

Software Engineering (ISE)

Übungsblatt 6

Abgabe in Zweiergruppen erlaubt

Aufgabe 6.1 (Spring Boot: Erweiterung von CampusCoffee: 5 Punkte)

In dieser Übung werden Sie eine Benutzerverwaltung zur *CampusCoffee*-Anwendung hinzufügen. Sie können sich dabei an dem vorhandenen Code für POS orientieren. In einem späteren Übungsblatt werden wir dann den resultierenden teilweise redundanten Code konsolidieren (“Refactoring”). Das Ziel ist, dass Sie ein tieferes Verständnis der Architektur erlangen und den in Woche 7 eingeführten Bean-Validation-Ansatz anzuwenden.

In this exercise, you will add user management to the *CampusCoffee* application. You may use the existing code for POS as a guideline. In a later assignments, we will consolidate the resulting partially redundant code (“refactoring”). The goal is for you to gain a deeper understanding of the architecture and to apply the Bean Validation approach introduced in Week 7.

Vorbereitung:

Preparation:

- Forken Sie das GitHub-Repository für dieses Übungsblatt:
[Fork the GitHub repository for this assignment.](#)
https://github.com/se-uhd/ise25-26_assignment06.
- Klonen Sie Ihren Fork in ein Verzeichnis Ihrer Wahl.
[Clone your fork to a directory of your choice.](#)
- Folgen Sie den Anweisungen in der README-Datei.
[Follow the instructions in the README file.](#)
- Führen Sie `mvn clean install` aus. Der Build sollte erfolgreich verlaufen.
[Run mvn clean install. The build should be successful.](#)
- Um sich einen Überblick über die TODOs im Code zu verschaffen, können Sie das IntelliJ TODO tool window verwenden.
[To get an overview of the TODOs in the code, you can use the IntelliJ TODO tool window.](#)

Aufgaben: Ihr Ziel in dieser Aufgabe ist es, Funktionalitäten zum Erstellen, Aktualisieren, Abrufen und Löschen von Benutzern zu implementieren. Ihr Fokus liegt dabei auf dem API- und Data-Modul. Das Datenbanktabellen-Schema, die entsprechende Datenbankentität und der Datenservice sind bereits implementiert. Ihre Implementierung sollte Folgendes umfassen:

Tasks: Your goal in this assignment is to implement functionality to create, update, retrieve, and delete users. Your focus will be on the API and Data modules. The database table schema, the corresponding database entity, and the data service are already implemented. Your implementation should include:

- Eine Repräsentation eines Benutzers im API-Modul (`UserDto`) und im Domain-Modul (`User`) sowie ein entsprechender Mapper. Benutzer sollten die folgenden Felder haben:

A representation of a user in the API module (`UserDto`) and the domain module (`User`) as well as a corresponding mapper. Users should have the following fields:

(1 Punkt)

- ID (`Long`).
- Creation timestamp (`LocalDateTime`, automatically set upon creation).
- Update timestamp (`LocalDateTime`, automatically set upon creation and update).
- Login name (`String`, must only contain valid word characters, see Pattern class).
- Email address (`String`, must be a valid email address).
- First name (`String`, at least one and at most 255 characters).
- Last name (`String`, at least one and at most 255 characters).

Die Einschränkungen der Felder sollten im API-Modul mit Hilfe von Bean Validation überprüft werden. Statische Null-Checks sollten sowohl im API- als auch im Domain-Modul stattfinden.

The field constraints should be checked in the API module using Bean Validation. Static null checks should happen both in the API and the domain module.

- Einen neuen `UserController`, der die folgenden Operationen unter dem Basispfad `/api/users` bereitstellt. Sie müssen keine OpenAPI-Annotationen hinzufügen.

A new `UserController` that offers the following operations under the base path `/api/users`. You do not need to add OpenAPI annotations.

(2 Punkte)

- Abrufen aller Benutzer.
[Retrieving all users](#).
- Abrufen eines Benutzers basierend auf einer ID.
[Retrieving a user by ID](#).
- Abrufen eines Benutzers nach Anmeldename über einen Filter-Endpunkt (analog zum `PosController`).
[Retrieving a user by login name using a filter endpoint \(analog to PosController\)](#).
- Erstellen eines neuen Benutzers.
[Creating a new user](#).
- Aktualisieren eines bestehenden Benutzers.
[Updating an existing user](#).
- Löschen eines bestehenden Benutzers basierend auf einer ID.
[Deleting an existing user by ID](#).

- Die fehlenden Schnittstellen und Service-Implementierungen im Domain-Modul.

The missing interfaces and service implementations in the domain module.

