Quantified Self API Documentation

Description:

Documentation for the RESTful api service for the Quantified Self Web Application. The name of the database is cs5500db and the three (3) collections in the database are entries, activities, and places.

Entries REST api

Description:

Entries REST api are tests for the entries collection. An entry in the entries collection is representative of a json element in the json array in storyline.json.

GET

Collection - All Entries

Request - http://localhost:8080/entries

- Retrieves all entries from the database and provides a json of all the entries.

GET

Collection - Activity Count by Year

Request - http://localhost:8080/entries/activity/count

- Retrieves the activity count by year.

GET

Collection - Calories per Year

Request - http://localhost:8080/entries/activity/calories

- Returns a collection of the total calories burned per year.

POST

Add Entry/ies as JsonArray

Request - http://localhost:8080/entries/save/jsonarray

- Adds an entry to the entries collection.
- POST request currently returns an error, and further needs to be debugged.

Input:

- Same input below is a json object, in a json array. This is inputted into the 'Body' - raw - json of Postman.

json
```
[{
  "date": "2013-02-09",
  "caloriesIdle": 1439.0,
  "lastUpdate": "22:55:57",
  "segments": [
    {
      "type": "place",
      "startTime": "20130209T063407-0800",
      "endTime": "20130209T132707-0800",
      "activities": [],
      "place": {
        "id": 6552482,
        "name": "Home",
        "type": "home",
        "location": {
          "lat": 47.67645,
          "lon": -122.32305
        }
      }
    },
    {
      "type": "move",
      "startTime": "20130209T132707-0800",
      "endTime": "20130209T133415-0800",
      "activities": [
        {
          "type": null,
          "manual": false,
          "startTime": "20130209T132707-0800",
          "endTime": "20130209T133415-0800",
          "duration": 428.0,
          "distance": 508.0,
          "trackPoints": [],
          "steps": 593.0,
```

```
      "calories": 25.0
    }
  ]
},
{
  "type": "place",
  "startTime": "20130209T133416-0800",
  "endTime": "20130209T144452-0800",
  "activities": [],
  "place": {
    "id": 7187779,
    "name": "Forza Coffee Co.",
    "type": "foursquare",
    "location": {
      "lat": 47.67877626216485,
      "lon": -122.3267572769863
    }
  }
},
{
  "type": "move",
  "startTime": "20130209T144452-0800",
  "endTime": "20130209T145054-0800",
  "activities": [
    {
      "type": null,
      "manual": false,
      "startTime": "20130209T144452-0800",
      "endTime": "20130209T145054-0800",
      "duration": 362.0,
      "distance": 378.0,
      "trackPoints": [],
      "steps": 479.0,
      "calories": 19.0
    }
  ]
},
{
  "type": "place",
  "startTime": "20130209T145055-0800",
  "endTime": "20130209T171309-0800",
  "activities": [
    {
      "type": null,
      "manual": false,
      "startTime": "20130209T162222-0800",
      "endTime": "20130209T162232-0800",
```

```
      "duration": 10.0,
      "distance": 3.0,
      "trackPoints": [],
      "steps": 6.0,
      "calories": 0.0
    }
  ],
  "place": {
    "id": 6552482,
    "name": "Home",
    "type": "home",
    "location": {
      "lat": 47.67645,
      "lon": -122.32305
    }
  }
},
{
  "type": "move",
  "startTime": "20130209T171309-0800",
  "endTime": "20130209T174745-0800",
  "activities": [
    {
      "type": "transport",
      "manual": false,
      "startTime": "20130209T171309-0800",
      "endTime": "20130209T174352-0800",
      "duration": 1843.0,
      "distance": 11701.0,
      "trackPoints": []
    },
    {
      "type": null,
      "manual": false,
      "startTime": "20130209T174352-0800",
      "endTime": "20130209T174745-0800",
      "duration": 233.0,
      "distance": 145.0,
      "trackPoints": [],
      "steps": 291.0,
      "calories": 7.0
    }
  ]
},
{
  "type": "place",
  "startTime": "20130209T174746-0800",
```

```
"endTime": "20130209T183422-0800",
"activities": [
  {
    "type": null,
    "manual": false,
    "startTime": "20130209T175022-0800",
    "endTime": "20130209T175222-0800",
    "duration": 120.0,
    "distance": 98.0,
    "trackPoints": [],
    "steps": 131.0,
    "calories": 5.0
  },
  {
    "type": null,
    "manual": false,
    "startTime": "20130209T175452-0800",
    "endTime": "20130209T175522-0800",
    "duration": 30.0,
    "distance": 20.0,
    "trackPoints": [],
    "steps": 41.0,
    "calories": 1.0
  },
  {
    "type": null,
    "manual": false,
    "startTime": "20130209T180223-0800",
    "endTime": "20130209T180253-0800",
    "duration": 30.0,
    "distance": 15.0,
    "trackPoints": [],
    "steps": 30.0,
    "calories": 1.0
  },
  {
    "type": null,
    "manual": false,
    "startTime": "20130209T181023-0800",
    "endTime": "20130209T181053-0800",
    "duration": 30.0,
    "distance": 25.0,
    "trackPoints": [],
    "steps": 51.0,
    "calories": 1.0
  },
  {
```

```
    "type": null,
    "manual": false,
    "startTime": "20130209T181223-0800",
    "endTime": "20130209T181253-0800",
    "duration": 30.0,
    "distance": 20.0,
    "trackPoints": [],
    "steps": 41.0,
    "calories": 1.0
  },
  {
    "type": null,
    "manual": false,
    "startTime": "20130209T181453-0800",
    "endTime": "20130209T181523-0800",
    "duration": 30.0,
    "distance": 25.0,
    "trackPoints": [],
    "steps": 51.0,
    "calories": 1.0
  },
  {
    "type": nul
```

