_HOME in the same way. Further, double click on the path and add "%MAVEN_HOME%\bin".

After installing VS Code, it is preferable to add some partially mandatory plugins for a pleasant development environment:

- Angular Extension Pack
- Extension Pack for Java
- Spring Boot Extension Pack
- Required: Lombok Annotations Support for VS Code

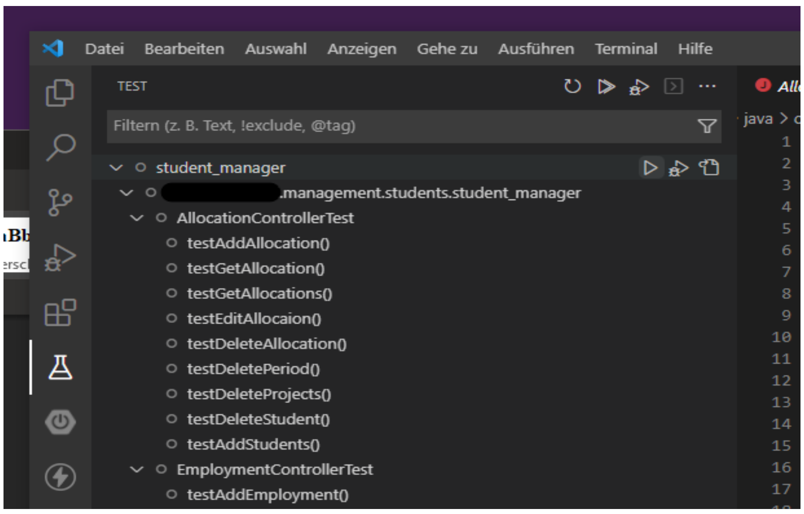
- Optional: Excel Viewer
- Optional: Thunder Client (to test API requests inside VSCode)

3. Spring Boot Setup

We can start importing the project to Visual Studio Code or another IDE from a Git repository or the source folder. Make sure to have installed Java, Spring Boot, and Lombok Plugins. Otherwise, the program probably is not startable. Moreover, developer tools improve the building environment, including live reload. F. ex., it needs several steps for activation in IntelliJ, while VS Code provides standard predefined configuration.

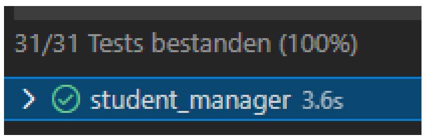
Further, check the "application.properties" configuration to see if it fits with the system (f. ex., port and database configuration). For example, the standard port is 8080.

Finally, test if everything works correctly. First, click on the test icon, hold the mouse over "student_manager", and click on the run button.



```
student_manager
  [redacted].management.students.student_manager
    AllocationControllerTest
      testAddAllocation()
      testGetAllocation()
      testGetAllocations()
      testEditAllocaion()
      testDeleteAllocation()
      testDeletePeriod()
      testDeleteProjects()
      testDeleteStudent()
      testAddStudents()
    EmploymentControllerTest
      testAddEmployment()
```

If all tests turn green, the project setup for Spring Boot is complete.



```
31/31 Tests bestanden (100%)
> student_manager 3.6s
```

4. Spring Boot Application Overview

First, one should always keep an eye on "pom.xml" and "application.properties". The first file manages all dependencies. For example, here, we can add and remove Spring Boot Plugins. It may also be interesting to adapt the Java version and add Maven plugins here later. Next, we need the "application.properties" file to handle system settings.

4.1 Test

The test folder includes all test files. In addition, the "StudentManagerTest.java" provides the main configuration and helping methods for the tests. Therefore, it is preferable to extend that class while writing Unit tests.

Further, the "@SpringBootTest" annotation lets the tests automatically start the server and then go through the tests. This action is necessary to avoid operating user errors. Therefore, we must ensure the server is down when starting the tests.

Generally, running all tests after changing the source code is a great idea to check if anything works correctly.

Due to the extensive preparation and multiple provided tests, the software is ready to develop comfortably and test-driven.

Write tests for as many use cases as possible to improve the quality of the software.

Generally, we can run these tests with Maven as well without using the testing plugin. Possibly that could support and secure an automatic deploying process.

