

Internetkommunikation – Nakeu

Übungsblatt 12 – RESTful API

Am Ende dieses Praktikums soll der Student in der Lage sein, eine REST-API zu erkennen, er soll eine REST-API designen können und alle URIs einer Ressource identifizieren können. Er soll nach dem Besuch des Praktikums den Einsatz anwendungsnaher Protokolle wie REST beurteilen können.

Für das Praktikum benötigen Sie folgenden Komponenten:

- Betriebssystem (nach ihrer Wahl)
- Node JS
- Editor
- Git
- Datenbank (MongoDB)
- Postman
- Docker

Betriebssystem

Für dieses Praktikumsblatt benötigen Sie einen Computer mit einem Betriebssystem Ihrer Wahl. Die im Laufe dieses Übungsblatt vorgestellten Programme laufen sowohl auf Windows, macOS, also auch Linux.

Node JS

Sie werden in den Praktikum **Node JS** verwenden, es ist ausreichend, wenn Sie sich einigen Basis vertraut machen.

Über den link: [hier](#), können Sie die Basis lernen.

Editorsystem

Installieren Sie ein Editor ihrer Wahl auf ihrem Betriebssystem. Ich persönlich verwende Visual Studio Code. Unter dem folgenden Link können Sie sich VS-Code herunterladen: [hier](#).

Datenbank

In diesem Übungsblatt wird die Datenbank-Technologie MongoDB verwendet. Damit Sie im Laufe dieses Übungsblattes MongoDB verwenden können, müssen Sie diese Datenbank-Technologie zunächst "installieren". Dazu haben sie drei Möglichkeiten.

Zum einen bietet sich an, MongoDB lokal auf ihren Computer zu installieren. Ein Tutorial diesbezüglich finden Sie unter dem folgenden Link [hier](#). Hierbei handelt es sich um eine Zeitintensive Variante.

Des Weiteren können Sie MongoDB als Cloudvariante verwenden. Unter der folgenden URL können Sie sich eine kostenfreie Instanz einer MongoDB installieren. Persönlich habe ich diese Variante verwendet.

Außerdem bietet es sich an, MongoDB als Docker Container zu betreiben. Informationen diesbezüglich finden Sie unter folgenden Link: [hier](#).

Damit Sie sich von ihrem Computer auf die MongoDB verbinden können, können Sie entweder die offizielle MongoDB-Bibliothek verwenden oder Sie benutzen die Mongoose-Bibliothek.

Webservice

Um RESTful-API zu verwenden, bietet der Markt eine Auswahl von verschiedenen Tools an (hier-link-einfügen). In diesem Übungsblatt verwenden wir die Software Postman. Sie können die Software unter dem folgenden Link herunterladen: [hier](#).

Git

Sie werden in diesem Praktikum Git verwenden. Sollten Sie bislang noch keine Erfahrung mit Git gesammelt haben, können Sie sich unter den folgenden Links Grundlagen aneignen:

- [link 1](#)

- [link 2](#)

Es gibt ein GIT von der Hochschule und ich empfehle Ihnen es zu verwenden: [hier](#) den Link

Docker

Des Weiteren verwenden wird in diesem Übungsblatt die Virtualisierungssoftware Docker eingesetzt. Bitte besuchen Sie den folgenden Link [hier](#) und verschaffen Sie sich Grundlagen zu Docker.

Besuchen Sie den link: [hier](#), um die Basis über Docker für die nächste Aufgaben zu haben.

Aufgabe 1 – Use Cases

Lösen Sie die folgenden Aufgaben:

Sie haben bereits in der Vorlesung einige Use-Cases der REST-API kennengelernt.

- Nennen Sie zwei Use Cases der REST-API.
- Beschreiben Sie eine REST-API Architektur.
- Nennen Sie drei REST Grundprinzipien.
- Warum soll die Kommunikation von REST statuslos erfolgen?
- Wie soll der Client und der Server verbunden werden?
- Nennen Sie drei HTTP-Methoden und beschreiben Sie diese mit Beispielen.

- Welcher Status-Code hat einen erfolgreicher Request bei der REST API?
- Geben Sie ein Beispiel eines POST Requests.

Aufgabe 2 – Design einer REST-API

Lösen Sie die folgenden Aufgaben:

- Was ist eine Ressource einer REST-API?
- Was sind die wichtigsten URIs beim Design einer REST-API?
- Beschreiben Sie, warum das Design einer REST-API so wichtig ist.
- Warum sollte eine REST-API die URIs benutzen?
- Warum sollen die URIs einer API stabil sein?
- Beschreiben Sie, warum die Methode „Connect“ nicht sicher, nicht idempotent und nicht cacheable ist.
- Identifizieren Sie alle Bestandteile der folgenden URI:
" <http://localhost:8080/username / 141203> "
- Was sind die sicheren Methoden und warum sind Sie als sicher definiert?

Aufgabe 3 – Fragen zu REST Komponenten

Lösen Sie die folgenden Aufgaben:

Um ein REST Szenario zu erstellen, benötigt man eine Datenbank.

- Nennen Sie drei verschiedene Datenbanken, die Sie kennen und geben Sie eine kurze Beschreibung von diesen.
- Welche API-Clients kennen Sie und welche Funktionsweise ermöglichen diese für eine REST-API?
- Was sind die syntaktischen Kernkomponenten einer HTTP-Anfrage?
- Beschreiben Sie, wie die syntaktische Kernkomponenten einer HTTP-Antwort aussehen.