Activity REST api

Description:

Activity REST api are tests for the activities collection. An activity in the activities collection is a collection of all activities that were parsed from storyline.json.

GET

Collection - All Activities

Request - http://localhost:8080/activities

- Returns all the activities from the activities collection.

GET

Collection - Filter by Activity

Request - http://localhost:8080/activities/cycling

- Returns a collection of activities when given an activity.
- In the request above, a collection of all activities that are cycling are returned in the HTTP request.

POST

Add Activity

Request - http://localhost:8080/activities/add?date=20210818&activity=kayaking&startTime=0.0&endTime=0.0&duration=60&distance=5&steps=0&calories=200

- This will allow the user or client to create an Activity object to add to the activities collection.
- This POST request will be an added feature in the future that will allow the user or client to complete a form, which will create an Activity object and add it to the activities collection.
- Response to this request is the unique id generated by MongoDB.

Request Params - these are query params rather than a direct path using path variable

date	20210818
activity	kayaking
startTime	0.0
endTime	0.0
duration	60
distance	5
steps	0
calories	200

PUT

Update Activity

Request - http://localhost:8080/activities/update/611de23ca557836537e198e5?date=20210818&activity=kayaking&startTime=0.0&endTime=0.0&duration=100&distance=1000&steps=0&calories=400

- This will allow the user or client to update an activity when given an id, which is unique to a MongoDB document, or object.
- This PUT request will be an added feature in the future that will allow the user or client to edit an activity.
- PUT request requires a direct path with the id to get the object from the collection and also utilizes query params to set the new values provided by the user or client.

Request Params

date	20210818
activity	kayaking
startTime	0.0
endTime	0.0
duration	100
distance	1000
steps	0
calories	400

DEL

Delete by Id via Path

Request - http://localhost:8080/activities/delete/611de23ca557836537e198e5

- Method deletes an activity from the activities collection when given the id, which is unique to a MongoDB document, or object.
- It is important to note that the id changes and randomized, thus in order to conduct an accurate test, will need to grab a new id each time when the database is reloaded.

Place REST api

Description:

Place REST api are tests for the places collection. A place in the places collection is a collection of all places that were parsed from storyline.json.

GET

Collection - All Places

Request - http://localhost:8080/places

- Returns all places in the places collection.

GET

Collection - Filter by Places

Request - http://localhost:8080/places/place/Home

- Returns a collection of places when given a place.
- In the request above, a collection of all places that are Home are returned in the HTTP request.

GET

Collection - Filter by Year

Request - http://localhost:8080/places/year/2016

- Returns a collection of places when given a year.
- In the request above, a collection of all places that were visited in 2016 are returned in the HTTP request.

GET

Collection - Place Count

Request - http://localhost:8080/places/count

- Returns a map, where the places visited are the keys, and the values are the number of times the places appear in the places collection.

POST

Add Place

Request - http://localhost:8080/places/add?date=20210821&place=The Container Store&type=retail&latitude=-14.5&longitude=56.7&fourSquareId=1234

- This will allow the user to create a Place object to add to the places collection.
- This POST request will be utilized to complete a form on the website that will add a Place to the places collection.
- Response to this request is the unique id generated by MongoDB.

Request Params

date	20210821
place	The Container Store
type	retail
latitude	-14.5
longitude	56.7
fourSquareId	1234

PUT

Update Place

Request - http://localhost:8080/places/update/611de29fa557836537e198e6?date=20130301&place=Northeastern University&type=School&latitude=-45.11&longitude=121.05&fourSquareId=1234

- This will allow the user or client to update a place when given an id, which is unique to a MongoDB document, or object.
- This PUT request will be an added feature in the future that will allow the user or client to edit a place.
- PUT request requires a direct path with the id to get the object from the collection and also utilizes query params to set the new values provided by the user or client.

Request Params

date	20130301
place	Northeastern University
type	School
latitude	-45.11
Longitude	121.05
fourSquareId	1234

DEL

Delete by Id via Path

Request - http://localhost:8080/places/delete/611de29fa557836537e198e6

- Method deletes a place from the places collection when given the id, which is unique to a MongoDB document, or object.
- It is important to note that the id changes and randomized, thus in order to conduct an accurate test, will need to grab a new id each time when the database is reloaded.