Journey-Service (J-S) routing basics

Contents

Public transportation routing

Journey-Service (https://developer.sbb.ch/apis/journey-service/documentation) is a routing service provided by SBB and is widely used by SBB public transportation (train, bus, tram,..) related applications, such as https://www.sbb.ch/en/home.html.

Goals of J-S:

- **Make it easy for developers to visualize routing use cases in end-user applications** (for e.g. itineraries, ticket fare sales, maps,..)
- Providing consistent journey data

  - travelling from A -> B -> C is based on complex and efficient routing algorithms and weighting parameters, considering various origin (above "A")/destination (above "C") "points" and available transport-products (specific vehicles operated by various transport organisations) within a given time-window.
- Passing vias (for e.g. "B" above) may impact a vehicle change for passengers, therefore switching information is provided (for e.g. duration, track, involved footpath).
- Provide a yearly managed schedule (typically from 1st decade of December) and enrich by current realtime-data. (Typically there are major routing changes/optimizations per period in Switzerland).

Who is this manual for?

It is meant for developers using J-S mainly to understand relevant concepts about provided data by its REST-API.

It is not a complete or very sophisticated reference about routing at all. Please keep to more detailed specifications if you need to drill-down deeper, for e.g.:

- https://dms.vdv.de/mitglieder/Seiten/ojp.aspx
- https://opendata.swiss/de/dataset/open-journey-planner-2020-beta
- Transmodel (or related implementations such as NeTex, Siri, ..)
- https://www.opentripplanner.org/

If you plan a route from A to C, possible trips rely on well scheduled trips, however things may go wrong and realtime aspects (like delays, broken vehicles, station changes, ..) may impact your trip effective routes or by time-behaviour.

Routing API

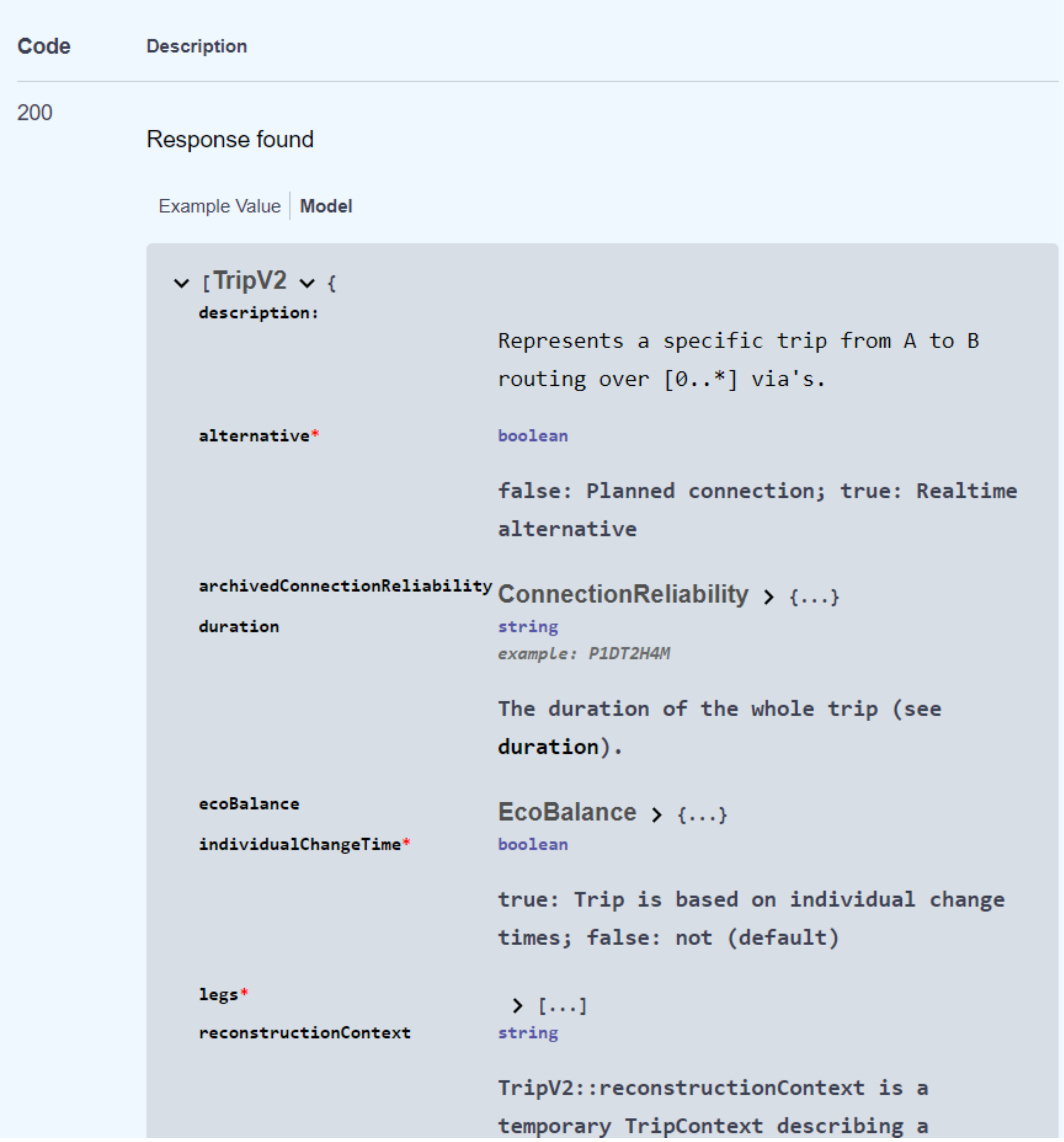
J-S v2 (and v1) is rather a SBB specific approach, optimized for a variety of SBB consumers and their requirements.

