Trip Exchange System

# TODO: Explain difference between users and drivers

# RESTful API Overview

The RESTful API is exposing web services to work with the server. The API has the following web service endpoints:

|  |  |  |
| --- | --- | --- |
| **HTTP Method** | **Web service endpoint** | **Description** |
| POST (public) | **api/users/register** | Registers a **new user** in the system. |
| POST (public) | **api/users/login** | **Logs in** a user in the system. |
| GET | **api/users/userInfo** | Gets the information about the **current logged user** |
| POST | **api/users/logout** | **Logs out** a user from the system. |
| GET (public) | **api/stats** | Gets **statistics** for the system – total trips, total users, total drivers and total finished trips. |
| GET (public) | **api/cities** | Gets list of strings with **all cities** sorted by name |
| POST | **api/trips** | **Creates a new trip** with current user as driver. |
| GET | **api/trips/{tripId}** | Gets **trip details** by its id. |
| GET (public) | **api/trips** | Gets **top 10 trips** where their departure time is in the future sorted by their time of departure in ascending order. |
| GET | **api/trips?page={P}** | Gets the trips at positions **from page {P}** where their departure time is in the future (upcoming only). All trips are sorted by their time of departure in ascending order and are at most 10. If page parameter is missing its default value is 1. |
| GET | api/trips?page={P}&orderBy ={S}&orderType={O} | Gets the trips at positions from page {P}. All trips are sorted by “**driver**”, “**date**”, “**seats**”, “**from**”, “**to**” and are at most 10. Order is “**asc**” or “**desc**”. If order is missing, it is assumed “date” as default in “asc” order. |
| GET | api/trips?page={P}&from={F}&to={T} | Gets the trips at positions from page {P}.  Trips are filtered by from town {F} and to town {T} (one of which may be missing). All trips are sorted by time of departure and are at most 10. |
| GET | api/trips?page={P}&finished=true | Gets the trips at positions from page {P}, includes all the already finished trips (with time of travel earlier than the current time). By default, if finished is missing, the server returns only the upcoming trips. |
| GET | api/trips?page={P}&onlyMine =true | Gets the trips at positions from page {P}. Filters the trips by only the current user’s created or joined trips. If onlyMine is missing, the server returns everyone’s trips. |
| PUT | **api/trips/{tripId}** | **Join a trip** by given id and with available seats – assigns the current user as passenger. |
| GET (public) | **api/drivers** | Get **top 10 drivers**, sorted by number of total trips (desc) and then by username (asc). |
| GET | **api/drivers?page={P}** | Gets the **drivers from page {P}**, sorted by number of total trips (desc) and then by username (asc). |
| GET | **api/drivers?page={P}&username={U}** | Gets the **drivers from page {P}**, sorted by number of total trips (desc) and then by username (asc) and **filtered by username** (filter is “*contains*” and case-insensitive). |
| GET | **api/drivers/{driverId}** | Gets **driver by its id**, along with his/her 50 most recent upcoming trips (in which this user is driver or passengers). |

**\*NOTE: Parameters *orderBy*, *orderType*, *from*, *to*, *finished* and *onlyMine* on the *trips* API point are optional and can be combined with the *page* parameter. For example *api/trips?page=5&sort=seats&to=Sofia* is valid request.**

**\*NOTE: Paging numeration always starts from 1 (there is no page 0).**

# RESTful API Details

## Users

### **Register**

|  |  |  |  |
| --- | --- | --- | --- |
| api/users/register | | | POST |
| **Request** | { "email": "test@test.com",  "password": "12345678",  "confirmPassword": "12345678",  "isDriver": "true",  "car": "Mazda 6 2014" } | | |
| **Response** |  | 200 OK | |

### **Login**

|  |  |  |
| --- | --- | --- |
| api/users/login | | POST |
| **Request** | { "grant\_type": "password",  "username": "test@test.com",  "password": "12345678" } | |
| **Response** | { "access\_token": "deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz",  "token\_type": "bearer",  "expires\_in": 1209599,  "userName": "test@test.com",  ".issued": "Fri, 26 Sep 2014 10:39:16 GMT",  ".expires": "Fri, 10 Oct 2014 10:39:16 GMT" } | |

### **User info**

|  |  |  |
| --- | --- | --- |
| api/users/userInfo | | GET |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Response** | { "email": "test@test.com",  "isDriver": true,  "car": "Mazda 6 2014" } | |

### **Logout**

|  |  |  |  |
| --- | --- | --- | --- |
| ****api/users/logout**** | | | POST |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | | |
| **Response** |  | 200 OK | |

## Statistics

|  |  |  |
| --- | --- | --- |
| api/stats | | GET |
| **Response** | { "trips": 1337,  "finishedTrips": 919,  "users": 654,  "drivers": 321 } | |

## Cities

|  |  |  |
| --- | --- | --- |
| api/cities | | GET |
| **Response** | ["Aheloy","Ahtopol","Aksakovo",...,"Zlataritsa","Zlatitsa","Zlatograd"] | |

