



# HANDBUCH **VAO** ReST API

Version 2025-05-22  
Verkehrsauskunft Österreich VAO GmbH



## Inhalt

<b>1</b>	<b>ALLGEMEINES ZUM HANDBUCH</b>	<b>5</b>
1.1	ÄNDERUNGSHISTORIE	5
1.2	ÜBER DAS HANDBUCH	7
<b>2</b>	<b>UMFANG &amp; DOKUMENTATION</b>	<b>8</b>
2.1	UMFANG	8
2.2	MASCHINENLESBARE BESCHREIBUNGEN	8
<b>3</b>	<b>VERSIONIERUNG UND BETRIEBSDAUER</b>	<b>9</b>
3.1	VERSIONEN	9
3.2	BETRIEBSDAUER	9
<b>4</b>	<b>AUFRUF &amp; AUTHENTIFIZIERUNG</b>	<b>10</b>
4.1	AUFRUF	10
4.2	VERWENDUNG VON URI SYNTAX	10
4.3	AUTHENTIFIZIERUNG	10
4.4	ERGEBNISFORMAT	10
<b>5</b>	<b>WICHTIGE INFORMATIONEN</b>	<b>13</b>
5.1	ZUSTANDSLOSER DIENST UND DATENAKTUALISIERUNG (GÜLTIGKEIT VON IDS UND REFS)	13
5.2	KOORDINATEN	13
5.3	DATUM- UND ZEITANGABEN	13
5.4	SPRACHEN	14
5.5	REQUEST VERFOLGUNG	14
5.6	HALTESTELLENMODELLIERUNG UND IDS	14
5.7	FAHRPLANDATEN (SOLL-DATEN)	16
5.8	ÖV PRODUKTKLASSEN (VERKEHRSMITTEL-KATEGORIEN)	17
5.9	ECHTZEITINFORMATIONEN IM ÖV (IST-DATEN)	19
5.10	ECHTZEITINFORMATIONEN IM IV	23
5.11	MAUTINFORMATIONEN	24
5.12	ORTSPUNKTE UND VERKEHRSNETZ	25
5.13	EINSCHRÄNKUNGEN BEI ROUTEN IM AUSLAND	25
5.14	WETTERINFORMATIONEN	25
5.15	DATEN ZU E-AUTO LADESTELLEN	26
5.16	POLYLINIE DER FAHRT	26
<b>6</b>	<b>SERVICE-ÜBERSICHT</b>	<b>27</b>
<b>7</b>	<b>LOCATION.NAME (ORTSPUNKTAUFLÖSUNG MIT NAMEN)</b>	<b>27</b>
7.1	REQUEST PARAMETER	28
7.2	VERWENDUNG ZUSÄTZLICHER KOORDINATEN (STANDORT-BASIERTE AUFLÖSUNG)	29
7.3	FILTER-MODUS (PARAMETER FILTERMODE)	30
7.4	HÖHER AUFLÖSBARE ORTE	30



<b>8</b>	<b>LOCATION.NEARBYSTOPS (UMKREISSUCHE MIT KOORDINATEN)</b>	<b>30</b>
8.1	REQUEST PARAMETER	31
8.2	META-FILTER FÜR BESTIMMTE INHALTE IN DER UMKREISSUCHE	32
8.3	META-FILTER FÜR POI-FILTERUNG	34
<b>9</b>	<b>LOCATION.DETAILS (ORTSPUNKTDETAILS)</b>	<b>37</b>
9.1	REQUEST PARAMETER	38
<b>10</b>	<b>ADDRESSLOOKUP (ADRESS-SUCHE UM KOORDINATE)</b>	<b>40</b>
10.1	REQUEST PARAMETER	40
<b>11</b>	<b>TRIP SERVICE</b>	<b>41</b>
11.1	REQUEST PARAMETER	44
11.2	MODALITÄTEN UND KOMBINATIONEN (GROUPFILTER)	52
11.3	ROUTEN-SUCHALGORITHMUS FÜR DEN ÖV	55
11.4	ROUTEN-SUCHALGORITHMUS FÜR DEN IV	56
11.5	META-PROFILE FÜR IV ROUTING (AUTO, RAD, E-AUTO)	57
11.6	META-PROFILE FÜR E-AUTO ROUTING	60
11.7	VIA-ORTE IN DER ÖV UND IV ROUTE	61
11.8	INTERPRETATION DER HINWEISE IN DEN SEGMENTEN EINER IV ROUTE	63
<b>12</b>	<b>GIS ROUTEN SERVICE</b>	<b>64</b>
12.1	REQUEST PARAMETER	64
<b>13</b>	<b>DEPARTUREBOARD SERVICE</b>	<b>64</b>
13.1	REQUEST PARAMETER	65
13.2	SCROLLING	68
<b>14</b>	<b>ARRIVALBOARD (ANKUNFTSTAFEL)</b>	<b>68</b>
14.1	REQUEST PARAMETER	68
14.2	SCROLLING	70
<b>15</b>	<b>JOURNEY DETAIL SERVICE</b>	<b>71</b>
15.1	REQUEST PARAMETER	71
<b>16</b>	<b>RECONSTRUCTION SERVICE</b>	<b>74</b>
16.1	REQUEST PARAMETER	74
<b>17</b>	<b>TIMETABLE INFO</b>	<b>75</b>
17.1	REQUEST PARAMETER	75
<b>18</b>	<b>DATA INFORMATION (DATAINFO)</b>	<b>75</b>
18.1	REQUEST PARAMETER	75
<b>19</b>	<b>LINE INFORMATION (LINEINFO) – BETA VERSION</b>	<b>77</b>
19.1	REQUEST PARAMETER	77
<b>20</b>	<b>FEHLERCODES UND WARNUNGEN</b>	<b>78</b>
20.1	REST REQUEST-FEHLER	78
20.2	WARNUNGEN	80



