



### Aufgabe 0.9 (1 Punkt)

Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

### Aufgabe 19 (8 Punkte)

Im von der Veranstaltungsseite ladbaren Projekt `qsAufgabeWebServiceTest` befindet sich u. a. im Verzeichnis `service` eine unter Windows 64Bit ausführbare Datei `WebService.exe`, die ein erstes unvollständiges Inkrement zur Verwaltung von Studierenden und deren Prüfungen für Module darstellt. Der Service hat bereits einige Daten. Genauer werden folgende Dienste bisher angeboten. Dabei werden übergebene Werte bisher nicht auf ihre Plausibilität geprüft, was Sie damit auch ignorieren sollen. Weiterhin sind Warnungen des Servers zu ignorieren. Werden die Dienste aber mit Parametern falschen Typs aufgerufen oder Studierende oder Module referenziert, die nicht existieren, liefern die Dienste einen HTTP-Fehlerstatus zurück.

Typ	Dienst	
GET	Aufruf	<code>/studi</code>
	Beschreibung	gibt die Liste aller vorhandenen Studierenden zurück
	Ergebnis	<code>{ "Studis": [ &lt;Array mit Studi-Objekten&gt; ] }</code> (Studi-Objekte weiter unten genauer spezifiziert)
GET	Aufruf	<code>/modul</code>
	Beschreibung	gibt die Liste aller vorhandenen Module zurück
	Ergebnis	<code>{ "Module": [ &lt;Array mit Modul-Objekten&gt; ] }</code> (Modul-Objekte weiter unten genauer spezifiziert)
GET	Aufruf	<code>/studi/{id}</code>
	Beschreibung	gibt den Studi mit der Matrikelnummer <code>{id}</code> (Typ <code>int</code> ) zurück
	Ergebnis	<code>{</code> <code>"id": 1,</code> <code>"name": "Leila",</code> <code>"pruefungen": [ &lt;Array mit Prüfung-Objekten&gt; ]</code> <code>}</code> oder Status 400 Bad Request, wenn Studi nicht existiert (Prüfung-Objekte weiter unten genauer spezifiziert)
GET	Aufruf	<code>/modul/{id}</code>
	Beschreibung	gibt das Modul mit der Modulnummer <code>{id}</code> (Typ <code>int</code> ) zurück
	Ergebnis	<code>{</code> <code>"id": 1,</code> <code>"name": "SQ",</code> <code>"cp": 5</code> <code>}</code> oder Status 400 Bad Request, wenn Modul nicht existiert
GET	Aufruf	<code>/studi/{sid}/modul/{mid}</code>
	Beschreibung	gibt für den Studi mit der Matrikelnummer <code>{sid}</code> und das Modul <code>{mid}</code> alle bisher durchgeführten Prüfungen zurück
	Ergebnis	<code>{ "Pruefungen": [ &lt;Array mit Prüfungsobjekten&gt; ] }</code> sollte der Studi noch keine Pruefung abgelegt haben, ist das Array leer oder Status 400 Bad Request, wenn Studi oder Modul nicht existiert (Prüfung-Objekte weiter unten genauer spezifiziert)
GET	Aufruf	<code>/studi/{sid}/modul/{mid}/pruefung/{pid}</code>
	Beschreibung	gibt für den Studi mit der Matrikelnummer <code>{sid}</code> das Modul <code>{mid}</code> und die Pruefungs-Id <code>{pid}</code> (Typ <code>int</code> ) dessen Prüfungsergebnis in diesem Modul zurück
	Ergebnis	<code>{</code> <code>"id": 6,</code> <code>"modul": {</code> <code>"id": 3,</code> <code>"name": "Mathe 1",</code> <code>"cp": 75</code> <code>}</code>



		<pre>},   "wert": 100 }</pre> oder Status 400 Bad Request, wenn Studi oder Modul oder Pruefung des Studis zu diesem Modul nicht existiert
PUT	Aufruf	<code>/studi/{sid}/modul/{mid}/pruefung/{pid}</code>
	Parameter	<pre>{   "wert": 230 }</pre>
	Beschreibung	ändert für den Studi mit der Matrikelnummer {sid} das Modul {mid} und die Pruefungs-Id {pid} dessen Prüfungsergebnis in diesem Modul den Wert (Typ int) auf den übergebenen Wert ab
	Ergebnis	bei Erfolg Status 200 OK und der Aufrufpfad steht im Header-Attribut Location sollte sid, pid oder mid nicht existieren oder zusammenpassen oder ein sonstiger Fehler auftreten, ist das Ergebnis Status 400 Bad Request
POST	Aufruf	<code>/studi</code>
	Parameter	<pre>{   "Name": "Sandra" }</pre>
	Beschreibung	erzeugt einen neuen Studi mit dem übergebenen Namen (Typ String)
	Ergebnis	bei Erfolg Status 201 Created und der Aufrufpfad für das neue Studi-Objekt steht im Header-Attribut Location sollte ein Fehler auftreten, ist das Ergebnis Status 400 Bad Request
POST	Aufruf	<code>/studi/{sid}/modul/{mid}/pruefung</code>
	Parameter	<pre>{   "wert": 230 }</pre>
	Beschreibung	erzeugt ein neues Pruefungsergebnis für den Studi {sid} und das Modul {mid} und der Note Wert (Typ int)
	Ergebnis	bei Erfolg Status 201 Created und der Aufrufpfad für das neue Pruefung-Objekt steht im Header-Attribut Location sollte ein Fehler auftreten, ist das Ergebnis Status 400 Bad Request
POST	Aufruf	<code>/modul</code>
	Parameter	<pre>{   "Name": "SW Projekt",   "Cp": 100 }</pre>
	Beschreibung	erzeugt ein neues Modul mit dem übergebenen Namen (Typ String) und den angegebenen Credit Points (Cp, Typ int)
	Ergebnis	bei Erfolg Status 201 Created und der Aufrufpfad für das neue Modul-Objekt steht im Header-Attribut Location sollte ein Fehler auftreten, ist das Ergebnis Status 400 Bad Request

Hinweis: Die Services vergeben nur positive Ids. Ids werden nur vom Service vergeben.

- Schreiben Sie Systemtests mit JUnit die alle angebotenen Dienste aufrufen. Nutzen Sie dabei u. a. Ihr Wissen über die systematische Testerstellung (u. a. Äquivalenzklassen), um positive, wie negative Fälle zu testen. Oftmals ist es sinnvoll, mehrere Dienstnutzungen in Tests zu kombinieren. Mindestens einer der Dienste ist auch nach obiger Spezifikation fehlerhaft.
- (freiwillig) Laden Sie das Programm cURL von <https://curl.haxx.se/download.html> und führen Sie für jeden Dienst jeweils eine erfolgreiche und eine erfolglose Nutzung durch. Dokumentieren Sie Ihre Anfragen und Ergebnisse.
- (freiwillig) Laden Sie das Programm Postman von <https://www.postman.com/downloads/> oder portabel von <https://portapps.io/app/postman-portable/> und führen Sie für jeden Dienst jeweils eine erfolgreiche und eine erfolglose Nutzung durch. Dokumentieren Sie Ihre Anfragen und Ergebnisse.

Hinweise: Schauen Sie auf die Hilfsmethoden der Klasse test.Execute. Diese Aufgabe eignet sich diesmal, dass jedes Gruppenmitglied einen Lösungsteil erstellt.