(1 Punkt)

- Mindestens zwei zusätzliche Tests in der Klasse `UsersSystemTests`.

At least two additional tests in class `UsersSystemTests`.

(1 Punkt)

Stellen Sie sicher, dass `mvn clean install` das Projekt weiterhin erfolgreich baut. Sie können die `curl`-Befehle in der README-Datei verwenden, um Ihre Implementierung zu testen. Pushen Sie Ihre Änderungen in Ihren Fork und fügen Sie der Abgabe einen Link zu Ihrem Fork hinzu. Ensure that `mvn clean install` successfully builds the project. You can use the `curl` commands in the README file to test your implementation. Push your changes to your fork and add a link to your fork to the submission.

Einreichungsfrist:

- Dienstag, 02.12.2025, 14:00 Uhr

Nächste Übungsgruppen:

- Dienstag, 25.11.2025
- Mittwoch, 26.11.2025
- Donnerstag, 27.11.2025

Hinweise zur Abgabe:

- Wir verwenden das Format dd.mm.yyyy für **Datumsangaben** und das 24-Stunden-Format für **Uhrzeiten**.
- Fassen Sie alle Texte, Diagramme und andere Bestandteile Ihrer Abgabe in **einer PDF-Datei** zusammen.
- Die erste Seite der PDF-Datei muss Ihren **Namen** und Ihre **Matrikelnummer** enthalten; bei Gruppenabgaben die Namen und Matrikelnummern beider Gruppenmitglieder. Ob eine **Einzel- oder Gruppenabgabe** gefordert ist, ist oben auf dem Übungsblatt vermerkt.
- Die Abgabe von **Plagiaten** führt zur Bewertung mit **0 Punkten** für das gesamte Übungsblatt. Bei **Verdacht** der Verwendung von **GenAI-Tools** ohne Quellenangabe wird die Abgabe ebenfalls mit **0 Punkten** bewertet.
- Bei **UML-Aufgaben**, reichen Sie Ihre **Diagramme** als Teil der **PDF-Datei** ein. Optional können Sie Ihren **PlantUML-Quellcode** reinreichen.
- Bei **Programmieraufgaben**, reichen Sie Ihren Quellcode oder Ihre Konfigurationsdateien als Textdateien mit entsprechender Dateiendung ein (z.B. `.java` für Java-Quellcode oder `.yaml` für YAML-Konfigurationsdateien). Reichen Sie keine kompilierten Dateien ein (z.B. `.class` in Java). Für Aufgaben, die das Forken eines GitHub-Repositories erfordern, ist der Standardweg zur Einreichung Ihrer Lösung, dass Sie einen **Link zu Ihrem Fork** in die oben genannte PDF-Datei einfügen.
- Falls Sie **Windows** verwenden, empfehlen wir dringend, für die Bearbeitung der technischen Aufgaben **WSL** zu verwenden. Wir können unseren Code nicht unter Windows testen. Wenn Sie Probleme feststellen, informieren Sie uns bitte.
- Falls Sie kein **GitHub-Konto** anlegen möchten, können Sie Aufgaben, für die ein GitHub-Fork erforderlich ist, auch einreichen, indem Sie einen Link zu einer Kopie des Git-Repositories mit Ihren Lösungen in einem anderen **öffentlich oder universitätsintern gehosteten Git-Repository** angeben (Aufgaben-Repository klonen, Änderungen implementieren, in das Repository pushen, Link teilen). Beachten Sie jedoch, dass Sie damit die CI-bezogenen Aufgaben nicht bearbeiten können. Es steht Ihnen frei, GitLab als Alternative zu verwenden und die GitHub-Actions-Aufgaben an GitLab CI anzupassen. Für diese alternativen Vorgehensweisen können wir jedoch **keine Unterstützung** anbieten.
- Setzen Sie sich frühzeitig mit den benötigten **Abhängigkeiten und Tools** und deren Installation auseinander. Wir stellen für die Übungsblätter `mise.toml`-Dateien bereit, die die Installation von Abhängigkeiten vereinfachen (siehe Dokumentation von **mise**). Sie können die Abhängigkeiten auch manuell installieren, was wir aber nicht empfehlen. Sie werden auch eine lokale **Docker**-Installation benötigen. Alternativ können Sie Rancher Desktop installieren. Als Entwicklungsumgebung (IDE) empfehlen wir **IntelliJ IDEA**. Sie können auch andere Editoren wie z.B. Visual Studio Code mit entsprechenden Erweiterungen verwenden, wofür wir aber keine Unterstützung anbieten können. Alle Beispiele und Anleitungen basieren auf IntelliJ. Für mise gibt es verschiedene **IDE-plugins**.