## 1 ALLGEMEINES ZUM HANDBUCH

### 1.1 Änderungshistorie

Version	Änderungen	Datum	Autor
1.0	Erstellung	22.03.2019	Christian Dorner
1.3	Diverse Anpassungen für VAO	19.04.2019	Christian Dorner
1.4	Ergänzungen VAO	08.05.2019	Christian Dorner
1.5	Ergänzungen Meta-Profile	09.05.2019	Christian Dorner
1.7	Parameter polyEnc eingefügt	04.09.2019	Christian Dorner
2.0	Überarbeitung/Einfügen der Versionierung	18.09.2019	Andreas Partusch
2.1	Update von neuen Schnittstellenfunktionen gemäß Version V1.1.0, Ergänzung von Journey Detail	21.02.2020	Florian Hennerbichler Florian Hennerbichler
2.2	Formatanpassungen, Ergänzung AdressLookup, Reconstruction Service	14.10.2020	
2021-03-04	Veraltete/aufgelassene Parameter (deprecated) Trip serach: unsharp, extID; Informationen über Versionen; Update der Texte	04.03.2021	Bettina Neuhäuser
202-04-20	Korrektur totalMeta / totalCar	20.04.2021	Bettina Neuhäuser
2021-05-17	Ergänzung Meta-Profile (Tabelle in 4.4.1.3) Parksuchzeiten deaktivieren	17.05.2021	Bettina Neuhäuser
2021-06-16	groupFilter für Bike & Ride erweitert	16.06.2021	Bettina Neuhäuser
2021-06-21	Kapitel 3.4. Zustandsloser Dienst vs. Datenabhängigkeit; rtMode Parameter in Station Board services deprecated	21.06.2021	Bettina Neuhäuser
2021-09-30	Ergänzung LocationDetails service Ergänzung groupFilter Parameter Ergänzung meta Parameter	30.09.2021	Bettina Neuhäuser
2021-11-22	Timetable Info Service Beschreibung JourneyDetail Service Beschreibung ergänzt Informationen zur Versionen (Betrieb)	22.11.2021	Bettina Neuhäuser
2022-12-10	Tariff Parameter (Trip) – Default Wert	10.12.2022	Bettina Neuhäuser
2022-02-15	IFOPT Ids (altId) als Input für Parameter Id	15.02.2022	Bettina Neuhäuser
2022-05-10	Änderung im Parameter Type in location.nearbystops ab V1.4.0. Unterscheidung zwischen Stationen (S) und Stationen mit Entry Points (E) im Parameter Type	10.05.2022	Bettina Neuhäuser
2022-05-30	Überarbeitung und Ergänzung der Texte in location.name und trip	30.05.2022	Bettina Neuhäuser
2022-07-20	Data info Service, Categories Parameter im Trip Service, groupFilter in Trip ergänzt,	20.07.2022	Bettina Neuhäuser



