

HFUwerdMobil!

Spezifikation RESTful API

Autor & Ansprechpartner

Tobias Straub

t.straub@hs-furtwangen.de

Internal Version: 3.3

External Version: 1.0

Authentifizierung

Bei GET-Requests: Durch Anhängen des Token an den Parameter ?token=

Bei POST- und PUT-Requests: Das Token befindet sich im JSON-Objekt unter "token"

Token Berechnung: $\text{sha256}(\{\text{user_email}\} + \{\text{API_KEY}\} + \{\text{API_ENDPUNKT}\} + \{\text{JSON_OBJEKT}\})$

Anmerkungen:

- Bildung des Hashes durch die Hashfunktion „SHA-2“ mittels SHA-256
- {user_email} ist beim Aufruf des API Endpunktes zur Änderung der Nutzerdaten bereits die neue E-Mail (falls geändert), ansonsten die bisherige (nicht geänderte)
- {API_KEY} lautet: `/u8teH}@A}_&&sQ%3DUsbW/m88Wk,K` (API_KEY muss „geheim“ bleiben)
- {API_ENDPUNKT} ist gesamter Endpunkt, z.B. /user/max.mustermann@hs-furtwangen.de/route/873
- {JSON_OBJEKT} ist das gesamte JSON-Objekt OHNE führende und OHNE endende geschweifte Klammer mit token Bezeichner einschließlich Doppelpunkt aber ohne Value, bei GET-Requests hat das JSON-Objekt den Wert "" (Leerstring)

Beispiel:

```
"user_email": "max.mustermann@gmail.com",
"firstname": "Max",
"lastname": "Mustermann",
"mobile_number": "49776523456",
"hometown_zip": "78098",
"token":
```

- Stimmt gelieferter Token nicht mit dem auf serverseite berechneten Token überein, muss der Request immer mit dem HTTP Response 401 abgelehnt werden

Verifizierungscode anfordern

API-Endpoint: /verify/{user_email} /* z.B. max.mustermann@hs-furtwangen.de */

HTTP-Method: GET

Erwartete Rückgabe:

- HTTP Response Code 200 (ok) + JSON-Objekt (mit Beispieldinhalt):

```
{
  "verify_code":123456
}
```

Anmerkung: Es wird ein zufälliger, sechs-stelliger Verifizierungscode generiert und an {user_email} verschickt, mit der Aufforderung, diesen Verifizierungscode in der App einzugeben. Zusätzlich wird dieser als JSON-Objekt dem Request zurückgegeben.

App Nutzer anlegen

API-Endpoint: /user/new

HTTP-Method: POST

Gelieferte Daten (mit Beispielinhalten):

```
{
  "user_email": "max.mustermann@gmail.com",
  "firstname": "Max",
  "lastname": "Mustermann",
  "mobile_number": "49776523456",
  "hometown_zip": "78098",
  "token": "ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad"
}
```

Erwartete Rückgabe:

- HTTP Response Code 200 (ok)
 - HTTP Response Code 409 (conflict): {user_email} schon vorhanden
 - HTTP Response Code 401 (unauthorized): ungültiger Token
-

App Nutzer ändern

API-Endpoint: /user/{(old_)user_email} /* z.B. max.mustermann@gmail.com */

HTTP-Method: POST

Gelieferte Daten (mit Beispielinhalten):

```
{
  "user_email": "max@mustermann.com", /* Bei keiner Änderung der E-Mail Adresse =>
old_user_email */
  "firstname": "Max",
  "lastname": "Mustermann",
  "mobile_number": "49776523456",
  "hometown_zip": "78098",
  "token": "ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad"
}
```

Erwartete Rückgabe:

- HTTP Response Code 200 (ok)
 - HTTP Response Code 424 (failed dependency): {(old_)user_email} nicht vorhanden
 - HTTP Response Code 401 (unauthorized): ungültiger Token
-

App Nutzer Daten abrufen

API-Endpoint: /user/{user_email}

HTTP-Method: GET

Erwartete Rückgabe:

- HTTP Response Code 200 (ok) + JSON-Objekt (mit Beispieldaten):

```
{
  "user_email": "max.mustermann@gmail.com",
  "firstname": "Max",
  "lastname": "Mustermann",
  "mobile_number": "49776523456",
  "hometown_zip": "78098"
}
```

