

Beschreibung Fahrplan-API

Die Deutsche Bahn stellt [erstmal] über eine offene Schnittstelle für den Zugriff auf ihre Fahrplandaten[bank] zur Verfügung - [zunächst] eingeschränkt auf den SOLL-Fahrplan des Fernverkehrs. Über eine XML/JSON basierte REST Schnittstelle werden die folgenden Dienste angeboten:

a) Bahnhofssuche (`Location.name Service`): liefert zum angefragten Suchstring ("Pattern") Bahnhöfe eine Liste Bahnhöfen mit ihrer jeweiligen API spezifischen Bahnhofs-IDs .

Beispiel XML: [https://open-api.bahn.de/bin/rest.exe/location.name?authKey=\[REDACTED\]&lang=de&input=Frankfurt+hbf](https://open-api.bahn.de/bin/rest.exe/location.name?authKey=[REDACTED]&lang=de&input=Frankfurt+hbf)

Beispiel JSON: [https://open-api.bahn.de/bin/rest.exe/location.name?authKey=\[REDACTED\]&lang=de&input=Frankfurt+hbf&format=json](https://open-api.bahn.de/bin/rest.exe/location.name?authKey=[REDACTED]&lang=de&input=Frankfurt+hbf&format=json)

b) Abfahrtstafel (`Stationboard Service`): zu einer gegebenen Bahnhof-ID (und optional Datum/Uhrzeit) werden z.B. die planmäßig nächsten <n> - Abfahrten ("Züge" mit ihrem jeweiligen API spezifischen "Zug-Handle") geliefert.

Beispiel XML: [https://open-api.bahn.de/bin/rest.exe/departureBoard?authKey=\[REDACTED\]&lang=de&id=008000105&date=2016-02-22&time=15%3a00](https://open-api.bahn.de/bin/rest.exe/departureBoard?authKey=[REDACTED]&lang=de&id=008000105&date=2016-02-22&time=15%3a00)

Beispiel JSON: [https://open-api.bahn.de/bin/rest.exe/departureBoard?authKey=\[REDACTED\]&lang=de&id=008000105&date=2016-02-22&time=15%3a00&format=json](https://open-api.bahn.de/bin/rest.exe/departureBoard?authKey=[REDACTED]&lang=de&id=008000105&date=2016-02-22&time=15%3a00&format=json)

c) Ankunftstafel (`Stationboard Service`): zu einer gegebenen Bahnhof-ID (und optional Datum/Uhrzeit) werden z.B. die planmäßig nächsten <n> - Ankünfte ("Züge" mit ihrem jeweiligen API spezifischen "Zug-Handle") geliefert.

Beispiel XML: [https://open-api.bahn.de/bin/rest.exe/arrivalBoard?authKey=\[REDACTED\]&lang=de&id=008000105&date=2016-02-22&time=15%3a00](https://open-api.bahn.de/bin/rest.exe/arrivalBoard?authKey=[REDACTED]&lang=de&id=008000105&date=2016-02-22&time=15%3a00)

Beispiel JSON: [https://open-api.bahn.de/bin/rest.exe/arrivalBoard?authKey=\[REDACTED\]&lang=de&id=008000105&date=2016-02-22&time=15%3a00&format=json](https://open-api.bahn.de/bin/rest.exe/arrivalBoard?authKey=[REDACTED]&lang=de&id=008000105&date=2016-02-22&time=15%3a00&format=json)

d) Zuglauf (`Journey Details - vehicle route`): Zuglauf Abfrage (nur als Folge-Request, die aufzurufenden URLs hängen an den Fahrten in der Abfahrtstafel):

Beispiel XML: [https://open-api.bahn.de/bin/rest.exe/journeyDetail?ref=428298%2F321450%2F811954%2F263211%2F80%3Fdate%3D2016-02-22%26station_evald%3D8000105%26station_type%3Darr%26authKey%3D\[REDACTED\]%26lang%3Dde%26](https://open-api.bahn.de/bin/rest.exe/journeyDetail?ref=428298%2F321450%2F811954%2F263211%2F80%3Fdate%3D2016-02-22%26station_evald%3D8000105%26station_type%3Darr%26authKey%3D[REDACTED]%26lang%3Dde%26)

Beispiel JSON: [https://open-api.bahn.de/bin/rest.exe/journeyDetail?ref=428298%2F321450%2F811954%2F263211%2F80%3Fdate%3D2016-02-22%26station_evald%3D8000105%26station_type%3Darr%26authKey%3D\[REDACTED\]%26lang%3Dde%26format%3Djson%26](https://open-api.bahn.de/bin/rest.exe/journeyDetail?ref=428298%2F321450%2F811954%2F263211%2F80%3Fdate%3D2016-02-22%26station_evald%3D8000105%26station_type%3Darr%26authKey%3D[REDACTED]%26lang%3Dde%26format%3Djson%26)



Foto: Max Lautenschläger

DB Vertrieb GmbH

REST Fahrplan-API Version **BETA**

Filename: Fpl-API-Doku-Open-Data-BETA-0_81.docx

Dokument-Version: 1.3_5

Anzahl Seiten: 23

Änderungshistorie

Version	Datum	Autor	Bemerkung
0.1	24.02.16	Doc HK DB Vertrieb and Hacon GmbH	Tayloring