	metaFilter in Location.nearby ergänzt, Fehlercodes ergänzt Geo-basierte Orspunktauflösung in location name		
2022-12-02	POI Filter in location.details Langversion der Wegbeschreibung aktivieren Allgemeine Hinweise zu Eckdaten und Gültigkeit	02.12.2022	Bettina Neuhäuser
2023-04-21	MetaFilter für WienMobil Auto, Paketboxen, Radservicestationen  GroupFilter für Wien Mobil Auto	22.02.2023	Bettina Neuhäuser
2023-06-30	Elektromobilitätsroutenplaner		Bettina Neuhäuser
2023-07-05	Überarbeitung der POI Filter Überarbeitung der Abfrage der Wetterinformationen	12.05.2023	Bettina Neuhäuser
2023-10-04	GroupFilter Sharetoo, Elektromonilität	04.10.2023	Bettina Neuhäuser
2023-10-04	Beschreibung der Verwendung von Echtzeitdaten und Solldaten (Kapitel 3.4 und 3.5)	18.12.2023	Bettina Neuhäuser
2023-02-20	Überarbeitung der Struktur des Handbuches Einfügen von 5.4, 5.5, 5.6, 5.7	20.02.2024	Bettina Neuhäuser
2023-03-21	Aktualisierung Kapitel 3.2 Betriebsdauer	23.03.2024	Bettina Neuhäuser
2023-05-15	Ergänzungen zur Elektromobilität	15.05.2024	Bettina Neuhäuser
2023-06-07	Korrektur Parameter Range für numF und poolld	07.06.2024	Bettina Neuhäuser
2023-07-23	Ergänzungen Meta-Filter Park&Ride, POI Kategorie „Sonstige“, Alternative Route und Ergänzungen zu Daten in Kapitel 5	23.07.2024	Bettina Neuhäuser
2024-12-09	Ergänzung Parameter „meta“ in location.name  Ergänzung Kapitel 5 (neue Inhalte 5.4, 5.5, 5.11, 5.13)  Ergänzung Kapitel 5.16	27.11.2024	Bettina Neuhäuser
2024-12-10	Kapitel Scrolling (13.2 und 14.2) im Departure Board und Arrival Board	10.12.2024	Bettina Neuhäuser
2025-02-11	Struktur überarbeitet	11.02.2025	Bettina Neuhäuser
2025-05-22	IncludeDrt Parameter	22.05.2025	Bettina Neuhäuser
2025-07-02	Update 5.9.2 Informationstexte bei Störungen	02.07.2025	Bettina Neuhäuser
2025-08-11	Line Info Service		Bettina Neuhäuser



## 1.2 Über das Handbuch

Das VAO Handbuch beschreibt die VAO ReST API mit den zur Verfügung stehenden Endpoints und Parametern, die vom VAO Routenplaner unterstützt werden. Das Dokument basiert auf der Beschreibung der Hafas ReST API der HaCon Ingenieurgesellschaft mbH und wurde um die Spezifika der Verkehrs Auskunft Österreich ergänzt.

Die Handbuch VAO ReST API gibt einen Überblick über die verfügbaren Funktionalitäten der Schnittstelle der Verkehrs Auskunft Österreich (VAO). Weiters stehen Beschreibungen der Schnittstelle in form von XSD Schema, JSON Beschreibung und Swagger Dokumentation zur Verfügung.

Das Handbuch wird laufend überarbeitet und im Zuge von Release-Information an den Kundenkreis verteilt. Das Handbuch ist ständig unter folgendem Link verfügbar:



## 2 UMFANG & DOKUMENTATION

### 2.1 Umfang

Die VAO ReST API ist ein spezielles Produkt der HaFas ReST API der HaCon Ingenieurgesellschaft mbH. Die VAO ReST API hat das Ziel, alle Funktionen zu ermöglichen, welche die VAO Routenauskunft über VAO WebApps und Apps bietet. Der Funktionsumfang und die Konfigurationsmöglichkeiten entsprechen daher mindestens der VAO WebApp. Die Weiterentwicklungen der VAO Produkte läuft parallel, sodass neue Funktionen in der WebApp auch in der ReST API nachgezogen werden (siehe Versionierung).

Es werden von VAO aber nur ausgewählte Requests und Request Parameter der HaFas ReST API verwendet. Daher werden nicht alle möglichen Requests und Parameter verwendet, welche die Standard Hafas ReST API umfasst. Die maschinenlesbare Dokumentation der VAO ReST API umfassen daher auch Parameter, die nicht von VAO unterstützt werden. Dies muss besonders bei der automatisierten Generierung von Clients auf Basis der OpenAPI Dokumentation beachtet werden.