Erwartete Rückgabe:

- HTTP Response Code 200 (ok)
- HTTP Response Code 424 (failed dependency): {user_email} nicht gefunden
- HTTP Response Code 401 (unauthorized): ungültiger Token

Bei Beginn einer Routen Aufzeichnung (User klickt auf Route aufzeichnen)

API-Endpoint: /user/{user_email}/route/new

HTTP-Method: POST

Gelieferte Daten (mit Beispieldaten):

```
{
  "route_id": "1cf2d0dfdf09e8e11421338948",
  "timestamp": "123456789",
  "start_point": {
    "latitude": "37.422005",
    "longitude": "-122.084095"
  },
  "end_point": "1", /* 0: User Heimatort; 1: Furtwangen; 2: Schwenningen; 3: Tuttlingen */
  "token": "ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad"
}
```

Erwartete HTTP-Rückgabewerte

- HTTP Response Code 200 (ok)
- HTTP Response Code 409 (conflict): {route_id} schon vorhanden
- HTTP Response Code 424 (failed dependency): {user_email} nicht vorhanden
- HTTP Response Code 401 (unauthorized): ungültiger Token

Während einer Routen Aufzeichnung

API-Endpoint: /user/{user_email}/route/{route_id}

HTTP-Method: POST

Gelieferte Daten (mit Beispieldaten):

```
{
  "positions": [
    {
      "timestamp": "12569537329",
```

```

    "latitude": "38.123456",
    "longitude": "-47.654321"
  },
  {
    "timestamp": "12569564253",
    "latitude": "42.123456",
    "longitude": "-66.654321"
  },
  ( ..... )
],
"token": "ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad"
}

```

Erwartete HTTP-Rückgabewerte

- HTTP Response Code 200 (ok)
- HTTP Response Code 424 (failed dependency): {route_id} nicht gefunden
- HTTP Response Code 401 (unauthorized): ungültiger Token

Anmerkung: Alle X-Minuten wird ein Satz von GPS-Daten an den API-Endpunkt versendet. Nach weiteren X-Minuten wird ein neuer Satz von GPS-Daten an den API-Endpunkt gesendet. Der Vorgang wiederholt sich solange, bis die Routen Aufzeichnung beendet wird (dafür wird ein entsprechender API-Endpunkt aufgerufen, siehe weiter unten)

Nach Ende der Routen Aufzeichnung (User klickt auf Route aufzeichnen beenden)

API-Endpoint: /user/{user_email}/route/end

HTTP-Method: POST

Gelieferte Daten (mit Beispieldaten):

```

{
  "route_id": "123456",
  "timestamp": "123456789",
  "total_duration_seconds": "1500",
  "total_kilometers": "25",
  "obd_info": { /* Sofern OBD aktiv war, ansonsten werden die OBD Values mit null gefüllt */
    "average_speed": "70",
    "average_consumption": "3"
  },
  "token": "ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad"
}

```

Erwartete HTTP-Rückgabewerte

- HTTP Response Code 200 (ok)
- HTTP Response Code 409 (conflict): Route wurde schon beendet
- HTTP Response Code 424 (failed dependency): {route_id} nicht vorhanden
- HTTP Response Code 401 (unauthorized): ungültiger Token

Anmerkungen

- {user_email} kann als Primärschlüssel verwendet werden
- {route_id} ist nur für den jeweiligen App-Nutzer eindeutig (nicht global), aber auf allen seinen Android Geräten
- Beim Start der Route (nach Aufruf von /user/{user_email}/route/new), beim Zufügen von Routen Daten (nach Aufruf von /user/{user_email}/route/end) und beim Beenden der Route (nach Aufruf von /user/{user_email}/route/end) muss die Websites-Gruppe über das Update informiert werden (Ping)! - Das kann ähnlich wie das die API-Designer bei Facebook implementiert haben, realisiert werden (siehe: <https://developers.facebook.com/docs/graph-api/real-time-updates/v2.2>). Oder einfacher als hardcodierte Ping URL. Mit dem Ping, werden die benötigten Daten per HTTP-GET als JSON mitgeliefert.