Projektaufgabe TINF15B

Deadlne:
Letzter Push: 31.07.2017
Prüfung gibt 60 Punkte. (Diese sind auf die Features verteilt)
Verteilung:
Funktionalitäten: 50 Punkte
Code: 10 Punkte

Aufgabenstellung:
Um einen Blog zu implementieren werden API-Routen benötigt:
Zu implementieren ist also eine Web-Api welche Daten im JSON-Format an einen Client sendet. **(Eine Implementierung eines Frontend, d.h. HTML/ CSS ist nicht notwendig).** Die Daten liegen als JSON-Datei vor:

- **User.json:** Username und Passwort eines Users
- **Data.json:** Alle Blogartikel

Zusätzlich soll mit Hilfe einer Middleware und einem JSON-Webtoken die User Authentifizierung und Autorisierung stattfinden. Hierbei gibt eine Schnittstelle die einen Login entgegennimmt, einen Webtoken zurück an den Clien. Dieser Token wiederrum wird bei jedem Request mitgeliefert.

Anforderungen Technologien:

- Node.js / Express
- JSON Web-Token (https://jwt.io/)

Funktionale Anforderungen (50 Punkte):

- Daten sollen mithilfe einer Web-API zur Verfügung gestellt werden./ Implementierung der Routen **(30 Punkte)**
- Hierbei sollen die Daten in JSON an einen Client gesendet werden. **(5 Punkte)**
- Eine Authentifizierung und Autorisierung soll mithilfe eines JSON-WebToken stattfinden **(10 Punkte)**
- Es soll eine Login Funktionalität in dem Backend vorhanden sein **(5 Punkte)**

Die Web-Api Routen:
User-spezifische Routen

- **Login: PUT** /api/V1/login

Input: Es wird ein Passwort und Username gesendet
Output: Ein JSON-WebToken

- **Passwort ändern: PUT** /api/V1/passwordRecovery

Input: Ein JSON-WebToken, neues Passwort, altes Passwort
Output: Success oder Fail Meldung und ein neuer JSON-WebToken

Blog-spezifische Routen

- **Alle Blogartikel GET /api/V1/blog**

Input: JWT
Output: Alle Blogartikel als JSON-Array.
Wenn die Route ohne JWT aufgerufen wird, sollen nur die Blogartikel übertragen werden, die als Attribut *hidden* als **false** haben

- **Ein spezifischer Blogartikel GET /api/V1/blog/:id**

Input: JWT
Output: Blogartikel mit der spezifischen Id
Wenn die Route ohne JWT aufgerufen wird und der Artikel als *hidden-value* allerdings ein **true** beinhaltet, soll ein 401 Statuscode übersendet werden.

- **Löschen eines Blogartikels DELETE /api/V1/blog/:id**

Input: JWT
Output: 200- erfolgreich gelöscht
Wenn die Route ohne JWT aufgerufen wird und der Artikel als *hidden-value* allerdings ein **true** beinhaltet, soll ein 401 Statuscode übersendet werden.

- **Editieren eines Blogartikels PUT /api/V1/blog/:id**

Input: JWT - neue Daten (Als Key Value Pair im Request Body)
Output: 200 - erfolgreich geupdatet - Im Response Body soll der Artikel nochmal als Objekt zurückgegeben werden.
Wenn die Route ohne JWT aufgerufen wird und der Artikel als *hidden-value* allerdings ein **true** beinhaltet, soll ein 401 Statuscode übersendet werden.

- **Anlegen eines Blogartikels POST /api/V1/blog**

Input: JWT
Output: 201- erfolgreich angelegt und eine neue ID des Artikels.
Ein Blog Artikel kann nur angelegt werden, wenn der User einen JWT übergeben hat.
Hierbei soll der neue Blog Artikel exakt dieselben Attribute haben wie die vorigen.
Das muss überprüft werden.