### 2.2 Maschinenlesbare Beschreibungen

#### Wichtiger Hinweis:

Bei den maschinenlesbaren Beschreibungen der ReST API, (WADL und Swagger (OpenAPI 3.0)) sind Parameter beschrieben, die NICHT von VAO genutzt werden. Es werden nur jene Parameter unterstützt, die im VAO Handbuch beschrieben sind. Daher eignen sich OpenAPI 3.0 und WADL Dokumentation nur bedingt zur automatisierten Client-Generierung. Die Entwicklung muss immer in Hinblick auf den tatsächlichen Funktionsumfang der VAO erfolgen.

#### Die Web Application Description Language (WADL) Beschreibung

WADL ist ein XML-basiertes Dateiformat, und unterstützt die Beschreibung der Services in einer maschinenlesbaren Form und damit die Entwicklung von Anwendungen zu vereinfachen. Für jedes verfügbare Service kann online eine Beschreibung der Request-Parameter in Form einer WADL- Datei abgefragt werden. Diese sind via Übersichtsliste verfügbar, oder können wie folgt aufgerufen werden.

allgemein:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/?wadl>

spezifisch:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/<service>?wadl>

Beispiel:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/trip?wadl>

#### OpenAPI 3.0 (Swagger) Beschreibung der API

Die Swagger Dokumentation ist unter folgenden Link abrufbar

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/api-doc>

Eine interaktive Swagger-Dokumentation (Swagger UI) ist unter folgender URL abrufbar

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/swagger-ui>

#### XML-Schemadefinition (XSD Service) zur Beschreibung der Response

Der XSD-Dienst gibt die XML-Schemadefinition einer bestimmten Version zurück. Beim Aufruf des XSD-Dienstes, wird eine Liste aller verfügbaren XSD-Dateien im HTML-Format zurückgegeben.

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}//xsd>





### 3 VERSIONIERUNG UND BETRIEBSDAUER

Durch stetige Weiterentwicklung der VAO Services und somit auch der Schnittstelle können sich Parameter und Schemata von Requests und Responses über die Zeit ändern. Um Schnittstellen-Nutzern entsprechende Flexibilität und Zeit für die Weiterentwicklung zu geben, werden für den Fall solcher Änderungen von der VAO unterschiedliche Versionen der Schnittstelle geführt. Somit werden neue Versionen immer parallel zu bereits bestehenden Versionen zur Verfügung gestellt. Die VAO hält sich jedoch das Recht vor, mit zeitgerechter Ankündigung und entsprechender Vorlaufzeit ältere Versionen zu deaktivieren.

Bei Änderungen bei Parametern und interpretationsrelevanten Schemabestandteilen, werden neue Versionen parallel zu bereits bestehenden zur Verfügung gestellt, um Breaking Changes an bestehenden Versionen zu vermeiden.

#### 3.1 Versionen

Die Herausgabe von Releases ist abhängig von den Kundenerfordernissen und Weiterentwicklungen bei VAO. Grundsätzlich sind maximal 3 Versionen pro Jahr eingeplant. Dies kann aber nach den aktuellen Erfordernissen variieren.

Die Versionsnummerierung folgt dem „semantic Versioning 2.0.0“ (nach <https://semver.org>). Die Bedeutung der Nummerierung folgt dem gängigen Schema MAJOR.MINOR.PATCH. Die Klassifizierung von Änderungen wird in der VAO Guideline für die Versionsnummerierung (siehe Dokument VAO ReST API Versionierung) beschrieben.

Aufgrund von Weiterentwicklungen an den Backend-Komponenten sind Änderungen beim Schnittstellen-Schema (also in der Response) bei bestehenden Versionen nicht komplett ausgeschlossen, daher wird empfohlen die Clients tolerant gegenüber kleinen Änderungen zu entwickeln. Solche Änderungen werden mit Vorlauf angekündigt. Inhaltliche Änderungen und Anpassungen (wie ein Fahrplanwechsel oder die Erweiterung durch neue Verkehrsmittel) im Rahmen des bestehenden Schemas sind jederzeit möglich.

