

API

Author: Sebastian Lienau

Table of Content

- [Endpoint](#)
- [Error handling](#)
 - [Error types](#)
 - [Error object](#)
- [Objects](#)
 - [Location](#)
 - [VirtualBusStop](#)
 - [Account](#)
 - [Order](#)
 - [Route](#)
 - [ActiveOrderStatus](#)
- [Resources](#)
 - [Authorization](#)
 - [/login](#)
 - [/account](#)
 - [/orders](#)
 - [/activeorder](#)
 - [/routes](#)
 - [/virtualbusstops](#)
 - [/leaderboard](#)
 - [/vans](#)

Endpoint

The base URL for this APIs endpoint is

```
http://3.120.249.73:8080
```

Error handling

This API uses standard HTTP status codes to indicate the status of a response.

Error types

Name	Code	Description
Bad request	400	The request was unacceptable
Unauthorized	401	The request has not been applied because it lacks valid authentication credentials for the target resource.
Forbidden	403	The server understood the request, but is refusing to fulfill it
Not Found	404	The server has not found anything matching the request URI
Server error	500	A technical error occured in the Cloud

Error object

This general error structure is used throughout this API.

Field	Type	Required	Description	Examples
code	Integer	yes	The error code.	400
message	String	No	(Long) description of the error.	The server understood the request, but is refusing to fulfill it

Example

```
{
  "code": 400,
  "message": "Bad query parameter [$size]: Invalid integer value [abc]"
}
```

Objects

Location

This object represents a geographical location.

Field	Type	Required	Description	Examples
latitude	Number	yes	The north coordinate.	52.478442
longitude	Number	yes	The east coordinate.	13.405938

Example

```
{
  "latitude": 52.478442,
  "longitude": 13.405938
}
```

VirtualBusStop

This object represents a virtual bus stop.

Field	Type	Required	Description	Examples
id	String	yes	The ID.	0e8cedd0-ad98-11e6-bf2e-47644ada7c0b
name	String	Yes	Name of the station, e.g. street name or point of interest name	Straße des 17. Juni 135
accessible	Boolean	Yes	True if the virtual bus stop is currently accessible, false if not.	true , false
location	Location	Yes	The location of the VirtualBusStop.	See Location

Example

```
{
  "id": "0e8cedd0-ad98-11e6-bf2e-47644ada7c0b",
  "name": "Straße des 17. Juni 135",
  "accessible": true,
  "location": {
    "latitude": 52.478442,
    "longitude": 13.405938
  }
}
```

Account

This object represents a user account.

Field	Type	Required	Description	Examples
<code>id</code>	<code>String</code>	yes	The ID.	<code>0e8cedd0-ad98-11e6-bf2e-47644ada7c0f</code>
<code>firstName</code>	<code>String</code>	Yes	The users first name.	<code>Max</code>
<code>lastName</code>	<code>String</code>	Yes	The users last name.	<code>Müller</code>
<code>email</code>	<code>String</code>	Yes	The users email.	<code>maxmueller@tu-berlin.de</code>
<code>username</code>	<code>String</code>	Yes	The username.	<code>maxiboy123</code>
<code>loyaltyPoints</code>	<code>Number</code>	Yes	The users total loyalty points.	<code>234</code>
<code>loyaltyStatus</code>	<code>String</code>	yes	The loyalty program status.	<code>gold</code>
<code>address</code>	<code>Object</code>	No	The users address. Contains <code>street</code> , <code>zipcode</code> and <code>city</code>	<i>See account example.</i>
<code>distance</code>	<code>Number</code>	yes	Total amount of kilometres driven by the user.	<code>423.34</code>
<code>co2savings</code>	<code>Number</code>	yes	Total amount of CO2 savings (in kilogram) for all van rides.	<code>70.4</code>

Example

```
{
  "id": "0e8cedd0-ad98-11e6-bf2e-47644ada7c0f",
  "firstName": "Max",
  "lastName": "Müller",
  "email": "maxmueller@tu-berlin.de",
  "username": "maxiboy123",
  "loyaltyPoints": 234,
  "loyaltyStatus": "gold",
  "address": {
    "street": "Salzufer 1",
    "zipcode": "10587",
    "city": "Berlin"
  },
  "distance": 423.34,
  "co2savings": 70.4,
}
```

Order

This object represents an order. It can be created by a user using the `POST /orders` endpoint.