## Trips

### **Create trip**

|  |  |  |
| --- | --- | --- |
| api/trips | | POST |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Request** | { "from": "Sofia",  "to": "Kyustendil",  "availableSeats": "4",  "departureTime": "2014-09-29 13:37:00" } | |
| **Response** | { "id": "bcf79ac5-0e77-4a21-aa21-279c93375231",  "driverId": "fe411b06-cef6-4c95-b95a-13a3310a1f05",  "driverName": "test@test.com",  "from": "Sofia",  "to": "Kyustendil",  "departureDate": "2014-09-29T13:37:00",  "numberOfFreeSeats": 4,  "isMine": true,  "passangers": ["test@test.com"] } | |

**\* NOTE: The logged user should be driver in order to be able to create trips.**

**\* NOTE: isMine value means that the currently logged user is part of the trip (either driver or passenger)**

**\* NOTE: All drivers are passengers too.**

### **Get trip detail**

|  |  |  |
| --- | --- | --- |
| api/trips/7362e399-d051-42bd-95a0-109a2673dfd5 | | GET |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Response** | { "id": "7362e399-d051-42bd-95a0-109a2673dfd5",  "driverId": "c73d0bc9-47b7-4cc4-abb2-cd5516b94596",  "driverName": "test02@test.com",  "from": "Ahtopol",  "to": "Etropole",  "departureDate": "2014-10-11T17:09:19.203",  "numberOfFreeSeats": 3,  "isMine": false,  "passangers": [  "test02@test.com",  "test06@test.com"  ]  } | |

### **Join trip**

|  |  |  |
| --- | --- | --- |
| api/trips/7362e399-d051-42bd-95a0-109a2673dfd5 | | PUT |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Response** | { "id": "7362e399-d051-42bd-95a0-109a2673dfd5",  "driverId": "c73d0bc9-47b7-4cc4-abb2-cd5516b94596",  "driverName": "test02@test.com",  "from": "Ahtopol",  "to": "Etropole",  "departureDate": "2014-10-11T17:09:19.203",  "numberOfFreeSeats": 2,  "isMine": true,  "passangers": [  "test02@test.com",  "test06@test.com",  "test@test.com"  ]  } | |

**\*NOTE: In case of any error (invalid trip id, past trip, user already in the trip, no seats available, etc.) you will receive status code 400 (with JSON object with property message, containing the text of the error). This applies for all other queries.**

### **Get trips**

|  |  |  |
| --- | --- | --- |
| api/trips | | GET |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Response** | Collection of { Id, driverId, driverName, from, to, departureTime, numberOfFreeSeats, isMine } | |

**\*NOTE: All other GET requests to trips return the same data but filtered, sorted and paged.**

## Drivers

### **Get drivers**

|  |  |  |
| --- | --- | --- |
| api/drivers?page=1&username=TEST.com | | GET |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Response** | [{  "id": "1c25d2fc-f2f5-4e63-bef1-01a2c4374e26",  "name": "test44@test.com",  "numberOfUpcomingTrips": 3,  "numberOfTotalTrips": 10  }, {  "id": "36f11831-545c-4228-b0b6-be54832884e6",  "name": "test01@test.com",  "numberOfUpcomingTrips": 1,  "numberOfTotalTrips": 8  }, {  "id": "52f9631d-0af7-49b0-81b1-5b0476131022",  "name": "test34@test.com",  "numberOfUpcomingTrips": 3,  "numberOfTotalTrips": 8  }] | |

**\*NOTE: If you send the request anonymously you will receive** **the top 10 drivers, sorted by number of trips (in descending order) and then by username (in ascending order) with no paging and filtering.**

### **Get driver detail**

|  |  |  |
| --- | --- | --- |
| api/drivers/52f9631d-0af7-49b0-81b1-5b0476131022 | | GET |
| **Headers** | "Authorization" = "Bearer deYtpuVHa4Ba-1t-...\_hdTI-1Pa-bNiASNQz" | |
| **Response** | {  "id": "e11ba9c5-d0cc-465a-aa55-354340e42baf",  "name": "test36@test.com",  "numberOfUpcomingTrips": 3,  "numberOfTotalTrips": 9,  "trips": [{  "id": "f466253d-a1ef-41c6-9f84-5c2722a01d73",  "driverId": "e11ba9c5-d0cc-465a-aa55-354340e42baf",  "driverName": "test36@test.com",  "from": "Botevgrad",  "to": "Simitli",  "departureDate": "2014-10-05T17:09:19.16",  "numberOfFreeSeats": 1,  "isMine": false  }, {  "id": "c639f81c-1f3e-4b00-9533-0c88e96a520e",  "driverId": "c73d0bc9-47b7-4cc4-abb2-cd5516b94596",  "driverName": "test02@test.com",  "from": "Ardino",  "to": "Belitsa",  "departureDate": "2014-10-31T17:09:19.27",  "numberOfFreeSeats": 1,  "isMine": false  }, {  "id": "080dc395-09ac-4f76-8e17-a188a9de0fae",  "driverId": "70bd20b5-2d78-427b-b359-3eb1b891d0d7",  "driverName": "test26@test.com",  "from": "Saedinenie",  "to": "Maglizh",  "departureDate": "2014-11-15T17:09:19.323",  "numberOfFreeSeats": 3,  "isMine": false  }]  } | |