#### Wichtiger Hinweis:

Mit folgendem Aufruf können die aktuell verfügbaren Versionen der VAO Schnittstelle abgefragt werden:  
<https://routenplaner.verkehrsauskunft.at/vao/restproxy>

Die Endpunkte der jeweiligen Schnittstellenversionen setzen sich wie folgt zusammen:  
<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}>

Das Trip-Service der Schnittstellenversion v1.8.0 wird somit beispielsweise mit folgender URL aufgerufen:  
<https://routenplaner.verkehrsauskunft.at/vao/restproxy/v1.8.0/trip>

#### 3.2 Betriebsdauer

Der Betrieb einer Version wird für mindestens ein Jahr, der Betrieb einer LTS (Long-Term-Service) Version für 1,5 Jahre garantiert. Nach einem Jahr behält sich VAO das Recht vor, eine Version außer Betrieb zu nehmen. Der Umstieg auf die aktuelle Version, sollte daher von den API Abnehmer in die Release-Zyklen eingeplant werden. Der Fahrplan für die geplanten Releases wird spätestens bei jedem Release einer neuen Version kommuniziert, um die Planung auf Seiten der API Abnehmer zu ermöglichen.

VAO API Version	Inbetriebnahme	Außerbetriebnahme
V1.11.0	23.05.2024	31.05.2027
V1.10.0	3.12.2024	31.12.2025
V1.9.0 LTS*	5.08.2024	31.08.2026
V1.8.0	4.10.2023	31.01.2025 (verlängert)
V1.7.0 LTS*	21.04.2023	30.04.2025 (verlängert)



V1.6.0	14.12.2022	30.04.2024 (verlängert)
V1.5.0	19.07.2022	31.07.2023
v1.4.0	11.03.2022	30.06.2023 (verlängert)
v1.3.2 LTS*	21.10.2021	30.06.2023 (verlängert)
v1.3.1	28.02.2021	30.06.2022
v1.3.0*	24.11.2020	30.06.2022
v1.2.0	27.07.2020	31.12.2021
v1.1.0	11.03.2020	31.12.2021
V1.0.0	08.04.2019	31.12.2021

\* Long-term Service

#### Wichtiger Hinweis:

Die VAO behält sich das Recht vor, eine Version frühestens nach einem Jahr und ein Long-Term-Service (LTS) Version nach 1,5 Jahren nach Release-Zeitpunkt der Version zu deaktivieren. Es wird empfohlen regelmäßig auf die letzte Version der ReST API umzusteigen. Weiterentwicklungen finden nur an der letzten Version statt.

## 4 AUFRUF & AUTHENTIFIZIERUNG

### 4.1 Aufruf

Der Zugriff auf die Schnittstelle erfolgt über untenstehenden Endpunkt. Die Authentifizierung erfolgt über einen mitzugebenden Access Key. Dieser wird nach Vertragsabschluss von der VAO zur Verfügung gestellt.

Aufruf der Schnittstelle: <https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}>

Das System implementiert read-only GET Anforderungen, welche über Service URLs und GET Parameter aufgerufen werden. Diese Parameterwerte müssen in UTF-8 kodiert sein. Das Ergebnis wird entweder als XML- oder JSON- Container geliefert. Bei inkorrektur Parameterkodierung kann es zu unvorhersehbaren Ergebnissen kommen.

### 4.2 Verwendung von URI Syntax

ReST verwendet das HTTP-Protokoll für die Kommunikation. Die Aufrufe der ReST API müssen daher eine URI konforme Syntax verwenden. Beim Absetzen eines Requests übernimmt häufig der Browser automatisch die URL Codierung. In einigen Fällen kann es aber zu einer falschen Interpretation kommen, wenn reservierte Zeichen (Nach URI Syntax) enthalten sind, die aber nicht interpretiert werden sollen. Dies ist zum Beispiel bei der Nutzung von komplexen Strukturen, wie beispielsweise der `GisRefs` zu beachten, da diese aufgrund des Namens des Ortpunktes (Adresse, Haltestelle, POI) reservierte Zeichen enthalten können, die in eine HTTP konforme Schreibweise (percent-encoding) übersetzt werden müssen. Die URL-Parameter sind alle case-sensitive und verlangen somit erhöhte Vorsicht bei den Eingaben.

### 4.3 Authentifizierung

Jeder Zugriff der API muss mit einem gültigen Authentifizierungsschlüssel erfolgen. Folgender Parameter muss beim Request mitgegeben werden: `accessId=<your_key_here>`.

Der Access Key wird nach Vertragsabschluss zur Nutzung der Schnittstelle von Seiten VAO zur Verfügung gestellt.

### 4.4 Ergebnisformat

Die Schnittstelle liefert Antworten entweder im Format XML oder JSON. Um einen JSON- Output zu erhalten, muss der GET-Parameter `format=json` gesetzt werden.



Wenn XML angefordert wird, hat die Antwort den Namensraum:  
„`http://hacon.de/hafas/proxy/hafasproxy`“.

