

Pflichtaufgabe 1: User Interface und REST-Server

1. Rahmenbedingungen

Es gelten folgende Rahmenbedingungen:

- Die Pflichtaufgaben sind - neben der Klausur - Teil der Prüfungsleistung und müssen somit erfolgreich absolviert werden, um die Lehrveranstaltung Webtechnologien zu bestehen.
- Lösen Sie die Pflichtaufgaben in Gruppen bis maximal 4 Personen. Dafür müssen Sie Ihre Gruppe bis zum 20.04.2022 23.59 Uhr per Moodle anmelden. Achten Sie auf eine gleichmäßige Verteilung der Aufgaben, jedes Gruppenmitglied sollte an jeder Teilaufgabe beteiligt sein.
- Die Aufgaben müssen von Ihnen selbstständig ohne fremde Hilfe gelöst werden.
- Laden Sie Ihre Lösung für die erste Pflichtaufgabe bis zum 15.05.2022 23:59 Uhr als ZIP-Datei per Moodle hoch. Informationen zum Inhalt der Abgabe finden Sie im Abschnitt 3.6. Bitte halten Sie sich an die Vorgaben für die Datei- und Verzeichnisnamen.

2. Inhalt

Mit verschiedenen Webtechnologien soll ein Websystem zur Verwaltung von TV-Serien entwickelt werden: Steam - Streaming Series Memory.

Steam bietet die folgenden Möglichkeiten:

- Eine Benutzerin kann sich am System mit ihrem Namen anmelden (mit Passwortüberprüfung). Nach erfolgter Anmeldung wird ihre Seriensammlung angezeigt.
- Eine Benutzerin kann neue Serien anlegen. Dabei werden Titel, Anzahl der Staffeln (seasons), Genre und Streaming-Plattform einer Serie erfasst.
- Darüberhinaus kann eine Benutzerin für jede Serie ein Rating (good, bad, ...) und eine Bemerkung vergeben.

- Serien können nach **User**, **Genre**, **Plattform** sowie **Rating** (der eingeloggten Benutzerin) **gesucht** werden.

Um Ihnen die Aufgabe zu erleichtern, ist ein einfaches Backendsystem bereits implementiert. Das Backendsystem stellt dem Websystem die folgende Fassade zur Verfügung:

<pre> <<facade>> <<service>> Steamservices </pre>
<pre> <<static>> exemplar(): SteamServices newUser(username: String, pwd: String): Bool login(username: String, pwd: String): Bool addOrModifySeries(title:String, nbOfSeasons:int, g: Genre, s: StreamingProvider, userName:String, sc: Score, remark: String) search(username: String, g: Genre, s: StreamingProvider, sc: Score): List<Series> getAllSeries(): List<Series> getAllSeriesWithTitle(title: String): List<Series> getAllSeriesOfUser(username: String): List<Series> getRating(seriesname: String, username: String): Rating </pre>

3. Aufgaben

Im Rahmen dieser Pflichtaufgabe sollen Sie ein User-Interface in Form einer Website gestalten, sowie eine entsprechende REST-Services entwerfen und auch umsetzen. In der zweiten Pflichtaufgabe, die Sie in der zweiten Hälfte des Semesters erhalten werden, sollen Sie das Websystem vervollständigen, in dem Sie die Website und die REST-Services miteinander verknüpfen.

3.1. Wireframe

Damit Sie in Ihrer Gruppe ein gemeinsames Verständnis der zu entwickelnden Website schaffen, sollten Sie zunächst Wireframes erstellen. Diese müssen nicht perfekt sein, es soll hauptsächlich der Kommunikation im Team dienen und Ihnen bei der Verteilung der Aufgaben helfen. Achten Sie darauf, dass alle im Abschnitt 2 genannten Möglichkeiten berücksichtigt werden. Eine einfache Einführung in die Gestaltung von Wireframes finden Sie unter <https://www.youtube.com/watch?v=mbnrta-BK1w>.

3.2. Umsetzung von HTML5 und CSS3

Im vorherigen Schritt entworfenen Seiten sollen nun in HTML5 umgesetzt werden. Alle Seiten sollen außerdem eine (oder mehrere) gemeinsame CSS3-Dateien referenzieren. Verwenden Sie gerne Frameworks, z.B. Bootstrap. Weitere Assets (z.B. Grafiken, Icons) binden Sie optimiert in die Website ein.

- Alle von Ihnen erstellen Dateien sollen den jeweiligen Standards entsprechen. Nutzen Sie folglich unbedingt die Validation Services des W3C.