Field	Type	Required	Description	Examples
<code>id</code>	<code>String</code>	yes	The order ID.	<code>13cf81ee-8898-4b7a-a96e-8b5f675deb3c</code>
<code>accountId</code>	<code>String</code>	yes	User account ID which placed this order.	<code>0e8cedd0-ad98-11e6-bf2e-47644ada7c0f</code>
<code>orderTime</code>	<code>Datetime</code>	yes	Time at which the order was created.	<code>2018-11-23T18:25:43.511Z</code>
<code>active</code>	<code>Boolean</code>	yes	<code>True</code> if this order is active (user is still travelling to the ending virtual bus stop). <code>False</code> if the user reached the ending virtual bus stop (finished the journey), or if the order was canceled.	<code>true</code>

canceled	Boolean	yes	True if the user canceled the order.	false
vanStartVBS	VirtualBusStop	yes	Virtual Bus Stop where the van departs.	See VirtualBusStop
vanEndVBS	VirtualBusStop	yes	Virtual Bus Stop where the van arrives.	See VirtualBusStop
vanEnterTime	Datetime	no	Time at which the user entered the van. Null if the order was canceled.	2018-11-23T18:30:25.000Z
vanExitTime	Datetime	no	Time at which the user left the van. Null if the order was canceled.	2018-11-23T18:45:48.000Z
vanId	Number	yes	The van which carries the user.	3
loyaltyPoints	Number	yes	The amount of loyalty points for this order.	200
distance	Number	yes	Amount of kilometres the van drives.	12.4
co2savings	Number	yes	Amount of CO2 savings (in kilogram) for this ride .	2.2
route	Route	only if active is true	The route object for this order.	See Route

Example

```
{
  "id": "13cf81ee-8898-4b7a-a96e-8b5f675deb3c",
  "accountId": "0e8cedd0-ad98-11e6-bf2e-47644ada7c0f",
  "orderTime": "2018-11-23T18:25:43.511Z",
  "active": true,
  "canceled": false,
  "vanStartVBS": {
    "id": "7416550b-d47d-4947-b7ec-423c9fade07f",
    "name": "Straße des 17. Juni 135",
    "accessible": true,
    "location": {
      "latitude": 52.515598,
      "longitude": 13.326860
    }
  },
  "vanEndVBS": {
    "id": "76d7fb2f-c264-45a0-ad65-b21c5cf4b532",
    "name": "Straße des 17. Juni 120",
    "accessible": true,
    "location": {
      "latitude": 52.512974,
      "longitude": 13.329145
    }
  },
  "vanEnterTime": "2018-11-23T18:30:25.000Z",
  "vanExitTime": "2018-11-23T18:45:48.000Z",
  "vanId": 3,
  "loyaltyPoints": 3980,
  "distance": 12.4,
  "co2savings": 2.2,
  "route": {}
}
```

Route

This object represents a travel route from journey start to the final destination of the journey.

Field	Type	Required	Description	Examples
<code>id</code>	<code>String</code>	yes	The route ID.	<code>13cf81ee-8898-4b7a-a96e-8b5f675deb3c</code>
<code>userStartLocation</code>	<code>Location</code>	yes	Location where the journey starts.	See Location
<code>userDestinationLocation</code>	<code>Location</code>	yes	Final destination of the journey.	See Location
<code>vanStartVBS</code>	<code>VirtualBusStop</code>	yes	The <code>VirtualBusStop</code> where the user should enter the van.	See VirtualBusStop
<code>vanEndVBS</code>	<code>VirtualBusStop</code>	yes	The <code>VirtualBusStop</code> where the user should exit the van.	See VirtualBusStop
<code>vanDepartureTime</code>	<code>Datetime</code>	Yes	The time the van will depart from <code>vanStartVBS</code> .	<code>2018-12-17T17:20:00.000Z</code>
<code>vanArrivalTime</code>	<code>Datetime</code>	Yes	The (expected) time the van will arrive at <code>vanEndVBS</code> .	<code>2018-12-17T17:35:00.000Z</code>
<code>guaranteedVanArrivalTime</code>	<code>Datetime</code>	Yes	The guaranteed arrival time of the van at <code>vanEndVBS</code> if the ride is shared with other passengers.	<code>2018-12-17T17:45:00.000Z</code>
<code>toDestinationWalkingTime</code>	<code>Number</code>	Yes	The time in seconds the user needs from <code>vanEndVBS</code> to <code>userDestinationLocation</code> by foot.	<code>274</code>
<code>toStartRoute</code>	<code>Object</code>	Yes	Route from <code>userStartLocation</code> to <code>vanStartVBS</code> . This is an object returned by the Google	