Die Bevorzugung von XML- oder JSON-Ausgaben kann über den Accept Header signalisiert werden:

- `Accept: application/json` oder `Accept: application/xml`
- wobei XML der Standard ist.

Alternativ setzt der Parameter `format` den angeforderten Typ außer Kraft, auch wenn der Accept Header gesetzt ist.

Um eine JSON-Antwort anzufordern, müssen Sie `Accept: application/json` setzen oder den folgenden Parameter an jeden Aufruf der Schnittstelle anhängen: `format=json`.

JSONP wird derzeit nur durch die Verwendung von `format=jsonp&jsonpCallback=<name_of_callback>` unterstützt. Der `jsonpCallback`-Parameter gibt den Namen des Funktions-Callbacks an, in den das json-Ergebnis verpackt wird.

Der JSON- Inhalt wird durch die Konvertierung aus XML automatisch erzeugt. Sie erfolgt nach folgenden einfachen Regeln.

- Elementnamen werden zu Objekteigenschaften
- Text (PCDATA) wird zu einer Objekteigenschaft mit dem Namen „value“  
Das Root-Element wird direkt in ein JSON-Objekt umgewandelt (wobei der Name des Root Elements weggelassen wird)  
`<root><a>foo</a></root>` becomes `{ "a": { "value" : "foo" } }`
- Verschachtelte Elemente werden zu verschachtelten Eigenschaften  
`<a><b>foo</b><c>foo</c></a>`  
wird zu  
`{ "a": { "b" : { "$" : "foo" }, "c": { "$": "foo" } } }`
- Wenn es mehrere Elemente mit demselben Namen gibt, enthält der JSON-Code ein Array für diese Elemente.  
`<root><a><b>foo1</b><b>foo2</b></a></root>`  
wird zu  
`{ "a": { "b" : [ { "value": "foo1" }, { "value": "foo2" } ] } }`
- Attributnamen werden zu Objekteigenschaften  
`<root><a atb="foo1">foo2</a></root>`  
wird zu  
`{ "a": { "atb" : "foo1", "value" : "foo2" } }`

Das folgende Beispiel zeigt die XML- Response einer Route und deren Konvertierung in JSON.

```
<Trip xmlns="hafas_rest_v1">
  <Leg name="1" type="WALK" duration="PT4M"
    dist="320">
    <Origin name="Wien Schottentor" type="ST" extId="490118400"
      routeIdx="0" time="11:11" date="2019-12-18" />
    <Destination name="Wien Börse" type="ST"
      extId="490019508" routeIdx="12" time="11:15"
      date="2019-12-18" />
  </Leg>
</Trip>
```

JSON:

```
"Trip": {
  "Leg": {
    "name": "1",
    "type": "WALK",
```



```
"duration": "PT4M",  
"dist": "320",  
"Origin": { "name": "Wien Schottentor",  
            "type": "ST", "extId": "490118400", "routeIdx": "0",  
            "time": "11:11", "date": "2019-12-18" },  
"Destination": { "name": "Wien Börse",  
                 "type": "ST", "extId": "7420483", "routeIdx": "12", "time":  
                 "11:15", "date": "2019-12-18" }  
}
```

Alle möglichen Response-Element sind im XSD-Schema beschrieben (siehe



## 5 WICHTIGE INFORMATIONEN

### 5.1 Zustandsloser Dienst und Datenaktualisierung (Gültigkeit von IDs und Refs)

Alle Dienste der bereitgestellten Schnittstelle sind zustandslos (stateless), wie es für ein ReST-Protokoll erforderlich ist. Dies hat aber seine Einschränkung bezüglich der Fahrplandaten und Adressen. Sobald die Fahrplandaten oder Adressdatensatz aktualisiert werden (Fahrplanwechsel, Adressen-Update durch das BEV), sind die IDs nicht mehr unbedingt gültig und können sich ändern.

#### Wichtiger Hinweis:

Es wird abgeraten Ortpunkt-IDs (`Id`) oder Referenz-IDs, welche für die Nutzung in anderen Services ausgegeben werden (`GisRef`, `ctxRecon`, `JourneyDetailRef`) langfristig zu speichern. Sie sollten immer in einer Session aktuell bezogen werden.

### 5.2 Koordinaten

Koordinaten werden immer in WGS84, in Dezimalwerten angegeben.

#### 5.2.1 Einschränkungen bei frei gewählten Koordinaten

Grundsätzlich ist eine Eingabe von freigewählten Koordinaten als Start und Ziel im Trip Service erlaubt. Koordinaten-Inputs werden auf den nächsten Wegabschnitt mit der entsprechenden Erlaubnis (für die gesuchte Modalität) gesnappt, wobei hier entsprechende Grenzen beim Snapping hinterlegt sind.