- Achten Sie darauf, dass Ihre CSS-Regeln möglichst wenig Redundanz und wiederholte Eigenschaftsdeklarationen aufweisen.
- Die von Ihnen gestalteten Seiten sollen “responsiv” sein, d.h. auf verschiedenen Endgeräten gut aussehen.

3.3. Installation des Backendsystems

Parallel zu dieser Datei steht Ihnen das steam-Projekt zur Verfügung.

- Machen Sie sich mit dem Quellcode vertraut.
- Verstehen Sie den Unterschied zwischen Entity-, Repository-, Service-, und Ressourcenklassen.
- Nutzen Sie die Testklasse, um die Methoden der Fassade zu verstehen.

3.3.1 Einrichten der Entwicklungsumgebung

Das Projekt steam verwenden Sie analog zu dem in der Übung verwendeten Beispiel-Projekt. Stellen Sie dazu insbesondere sicher, dass Sie aktuelle Versionen von Java, Maven und Glassfish verwenden.

Entpacken Sie das Paket parallel zu dem Beispiel-Projekt, sodass Ihre Verzeichnisstruktur wie folgt aussieht:

```
verzeichnis
|- glassfish6
|- pizza #Beispiel-Projekt
|- steam #Abschluss-Projekt
```

3.4. Entwurf einer REST-Schnittstelle

Die Funktionen des Backendsystems sollen über eine REST-Schnittstelle verfügbar gemacht werden. Dafür benötigen Sie zunächst einen sauberen Entwurf unter Berücksichtigung der Grundprinzipien von REST, welche Sie in einem Dokument beschreiben sollen.

- Definieren Sie, welche Ressourcen Ihre Schnittstelle anbieten soll. Orientieren Sie sich dabei an den von der Fassade des Backendsystems zur Verfügung gestellten Funktionen.
- Definieren Sie für jede Ressource ein URI-Template und die möglichen Verben. Beschreiben Sie für jedes Verb kurz dessen Funktion.
- Als Repräsentationsformate sollen Sie JSON und XML unterstützen. Beschreiben Sie für alle möglichen Anfragen und Antworten das Datenschema:

- Für das Repräsentationsformat JSON müssen Sie für alle Anfragen und Antworten Beispieldaten mit Erklärungen zu den einzelnen Datenfeldern erstellen.
- Für das Datenformat XML müssen Sie für alle möglichen Anfragen und Antworten jeweils eine DTD- oder XSD-Datei erstellen.

3.5. Entwicklung einer REST-Schnittstelle

Die im vorherigen Schritt entworfene REST-Schnittstelle muss nun umgesetzt werden.

- Fügen Sie dem Projekt, Ihrem Entwurf folgend, die erforderlichen RESTful Services hinzu.
- Verwenden Sie für die Umsetzung dieser Services die vorhandene Fassade des Backendsystems.
- Testen Sie Ihre RESTful Services mit einem geeigneten Client. Beschreiben Sie Ihre Testmethode in der Datei README.md

3.6. Abgabe

Erstellen Sie eine ZIP-Datei mit dem Namen Ihrer Gruppe (z.B. Gruppe01.zip) und folgendem Inhalt:

- steam: In diesem Verzeichnis soll sich Ihr Java-Projekt befinden. Bitte geben Sie nur ein sauberes Projekt ohne automatisch generierte Artefakte (z.B. target Verzeichnis) ab.
- steam/README.md: Listen Sie in dieser Markdown-Datei alle Mitglieder Ihrer Gruppe auf (Name, Vorname, Matrikelnummer) und beschreiben Sie kurz den Beitrag jedes einzelnen Mitglieds zur Lösung der Aufgabe. hier sollen Sie auch weitere allgemeine Informationen zu Ihrer Abgabe hinterlegen (z.B. Hinweise für den Test)
- steam/doc/wireframe.pdf: Diese Datei enthält die von Ihnen erstellen Wireframes. Wenn Sie diese auf Papier gezeichnet haben, können Sie diese einscannen.
- steam/src/main/webapp: In diesem Verzeichnis legen Sie bitte alle erstellen HTML, JavaScript, CSS und Asset-Dateien für das User Interface ab.
- steam/doc/api.pdf: Diese Datei enthält die von Ihnen erstellte Beschreibung Ihrer REST-Schnittstelle. Im Verzeichnis steam/doc/api legen Sie bitte die zugehörigen JSON- und DTD- bzw. XSD-Dateien ab.
- steam/src/main/java/de/hsh/steam: Bitte legen Sie den Quellcode für die REST-Services in diesem Verzeichnis bzw. in passenden Unterverzeichnissen ab.