Aufgabe 4 – Best Practices

Bearbeiten Sie die folgenden Aufgaben:

- Installieren Sie auf ihrem Computer die folgenden Applikationen:
 - NodeJS([Hier](#)), Postman([Hier](#)), Git([Hier](#)).
- Erstellen Sie eine Datenbank über den link <https://www.mongodb.com/cloud>.
- Wählen Sie „Kostenlos testen“ und geben Sie die folgenden Informationen an:
 - Warum Sie MongoDB verwenden "Ich lerne MongoDB"
 - E-Mail, Vor- und Nachname und dann Passwort

- Benennen Sie Ihre Organisation und Ihr Projekt (müssen Sie nicht!)

Und **klicken** Sie auf „Überspringen“

- Gehen Sie auf „Cluster Erstellen KOSTENLOS“
 - Cloud-Anbieter und Region → [AWS](#)
 - Empfohlene Region -> [Frankfurt](#)
 - Auf Cluster erstellen **klicken**
 - (Bis 3 Minuten warten)
- Gehen Sie auf **Verbinden**, dann auf **1- Fügen Sie Ihre aktuelle IP-Adresse hinzu** und dann auf **IP-Adresse hinzufügen**
- **2-Erstellen Sie einen Datenbankbenutzer** (Nutzername und password) und klicken Sie auf **Datenbankbenutzer erstellen**
- Klicken Sie jetzt auf **Wählen Sie eine Verbindungsmethode**
- Gehen Sie auf Verbinden Sie Ihre Anwendung.
- Ändern Sie die folgenden Attribute „Password, durch das Kennwort für den Benutzer (Nutzername) und ändern Sie auch myFirstDatabase“ durch den Namen der Datenbank und speichern Sie den Link in einer Notiz. Schließen sie anschließend den Dialog.
- Gehen Sie auf Collection



- erstellen Sie Ihre eigene Collection: -> Eigene Datenbank Hinzufügen



- Geben Sie eine DATENBANK NAME und eine KOLLEKTIONSNAME (Hinweis: Sie müssen zwei Kollektions erstellen) an.

Create Database

DATABASE NAME ?

Enter database name

COLLECTION NAME ?

Enter collection name

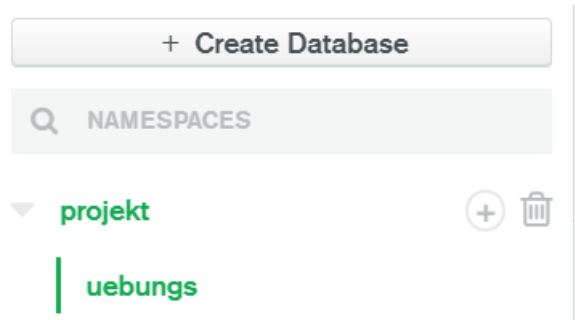
Capped Collection

Before MongoDB can save your new database, a collection name must be specified at the time of creation.

Cancel

Create

- Erstellen Sie die zweite Kollektion (drücken Sie auf „+“ und geben Sie den Name der zweiten Kollektion an).



Sie sollten am Ende die folgende Kollektion erstellt haben: (**uebungs, loesungs**).

- Sie haben die Möglichkeit, das Projekt per SSH zu clonen mit den Link:

<git@git.fslab.de:pnakeu2s/praktikumblatt-restful-api.git>

oder mit HTTPS mit dem Link

<https://git.fslab.de/pnakeu2s/praktikumblatt-restful-api.git>

Ersetzen Sie in der Datei **apps.js** `process.env.DB_CONNECTION` durch die gespeicherter Datenbankverbindungslink, so dass Sie:

```
//connected to the DB
mongoose.connect(="mongodb+srv://dbUser:dbUserpass@cluster0.vyuyh.mongodb.net/Project?retryWrites=true&w=majority", {useUnifiedTopology: true}, () =>
  console.log('connected to DB'));
```

haben.

- Öffnen Sie lokal ein Terminal und installieren Sie folgenden Packages:
 - `npm init`
 - `npm update`
 - starten Sie das Projekt mit dem Befehl "`npm run start`" und beschreiben sei das Ergebnis.
 - Beschreiben Sie das Ergebnis.
- Starten Sie nun Postman und führen Sie folgende Requests aus:
 - GET localhost:6000/ version
 - GET localhost:6000/loesung/all
 - DELETE localhost:6000/aufgabe/delete/94586

Beschreiben Sie die Response.

- Laden Sie eines der **.pdf-Datei** hoch, die Sie nicht verwenden mit den folgenden Request und beschreiben Sie die response.
 - ✓ POST localhost:6000/loesung/upload
(Body -> form-data,
Key „file“ ->Value „ihre Datei“

Key „id“ -> Value „Id auf ihrer Wahl“)
 - ✓ POST localhost:6000/aufgabe/upload
(Body -> form-data,
Key „file“ ->Value „ihre Datei“

Key „id“ -> Value „Id auf ihrer Wahl“)
 - ✓ DELETE localhost:6000/loesung/delete/“die Erste ID auf ihrer Wahl“

Öffnen Sie einen Browser auf ihren Computer und rufen Sie die beschriebenen URLs auf. Weil das Projekt online gebracht worden ist.

- <http://api.internetkommunikation-hbrs.de/version>
- <http://api.internetkommunikation-hbrs.de/loesung/all>
- <http://api.internetkommunikation-hbrs.de/aufgabe/all>
- Welche Schwierigkeiten hatten Sie bei der Bearbeitung der Aufgabe?