v6.bvg.transport.rest API documentation

v6.bvg.transport.rest is a REST API. Data is being returned as JSON.

You can just use the API without authentication. There's a rate limit of 100 request/minute (burst 200 requests/minute) set up.

OpenAPI playground

Note: The examples snippets in this documentation uses the url-encode CLI tool of the url-decode-encode-cli package for URL-encoding.

Routes

Note: These routes only wrap hafas-client@6 methods, check their docs for more details.

GET /stops/reachable-from

GET /stops/:id

GET /stops/:id/departures

GET /stops/:id/arrivals

GET /journeys

GET /trips/:id

GET /trips

GET /locations/nearby

GET /locations

GET /radar

GET /journeys/:ref

GET /stops

date/time parameters

v6.bvg.transport.rest is a REST API for the Berlin public transportation system, BVG.

Because it wraps a BVG API of BVG, it **includes all local traffic of Berlin & Brandenburg, as well as some long-distance trains running in the area**. Essentially, it returns whatever data the BVG app shows, **including realtime delays and disruptions**.

- Getting Started
- API documentation
- OpenAPI playground

Why use this API?

Realtime Data

This API returns realtime data whenever its upstream, the API for BVG's mobile app, provides it.

No API Key

You can just use the API without authentication. There's a rate limit of 100 requests/minute set up.

CORS

This API has CORS enabled, so you can query it from any webpage.

Caching-friendly

This API sends ETag & Cache-Control headers, allowing clients cache responses properly.

v6.bvg.transport.rest is a REST API. Data is being returned as JSON.

You can just use the API without authentication. There's a rate limit of 100 request/minute (burst 200 requests/minute) set up.

OpenAPI playground

Note: The examples snippets in this documentation uses the url-encode CLI tool of the url-decode-encode-cli package for URL-encoding.

Routes

Note: These routes only wrap hafas-client@6 methods, check their docs for more details.

- GET /stops/reachable-from
- GET /stops/:id
- GET /stops/:id/departures
- GET /stops/:id/arrivals
- GET /journeys
- GET /trips/:id
- GET /trips
- GET /locations/nearby
- GET /locations
- GET /radar
- GET /journeys/:ref
- GET /stops
- date/time parameters

GET /locations

Uses hafasClient.locations() to find stops/stations, POIs and addresses matching query.

Query Parameters

parameter	description	type	default value
query	Required.	string	–
fuzzy	Find more than exact matches?	boolean	true
results	How many stations shall be shown?	integer	10
stops	Show stops/stations?	boolean	true
addresses	Show addresses?	boolean	true
poi	Show points of interest?	boolean	true

linesOfStops Parse & return lines of each stop/station? boolean false

language Language of the results. string en

pretty Pretty-print JSON responses? boolean true

Example

```
curl 'https://v6.bvg.transport.rest/locations?query=alexanderplatz&results=1' -s | jq
[
	{
		"type": "stop",
		"id": "900100003",
		"name": "S+U Alexanderplatz",
		"location": {
			"type": "location",
			"id": "900100003",
			"latitude": 52.521508,
			"longitude": 13.411267
		},
		"products": {
			"suburban": true,
			"subway": false,
			"tram": false,
			// ...
		}
	}
]
```

GET /locations/nearby

Uses hafasClient.nearby() to find stops/stations & POIs close to the given geolocation.

Query Parameters

parameter	description	type	default value
latitude	Required.	number	–
longitude	Required.	number	–
results	maximum number of results	integer	8
distance	maximum walking distance in meters	integer	–
stops	Return stops/stations?	boolean	true
poi	Return points of interest?	boolean	false
linesOfStops	Parse & expose lines at each stop/station?	boolean	false
language	Language of the results.	string	en
pretty	Pretty-print JSON responses?	boolean	true

Example

```
curl 'https://v6.bvg.transport.rest/locations/nearby?latitude=52.52725&longitude=13.4123' -s | jq
[
  {
    "type": "stop",
    "id": "900100016",
    "name": "U Rosa-Luxemburg-Platz",
    "location": {
      "type": "location",
      "id": "900100016",
      "latitude": 52.528187,
      "longitude": 13.410405
    },
```

```
        "products": { /* … */ },
        "distance": 165
    },
    // …
    {
        "type": "stop",
        "id": "900110005",
        "name": "U Senefelderplatz",
        "location": { /* … */ },
        "products": { /* … */ },
        "distance": 597
    },
    // …
]
```


GET /stops

Uses vbb-stations-autocomplete@4 to find stops/stations matching query. If you don't pass query, it will just return all stops from vbb-stations@7.

