

Aufgabenblatt 5: Entwicklung eines REST-Service

Ziel dieses Aufgabenblattes ist es, die in der Vorlesung besprochenen Konzepte zu REST an einem kleinen praktischen Beispiel einzuüben und dabei Angular (Client-Seite) sowie PHP (Server-Seite) zu vertiefen.

Aufgabenstellung

Die mit Angular entwickelte Seite aus Aufgabenblatt 3 soll nun um einen Server-Anteil ergänzt werden. d.h. alle Daten zur Person bzw. die Login-Informationen sollen anstelle von LocalStorage in entsprechenden Dateien auf Serverseite gespeichert werden. Über einen mit PHP realisierten REST-Service (siehe hierzu den Beispiel-Service zur Bestellungsverwaltung aus der Vorlesung) können die Daten gelesen bzw. gespeichert werden.

Folgende Funktionalitäten sollen durch den REST-Service zur Verfügung gestellt werden:

1. GET <Service-Pfad>/person
Liefert die Personenbeschreibung als JSON-Objekt. Die Beschreibung sollte durch den Service aus der Datei „person.txt“ gelesen werden. Ein Beispiel für eine Personenbeschreibung wäre etwa:

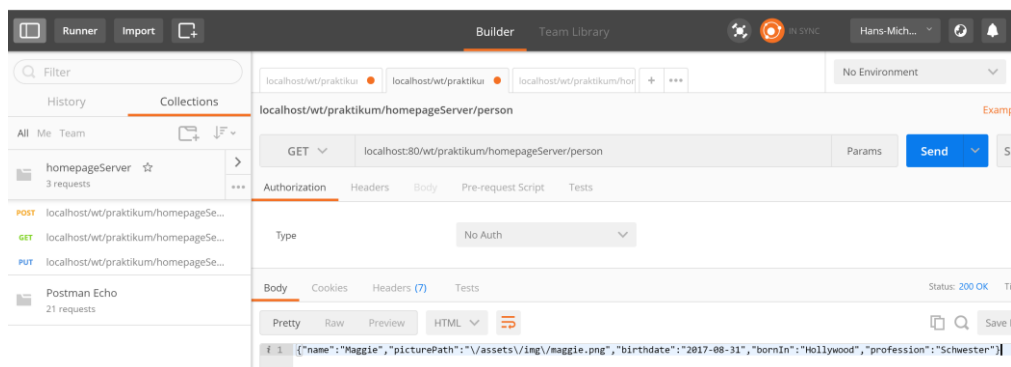
```
{ "name": "Bart S.", "birthdate": "15.7.2004, 00:00:00", "ort": "Hollywood",  
  "beruf": "who cares!", "bildpfad": "dataVbart.jpeg" }
```
2. PUT <Service-Pfad>/person
Speichert die geänderte Personenbeschreibung (übergeben als JSON-Objekt im Nachrichtenkörper) in der Datei „person.txt“.
3. POST <Service-Pfad>/login <Benutzername/Passwort> (übergeben als JSON-Objekt im Nachrichtenkörper)
Prüft anhand der in der Datei „benutzer.txt“ abgelegten Daten, ob Benutzername und Passwort korrekt sind. Liefert im Erfolgsfall den HTTP-Fehlercode 200 bzw. andernfalls den Code 404 zurück. Anmerkung: Der POST-Befehl passt hier, weil im Erfolgsfall typischerweise (nicht hier!) eine Session-ID erzeugt und zurückgeliefert wird (POST soll ja verwendet werden, um Objekte zu erzeugen).
Die Datei „benutzer.txt“ soll so aussehen (hier mit Beispieldaten):

```
benutzername=Hugo  
password=123
```

Vorgehensweise

1. Entwickeln Sie die Serverseite, indem Sie das Beispiel aus der Vorlesung als Grundlage verwenden.
2. Passen Sie dann den Dispatcher-Teil an die oben aufgeführte Schnittstelle an und definieren Sie die Service-Methoden.

3. In diesen können Sie dann Methoden der Persistenzklasse nach Bedarf aufrufen.
4. Die Persistenzklasse aus dem Beispiel sollte ebenfalls nach Bedarf angepasst werden. Sie benötigen im Wesentlichen 3 Methoden zum Laden und Speichern der Personendaten sowie zum Laden der Benutzerdaten (Benutzername, Passwort).
5. Zur Entwicklung der Serverseite ist es empfehlenswert, diese zunächst getrennt vom Client zu entwickeln. Zum **Testen ihrer REST-Schnittstelle** können Sie „Postman“ verwenden. Postman ist eine kostenlose Chrome-App, mit der man REST-Services sehr einfach aufrufen und damit testen kann (s. Abbildung unten). In der Datei „homepageServer.postman_collection.json“ sind Beispiel-Aufrufe für die REST-Schnittstelle enthalten (passen Sie bitte ggf. den Port und die Pfade an!). Damit können per PUT-Befehl Personendaten erzeugt und per GET-Befehl die Daten gelesen werden. Der POST-Befehl dient zum Testen der Authentifizierung. Die Datei kann von Postman eingelesen werden (Import-Button und z.B. Drag&Drop).



Hinweise

Entpacken und Ausführen von Client und Server

Für Client und Server gibt es im Zip-Archiv „fragment.zip“ zwei Lösungsfragmente, welche Sie entsprechend vervollständigen können. Im Ordner homepageClient finden Sie den angepassten AngularClient, im Ordner homepageServer finden Sie ein Fragment für ihren PHP-REST-Server.

- Client:
 - Verzeichnis mit VS Code öffnen, Terminalfenster öffnen (analog Aufgabe 3)
 - „npm install“ ausführen, dann „npm start“ (kein ng serve!)
- Server: Fragment unter htdocs in einem Verzeichnis entpacken und vervollständigen

Angular-Client

Der Angular-Client wurde analog zu den Beispielen aus der Vorlesung folgendermaßen angepasst:

1. Der SpeicherService soll zum Laden und Speichern der Personen-Beschreibung nicht mehr den LocalStorage, sondern den neuen REST-Service nutzen.
2. Es wurde ein neuer Service „AuthentifizierungsService“ mit einer Methode „authentifiziere(benutzername, password): boolean“ entwickelt, der aufgerufen wird, wenn eine Authentifizierung durch den REST-Server durchgeführt werden soll.
3. Zum Aufruf des REST-Service wurde der vordefinierte Angular-Service „HttpClient“ verwendet. Unter <https://angular.io/guide/http> finden Sie Informationen zur Nutzung dieser API.