

---

**ÜBUNG 1 – MEILENSTEIN 3 - FINALE ABGABE**

---

LV	WEB-ENGINEERING II		
THEMA	REST-SERVER	FÄLLIGKEIT	SIEHE MOODLE

---

## FINALE ABGABE/ MEILENSTEIN 3

Grundlage für die Umsetzung und Benotung dieser finalen Abgabe von Übung 1 sind die „Allgemeinen Regelungen für Übungsaufgaben“ sowie die Aufgabenstellung für Übung 1, die Sie beide im Moodle finden.

Die Vorgaben für Meilenstein 1 und 2 gelten auch für diese finale Abgabe. Sollten Sie Anforderungen von einem Meilenstein bei den letzten Abgaben nicht erfüllt haben, müssen sie zu diesem Meilenstein umgesetzt sein.

Die funktionalen Anforderungen und allgemeinen Anforderungen werden vor allem in der Aufgabenstellung von Übung 1 beschrieben. Nachfolgend gibt es noch einige Aspekte, die Sie bei der finalen Abgabe beachten sollten.

### FUNKTIONALE ANFORDERUNGEN

Für die finale Abgabe sollte nun alle Anforderungen der Aufgabenstellung umgesetzt werden. Das bedeutet insbesondere, dass der noch fehlende Endpoint für die Studienbewerbungen hinzugefügt wird.

Bei der Umsetzung des Endpoints sind die folgenden Aspekte zu berücksichtigen:

- Die Resource-ID für den REST-Service ist „degreeCourseApplications“.
- Der Server soll nun auf HTTPS umgestellt werden.

Neben den CRUD-Methoden für Studienbewerbungen sollen auch die folgenden funktionalen Anforderungen erfüllt werden. Die entsprechenden Tests für die funktionalen Anforderungen finden Sie wie bei den anderen Meilensteinen im Moodle.

### ***Anlegen einer Studienbewerbung durch einen User***

Ein Nutzer kann eine Studienbewerbung für sich anlegen. Dabei wird die User-ID über den Token bestimmt.

```
POST https://localhost/api/degreeCourseApplications
Authorization: {{manfredToken}}
Content-Type: application/json

{
  "degreeCourseID": {{degreeCourseID}},
  "targetPeriodYear": 2024,
  "targetPeriodShortName": "WiSe"
}
```

### ***Suche nach den Studienbewerbungen des aktuellen Nutzers***

In der Web-Anwendung soll der Studierende seine eigenen Studienbewerbungen ansehen können. Hierzu soll der folgende Request umgesetzt werden:

```
### Studienbewerbungen für manfred auflisten (es sollte 1 sein)
GET https://localhost/api/degreeCourseApplications/myApplications
Authorization: {{manfredToken}}
```

Die ID des aktuellen Users wird über den Token bestimmt.

### ***Suche nach den Studienbewerbungen für einen beliebigen Nutzer***

Der Administrator soll die Möglichkeit haben, die Studienbewerbungen für einen beliebigen User abzurufen. Hierzu soll die folgende Suche umgesetzt werden.

```
### Der Admin sucht nach Bewerbungen von Manfred
GET https://localhost/api/degreeCourseApplications?applicantUserID=manfred
Authorization: {{adminToken}}
```

Diese Suche sollen nur User durchführen können, die Administrator sind.

### ***Suche nach Studienbewerbungen für einen Studiengang***

Der Administrator soll die Möglichkeit haben, die Bewerbungen für einen bestimmten Studiengang abzurufen.

```
### Auflisten der Bewerbungen für einen Studiengang
GET https://localhost/api/degreeCourseApplications?degreeCourseID={{degreeCourseID}}
Authorization: {{adminToken}}
```

Auch diese Suche sollen nur Administratoren ausführen können.

Ergänzend zu diesen Zusatzfunktionen sollten alle weiteren Funktionen, die in den REST-Client-Tests für Meilenstein 3 enthalten sind, umgesetzt werden. Hierzu gehört insbesondere auch die nachgelagerte Suche nach Studienbewerbungen über den Endpoint von Studiengängen:

```
### Nachgelagerte Suche für Studienbewerbungen
GET https://localhost/api/degreeCourses/{{degreeCourseID}}/degreeCourseApplications
Authorization: {{adminToken}}
```

## **ALLGEMEINE ANFORDERUNGEN**

Abweichend von den Meilensteinen 1 und 2 sollte der REST-Server jetzt nicht mehr über http, sondern über HTTPS erreichbar sein. Hierzu sollten Sie sich ein Zertifikat erstellen und dann entsprechend einbinden. Der Server sollte nun über den Standard-Port von HTTPS 443 erreichbar sein.