See also: Developer Manual

HTTP GET Requests

Most requests return a single or a list of a specific routing data type.

For a Request parameter and response models description please refer to API-doc (Swagger) https://developer.sbb.ch/apis/journey-service/documentation for .e.g:



Code	Description
200	Response found

Example Value | Model

```
[TripV2 {
description:                      Represents a specific trip from A to B
                                  routing over [0..*] via's.
alternative*                      boolean
                                  false: Planned connection; true: Realtime
                                  alternative
archivedConnectionReliability     ConnectionReliability > {...}
duration                          string
                                  example: P1DT2H4M
                                  The duration of the whole trip (see
                                  duration).
ecoBalance                        EcoBalance > {...}
individualChangeTime*             boolean
                                  true: Trip is based on individual change
                                  times; false: not (default)
legs*                              > [...]
reconstructionContext             string
                                  TripV2::reconstructionContext is a
                                  temporary TripContext describing a
```

JourneyDetail

Describes the **whole route**/path **a given transport-product** (physical vehicle) **operates in a given time-window**.

A JourneyDetail consists of an ordered list of Stop’s (increasing routeIndex per Stop).

Partial JourneyDetails (~subtypes) are:

- Departure
- Arrival
- Leg

Trip

A Trip represents a complete ride from the perspective reps. search-criterias of a traveler/passenger, therefore in the example above it consists of 2 Legs (A->B and B->C) and various other data.

Trip::isValid

(de: Gültigkeit)

Basically this is a complex value depending on when you update your trip-Request!

1. Initially you get a specific Trip(V2) by calling /trips
    a. isValid=true (otherwise the trip won't probably make it into the response at all)
2. Anytime later you request an update for that specific Trip by /reconstructionContext
    a. isValid=true remains most often as totally rideable (even if some realtime fields are set)
    b. isValid=false toggled from previous true to false, by means there is any sudden realtime aspect that **makes the whole Trip impossible at the time of request** (for e.g. the connecting train at a Via is not reachable any more)
        i. analyse this case very carefully! Your passenger will probably not reach the destination as planned (separate alternatives must be searched explicitely!)
        ii. as soon as flag toggles to false, any fields might fall back to planned data only (because realtime updates might be irrelevant by now)
        iii. for SBB Mobile users there is a Push-Mechanism which might trigger observed trips directly into the App

Trip::isAlternate

(de: Ersatzzug)

Trip::isCancelled

(de: Ausfall)

Trip::legs

Each leg is assigned to a physical transport-product (vehicle), by means passengers may get-on and get-off. If you reach a via (change to next Leg) a further alighting/boarding will be necessary to another transport-product.

Leg

(In J-S::B2P also called Segment.)

Leg::isCancelled or ::partiallyCancelled

Leg::isReachable

Stop

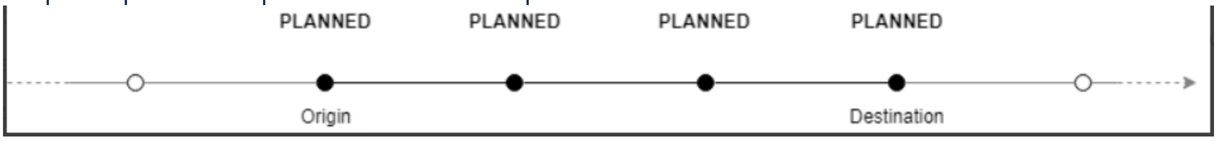
Departure/Arrival times

Platform (changed)

Each Stop has a proper StopStatus, which might be visualized on a JourneyDetail:

PLANNED

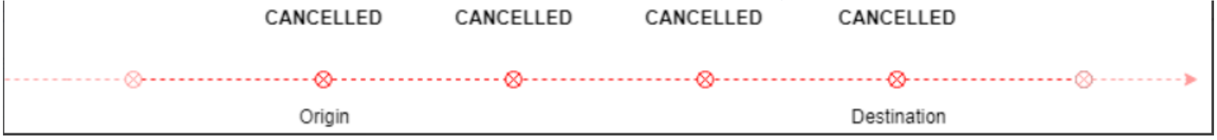
Stop was planned in plan data and is a stop in realtime data. This is the default value.



