

FriendsTrackerWorldwide API

Version 2.0.0

Authors: Philipp Bahnmueller, Felix Rosa, Kevin Schrötter

Description

The FriendsTrackerWorldwide API is an API which provides the communication between the FriendsTrackerWorldwide Mobile APP (or any other kind of frontend) and an external database from MongoDB. The API delivers the main functions for adding registration, login, creating users and marker objects as well as calculating distances between a specific users location and all other GPS related objects that are connected to that user. The results typically are in a JSON format sent to the frontend and can then be used. In the FriendsTrackerWorldwide Project overall, the FriendsTrackerWorldwide Mobile App uses the JSON data retrieved from the FriendsTrackerWorldwide API to draw a map using elements and style from OpenStreetMap (using osmdroid together with osmbonuspack) and for managing a friendlist within the application.

Accessing the API

There is no server-address yet available for using the FriendsTrackerWorldwide API (FTW API) over internet. However, you can download the FTW API Server source code and run it on your own server.

Make sure to have node.js installed on the computer and run the “npm install” command from the shell before starting the node server using “npm start”.

Downloading the API

To download the API and run it on your own server, simply visit https://github.com/KevinSchroetter/FriendsTrackerWorldwide_API and download the repository.

MongoDB Database

The API uses a specific mongoDB database that. However, you can simply add another mongoDB database by creating a new external database on <https://mlab.com/> and changing the `db.connect(url,function(){})` in the server.js file of the ftwapi folder so that it matches the mongoDB link of the new database. MLab will then automatically create all required Collections.

FriendsTrackerWorldwide API Collections

The API uses two collections. Both are used for drawing objects in an OpenStreetMap:

- User
 - Used for saving user information, user GPS-locations and friend list-management
- Marker
 - Used for saving user-generated GPS-Locations with a description and a creator name

Collection Models

In the past, node.js could use mongoose for establishing a connection to the mongoDB database which also added the possibility to generate a mongoose.Schema for collection. By now mongoose is deprecated and it is better to use MongoClient instead. However, mongo client can no longer define Schemas as mongoose used to do. The following display of the collections uses old mongoose to see how the data-structure should be to work properly in the API:

user

```
{
  "_id": {
    "$oid": AUTOGENERATED,
  },
  "username": {Type: String},
  "password": {Type: String (hashed Value!) },
  "geoLocation": {
    "latitude": {Type: Number},
    "longitude": {Type: Number}
  },
  "friends": {Type: String[ ] },
  "description": {Type: String}"
}
```

marker

```
{
  "_id": {
    "$oid": AUTOGENERATED
  },
  "owner": {Type: String},
  "geoLocation": {
    "latitude": {Type: Number},
    "longitude": {Type: Number}
  },
  "description": {Type: String}
}
```

required Node Modules

Here you can see which node_modules are required for correct functionality of the API as well as a short description of what they are used for.

- **body-Parser**
 - o **Version 1.17.2**
 - o **Used for working with parameters**
- **express**
 - o **Version 4.15.3**
 - o **Used mainly for the API itself**
- **bcrypt**
 - o **Version 1.0.2**
 - o **Used for hashing and saving hashed passwords in Database**
- **Geopoint**
 - o **Version 1.0.1**
 - o **Used for calculating distances between different GPS-locations**

ftw API Routes

Here is an overview of the possibly FriendsTrackerWorldwide API routes used in the project.

ROUTE	METHO	PARAMS	TYPE	RETURNS
Ip:port/api/user/test	GET	-	-	Msg
Ip:port/api/user/getFriends/:username	GET	username	String	FriendsArr
Ip:port/api/user/getMyLocation/:username	GET	username	String	userObj
Ip:port/api/user/getFriedsLocation/:username	GET	username	String	FriendsUserObjArr
Ip:port/api/user/loginUser	POST	username password	String String	SuccessMsg
Ip:port/api/user/addUser	POST	username password latitude longitude	String String Number Number	successMsg
Ip:port/api/user/addFriend	PUT	username friend	String String	successMsg
Ip:port/api/user/updateUser	PUT	username longitude latitude	String Number Number	successMsg
Ip:port/api/user/deleteFriend	DELETE	username friend	String String	successMsg
Ip:port/api/marker/test	GET	-	-	Msg
Ip:port/api/marker/getMyMarker/:username	GET	username	String	MarkerObjArr
Ip:port/api/marker/getFriendsMarker/:username	GET	username	String	FriendsMarkerObjArr
Ip:port/api/marker/addMarker	POST	owner latitude longitude description	String Number Number String	MarkerID
Ip:port/api/marker/deleteMarker	DELETE	Id	Number	successMsg