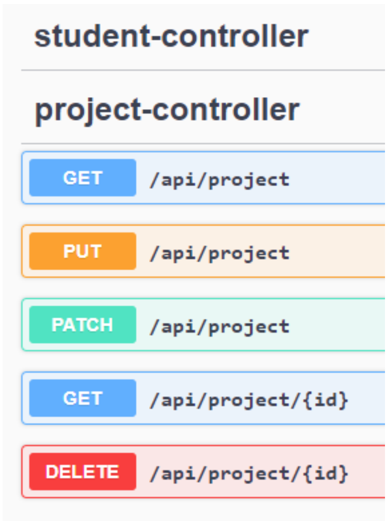
4.2 Resources

We can provide the resources folder with any external files, like Excel files or pictures, needed to run the program.

4.3 Controller

Each entity has a separate controller, which provides a Rest API for universal access. Of course, we can access this via our Angular application, but we can also write other frontend software like mobile or desktop applications.

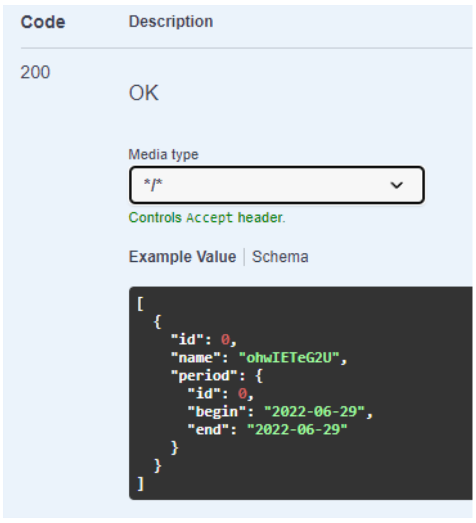
To observe and test the possibilities of the API, start the server and look at the Swagger-UI: http://localhost:8080/swagger-ui/index.html. The Swagger-UI provides an overview of all request mappings, including the ability to test it. Nevertheless, we should only activate that plugin in development and never in a productive environment.



student-controller

project-controller

```
GET     /api/project
PUT     /api/project
PATCH   /api/project
GET     /api/project/{id}
DELETE  /api/project/{id}
```



Code	Description
200	OK

Media type: */*

Controls Accept header.

Example Value | Schema

```
[
  {
    "id": 0,
    "name": "ohwIETeG2U",
    "period": {
      "id": 0,
      "begin": "2022-06-29",
      "end": "2022-06-29"
    }
  }
]
```

Remember to return adequate and understandable exceptions and errors. The more detailed these are, the easier the handling will be. However, it can be challenging for users to understand, so limit these exceptions to usability and background debugging.

When developing, choose the correct mapping and check the entries with the "@Valid" annotation. Again, keep the business logic to a minimum as the service processes it. For security reasons, the controllers only directly access the services and never the repositories.

4.4 Entity

The Lombok plugin (https://projectlombok.org/) generates the entities constructor, getter, and setter. Therefore, entities only need class parameters, validations, and database mappings. Furthermore, we can create custom objects with constructor parameters using the Lombok builder (https://projectlombok.org/features/Builder).

Ensure to include custom error messages for validation to show them in the frontend user input validation.

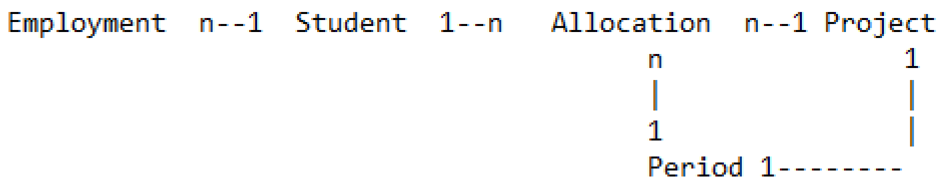
4.5 Repository

Each repository provides an interface for database interactions. Through inheritance, it is only necessary to include requests not already existing in the CrudRepository.

4.6 Service

The business logic happens in the service, which connects the controller, repository, and entities. Whenever possible, return entities as Optional and leave exception management to the controller. Think about the relations to other entities when editing something.