Table of Content

Änderungshistorie	2
1 Introduction	4
2 General principles	4
2.1 Coordinates	4
2.2 Date and time formats	4
2.3 Stateless service vs. data dependency	4
2.4 Route index (not supported)	5
2.5 Realtime information (not supported)	5
2.6 Versioning	5
2.7 Response Format	5
2.8 Authentication	5
2.9 Languages	6
3 Services	7
3.1 Location Service	7
3.1.1 Location.name Service	7
3.2 Stationboard services	7
3.3 Journey detail service	7
3.4 XSD service	8
4 Response formats	8
4.1 Location response	8
4.2 Departure board response	8
4.3 Arrival board response	8
4.4 Journey detail response	8
5 Error codes and messages	9
5.1 ReST Request Errors	9
5.2 Backend Server Errors	9
5.3 Trip Search Errors	9
5.4 Departure and Arrival Board Errors	9
5.5 Journey Details Errors	9

6	Appendix A: location.txt	10
7	Appendix B: departure.txt	16
8	Appendix C: JourneyDetails.txt	21

1 Introduction

The public interface is implemented as a ReST¹ (**R**epresentational **S**tate **T**ransfer) interface which provides different methods for the different functionalities of the journey planner, which are the following services:

- Location
- DepartureBoard
- ArrivalBoard
- JourneyDetail
- XSD

While Location, Trip, ArrvialBoard and DepartureBoard can be called directly the JourneyDetail-Method can only be called by a reference given in a result of the DepartureBoard service.

The XSD-Method might be called directly to download the XSD file with response specification of a certain service method.

The system only implements read-only GET requests which are called by given service URLs and multiple GET parameters to specify the requested journey planner information. The parameter values need to be in ISO-8859-1 URL encoded. The result of each request will be delivered either as XML or JSON (see 2.7) response. If the encoding of URL parameters is not right, the behaviour of the system might deliver unexpected results.

From now on it is assumed, that you have been provided with a base URL of the Deutsche Bahn system. The following documentation of the different requests are described based on this given base URL *<baseurl>*.

2 General principles

There are some general principles which are valid for the different services which are described in this section.

Please note: Especially response structures are prepared to hold further information for potential further use. However - the open data api will provide only selected data on response. Therefore such elements have to be treaded optional.

2.1 Coordinates

Coordinates are always in the WGS84 system, represented as decimal degrees in the interval -90 to 90 for the latitude (lat) and -180 to 180 for the longitude (long).

2.2 Date and time formats

Dates are always represented in the format YYYY-MM-DD. This applies both for request parameters as for dates in responses. Times are always represented in the format HH:MM in 24h nomenclature.

2.3 Stateless service vs. data dependency

All services of the provided interface are stateless as it is required for a ReST protocol. But this has its limitation concerning the journey planner's timetable data. As soon as the timetable data is exchanged (in most cases daily on weekdays), IDs of stops/stations are not necessary valid anymore. The same applies for reference URLs to retrieve journey details. The storage of stop/station IDs and reference URLs to journey details for a longer period except the current user session is not recommended therefore and can only be done on own risk for undetermined behaviour when reusing these IDs or references.

¹ See <http://rest.elkstein.org/> for a tutorial on ReST interfaces.

2.4 Route index (not supported)

A route is the list of stops/stations where a vehicle like a train or bus stops. Every stop/station on a route has its own index which can be used as a reference. This index is also used to identify distinctively if the same stop/station if it is contained several times in one route.

2.5 Realtime information (not supported)

Realtime information will be included in the service as far as it is available in the web based journey planner. It is delivered in addition to the planned departures and arrivals.

2.6 Versioning

Due to enhancements of the API the input parameters and the results can change over time. Different Versions of the API might be available at the same time.

The requested version can be specified by using the version number in the path info:

`https://<baseUrl>/<version>/<servicename>`

The version part is optional, if it is omitted, the latest version will be used. Be aware that omitting the version can break your client when a new API version is introduced. If your client should always use a special version of the API (v2 for example), your URL would look like this: **Fehler! Hyperlink-Referenz ungültig.**

Caveat – No warranty: The open service might change versions and API structure on short notice. However, Deutsche Bahn tries to limit changes if possible.

2.7 Response Format

The interface returns responses either in XML (default) or JSON format.

In order to request a JSON response you have to append the following parameter to each call of the interface: **format=json**.

The JSON content is generated by converting the xml content to JSON automatically. The conversion is done by the following simple rules:

- Element names become object properties
- Text (PCDATA) becomes an object property with name "\$"
`<a>foo` becomes `{ "a": { "$": "foo" } }`
- Nested elements become nested properties
`<a>foo<c>foo</c>`
becomes
`{ "a": { "b": { "$": "foo" }, "c": { "$": "foo" } } }`
- If there are multiple elements with the same name the JSON code contains an array for these element.
`<a>foo1foo2`
becomes
`{ "a": { "b": [{ "$": "foo1" }, { "$": "foo2" }] } }`
- Attribute names become object properties
`foo2`
becomes
`{ "a": { "atb": "foo1", "$": "foo2" } }`

2.8 Authentication

Every client using the API needs to pass a valid authentication key in every request. This parameter has to be appended to the URL: **authKey=<your_key_here>**.

2.9 Languages

The REST API supports multiple languages. The language can be specified by the optional URL parameter `lang=<code>`.

The default language is English and it is used if no language parameter is delivered. The language code has to be lower case.

The supported languages depend on the plan data of the underlying system. The chosen language only influences the returned Notes in the ReST responses.

Code	Language	Code	Language	Code	Language
De	German	fr	French		
Da	Danish			pl	Polish
En	English	it	Italian		
Es	Spanish	nl	Dutch		

3 Services

3.1 Location Service

There are different types of the location service which can be used to get a list of locations using different input parameters.

