Schnittstellen Beschreibung Backend

Die URL fängt immer mit <https://yourChoice.de/api/v1/> an (wird mit … abgekürzt).

War der Request erfolgreich wird als Code 200 zurückgegeben, bei Misserfolg 404.

Greift man auf etwas zu, auf das man keine Rechte hat, wir 401 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:

Ein **POST** Request über folgende

**URL:**

…*/login*

**Daten:**

* hash: … (Der Login Hash Wert)

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: …

Sollte dies (aus irgendwelchen Gründen) nicht funktionieren, kann es auch über den expliziten Aufruf der save Funktion realisiert werden:

**URL:**

*…/<model\_name>/save*

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 **POST** Request über folgende

**URL:**

*…/election/<id>/evaluate*

**Daten:**

* token: …

Wählen:

Ein **POST** Request über folgende

**URL:**

…/vote/

**Daten:**

* election\_id: …
* first\_vote: …
* second\_vote: …
* valid: …

Oder ein **POST** Request über folgende

**URL:**

…/election/<id>/vote

**Daten:**

* first\_vote: …
* second\_vote: …
* valid: …