			Maps API.	
<code>vanRoute</code>	<code>Object</code>	Yes	Van route from <code>vanStartVBS</code> to <code>vanEndVBS</code> . This is an object returned by the Google Maps API.	
<code>toDestinationRoute</code>	<code>Object</code>	Yes	Route from <code>vanEndVBS</code> to <code>userDestinationLocation</code> . This is an object returned by the Google Maps API.	
<code>vanId</code>	<code>Number</code>	yes	The ID of the van which drives this route.	<code>3</code>
<code>validUntil</code>	<code>Datetime</code>	Only for <code>POST /routes</code> request	Time until this route can be confirmed (create an order out of this route). After this time has elapsed, a new route must be requested, otherwise no order can be created.	<code>2018-12-17T17:10:00.000Z</code>

Example

```
{
  "id": "13cf81ee-8898-4b7a-a96e-8b5f675deb3c",
  "userStartLocation": {
    "latitude": 52.516639,
    "longitude": 13.331985
  },
  "userDestinationLocation": {
    "latitude": 52.513245,
    "longitude": 13.332684
  },
}
```

```
"vanStartVBS": {
  "id": "7416550b-d47d-4947-b7ec-423c9fade07f",
  "name": "Straße des 17. Juni 135",
  "accessible": true,
  "location": {
    "latitude": 52.515598,
    "longitude": 13.326860
  }
},
"vanEndVBS": {
  "id": "76d7fb2f-c264-45a0-ad65-b21c5cf4b532",
  "name": "Straße des 17. Juni 120",
  "accessible": true,
  "location": {
    "latitude": 52.512974,
    "longitude": 13.329145
  }
},
"vanDepartureTime": "2018-12-17T17:20:00.000Z",
"vanArrivalTime": "2018-12-17T17:35:00.000Z",
"guaranteedVanArrivalTime": "2018-12-17T17:45:00.000Z",
"toDestinationWalkingTime": 274,
"toStartRoute": {},
"vanRoute": {},
"toDestinationRoute": {},
"vanId": 3,
"validUntil": "2018-12-17T17:10:00.000Z"
}
```

ActiveOrderStatus

All fields are *not* required. Only modified values since the last request must be included in the response.

Field	Type	Description
<code>vanId</code>	<code>Number</code>	The van ID.
<code>userAllowedToEnter</code>	<code>Boolean</code>	<code>true</code> if the user can enter the van (if the user is close enough to the van), <code>false</code> otherwise.
<code>userAllowedToExit</code>	<code>Boolean</code>	<code>true</code> if the user can exit the van, <code>false</code> otherwise.
<code>vanLocation</code>	<code>Location</code>	The current position of the van.
<code>vanDepartureTime</code>	<code>Datetime</code>	The time the van will depart from <code>vanStartVBS</code>
<code>vanArrivalTime</code>	<code>Datetime</code>	The (expected) time the van will arrive at the <code>vanEndVBS</code>
<code>guaranteedVanArrivalTime</code>	<code>Datetime</code>	The guaranteed arrival time of the van at <code>vanEndVBS</code> if the ride is shared with other passengers.
<code>otherPassengers</code>	<code>Array</code>	Array containing usernames (Strings) of other passengers.
<code>message</code>	<code>String</code>	The status message.
<code>nextStops</code>	Array of <code>VirtualBusStop</code>	This array contains the next virtual bus stops that the van will drive up in case of pooling.
<code>nextRoutes</code>	Array of <code>Object</code>	This array contains Google Maps Route objects. Every element of the array contains the route from one virtual bus stop to the next.

Example

```
{
  "vanId": 3,
  "userAllowedToEnter": false,
  "userAllowedToExit": false,
  "vanLocation": {
    "latitude": 52.515598,
    "longitude": 13.32686
  },
  "vanDepartureTime": "2018-11-23T18:20:25.000Z",
  "vanArrivalTime": "2018-11-23T18:30:25.000Z",
  "guaranteedVanArrivalTime": "2018-11-23T18:37:25.000Z",
  "otherPassengers": [
    "sarah",
    "wilbert"
  ],
  "message": "Van has not arrived yet",
  "nextStops": [],
  "nextRoutes": []
}
```

Resources

All request and response bodys are of type `application/json` .