The response format for all services is defined in `hafasRestLocation.xsd` (see also 4.1 for further details).

3.1.1 Location.name Service

The `location.name` service can be used to perform a pattern matching of a user input and to retrieve a list of possible matches in the journey planner database. Possible matches might be stops/stations, points of interest and addresses. For reasons of backward compatibility the service name `location` can be used as an alias for `location.name`.

The service has only one GET parameter which is called `input`. This parameter contains a string with the user input. The result is a list of possible matches (locations) where the user might pick one entry to perform a trip request with this location as origin or destination or to ask for a departure board or arrival board of this location (stops/stations only) .

The URL to call the service is the following:

https://<baseurl>/location.name?input=<your_user_input>

3.2 Stationboard services

The station board can be retrieved by a call to the `departureBoardService`. This method will return the next 20 departures (or less if not existing) from a given point in time. The service can only be called for stops/stations by using according ID retrieved by the `location` method. The parameter is called `id`. The time and date are defined with the parameters `date` and `time`. In order to get only departures of vehicles with a specified direction you can use the parameter `direction=<stopid>`.

A departure board for Frankfurt(Main)Hbf for the next 20 departures on 19th September, 2011 at 07:02am can be retrieved by calling

<https://<baseurl>/departureBoard?id=800105&date=2011-11-19&time=07:02>

As a response the service will return a result according to `hafasRestDepartureBoard.xsd`. This will contain a list of departures with train/line number, type of transport, departure times (incl. realtime), departure stop/stations (might be different from requested stop), direction text and a track information if available. Every departure will also contain a reference to the journey detail service.

In addition to departure boards the service `arrivalBoard` delivers arriving journeys at a specified stop. The parameters are identical to the parameters of the `departureBoard` service.

As a response the service will return a result according to `hafasRestArrivalBoard.xsd`.

This will contain a list of arrival with train/line number, type of transport, arrival times (incl. realtime), departure stop/stations (might be different from requested stop), the name of the origin stop and a track information if available. Every arrival will also contain a reference to the journey detail service.

3.3 Journey detail service

The `journeyDetail` service will deliver information about the complete route of a vehicle. This service can't be called directly but only by reference URLs in a result of a `departureBoard` request. It contains a list of all stops/stations of this journey including all departure and arrival times (with realtime data if available / not supported right now) and additional information like specific attributes about facilities and other texts.

The response will be returned a result according to the format described in `hafasRestJourneyDetails.xsd`.

3.4 XSD service

The XSD service will return a certain XSD of a certain version. The URL parameter `name` specifies the requested XSD file. The version has to be specified in the rest call as usual. Example: `https://<baseUrl>/v2/xsd?name=hafasRestLocation.xsd`

Each xml response includes these calls as a link in the `NameSpaceSchemaLocation`.

4 Response formats

All services return their responses either in XML or JSON format (see 2.7). Every response is defined in a separate XSD file. The following sections will describe the responses in detail. The formats might be enhanced in the future so the implementation of the parsing should be implemented in view of future possible changes.

4.1 Location response

The location consists of a list of entries, which are either stops/stations or named coordinates. The root element of the response is `LocationList`.

Example



location.txt (Appendix A: location.txt)

4.2 Departure board response

The departure board response contains a list of departures incl. all information concerning times, tracks, realtime data (not supported right now) and journey. It also contains reference URLs to get more details for the different journeys. The root element is `DepartureBoard`.



departure.txt (Appendix B: departure.txt)

4.3 Arrival board response

The arrival board response contains a list of arrivals incl. all information concerning times, tracks, realtime data and journey. It also contains reference URLs to get more details for the different journeys. The root element is `ArrivalBoard`.

4.4 Journey detail response

The journey detail response delivers all information about a single journey (vehicle route). It contains a list of stops including their indexes on the route and their coordinates. It contains also all times, tracks and realtime information if available for the whole route. It also contains the journeys name and type (there might be different names and types on parts of the journey). Finally it contains notes including information about their validity on segments of the total route.

Example



JourneyDetails.txt (Appendix C: JourneyDetails.txt)

5 Error codes and messages

If a request failed an error code and textual description will be returned. The error can be classified into several categories. The Rest API and Backend Server related errors are independent from the called service. Other errors depend on the called service.

5.1 ReST Request Errors

Code	Text
------	------

R0001	Unknown service method
R0002	Invalid or missing request parameters
R0007	Internal communication error

5.2 Backend Server Errors

Code	Text
------	------

S1	The desired connection to the server could not be established or was not stable.
----	--

5.3 Trip Search Errors

Code	Text
------	------

H9380	Dep./Arr./Intermed. or equivalent stations defined more than once
H9360	Error in data field
H9320	The input is incorrect or incomplete
H9300	Unknown arrival station
H9280	Unknown intermediate station
H9260	Unknown departure station
H9250	Part inquiry interrupted
H9240	Unsuccessful search
H9230	An internal error occurred
H9220	Nearby to the given address stations could not be found
H900	Unsuccessful or incomplete search (timetable change)
H892	Inquiry too complex (try entering less intermediate stations)
H891	No route found (try entering an intermediate station)
H890	Unsuccessful search.
H500	Because of too many trains the connection is not complete
H460	One or more stops are passed through multiple times.
H455	Prolonged stop
H410	Display may be incomplete due to change of timetable
H390	Departure/Arrival replaced by an equivalent station
H895	Departure/Arrival are too near
H899	Unsuccessful or incomplete search (timetable change)

5.4 Departure and Arrival Board Errors

Code	Text
------	------

SQ001	No station board available.
SQ002	There was no journey found for the requested board or time.