Query Parameters

parameter	description	type	default value
query	Filter by name, e.g. mehringd with completion=true, or mehringdamm with completion=false.	string	–
results	How many stops/stations?	number	5
fuzzy	Find other than exact matches?	boolean	false
completion	Search by prefix?	boolean	true

todo

GET /stops/reachable-from

Uses hafasClient.reachableFrom() to find stops/stations reachable within a certain time from an address.

Query Parameters

parameter	description	type	default value
latitude	Required.	number	–
longitude	Required.	number	–
address	Required.	string	–
when	Date & time to compute the reachability for. See date/time parameters. date+time		now
maxTransfers	Maximum number of transfers.	integer	5
maxDuration	Maximum travel duration, in minutes.	integer	infinite
language	Language of the results.	string	en
suburban	Include S-Bahn (S)?	boolean	true
subway	Include U-Bahn (U)?	boolean	true
tram	Include Tram (T)?	boolean	true
bus	Include Bus (B)?	boolean	true
ferry	Include Fähre (F)?	boolean	true
express	Include IC/ICE (E)?	boolean	true
regional	Include RB/RE (R)?	boolean	true
pretty	Pretty-print JSON responses?	boolean	true

Example

```
curl 'https://v6.bvg.transport.rest/stops/reachable-from?latitude=52.52446&longitude=13.40812&address=10178+Berlin-Mitte,+Münzstr.+12' -s | jq
[
```

```
{
	"duration": 4,
	"stations": [
		{
			"type": "stop",
			"id": "900100051",
			"name": "U Weinmeisterstr.",
			"location": { /* ... */ },
			"products": { /* ... */ },
		}
	]
},
// ...
{
	"duration": 7,
	"stations": [
		{
			"type": "stop",
			"id": "900007110",
			"name": "U Bernauer Str.",
			"location": { /* ... */ },
			"products": { /* ... */ }
		},
		{
			"type": "stop",
			"id": "900100004",
```

```
                "name": "S+U Jannowitzbrücke",
                "location": { /* … */ },
                "products": { /* … */ }
            },
            // …
        ]
    },
    // …
]
```


GET /stops/:id

Uses hafasClient.stop() to find a stop/station by ID.

Query Parameters

parameter	description	type	default value
linesOfStops	Parse & expose lines at each stop/station?	boolean	false
language	Language of the results.	string	en
pretty	Pretty-print JSON responses?	boolean	true

Example

```
curl 'https://v6.bvg.transport.rest/stops/900017101' -s | jq
```


```
{
    "type": "stop",
    "id": "900017101",
    "name": "U Mehringdamm",
    "location": {
        "type": "location",
        "id": "900017101",
```

```
            "latitude": 52.49357,
            "longitude": 13.388138
        },
        "products": { /* … */ },
}
```


GET /stops/:id/departures

Uses hafasClient.departures() to get departures at a stop/station.

Query Parameters

parameter	description	type	default value
when	Date & time to get departures for. See date/time parameters.	date+time	now
direction	Filter departures by direction.	string	
duration	Show departures for how many minutes?	integer	10
results	Max. number of departures.	integer	*whatever HAFAS wants
linesOfStops	Parse & return lines of each stop/station?	boolean	false
remarks	Parse & return hints & warnings?	boolean	true
language	Language of the results.	string	en
suburban	Include S-Bahn (S)?	boolean	true
subway	Include U-Bahn (U)?	boolean	true
tram	Include Tram (T)?	boolean	true
bus	Include Bus (B)?	boolean	true
ferry	Include Fähre (F)?	boolean	true
express	Include IC/ICE (E)?	boolean	true
regional	Include RB/RE (R)?	boolean	true
pretty	Pretty-print JSON responses?	boolean	true

Example

```
# at U Kottbusser Tor, in direction U Görlitzer Bahnhof

curl 'https://v6.bvg.transport.rest/stops/900013102/departures?direction=900014101&duration=10' -s | jq

[
	{
		"tripId": "1|61154|54|86|29042020",
		"direction": "Ersatz S+U Warschauer Str.",
		"line": {
			"type": "line",
			"id": "u1",
			"name": "U1",
			"mode": "bus",
			"product": "bus",
			// ...
		},

		"when": "2020-04-29T19:31:00+02:00",
		"plannedWhen": "2020-04-29T19:30:00+02:00",
		"delay": 60,
		"platform": null,
		"plannedPlatform": null,

		"stop": {
			"type": "stop",
			"id": "900013102",
			"name": "U Kottbusser Tor",
```

```
            "location": { /* … */ },
            "products": { /* … */ },
            // …
        },

        "remarks": [ /* … */ ],
    },
    // …
]
```


GET /stops/:id/arrivals

Works like /stops/:id/departures, except that it uses hafasClient.arrivals() to arrivals at a stop/station.