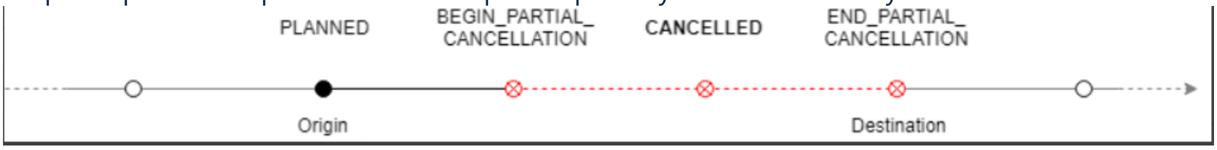
CANCELLED/PARTIALLY_CANCELLED

Possible cases:

Stop was planned in plan data but it is part of cancelled Journey in realtime data:



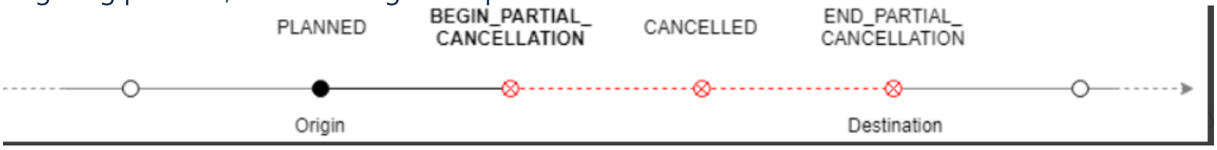
Stop was planned in plan data but it is part of partially cancelled Journey in realtime data:



BEGIN_PARTIAL_CANCELLATION

Stop was planned in plan data but is first stop of partially cancellation on Journey in realtime data.

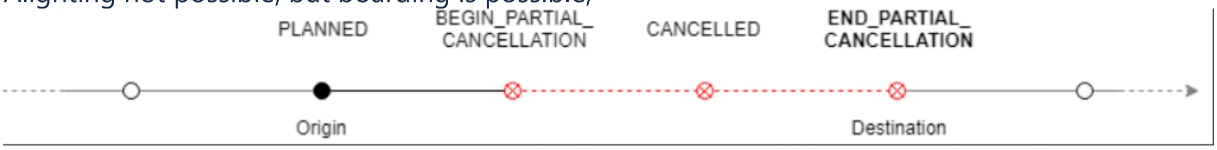
Alighting possible, but boarding is not possible.



END_PARTIAL_CANCELLATION

Stop was planned in plan data but is last stop of partially cancellation on Journey in realtime data.

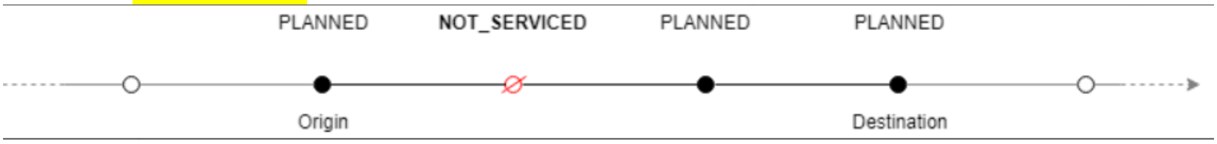
Alighting not possible, but boarding is possible,



NOT_SERVICED

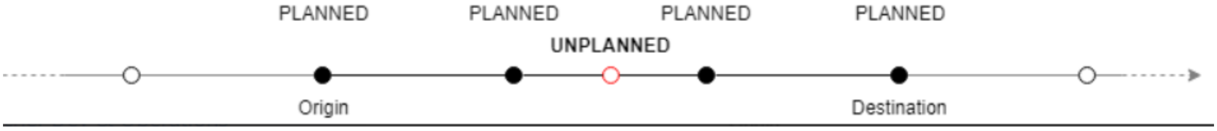
A station which was planned in plan data but is no stop in realtime data. (de: ausserordentliche Durchfahrt).

Condition ::**isRedirected** **==** **true**



UNPLANNED

A station which was not planned in plan data bus is a stop in realtime data. (de: ausserordentlicher Halt)



Other realtime aspects

Delays

For delayed transport-products typically operation-management

Messages

Summaries