5.5 Journey Details Errors

Code	Text
------	------

TI001	No trip journey information available.
-------	--

6 Appendix A: location.txt

Request:

JSON:

https://DOMAINE-TOBE-

DEFI-

NED.de/bin/dev/v4/r16.04/rest.exe/location.name?authKey=ZUGRIFFSCODE&lang=de&input=Frankfurt+hbf&format=json

XML:

https://DOMAINE-TOBE-

DEFI-

NED.de/bin/dev/v4/r16.04/rest.exe/location.name?authKey=ZUGRIFFSCODE&lang=de&input=Frankfurt+hbf&

JSON:

Response:

```
{
  "LocationList":{
    "noNamespaceSchemaLocation":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestLocation.xsd"
  },
  "StopLocation":[{
    "name":"Frankfurt (Main) Hbf",
    "lon":"8.663785",
    "lat":"50.107149",
    "id":"008000105"
  },{
    "name":"Frankfurt (M) Flughafen",
    "lon":"8.570971",
    "lat":"50.051209",
    "id":"008000281"
  },{
    "name":"FRANKFURT (MAIN) ",
    "lon":"8.663785",
    "lat":"50.107149",
    "id":"008096021"
  },{
    "name":"Frankfurt (Main) Süd",
    "lon":"8.686456",
    "lat":"50.099364",
    "id":"008002041"
  },{
    "name":"Frankfurt (M) Flughafen Fernbf",
    "lon":"8.570180",
    "lat":"50.053169",
    "id":"008070003"
  },{
    "name":"Frankfurt (Oder) ",
    "lon":"14.546497",
    "lat":"52.336420",
    "id":"008010113"
  },{
    "name":"Frankfurt am Main Hauptbahnhof",
    "lon":"8.664765",
```

```

"lat": "50.107598",
"id": "000100010"
}, {
"name": "Frankfurt (M) Flughafen Regionalbf",
"lon": "8.571250",
"lat": "50.051218",
"id": "008070004"
}, {
"name": "Frankfurt (Main) West",
"lon": "8.639334",
"lat": "50.118862",
"id": "008002042"
}, {
"name": "Frankfurt-Höchst",
"lon": "8.542368",
"lat": "50.102636",
"id": "008000106"
}, {
"name": "Frankfurt-Niederrad",
"lon": "8.637078",
"lat": "50.081287",
"id": "008002050"
}, {
"name": "Frankfurt-Rödelheim",
"lon": "8.607027",
"lat": "50.124615",
"id": "008002052"
}, {
"name": "Frankfurt (Main) Ost",
"lon": "8.708641",
"lat": "50.112569",
"id": "008002039"
}, {
"name": "Frankfurt-Mainkur",
"lon": "8.767440",
"lat": "50.134602",
"id": "008002048"
}, {
"name": "Frankfurt (Oder) -Neuberesinchen",
"lon": "14.546047",
"lat": "52.322325",
"id": "008011557"
}, {
"name": "Frankfurt (Oder) -Rosengarten",
"lon": "14.472632",
"lat": "52.337768",
"id": "008011559"
}, {
"name": "Frankfurt-Sossenheim",
"lon": "8.553523",
"lat": "50.115077",
"id": "008002054"
}, {
"name": "Frankfurt-Unterliederbach",
"lon": "8.528021",
"lat": "50.107913",
"id": "008007158"
}, {

```

```

"name":"Frankfurt am Main - Stadion",
"lon":"8.633249",
"lat":"50.068226",
"id":"008002040"
},{
"name":"Frankfurt-Eschersheim",
"lon":"8.655434",
"lat":"50.158585",
"id":"008002045"
},{
"name":"Frankfurt-Griesheim",
"lon":"8.606398",
"lat":"50.094330",
"id":"008002046"
},{
"name":"Frankfurt-Höchst Farbwerke",
"lon":"8.528731",
"lat":"50.098555",
"id":"008002051"
},{
"name":"Frankfurt-Nied",
"lon":"8.570738",
"lat":"50.102519",
"id":"008002049"
},{
"name":"Frankfurt-Berkersheim",
"lon":"8.697827",
"lat":"50.176483",
"id":"008002043"
},{
"name":"Frankfurt-Frankfurter Berg",
"lon":"8.676550",
"lat":"50.170262",
"id":"008002044"
},{
"name":"Frankfurt-Louisa",
"lon":"8.670365",
"lat":"50.083957",
"id":"008002047"
},{
"name":"Frankfurt (M) Galluswarte",
"lon":"8.644719",
"lat":"50.104030",
"id":"008006690"
},{
"name":"Frankfurt (M) Hauptwache",
"lon":"8.678950",
"lat":"50.113927",
"id":"008006692"
},{
"name":"Frankfurt (M) Konstablerwache",
"lon":"8.687094",
"lat":"50.114619",
"id":"008004429"
},{
"name":"Frankfurt (M) Lokalbahnhof",
"lon":"8.693027",
"lat":"50.102169",