Query Parameters

parameter	description	type	default value
when	Date & time to get departures for. See date/time parameters.	date+time	now
direction	Filter departures by direction.	string	
duration	Show departures for how many minutes?	integer	10
results	Max. number of departures.	integer	whatever HAFAS wants
linesOfStops	Parse & return lines of each stop/station?	boolean	false
remarks	Parse & return hints & warnings?	boolean	true
language	Language of the results.	string	en
suburban	Include S-Bahn (S)?	boolean	true
subway	Include U-Bahn (U)?	boolean	true
tram	Include Tram (T)?	boolean	true
bus	Include Bus (B)?	boolean	true

ferry Include Fähre (F)? boolean true

express Include IC/ICE (E)? boolean true

regional Include RB/RE (R)? boolean true

pretty Pretty-print JSON responses? boolean true

Example

at U Kottbusser Tor, 10 minutes

curl 'https://v6.bvg.transport.rest/stops/900013102/arrivals?duration=10' -s | jq

GET /journeys

Uses hafasClient.journeys() to find journeys from A (from) to B (to).

from (A), to (B), and the optional via must each have one of these formats:

as stop/station ID (e.g. from=900017101 for U Mehringdamm)

as a POI (e.g. from.id=900980720&from.latitude=52.54333&from.longitude=13.35167&from.name=ATZE+Musiktheater for ATZE Musiktheater)

as an address (e.g. from.latitude=52.543333&from.longitude=13.351686&from.address=Voltastr.+17 for Voltastr. 17)

Pagination

Given a response, you can also fetch more journeys matching the same criteria. Instead of from*, to* & departure/arrival, pass earlierRef from the first response as earlierThan to get journeys "before", or laterRef as laterThan to get journeys "after".

Check the hafasClient.journeys() docs for more details.

Query Parameters

parameter description type default value

departure	Compute journeys departing at this date/time. Mutually exclusive with arrival. See date/time parameters.	date+time	now
arrival	Compute journeys arriving at this date/time. Mutually exclusive with departure. See date/time parameters.	date+time	now
earlierThan	Compute journeys "before" an ealierRef.	string	
laterThan	Compute journeys "after" an laterRef.	string	
results	Max. number of journeys.	integer	3
stopovers	Fetch & parse stopovers on the way?	boolean	false
transfers	Maximum number of transfers.	integer	let HAFAS decide
transferTime	Minimum time in minutes for a single transfer.	integer	0
accessibility	partial or complete.	string	not accessible
bike	Compute only bike-friendly journeys?	boolean	false
startWithWalking	Consider walking to nearby stations at the beginning of a journey?	boolean	true
walkingSpeed	slow, normal or fast.	string	normal
tickets	Return information about available tickets?	boolean	false
polylines	Fetch & parse a shape for each journey leg?	boolean	false
subStops	Parse & return sub-stops of stations?	boolean	true
entrances	Parse & return entrances of stops/stations?	boolean	true
remarks	Parse & return hints & warnings?	boolean	true
scheduledDays	Parse & return dates each journey is valid on?	boolean	false
language	Language of the results.	string	en
suburban	Include S-Bahn (S)?	boolean	true
subway	Include U-Bahn (U)?	boolean	true
tram	Include Tram (T)?	boolean	true
bus	Include Bus (B)?	boolean	true
ferry	Include Fähre (F)?	boolean	true

express Include IC/ICE (E)? boolean true

regional Include RB/RE (R)? boolean true

pretty Pretty-print JSON responses? boolean true

Examples

stop/station to POI

curl 'https://v6.bvg.transport.rest/journeys?from=900023201&to.id=900980720&to.name=ATZE+Musiktheater&to.latitude=52.54333&to.longitude=13.35167' -s | jq

without buses, with ticket info

curl 'https://v6.bvg.transport.rest/journeys?from=...&to=...&bus=false&tickets=true' -s | jq

GET /journeys/:ref

Uses hafasClient.refreshJourney() to "refresh" a journey, using its refreshToken.

