

TPT Commuter API

Note:

This documentation applies to server version 0.4.4

General API Notes

All server endpoints only accept and return JSON Objects. One or more of the following HTTP Header entries should be added to every request:

```
Accept: application/json
Authorization: Bearer <secret_token>
Content-type: application/json
```

For unauthorized requests, the *Authorization* header may be omitted. If a request requires authorization, then the *Authorization* header is mandatory, the is obtained from the server maintainer.

The *Content-type* header must be omitted from requests that do not carry a payload (e.g. all GET and DELETE requests).

Every endpoint response is delivered in a JSON object along with the status of the request (either success or error). Sample responses for a successful request and a failed request:

```
{'status': 'success', 'data': <JSON Object or value>}

{'status': 'error', 'hint': 'email', 'message': 'a hint may be provided'}
```

Failed requests also return a proper HTTP status code (e.g. 400, 401, 403, 404, 500, etc).

- 400:** The server received a malformed request. Commonly seen when the JSON payload was incomplete, had unexpected entries or is simply malformed.
- 401:** Returned by requests that require authorization if no secret token was provided or the provided token is invalid.
- 403:** Returned by requests that require authorization if the provided secret token does not have access to the requested resources.
- 404:** Returned when the requested resources are not found (never existed or have been deleted in the mean time)
- 409:** Returned when the requested action violates a database integrity constrain: e.g. uniqueness, bad foreign key references etc. The value of the hint entry is the offending entry from the POST payload.
- 422:** Returned when the requested change violates some semantic data check or soft correlation between entities.
- 500:** Returned when a server error has occurred.

Successful requests will return the following HTTP status codes:

- 200:** The request was successful.
- 201:** Returned by POST requests when a new record was successfully persisted. The *Location* response header is also set and contains a reference to the created record.
- 304:** Returned when the server record has not been modified after the timestamp the client provided in the request.

Timestamps follow the ISO 8601 format without timezone specification. The API assumes that all received timestamps are in UTC and will always return in UTC. Examples:

```
Correct:    2016-01-05 09:00:00.00
Incorrect: 2016-01-05 09:00:00.00+02:00
Incorrect: 2016-01-05 09:00:00.00+02
Incorrect: 2016-01-05 09:00:00.00+00:00
```

It is advised that downstream clients also work with UTC timestamps and convert to localtime only when displaying information to the user.

Data synchronization is supported using the **from** query argument accepted by most bulk endpoints (e.g. endpoints that return lists of records). Note that synchronizing clients will also have to use the **start** and **count** query parameters because all bulk endpoints will only return at most 100 records per request.

Places, Routes and RouteSteps

GET /v1/ttt/routes

Retrieves Routes in a list. No authorization required.

Query parameters:

count: (optional, integer) includes in the returned list only this many records (max. 100).
start: (optional, integer) skips this many records and only includes in the result records after this value.
from: (optional, [timestamp](#)) includes in the returned list only items that have been modified after this date.
returns: records in insertion order (FIFO)

Sample *Route* list:

```
[
{
  "route_id": 87654321,
  "created": "2017-01-05 09:00:00.00",
  "modified": "2017-01-05 09:00:00.00",
  "deleted": null,
  "route_name": "Tramvai 6 to Piața Traian",
  "vehicle_kind": "tram"
  "ext": {
    "ratt_id": "4231",
    // route ids from external/upstream data provider (if any)
  },
  "attrib": {
    "square_name": "6",
    // other misc route attributes ...
  }
},

{
  "route_id": 17654321,
  "created": "2017-01-05 09:10:00.00",
  "modified": "2017-01-05 09:10:00.00",
  "deleted": null,
  "route_name": "Tramvai 6 to Piața Sf. Maria",
  "vehicle_kind": "tram"
  "ext": {
    "ratt_id": "4231",
    // route ids from external/upstream data provider (if any)
  },
  "attrib": {
    "square_name": "6",
    // other misc route attributes ...
  }
},

{
  "route_id": 43218765,
  "created": "2017-01-05 19:00:00.00",
  "modified": "2017-01-05 19:00:00.00",
  "deleted": null,
  "route_name": "Autobuz 33 to Pod Cl. Șagului",
  "vehicle_kind": "bus"
  "ext": {
    "ratt_id": "4237",
    // route ids from external/upstream data provider (if any)
  },
  "attrib": {
    "square_name": "33",
    // other misc route attributes ...
  }
},

{
  "route_id": 87654321,
  "created": "2017-01-05 19:00:10.00",
  "modified": "2017-01-05 19:00:10.00",
  "deleted": null,
  "route_name": "Autobuz 33 to Catedrala Mitropolitană",
  "vehicle_kind": "bus"
  "ext": {
    "ratt_id": "5231",
    // route ids from external/upstream data provider (if any)
  },
  "attrib": {
    "square_name": "33",
    // other misc route attributes ...
  }
},

// ... and so on for every route in the list
]
```

GET /v1/ttt/routes/(route_id)

Retrieves the *Route* object referenced by the given route_id. No authorization required.