```

```

"id": "008002038"
}, {
"name": "Frankfurt (M) Mühlberg",
"lon": "8.701504",
"lat": "50.101612",
"id": "008002034"
}, {
"name": "Frankfurt (M) Ostendstraße",
"lon": "8.696982",
"lat": "50.112560",
"id": "008002058"
}, {
"name": "Frankfurt (M) Stresemannallee",
"lon": "8.671246",
"lat": "50.094438",
"id": "008002059"
}, {
"name": "Frankfurt (M) Taunusanlage",
"lon": "8.668864",
"lat": "50.113567",
"id": "008006691"
}, {
"name": "Frankfurt (Main) Messe",
"lon": "8.643370",
"lat": "50.112003",
"id": "008079632"
}, {
"name": "Frankfurt-Sindlingen",
"lon": "8.512263",
"lat": "50.087499",
"id": "008002053"
}, {
"name": "Frankfurt-Zeilsheim",
"lon": "8.506006",
"lat": "50.090123",
"id": "008002055"
}, {
"name": "Frankfurt (M) Flughafen Terminal 2",
"lon": "8.588159",
"lat": "50.052683",
"id": "008089361"
}, {
"name": "Frankfurt (Oder) Bahnhof",
"lon": "14.545742",
"lat": "52.338640",
"id": "000740167"
}, {
"name": "Frankfurt (Oder) Friedhof",
"lon": "14.527368",
"lat": "52.331260",
"id": "000740197"
}, {
"name": "Festhalle/Messe, Frankfurt am Main",
"lon": "8.654859",
"lat": "50.112183",
"id": "000100210"
}, {
"name": "Frankensteiner Platz, Frankfurt am Main",

```

```

"lon": "8.692461",
"lat": "50.106915",
"id": "000100902"
}, {
"name": "Frankfurter Berg S-Bahn-Station, Frankfurt am Main",
"lon": "8.676433",
"lat": "50.170703",
"id": "000101391"
}, {
"name": "Friedberger Platz, Frankfurt am Main",
"lon": "8.691750",
"lat": "50.123509",
"id": "000100505"
}, {
"name": "Frankfurt, Markt Taschendorf",
"lon": "10.525685",
"lat": "49.680287",
"id": "000677135"
}, {
"name": "Fechenheim Schießhüttenstraße, Frankfurt am Main",
"lon": "8.773534",
"lat": "50.121424",
"id": "000101548"
}, {
"name": "Flaschenburgstraße, Frankfurt am Main",
"lon": "8.730880",
"lat": "50.100479",
"id": "000101607"
}, {
"name": "Fischstein, Frankfurt am Main",
"lon": "8.624844",
"lat": "50.129280",
"id": "000101223"
}, {
"name": "Fritz-Tarnow-Straße, Frankfurt am Main",
"lon": "8.668684",
"lat": "50.142908",
"id": "000101303"
}, {
"name": "Friedberger Warte, Frankfurt am Main",
"lon": "8.698438",
"lat": "50.139636",
"id": "000101430"
}]
}
}

```

XML:

```

<LocationList xsi:noNamespaceSchemaLocation="https://DOMAINE-TOBE-DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestLocation.xsd"
>
  <StopLocation name="Frankfurt (Main)Hbf" lon="8.663785"
lat="50.107149" id="008000105"/>
  <StopLocation name="Frankfurt (M) Flughafen" lon="8.570971"
lat="50.051209" id="008000281"/>

```

```

    <StopLocation name="FRANKFURT (MAIN)" lon="8.663785"
lat="50.107149" id="008096021"/>
    <StopLocation name="Frankfurt (Main) Süd" lon="8.686456"
lat="50.099364" id="008002041"/>
    <StopLocation name="Frankfurt (M) Flughafen Fernbf" lon="8.570180"
lat="50.053169" id="008070003"/>
    <StopLocation name="Frankfurt (Oder)" lon="14.546497"
lat="52.336420" id="008010113"/>
    <StopLocation name="Frankfurt am Main Hauptbahnhof" lon="8.664765"
lat="50.107598" id="000100010"/>
    <StopLocation name="Frankfurt (M) Flughafen Regionalbf"
lon="8.571250" lat="50.051218" id="008070004"/>
    <StopLocation name="Frankfurt (Main) West" lon="8.639334"
lat="50.118862" id="008002042"/>
    <StopLocation name="Frankfurt-Höchst" lon="8.542368"
lat="50.102636" id="008000106"/>
    <StopLocation name="Frankfurt-Niederrad" lon="8.637078"
lat="50.081287" id="008002050"/>
    <StopLocation name="Frankfurt-Rödelheim" lon="8.607027"
lat="50.124615" id="008002052"/>
    <StopLocation name="Frankfurt (Main) Ost" lon="8.708641"
lat="50.112569" id="008002039"/>
    <StopLocation name="Frankfurt-Mainkur" lon="8.767440"
lat="50.134602" id="008002048"/>
    <StopLocation name="Frankfurt (Oder) -Neuberesinchen"
lon="14.546047" lat="52.322325" id="008011557"/>
    <StopLocation name="Frankfurt (Oder) -Rosengarten" lon="14.472632"
lat="52.337768" id="008011559"/>
    <StopLocation name="Frankfurt-Sossenheim" lon="8.553523"
lat="50.115077" id="008002054"/>
    <StopLocation name="Frankfurt-Unterliederbach" lon="8.528021"
lat="50.107913" id="008007158"/>
    <StopLocation name="Frankfurt am Main - Stadion" lon="8.633249"
lat="50.068226" id="008002040"/>
    <StopLocation name="Frankfurt-Eschersheim" lon="8.655434"
lat="50.158585" id="008002045"/>
    <StopLocation name="Frankfurt-Griesheim" lon="8.606398"
lat="50.094330" id="008002046"/>
    <StopLocation name="Frankfurt-Höchst Farbwerke" lon="8.528731"
lat="50.098555" id="008002051"/>
    <StopLocation name="Frankfurt-Nied" lon="8.570738" lat="50.102519"
id="008002049"/>
    <StopLocation name="Frankfurt-Berkersheim" lon="8.697827"
lat="50.176483" id="008002043"/>
    <StopLocation name="Frankfurt-Frankfurter Berg" lon="8.676550"
lat="50.170262" id="008002044"/>
    <StopLocation name="Frankfurt-Louisa" lon="8.670365"
lat="50.083957" id="008002047"/>
    <StopLocation name="Frankfurt (M) Galluswarte" lon="8.644719"
lat="50.104030" id="008006690"/>
    <StopLocation name="Frankfurt (M) Hauptwache" lon="8.678950"
lat="50.113927" id="008006692"/>
    <StopLocation name="Frankfurt (M) Konstablerwache" lon="8.687094"
lat="50.114619" id="008004429"/>
    <StopLocation name="Frankfurt (M) Lokalbahnhof" lon="8.693027"
lat="50.102169" id="008002038"/>
    <StopLocation name="Frankfurt (M) Mühlberg" lon="8.701504"
lat="50.101612" id="008002034"/>