## **ZUSATZLEISTUNGEN**

In der finalen Abgabe können Sie Zusatzleistungen umsetzen und sich anrechnen lassen. Es können dafür bis zu 20 Punkte angerechnet werden. Mögliche Zusatzleistungen sind:

- **Hinzufügen des Bewerbungsstatus mit Benachrichtigungen, wenn der Status geändert wird:** Die Bewerbung kann ergänzt werden um den Status der Bewerbung. Beispielsweise könnte die Stati „angelegt“, „angenommen“ und „abgelehnt“ umgesetzt werden. Wenn der Status geändert wird, sollte eine Nachricht an den Bewerber gesandt werden.
- **Registrierung:** Einen Service zum Registrieren von neuen Nutzern. Hierzu können sich beispielsweise Nutzer mit ihrer E-Mail-Adresse registrieren. Der Service schickt dann eine E-Mail an den User mit einem Aktivierungslink. Wenn der User dann den Aktivierungslink anklickt, wird der Account aktiviert. Wenn alle Aspekte inklusive der E-Mail umgesetzt sind, können bis zu 5 Zusatzpunkte angerechnet werden. Hinweis: Ein einfaches Anlegen von Usern ohne die Aktivierung über einen Rest-Call kann nicht als Registrierung angerechnet werden.
- **Validierung von Entitäten:** Es gibt zahlreiche Module für die Validierung der Daten, die bei Anfragen übertragen und vom REST-Server verarbeitet werden sollen. Die Umsetzung einer Datenvalidierung ist sehr sinnvoll. Die Zusatzpunkte hängen vom Umfang der umgesetzten Validierungen ab.
- **Newsletter:** Sie können einen Service zum Verteilen von Newslettern oder anderen Benachrichtigungen an ausgewählte User umsetzen. Je nach Umfang der Umsetzung können bis zu 5 Zusatzpunkte angerechnet werden.
- **Automatisierte Tests:** In Ergänzung zu den manuellen Tests, die über die REST-Client-Extension ausgeführt werden, können automatisierte Tests über Test-Suiten (z.B. Jest, Mocha, Chai, etc.) umgesetzt werden. Je nachdem wie umfangreich die Tests sind, können in der Regel 5, bei sehr umfangreichen Tests bis zu 10 Zusatzpunkte angerechnet werden.
- **Continuous Integration:** In Ergänzung zu den automatisierten Tests kann im GIT eine CI-Pipeline definiert und bei jedem Commit ausgeführt werden. In diesem Fall können weitere 3 Zusatzpunkte angerechnet werden.
- **Bilder-Upload:** Es können Profilbilder zu den Usern hochgeladen und mit dem User verbunden werden. Es sollte dann noch einen Endpoint zum Abrufen der Bilder wie bei einem Web-Server umgesetzt werden, um sie in Übung 2 zu nutzen. Je nach Umsetzung können hierfür bis zu 5 Zusatzpunkte angerechnet werden. Dafür sind jedoch entsprechende Tests per REST-Client für das Hochladen und Abrufen umzusetzen! Es reicht nicht, den File-Upload aus dem Vorlesungsvideo zu übernehmen.
- **Umsetzung von weiteren Entitäten:** Wenn Sie weitere Entitäten, die zu der Anwendung passen, umsetzen, können je nach Qualität der Umsetzung bis zu 5 Zusatzpunkte je Entität angerechnet werden.

**Anmerkung:** Es können nur Zusatzpunkte vergeben werden für Implementierungen, die nicht in meinen Vorlesungsfolien oder -videos beschrieben werden!

Das sind nur einige Vorschläge für Zusatzleistungen, die Sie umsetzen können. Sollten Sie noch andere Ideen für Zusatzleistungen haben, die besser zu Ihrer Anwendung passen, können Sie mir diese in einer Übung vorschlagen.

Beschreiben Sie bei der Abgabe des Meilensteins im Moodle Ihre Zusatzleistungen in einem separaten Dokument. Das Dokument sollte korrekt formatiert sein (Titelseite, korrekt formatierte Abschnitte, etc.). Erläutern Sie in dem Dokument Ihre Zusatzleistungen kurz aber gut verständlich. Ergänzen Sie ihre Beschreibung gegebenenfalls mit Screen Shots.

**Wichtig:** Wenn Sie Zusatzleistungen erbringen, muss in jedem Fall die abgegebene Lösung in einer persönlichen Abnahme angenommen werden. In dieser Abnahme werden dann die Zusatzleistungen gemeinsam durchgesprochen und Fragen zur Umsetzung gestellt.

**Viel Erfolg!**