Authorization

Authorization is required for all requests except to the `/login` endpoint. If not authorized, the server will respond with `HTTP 401` .

Authorization Header

Parameter	Value	Required	Description
<code>Authorization</code>	<code>Bearer <TOKEN></code>	Yes	The JWT token from the <code>POST /login</code> response.

/login

POST /login

Request Body

```
{
  "username": "admin",
  "password": "xyz"
}
```

Response Body

```
{
  "userId": "64e0d993-2867-44cf-872d-9a78a5c212a0",
  "token": "xxxxx.yyyyy.zzzzz"
}
```

/account

GET /account

Get the user account information.

Responses

Code	Body Type	Description
200	Account	See Account .
400	Error	Bad request. See Error .
401	Error	If a wrong username or password has been entered. See Error .

////////////////////////////////////

PUT /account

To update user account information.

Body

Type	Required	Description
Account	Yes	Containing the updated user account information. See Account .

Responses

Code	Body Type	Description
200	Account	Containing the updated user account information. See Account .
400	Error	See Error .

////////////////////////////////////

/orders

////////////////////////////////////

GET /orders

Get the orders of a user.

Request Query Parameters

Property	Type	Required	Description
fromDate	Datetime	No	If set, the response will only contain orders which were placed <i>after</i> this date.
toDate	Datetime	No	If set, the response will only contain orders which were placed <i>before</i> this date.

Example

- /orders
- /orders?fromDate=2018-01-01T00:00:00.000Z&toDate=2018-10-01T00:00:00.000Z

Responses

Code	Body Type	Description
200	Array of Order	Orders which match the request parameters. See Order .
400	Error	See Error .

Example


```
[
  {
    "id": "13cf81ee-8898-4b7a-a96e-8b5f675deb3c",
    "accountId": "0e8cedd0-ad98-11e6-bf2e-47644ada7c0f",
    "orderTime": "2018-02-23T18:25:43.511Z",
    "active": false,
    "canceled": false,
    ...,
  },
  {
    "id": "13cf81ee-4b7a-8898-a96e-8b5f675deb3c",
    "accountId": "0e8cedd0-ad98-11e6-bf2e-47644ada7c0f",
    "orderTime": "2018-02-24T18:25:43.511Z",
    "active": false,
    "canceled": false,
    ...,
  }
]
```



POST /orders

To create (place) a new van order.

Request

Body

Property	Type	Required	Description
routeId	String	Yes	ID of the route.

Example

```
{
  "routeId": "d79ab15d-39e8-4817-83d0-ed21d395dded",
}
```

Responses

Code	Body Type	Description
200	Order	The new created order. See Order .
400	Error	If the order couldn't be created. See Error .
404	Error	If the <code>routeId</code> wasn't found or if the route has expired. See Error .

/activeorder

GET /activeorder

Get the current active order.

Responses

Code	Body Type	Description
200	Order	The current active order. See Order .
400	Error	See Error .
404	Error	If there is no active order at the moment. See Error .

PUT /activeorder

Update an active order.

Request Body

The request body contains a JSON object with the properties `action` , which describes what should be done/changed in the active order, and `userLocation` , which contains the users current location coordinates (See [Location](#)).

Possible `action` -types are:

Value	Description
<code>startride</code>	User has entered the van. After this request has been sent, the van should lock the doors and start the ride. <code>403</code> Error if the user isn't close enough to the van, so he didn't enter the van.
<code>endride</code>	User left the van. <code>403</code> error if the van hasn't arrived at the destination virtual bus stop yet.
<code>cancel</code>	User wants to cancel the order. <code>403</code> error if the order can not be canceled anymore (ride already started or ended).

Example

```
{
  "action": "startride",
  "userLocation": {
    "latitude": 52.123456,
    "longitude": 13.123456
  }
}
```

Responses

Code	Body Type	Description
200	Order	The updated Order. See Order .
400	Error	See Error .
403	Error	If the <code>action</code> requested by the user is not allowed. See request body and Error .



GET /activeorder/status

Get information about the current active order.