Wenn in einem bestimmten Umkreis kein Wegabschnitt mit entsprechender Erlaubnis für die gewählte Modalität vorhanden ist, wird keine Route gefunden. Dies trifft häufig auf Start/Zielpunkte für den PKW zu, die auf abgelegene Ortpunkte, wie Gipfel gesetzt werden, kann aber auch im Siedlungsgebiet der Fall sein, wenn die Koordinate auf „Inseln“ im Netz gesetzt werden. Inseln entstehen, wenn kein durchgängiges Netz mit entsprechender Erlaubnis zu dieser Koordinate vorhanden ist. Daher kann nicht garantiert werden, dass freigewählte Koordinaten immer zu einem Routingergebnis führen. Bei automatisierten Tests ist daher besonders zu beachten, dass Koordinaten sinnvoll gesetzt werden.

#### Wichtiger Hinweis:

Bei der Eingabe von Koordinaten muss daher auf die Position im Verkehrsnetz geachtet werden. Es wird empfohlen im Regelfall existierende Ortpunkte (Haltestellen, Adressen, POIs) anstelle von Koordinaten zu verwenden, indem eine Request-Abfolge immer mit einer `Location.name` Abfrage startet, um gültige Ortpunkt-IDs zu beziehen. Bei frei gewählten Koordinaten ist zu beachten, dass es kein unbegrenztes automatisches Snapping auf das Verkehrsnetz gibt und daher keine Garantie abgegeben werden kann, dass der Standort an das Netz angebunden werden kann. Falls eine Koordinate nicht ans Netz angebunden werden kann, antwortet die ReST API mit einer `INT_ERR`.

### 5.3 Datum- und Zeitangaben

Datumswerte werden immer im Format `JJJJ-MM-TT` angegeben. Dies gilt sowohl für Angaben in Requests als auch für Ausgaben in den Ergebnissen.

Zeitangaben werden als local time im Format `hh:mm[:ss]` im 24h-Format angegeben. Sekunden sind optional. Für ÖV Requests werden die Sekunden ignoriert. Z. B. eine Abfahrtszeitangabe von `14:37:52`, wird als `14:37:00` interpretiert.



## 5.4 Sprachen

Die unterstützten Sprachen sind von den Soll-Daten abhängig. Aktuell werden Deutsch und Englisch unterstützt. Die Sprachen können über den optionalen URL Parameter **lang=<code>** spezifiziert werden. Die Standardsprache ist Deutsch (code=de). Aufgrund der Datenlage ist neben Deutsch die Sprache Englisch (code=en) sinnvoll, allerdings sind nicht alle Dateninhalte auch in beiden Sprachen durchgehend verfügbar. In der ReSt-API hat die gewählte Sprache nur Einfluss auf die Notes (wie Verkehrsmitteltexte) Einfluss.

## 5.5 Request Verfolgung

Um eine bessere Verfolgung der Requests in den Log-Files zu ermöglichen (z.B. für Fehleranalysen) kann der Parameter requestId genutzt werden, um eine frei gewählte Kennung zu transportieren. Dieser Wert ist sowohl in den Logs vorhanden, als auch in der Response.

## 5.6 Haltestellenmodellierung und IDs

Die Haltestellen sind Teil der SOLL-Daten, die VAO von den Mobilitätsverbünden Österreich zur Verfügung gestellt werden.

### 5.6.1 Prozesskette und Aktualisierung der Solldaten

Haltestellen IDs werden im Rahmen der Zulieferung der Solldaten durch die MVO aktualisiert. Für weitere Informationen zu Daten-Updates siehe [Abschnitt 5.7.1 Prozesskette und Aktualisierung](#). Die Daten können grundsätzlich täglich aktualisiert werden, wenn ein Verkehrsverbund neue Daten publizieren möchte. Bei einem Daten-Update können sich Haltestellen IDs daher ändern. Wenn neue Haltestellen eingefügt werden oder weggelassen hat dies Auswirkung auf andere IDs in diesem Teilnetz. Daher sollen die API Nutzern die Haltestellen IDs nicht langfristig zwischenspeichern.