```



```

    <StopLocation name="Frankfurt (M) Ostendstraße" lon="8.696982"
lat="50.112560" id="008002058"/>
    <StopLocation name="Frankfurt (M) Stresemannallee" lon="8.671246"
lat="50.094438" id="008002059"/>
    <StopLocation name="Frankfurt (M) Taunusanlage" lon="8.668864"
lat="50.113567" id="008006691"/>
    <StopLocation name="Frankfurt (Main) Messe" lon="8.643370"
lat="50.112003" id="008079632"/>
    <StopLocation name="Frankfurt-Sindlingen" lon="8.512263"
lat="50.087499" id="008002053"/>
    <StopLocation name="Frankfurt-Zeilsheim" lon="8.506006"
lat="50.090123" id="008002055"/>
    <StopLocation name="Frankfurt (M) Flughafen Terminal 2"
lon="8.588159" lat="50.052683" id="008089361"/>
    <StopLocation name="Frankfurt (Oder) Bahnhof" lon="14.545742"
lat="52.338640" id="000740167"/>
    <StopLocation name="Frankfurt (Oder) Friedhof" lon="14.527368"
lat="52.331260" id="000740197"/>
    <StopLocation name="Festhalle/Messe, Frankfurt am Main"
lon="8.654859" lat="50.112183" id="000100210"/>
    <StopLocation name="Frankensteiner Platz, Frankfurt am Main"
lon="8.692461" lat="50.106915" id="000100902"/>
    <StopLocation name="Frankfurter Berg S-Bahn-Station, Frankfurt am
Main" lon="8.676433" lat="50.170703" id="000101391"/>
    <StopLocation name="Friedberger Platz, Frankfurt am Main"
lon="8.691750" lat="50.123509" id="000100505"/>
    <StopLocation name="Frankfurt, Markt Taschendorf" lon="10.525685"
lat="49.680287" id="000677135"/>
    <StopLocation name="Fechenheim Schießhüttenstraße, Frankfurt am
Main" lon="8.773534" lat="50.121424" id="000101548"/>
    <StopLocation name="Flaschenburgstraße, Frankfurt am Main"
lon="8.730880" lat="50.100479" id="000101607"/>
    <StopLocation name="Fischstein, Frankfurt am Main" lon="8.624844"
lat="50.129280" id="000101223"/>
    <StopLocation name="Fritz-Tarnow-Straße, Frankfurt am Main"
lon="8.668684" lat="50.142908" id="000101303"/>
    <StopLocation name="Friedberger Warte, Frankfurt am Main"
lon="8.698438" lat="50.139636" id="000101430"/>
</LocationList>

```

7 Appendix B: departure.txt

Request:

JSON:

<https://DOMAINE-TOBE-DEFI->

[NED.de/bin/dev/v4/r16.04/rest.exe/departureBoard?authKey=ZUGRIFFCODE&lang=de&id=008000105&date=2016-02-22&time=15%3a00&format=json](https://DOMAINE-TOBE-DEFI-NED.de/bin/dev/v4/r16.04/rest.exe/departureBoard?authKey=ZUGRIFFCODE&lang=de&id=008000105&date=2016-02-22&time=15%3a00&format=json)

XML:

<https://DOMAINE-TOBE-DEFI->

[NED.de/bin/dev/v4/r16.04/rest.exe/departureBoard?authKey=ZUGRIFFCODE&lang=de&id=008000105&date=2016-02-22&time=15%3a00&](https://DOMAINE-TOBE-DEFI-NED.de/bin/dev/v4/r16.04/rest.exe/departureBoard?authKey=ZUGRIFFCODE&lang=de&id=008000105&date=2016-02-22&time=15%3a00&)

Response:

```

{
  "DepartureBoard": {

```

```

    "noNamespaceSchemaLocation":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestDepartureBoar
d.xsd",
    "Departure":[{
        "name":"RE 15306",
        "type":"RE",
        "stopid":"8000105",
        "stop":"Frankfurt (Main) Hbf",
        "time":"15:01",
        "date":"2016-02-22",
        "direction":"Limburg (Lahn) ",
        "track":"3",
        "JourneyDetailRef":{
            "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=832026%2F4665
06%2F733370%2F89343%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
        }
    },{
        "name":"ICE 1631",
        "type":"ICE",
        "stopid":"8000105",
        "stop":"Frankfurt (Main) Hbf",
        "time":"15:02",
        "date":"2016-02-22",
        "direction":"Berlin Gesundbrunnen",
        "track":"13",
        "JourneyDetailRef":{
            "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=459927%2F3302
79%2F329704%2F11543%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
        }
    },{
        "name":"S 4",
        "type":"S",
        "stopid":"8098105",
        "stop":"Frankfurt Hbf (tief)",
        "time":"15:02",
        "date":"2016-02-22",
        "direction":"Langen (Hess) ",
        "track":"101",
        "JourneyDetailRef":{
            "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=208347%2F2594
96%2F130674%2F4127%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
        }
    },{
        "name":"S 9",
        "type":"S",