5. Entity Relations



```
Employment  n--1  Student  1--n   Allocation  n--1 Project
                                        n               1
                                        |               |
                                        1               |
                                        Period 1--------
```

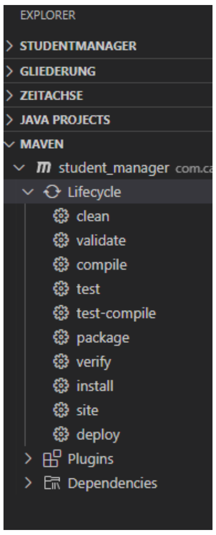
- When adding entities with non-existent relations, the service also adds the relations.
- Deleting employment is not possible when there is a student with that employment.
- It is not necessary to delete periods separately. Instead, they automatically are deleted by the deletion of allocations or projects.
- Student or project deletion triggers allocation deletion.

6. Documentation

The application uses JavaDoc (https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html) for documentation. Additionally, there are developer explanations of the source code inside the methods. JavaDoc helps to keep the overview through the IDE but can also generate documentation in the form of HTML.

7. Maven

In rare cases, cleaning up Maven and regenerating the project is necessary. Again, VS Code offers easily usable tools for these cases.

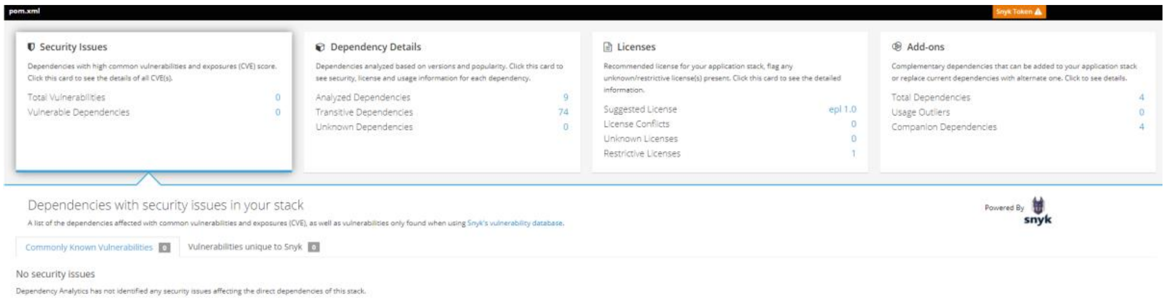


When Maven has starting issues, insert the following statement in the console being are in the root folder:

```
mvn -N io.takari:maven:wrapper
```

After making changes in "pom.xml", sometimes it is necessary to run the install script. One possible use-case can be the update/upgrade of Spring Boot or a plugin. The clean command can also help with serious program errors.

Verify that there are no known vulnerabilities in the included plugins f. ex., by using VS Codes' internal vulnerability report.



Security Issues
Dependencies with high common vulnerabilities and exposures (CVE) score. Click this card to see the details of all CVE(s).

Total Vulnerabilities	0
Vulnerable Dependencies	0

Dependency Details
Dependencies analyzed based on versions and popularity. Click this card to see security, license and usage information for each dependency.

Analyzed Dependencies	9
Transitive Dependencies	74
Unknown Dependencies	0

Licenses
Recommended license for your application stack, flag any unknown/restrictive license(s) present. Click this card to see the detailed information.

Suggested License	epl 1.0
License Conflicts	0
Unknown Licenses	0
Restrictive Licenses	1

Add-ons
Complementary dependencies that can be added to your application stack or replace current dependencies with alternate one. Click to see details.

Total Dependencies	4
Usage Outliers	0
Companion Dependencies	4

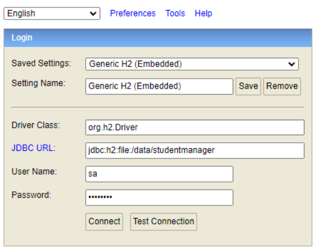
Dependencies with security issues in your stack
A list of the dependencies affected with common vulnerabilities and exposures (CVE), as well as vulnerabilities only found when using Snyk's vulnerability database.