The journey will be the same (equal from, to, via, date/time & vehicles used), but you can get up-to-date realtime data, like delays & cancellations.

Query Parameters

parameter description type default value

stopovers Fetch & parse stopovers on the way? boolean false

tickets Return information about available tickets? boolean false

polylines Fetch & parse a shape for each journey leg? boolean false

subStops Parse & return sub-stops of stations? boolean true

entrances Parse & return entrances of stops/stations? boolean true

remarks Parse & return hints & warnings? boolean true

scheduledDays Parse & return dates the journey is valid on? boolean false

language Language of the results. string en

pretty Pretty-print JSON responses? boolean true

Example

```
# get the refreshToken of a journey

journey=$(curl 'https://v6.bvg.transport.rest/journeys?from=...&to=...&results=1' -s | jq '.journeys[0]')

refresh_token=$(echo $journey | jq -r '.refreshToken')


# refresh the journey

curl "https://v6.bvg.transport.rest/journeys/$(echo $refresh_token | url-encode)" -s | jq
```

GET /trips/:id

Uses hafasClient.trip() to fetch a trip by ID.

A trip is a specific vehicle, stopping at a series of stops at specific points in time. Departures, arrivals & journey legs reference trips by their ID.

Query Parameters

parameter	description	type	default value
stopovers	Fetch & parse stopovers on the way?	boolean	true
remarks	Parse & return hints & warnings?	boolean	true
polyline	Fetch & parse the geographic shape of the trip?	boolean	false
language	Language of the results.	string	en
pretty	Pretty-print JSON responses?	boolean	true

Example

```
# get the trip ID of a journey leg

journey=$(curl 'https://v6.bvg.transport.rest/journeys?from=...&to=...&results=1' -s | jq '.journeys[0]')

journey_leg=$(echo $journey | jq -r '.legs[0]')

trip_id=$(echo $journey_leg | jq -r '.tripId')
```

fetch the trip

curl "https://v6.bvg.transport.rest/trips/$(echo $trip_id | url-encode)" -s | jq

GET /radar

Uses hafasClient.radar() to find all vehicles currently in an area, as well as their movements.

Query Parameters

parameter	description	type	default value
north	Required. Northern latitude.	number	–
west	Required. Western longitude.	number	–
south	Required. Southern latitude.	number	–
east	Required. Eastern longitude.	number	–
results	Max. number of vehicles.	integer	256
duration	Compute frames for the next n seconds.	integer	30
frames	Number of frames to compute.	integer	3
polylines	Fetch & parse a geographic shape for the movement of each vehicle?	boolean	true
language	Language of the results.	string	en
pretty	Pretty-print JSON responses?	boolean	true

Example

bbox='north=52.52411&west=13.41002&south=52.51942&east=13.41709'

curl "https://v6.bvg.transport.rest/radar?$bbox&results=10" -s | jq

GET /trips

Query Parameters

parameter	description	type	default value
query	line name or Fahrtnummer	string	*

when Date & time to get trips for. See date/time parameters. date+time now

fromWhen Together with untilWhen, forms a time frame to get trips for. Mutually exclusive with when. See date/time parameters. date+time now

untilWhen Together with fromWhen, forms a time frame to get trips for. Mutually exclusive with when. See date/time parameters. date+time now

onlyCurrentlyRunning Only return trips that run within the specified time frame. boolean true

currentlyStoppingAt Only return trips that stop at the specified stop within the specified time frame. string

lineName Only return trips with the specified line name. string

operatorNames Only return trips operated by operators specified by their names, separated by commas. string

stopovers Fetch & parse stopovers of each trip? boolean true

remarks Parse & return hints & warnings? boolean true

subStops Parse & return sub-stops of stations? boolean true

entrances Parse & return entrances of stops/stations? boolean true

language Language of the results. string en

suburban Include S-Bahn (S)? boolean true

subway Include U-Bahn (U)? boolean true

tram Include Tram (T)? boolean true

bus Include Bus (B)? boolean true

ferry Include Fähre (F)? boolean true

express Include IC/ICE (E)? boolean true

regional Include RB/RE (R)? boolean true

pretty Pretty-print JSON responses? boolean true

Date/Time Parameters

Possible formats:

anything that parse-human-relative-time can parse (e.g. tomorrow 2pm)

ISO 8601 date/time string (e.g. 2020-04-26T22:43+02:00)

UNIX timestamp (e.g. 1587933780)