```

```

    "stopid":"8098105",
    "stop":"Frankfurt Hbf (tief)",
    "time":"15:02",
    "date":"2016-02-22",
    "direction":"Wiesbaden Hbf",
    "track":"103",
    "JourneyDetailRef":{
      "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=153609%2F2415
74%2F69160%2F16632%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
    }
  },{
    "name":"S 6",
    "type":"S",
    "stopid":"8098105",
    "stop":"Frankfurt Hbf (tief)",
    "time":"15:04",
    "date":"2016-02-22",
    "direction":"Friedberg (Hess)",
    "track":"104",
    "JourneyDetailRef":{
      "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=847389%2F4726
03%2F990992%2F213048%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
    }
  },{
    "name":"S 1",
    "type":"S",
    "stopid":"8098105",
    "stop":"Frankfurt Hbf (tief)",
    "time":"15:04",
    "date":"2016-02-22",
    "direction":"Rödermark-Ober Roden",
    "track":"102",
    "JourneyDetailRef":{
      "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=686085%2F4191
16%2F542178%2F42394%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
    }
  },{
    "name":"S 2",
    "type":"S",
    "stopid":"8000105",
    "stop":"Frankfurt (Main) Hbf",
    "time":"15:04",
    "date":"2016-02-22",
    "direction":"Hofheim (Taunus)",
    "track":"22",
    "JourneyDetailRef":{

```

```

    "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=163092%2F2447
95%2F361412%2F126342%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
    }
  },{
    "name":"ICE 577",
    "type":"ICE",
    "stopid":"8000105",
    "stop":"Frankfurt (Main) Hbf",
    "time":"15:05",
    "date":"2016-02-22",
    "direction":"Stuttgart Hbf",
    "track":"6",
    "JourneyDetailRef":{
      "ref":"https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=953499%2F4965
17%2F127984%2F253841%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26format%3Djson%26"
    }
  }
}
}
}
}

```

XML:

```

<DepartureBoard xsi:noNamespaceSchemaLocation="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestDepartureBoar
d.xsd">
  <Departure name="RE 15306" type="RE" stopid="8000105"
stop="Frankfurt (Main) Hbf" time="15:01" date="2016-02-22" direc-
tion="Limburg (Lahn) " track="3">
    <JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=46584%2F20469
2%2F311190%2F67%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
  </Departure>
  <Departure name="ICE 1631" type="ICE" stopid="8000105"
stop="Frankfurt (Main) Hbf" time="15:02" date="2016-02-22" direc-
tion="Berlin Gesundbrunnen" track="13">
    <JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=194082%2F2416
64%2F239690%2F55151%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
  </Departure>
  <Departure name="S 4" type="S" stopid="8098105" stop="Frankfurt
Hbf (tief)" time="15:02" date="2016-02-22" direction="Langen (Hess)"
track="101">
    <JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-

```

```

NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=272601%2F2809
14%2F347046%2F82671%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
</Departure>
<Departure name="S 9" type="S" stopid="8098105" stop="Frankfurt
Hbf (tief)" time="15:02" date="2016-02-22" direction="Wiesbaden Hbf"
track="103">
<JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=140445%2F2371
86%2F250478%2F78433%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
</Departure>
<Departure name="S 6" type="S" stopid="8098105" stop="Frankfurt
Hbf (tief)" time="15:04" date="2016-02-22" direction="Friedberg (Hess)"
track="104">
<JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=364896%2F3117
72%2F321266%2F39016%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
</Departure>
<Departure name="S 1" type="S" stopid="8098105" stop="Frankfurt
Hbf (tief)" time="15:04" date="2016-02-22" direction="Rödermark-Ober
Roden" track="102">
<JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=955503%2F5089
22%2F270568%2F183217%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8098105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
</Departure>
<Departure name="S 2" type="S" stopid="8000105"
stop="Frankfurt (Main) Hbf" time="15:04" date="2016-02-22" direc-
tion="Hofheim (Taunus)" track="22">
<JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=218745%2F2633
46%2F157844%2F6007%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
</Departure>
<Departure name="ICE 577" type="ICE" stopid="8000105"
stop="Frankfurt (Main) Hbf" time="15:05" date="2016-02-22" direc-
tion="Stuttgart Hbf" track="6">
<JourneyDetailRef ref="https://DOMAINE-TOBE-
DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=20532%2F18552
8%2F190186%2F88249%2F80%3Fdate%3D2016-02-
22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%3DZUGRIFFS
CODE%26lang%3Dde%26"/>
</Departure>
</DepartureBoard>

```

8 Appendix C: JourneyDetails.txt

Request:

JSON:

https://DOMAINE-TOBE-

DEFI-

NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=459927%2F330279%2F329704%2F11543%2F80%3Fdate%3D2016-02-

22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%ZUGRIFFSKEY%26lang%3Dde%26format%3Djson%26

XML:

https://DOMAINE-TOBE-

DEFI-

NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/journeyDetail?ref=459927%2F330279%2F329704%2F11543%2F80%3Fdate%3D2016-02-