Commonly Known Vulnerabilities 0 | Vulnerabilities unique to Snyk 0

No security issues
Dependency Analytics has not identified any security issues affecting the direct dependencies of this stack.

8. H2 Database Access

To open the H2-Database console, go to http://localhost:8080/h2-console/. Then, insert the password "password". Further, always click on "Run" to open an SQL statement.



Saved Settings:	Generic H2 (Embedded)
Setting Name:	Generic H2 (Embedded)
Driver Class:	org.h2.Driver
JDBC URL:	jdbc:h2:file:/data/studentmanager
User Name:	sa
Password:	••••••••

9. Angular Setup

Open the folder or clone the Git repository. Then open the IDE terminal and run

```
ng serve
```

Open the browser page http://localhost:4200/ to begin developing the Angular frontend. Again, with developer tools, the Angular server automatically restarts if there are any changes.

10. Angular Application Overview

Unlike the Spring Boot application, we make changes via the terminal, like adding or removing plugins. Creating components, classes, modules, and more is possible through terminal statements or visually supported with IDE plugins; use the preferred option.

For new Angular users, Angular provides excellent documentation on https://angular.io/. In addition, the project includes Bootstrap 5 for comfortable design development (https://getbootstrap.com/docs/5.0/getting-started/introduction/).

10.1 entity.component.html

The HTML files include the frontend code. Through the routing module, we can include sites via "<app-entity></app- entity >". Further, Angular provides dynamic content access. Therefore, we only have one site with dynamically loaded content. The main site is "app" and directs to the entities. Therefore, it is necessary to get an overview of the structure before developing new functions.

10.2 entity.component.ts

The TypeScript files include the business logic we forward to the HTML file. In this case, the "app.component.ts" delivers lists with all required entities and is publically accessible from the other entities. This strategy reduces the number of server requests.

Angular uses dependency injection like Spring Boot also does. Dependency injection allows us to outsource server requests to the services effortlessly.

10.3 entity.service.ts

In this class, we send server requests in JSON format to our backend Spring Boot server. The Spring Boot server uses another port; therefore, it is necessary to configure that port at "proxy.conf.json". If using another port than 8080, please adjust the number.

API requests should return Observables from the "rxjs"-framework (https://rxjs.dev/guide/overview), which we need for concurrent data access.

10.4 entity.ts

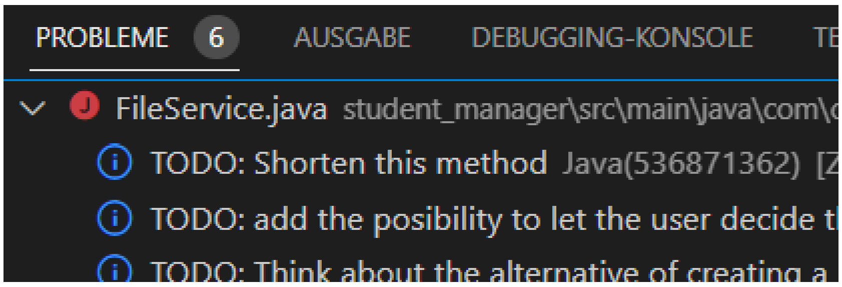
The class reflects the associated Spring Boot entity. The ID is optional because we do not know it in some cases (f.ex. creating an entity).

11. Angular Development Tips

- Avoid the "any" type.
- Never use "ts-ignore".
- Use "const" until "let" is necessary.
- Use lazy loading (like configured in "app-module.ts".
- Use "ng update" and "npm update" to update dependencies via console statement.
- Press CTRL + C to abort console statements.

12. Where To Start Developing?

- Search for the project todos. In VS Code, they are all accessible under the Issues/problems tab next to the console.



FileService.java student_manager\src\main\java\com\c
TODO: Shorten this method Java(536871362) [Z
TODO: add the posibility to let the user decide t
TODO: Think about the alternative of creating a

- Check if all tests work correctly. If they do, write further tests, or find improvements.
- If there is external planning or tasks from the requirement gathering phasis, get them done.
- Optimize the existing code: shorten, simplify, make it more performant, optimize the structure, add features, and write documentation.