Schnittstellen Beschreibung Backend

Die URL fängt immer mit https://yourChoice.de/api/ an (wird mit … abgekürzt).
War der Request erfolgreich wird als Code bei GET eine 200 und bei POST 201 zurückgegeben, bei Misserfolg 404.
Greift man auf etwas zu, auf das man keine Rechte hat, wir 403 zurückgeben.
Will man mit seinem Request ein neues Objekt anlegen, wir dieses als Antwort/Response zurückgegeben. Zusätzlich dazu, wird allerdings die *ID* zurückgeben. Im Umkehrschluss bedeutet dies, dass beim Request keine ID angeben wird, da diese bei einem neuen Objekt nicht bekannt ist.

Login User:

Ein **POST** Request über folgende
URL:

…/login/user

Daten:

- password: …
- username:…

Response:

- token:…
- role:…

Login Voter:

Ein **POST** Request über folgende
URL:

…/login/voter

Daten:

- password: …
- hash:…

Response:

- token:…
- role:…

Logout:

Ein **POST** Request über folgende
URL:

…/logout

Daten:

- token: …

Neues Objekt anlegen (speichern):

Hierfür wird ein **POST** request an das *entsprechende* Model geschickt:
URL:

.../<model_name>
.../party

Daten. Hier werden alle Daten des Objektes [siehe Datenbankschema] übergeben, Beispiel party:

- name: …
- text: …

- consituency: …
- election_id: …
- token: …

Objekt bearbeiten:

Hierfür wird ein **PUT** request an das *entsprechende* Model geschickt:

URL:

.../<model_name>/<id>

Daten. Hier werden alle Daten des Objektes [siehe Datenbankschema] übergeben, Beispiel party:

- name: …
- text: …
- consituency: …
- election_id: …
- token: …

Objekt auslesen:

Hierfür wird ein **GET** request an das entsprechende Model geschickt:

URL:

.../<model_name>/<id>

Daten:

- token: …

Objekt löschen:

Hierfür wird ein **DELETE** request an das entsprechende Model geschickt:

URL:

.../<model_name>/<id>

Daten:

- token: …

Wahl auswerten:

Ein **GET** Request über folgende

URL:

.../election/<id>/evaluate

Daten:

- token: …

Wählen:

Ein **POST** Request über folgende

URL:

…/election/<id>/vote

Daten:

- first_vote: …
- second_vote: …
- voter_id: …
- candidate_id: …
- party_id: …
- referendum: <yes|no>
- valid: …

CSV File hochladen:

Ein **POST** Request:

URL:

…/election/{id}/addParties (Parteien)
…/election/{id}/addCandidates (Kandidaten)
…/election/{id}/addVoters (Wähler)

Daten:

- upload:… (die hochzuladende Datei)
- checksum:… (die eingegebene Checksumme der CSV Datei)

Response:

Die angelegten Objekte als Array

Zusätzliche Funktionen:

URLs:

POST: …/election/{id}/parties (bekomme alle Parteien der Wahl
POST: …/election/{id}/candidates (bekomme alle Kandidaten der Wahl)
POST: …/election/{id}/referendums (bekomme alle Referendums der Wahl)