22%26station_evaId%3D8000105%26station_type%3Ddep%26authKey%ZUGRIFFSKEY%26lang%3Dde%26

Response:

```
{
```

```
"JourneyDetail":{
```

```
  "noNamespaceSchemaLocation":"https://DOMAINE-TOBE-DEFI-
```

```
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestJourneyDetail.xsd",
```

```
  "Stops":{
```

```
    "Stop":[{
```

```
      "name":"Frankfurt (Main) Hbf",
```

```
      "id":"8000105",
```

```
      "lon":"8.663785",
```

```
      "lat":"50.107149",
```

```
      "routeIdx":"0",
```

```
      "depTime":"15:02",
```

```
      "depDate":"2016-02-22",
```

```
      "track":"13"
```

```
    },{
```

```
      "name":"Erfurt Hbf",
```

```
      "id":"8010101",
```

```
      "lon":"11.038501",
```

```
      "lat":"50.972549",
```

```
      "routeIdx":"1",
```

```
      "arrTime":"17:10",
```

```
      "arrDate":"2016-02-22",
```

```
      "depTime":"17:12",
```

```
      "depDate":"2016-02-22",
```

```
      "track":"10"
```

```
    },{
```

```
      "name":"Halle (Saale) Hbf",
```

```
      "id":"8010159",
```

```
      "lon":"11.987088",
```

```
      "lat":"51.477509",
```

```
      "routeIdx":"2",
```

```
      "arrTime":"17:46",
```

```
      "arrDate":"2016-02-22",
```

```
      "depTime":"17:48",
```

```
      "depDate":"2016-02-22",
```

```
      "track":"6"
```

```
    },{
```

```

    "name":"Berlin Südkreuz",
    "id":"8011113",
    "lon":"13.365314",
    "lat":"52.475042",
    "routeIdx":"3",
    "arrTime":"18:54",
    "arrDate":"2016-02-22",
    "depTime":"18:56",
    "depDate":"2016-02-22",
    "track":"8"
  },{
    "name":"Berlin Hbf (tief)",
    "id":"8098160",
    "lon":"13.369548",
    "lat":"52.525589",
    "routeIdx":"4",
    "arrTime":"19:01",
    "arrDate":"2016-02-22",
    "depTime":"19:04",
    "depDate":"2016-02-22",
    "track":"6"
  },{
    "name":"Berlin Gesundbrunnen",
    "id":"8011102",
    "lon":"13.388515",
    "lat":"52.548961",
    "routeIdx":"5",
    "arrTime":"19:09",
    "arrDate":"2016-02-22",
    "track":"9"
  ]
},
"Names":{
  "Name":{
    "name":"ICE 1631",
    "routeIdxFrom":"0",
    "routeIdxTo":"5"
  }
},
"Types":{
  "Type":{
    "type":"ICE",
    "routeIdxFrom":"0",
    "routeIdxTo":"5"
  }
},
"Operators":{
  "Operator":{
    "name":"DPN",
    "routeIdxFrom":"0",
    "routeIdxTo":"5"
  }
},
"Notes":{
  "Note":{
    "key":"BR",
    "priority":"450",
    "routeIdxFrom":"0",

```

```

    "routeIdxTo": "5",
    "$": "Bordrestaurant"
  }
}
}
}

```

XML:

```

<JourneyDetail xsi:noNamespaceSchemaLocation="https://DOMAINE-TOBE-DEFI-
NED.de/bin/dev/v4/r16.04/rest.exe/v1.0/xsd?name=hafasRestJourneyDetail
.xsd">
  <Stops>
    <Stop name="Frankfurt(Main)Hbf" id="8000105" lon="8.663785"
lat="50.107149" routeIdx="0" depTime="15:02" depDate="2016-02-22"
track="13"/>
    <Stop name="Erfurt Hbf" id="8010101" lon="11.038501"
lat="50.972549" routeIdx="1" arrTime="17:10" arrDate="2016-02-22" dep-
Time="17:12" depDate="2016-02-22" track="10"/>
    <Stop name="Halle(Saale)Hbf" id="8010159" lon="11.987088"
lat="51.477509" routeIdx="2" arrTime="17:46" arrDate="2016-02-22" dep-
Time="17:48" depDate="2016-02-22" track="6"/>
    <Stop name="Berlin Südkreuz" id="8011113" lon="13.365314"
lat="52.475042" routeIdx="3" arrTime="18:54" arrDate="2016-02-22" dep-
Time="18:56" depDate="2016-02-22" track="8"/>
    <Stop name="Berlin Hbf (tief)" id="8098160" lon="13.369548"
lat="52.525589" routeIdx="4" arrTime="19:01" arrDate="2016-02-22" dep-
Time="19:04" depDate="2016-02-22" track="6"/>
    <Stop name="Berlin Gesundbrunnen" id="8011102" lon="13.388515"
lat="52.548961" routeIdx="5" arrTime="19:09" arrDate="2016-02-22"
track="9"/>
  </Stops>
  <Names>
    <Name name="ICE 1631" routeIdxFrom="0" routeIdxTo="5"/>
  </Names>
  <Types>
    <Type type="ICE" routeIdxFrom="0" routeIdxTo="5"/>
  </Types>
  <Operators>
    <Operator name="DPN" routeIdxFrom="0" routeIdxTo="5"/>
  </Operators>
  <Notes>
    <Note key="BR" priority="450" routeIdxFrom="0" rou-
teIdxTo="5">Bordrestaurant</Note>
  </Notes>
</JourneyDetail>

```