

Implementierung REST-Client

Gegeben war ein Server mit REST-Schnittstelle (*.jar file auf dem Desktop, nach dem Starten war die genaue Doku auf dem Server unter localhost verfügbar).

Folgender Ablauf:

1. **Authentifizierung** mittels Username und Passwort per POST, als Antwort gibts einen X-Auth-Token welcher bei den folgenden Operationen im Header mitgeschickt werden muss.
2. Ein Menü war zu implementieren mit den Funktionen **GetAllJokes**, **GetJokesWithHighestRating** und **GetJokeOfTheDay**
3. **GetAllJokes**: GET auf die URL localhost/api/jokes?page={}&size={}. Es mussten die retournierten Jokes angezeigt werden. Mittels der UI waren die Funktionen *PreviousPage* und *NextPage* zu realisieren. D.h. page gibt an welche Page zurückgeliefert werden soll und size die Anzahl an Jokes pro Page.
4. **GetJokesWithHighestRating**: im Prinzip das gleiche wie GetAllJokes nur eine andere URL (wiederum die gleiche Navigation)
5. **GetJokeOfTheDay**: GET ohne Parameter

Die Kommunikation mit dem Server war mittels JSON. Es musste der JSON String in das entsprechende In-Memory Objekt umgewandelt werden um es anzuzeigen.

Das **Joke** Objekt hatte folgende Felder:

- id (int)
- Date (String)
- Text (String)
- Rating (double)

Es konnte frei gewählt werden wie der Client realisiert werden soll. Command-Line war genauso okay wie ein Web-Interface oder eine Desktop-GUI. Die Tutoren schauen nur ob alle Funktionen realisiert wurden.

Commandline-Bsp:

```
Username: 07032016
Passwort: 07032016
Authentication Successfull. Recieved Token: lasidfjoi34u098aj0sfaasldkf302
```

====Menü====

- [1] All Jokes
- [2] Jokes with highest Rating
- [3] Joke of the Day
- [4] Exit

====All Jokes====

```
Page 1/3
<Joke ID>
<Date>
```

```
<Text>
<Rating>
...
...
[N] Next, [P] Previous, [M] Menu
```

Alternativ kann natürlich auch eine GUI implementiert werden.