Sample *Route* object:

```
{
  "route_id": 87654321,
  "created": "2017-01-05 09:00:00.00",
  "modified": "2017-01-05 09:00:00.00",
```

```

"deleted": null,
"route_name": "Tramvai 6 to Piața Traian",
"vehicle_kind": "bus|tram|trolley|express|metro|river"
"ext": {
  "ratt_id": "4231",
  // route ids from external/upstream data provider (if any)
},
"attrib": {
  "square_name": "6|E8|22|11",
  "short_name": "6|E8|M22|11",
  "heading": "Piața Traian",
  "from": "Piața Sf. Maria"
  // other misc route attributes ...
}
"route_steps": [
  {
    "step_id": 1234321,
    "step_order": 0,
    "is_stop": true, "is_enabled": true,
    "place": {
      "place_id": 12345678,
      "place_name": "Piața Sf. Maria",
      "timezone": "+02:00",
      "latitude": "45.754123",
      "longitude": "21.2123",
      "ext": {
        "ratt_id": "4312",
        "ratt_crypto_name": "garble garble",
        "ratt_fancy_name": "lorem ipsum"
        // place ids from external/upstream data provider
      },
      "attrib": {
        "poi": "shopping",
        // other misc place attributes ...
      }
    },
    "ext": {},
    "attrib": {
      "loading_side": "right"
      // other misc route step attributes ...
    }
  },
  {
    "step_id": 2224321,
    "step_order": 1,
    "is_stop": true, "is_enabled": true,
    "place": {
      "place_id": 32345678,
      "place_name": "Catedrala Mitropolitană",
      "timezone": "+02:00",
      "latitude": "45.754741",
      "longitude": "21.2337",
      "ext": {
        // place ids from external/upstream data provider
      },
      "attrib": {
        // other misc place attributes ...
      }
    },
    "ext": {},
    "attrib": {
      "loading_side": "right"
      // other misc route step attributes ...
    }
  },
  {
    "step_id": 444321,
    "step_order": 2,
    "is_stop": true, "is_enabled": true,
    "place": {
      "place_id": 52345678,
      "place_name": "Catedrala Mitropolitană",
      "timezone": "+02:00",
      "latitude": "45.754741",
      "longitude": "21.2337",
      "ext": {
        // place ids from external/upstream data provider
      },
      "attrib": {
        // other misc place attributes ...
      }
    }
  }
]

```

```

    },
    "ext": {},
    "attrib": {
        "loading_side": "right"
        // other misc route step attributes ...
    }
},
// and so on until the end of the route travel path
]
}

```

POST /v1/ttt/routes

Creates a new route step from the POSTed *RouteStep* object. Returns the UUID and creation timestamp of the newly created object. Requires admin p

Sample *RouteStep* object:

```

{
  "route_name": "Tramvai 6 to Piața Traian",
  "vehicle_kind": "bus|tram|trolley|express|metro|river"
  "ext": {
    "ratt_id": "4231",
    "ratt_crypto_name": "Tv6a|E8|M22|Tb11",
    "ratt_fancy_name": "6|E8|M22|11"
    // ids from external/upstream data provider (if any)
  },
  "attrib": {
    "renovated": true,
    "square_name": "6|E8|22|11",
    "short_name": "6|E8|M22|11",
    "heading": "Piața Traian",
    "from": "Piața Sf. Maria"
    // other misc attributes ...
  }
}

```

POST /v1/ttt/places

Creates a new place from the POSTed *Place* object. Returns the UUID and creation timestamp of the newly created object. Requires admin permission

Sample *Place* object:

```

{
  "place_name": "Catedrala Mitropolitană",
  "timezone": "+02:00",
  "latitude": "45.754741",
  "longitude": "21.2337",
  "ext": {
    "ratt_id": "4312",
    "ratt_crypto_name": "Badea Cartan 2tb",
    "ratt_fancy_name": "Piața Badea Cârțan"
    // ids from external/upstream data provider
  },
  "attrib": {
    "poi": "religious",
    // other misc attributes ...
  }
}

```

POST /v1/ttt/route_steps

Creates a new route step from the POSTed *RouteStep* object. Returns the UUID and creation timestamp of the newly created object. Requires admin p

Sample *RouteStep* object:

```

{
  "route_id": 87654321,
  "place_id": 12345678,
  "step_order": 3,
  "is_stop": true,
  "is_enabled": true,
  "ext": {
    // ids from external/upstream data provider (if any)
  },
  "attrib": {
    "loading_side": "right"
    // other misc attributes ...
  }
}

```

DELETE /v1/ttt/routes

Deletes all routes returning the number of removed records. Requires admin permission.

DELETE /v1/ttt/places

Deletes all places returning the number of removed records. Requires admin permission.

DELETE /v1/ttt/route_steps

Deletes all route steps returning the number of removed records. Requires admin permission.