Request Query Parameters

Property	Type	Required	Description
<code>passengerLatitude</code>	Number	Yes	The users current latitude.
<code>passengerLongitude</code>	Number	Yes	The users current longitude.

Example

```
GET /activeorder/status?passengerLatitude=52.123456&passengerLongitude=13.123456
```

Response

Code	Body Type	Description
200	ActiveOrderStatus	See ActiveOrderStatus .
400	Error	See Error .



/routes

POST /routes

Get suggested routes from starting point to destination.

Request

Body

Property	Type	Required	Description
start	Location	Yes	Journey start location. See Location .
destination	Location	Yes	Journey destination. See Location .
passengers	Number	No	The amount of passengers that want to take the ride. Default: 1
startTime	Datetime	No	Route start time. Current time if not specified.

Example

```
{
  "start": {
    "latitude": 52.512974,
    "longitude": 13.329145
  },
  "destination": {
    "latitude": 52.285946,
    "longitude": 13.317390
  },
  "startTime": "2018-12-15T18:30:00.000Z"
}
```

Responses

Code	Body Type	Description
200	Array of <code>Route</code>	Array of possible routes from desired start location to destination. See Route .
400	<code>Error</code>	See Error
404	<code>Error</code>	If no route was found. See Error

Example

```
[
  {
    "startLocation": {
      "latitude": 52.516639,
      "longitude": 13.331985
    },
    "startStation": ...
    ...
  }, {
    "startLocation": {
      "latitude": 52.516639,
      "longitude": 13.331985
    },
    "startStation": ...
    ...
  }
]
```

////////////////////////////////////

/virtualbusstops

////////////////////////////////////

GET /virtualbusstops

Get nearby virtual bus stops.

- Will only return virtual bus stops whose `accessible` property is `true` .
- Returns an array of virtual bus stops which are *inside of the radius* from the given location.
- Returns an empty array if no virtual bus stops are found inside that area.

Request

Query Parameters

Property	Type	Required	Description
<code>latitude</code>	Number	Yes	The latitude of the center position
<code>longitude</code>	Number	Yes	The longitude of the center position
<code>radius</code>	Number	No	Maximum distance of the virtual bus stops to the location parameter. <i>Unit: meter</i> Min: <code>100</code> ; Max: <code>10000</code> ; Default: <code>1000</code>

Examples

```
/virtualbusstops?radius=2000&latitude=52.512974&longitude=13.329145
```

```
/virtualbusstops?latitude=52.512974&longitude=13.329145
```

Responses

Code	Body Type	Description
<code>200</code>	Array of <code>VirtualBusStop</code>	Array of virtual bus stops which are inside of the radius from the given location. See VirtualBusStop
<code>400</code>	<code>Error</code>	See Error

Examples

```
[ ]
```

```
[
  {
    "id": "d79ab15d-39e8-4817-83d0-ed21d395dded",
    "name": "Straße des 17. Juni",
    "accessible": true,
    "location": {
      "latitude": 52.515729,
      "longitude": 13.323373
    }
  },
  {
    "id": "7416550b-d47d-4947-b7ec-423c9fade07f",
    "name": "Straße des 17. Juni",
    "accessible": true,
    "location": {
      "latitude": 52.515598,
      "longitude": 13.326860
    }
  }
]
```

/leaderboard

GET /leaderboard

Get the top 10 users with the most loyalty points.

Response

Code	Body Type	Description
200	Array of Objects	Every object contains <code>username</code> , <code>loyaltyPoints</code> and <code>status</code> (see example). The array is sorted by the property <code>loyaltyPoints</code> (highest first).
400	Error	See Error

Example

```
[
  {
    "loyaltyPoints": 175,
    "username": "antonio",
    "status": "platin"
  },
  {
    "loyaltyPoints": 125,
    "username": "philipp",
    "status": "gold"
  },
  {
    "loyaltyPoints": 75,
    "username": "alex",
    "status": "silver"
  }
]
```

/vans

GET /vans

Get all available vans and their positions.

Response

Code	Body Type	Description
200	Array of Objects	Every object contains the properties <code>vanId</code> and <code>location</code> (see example) .
400	Error	See Error

Example

```
[
  {
    "vanId": 1,
    "location": {
      "latitude": 52.5197098,
      "longitude": 13.3882533
    }
  },
  {
    "vanId": 2,
    "location": {
      "latitude": 52.507129199999999,
      "longitude": 13.3305731
    }
  }
]
```