### 5.6.2 Haltestellen-Topologie (Metaort-Hauptmast-Mast)

Die Modellierung von Haltestellen erfolgt durch die Verkehrsverbünde auf Grundlage eines hierarchischen Haltestellenmodells. Haltestellen müssen steigscharf modelliert werden. Bei komplexeren oder größeren Haltestellen wie z.B. Bahnhöfen, erfolgt zur speziellen Definition von Fußwegs- oder Zugangsmodellierungen die Erweiterung der Modellierung um Bereiche. Bei Haltestellen, für die keine Bereiche definiert wurden, gilt die Bereichsnummer 0. Diese Bereiche, die als HoB (=Haltestelle ohne Bereich / Oer Bereiche) bezeichnet werden, erhalten keine Bereichs-ID.

Für die Interpretation der Daten in der API ist relevant, dass eine hierarchische Haltestellenmodellierung den Daten zugrunde liegt. Eine Hauptmast (Haltestelle) untergliedert sich in zugeordnete Masten (Steige). Fahrten sind nur den Masten zugeordnet. In der Response der API wird diese Hierarchie so dargestellt, dass jeder Mast die Hauptmastäquivalenz besitzt (entspricht einer partenRef). Siehe dazu [Abschnitt 5.4.1 Hauptmasten und Masten \(IDs\)](#).

Ein Spezialfall sind sogenannte Metaort-Haltestellen, wie „Wien“ oder „Linz“. Dies sind Orte, welche von den Verkehrsverbünden definiert werden, um eine Route auch ohne Angabe eines bestimmten Haltestellennamens durchführen zu können. Dies ist besonders für Nutzer ohne Ortskenntnis relevant. Metaort-Haltestellen besitzen eine Koordinate und sind mit einer oder mehreren zentralen Haltestellen verknüpft. Metaort-Haltestellen enthalten keine Fahrten. Ein Metaort-Haltestelle wird mit dem Attribut meta="true" gekennzeichnet.

```
<mainMast id="A=1@O=Hollabrunn  
Bahnhof@X=16072227@Y=48562911@U=81@L=430378000@i=A×at:43:3780@" extId="430378000"  
isMainMast="true" name="Hollabrunn Bahnhof" lon="16.072227" lat="48.562911"  
products="1219" meta="true">
```



### 5.6.3 Hauptmasten und Masten (IDs)

Im `Location.name` Service werden nur Hauptmasten ausgegeben, die anschließend im `Trip` Service als Start oder Ziel eingesetzt werden können. In einer `Trip` Abfrage von einem Hauptmast werden automatisch alle Masten einbezogen, da die Masten die Fahrten enthalten. Die Routenberechnung ist allerdings mastscharf und gibt den exakten Weg zum in der Route relevanten Mast (Steig) an. Der Mast scheint daher in der Wegbeschreibung mit eigener ID auf.

Auch im `location.nearbystops` Service werden neben den Hauptmasten auch Masten und Zugangspunkte ausgegeben. Daher enthält die `Location.nearbystops` Response auch `MastIDs`. Ein Hauptmast wird durch das Attribut `hasMainMast="true"` erkenntlich gemacht.

Ein Mast, der einem Hauptmast zugeordnet hat (also eine Hauptmast-Äquivalenz besitzt) besitzt eine „Parent“-Referenz (also einen Verweis auf die übergeordnete Haltestelle), die sogenannte `MainMastId`. Damit wird eine eindeutige Zuordnung des Mastes zum Hauptmasten ermöglicht.

Die `id` sowie die `MainMastId` sind interne IDs innerhalb der API Schnittstelle. Diese IDs werden bei jedem Datenupdate neu erzeugt und sollen daher nicht längerfristig gespeichert werden. Der korrekte Request-Fluss startet daher immer mit einer `location`-Abfrage. Die aus der `location`-Response bezogenen `ids` sollen dann im `Trip` Service oder `Stationboard` Service verwendet werden.

Weiters werden neben den internen IDs auch die globale ID ausgegeben, die bei den Verkehrsverbünden übergreifend in Österreich verwendet werden. Die sogenannte `altId` enthält die externe ID der Verbünde.

### 5.6.4 Haltestellen IDs im Request

Die Rest-API unterstützt mehrere ID Spaces / globale IDs für einen Datensatz.

#### id (interne ID)

Sie ist eine API interne ID. Es wird empfohlen die `id` der Haltestelle für die Anfrage der Services zu verwenden. Die `id` ist eine komplexe Struktur, die neben der eigentlichen `id` der Haltestelle auch eine Versionierung der `id` enthält., welche die Haltestelle über die Datenupdates hinweg eindeutig macht. Die `id` ist die empfohlene ID zur Nutzung in allen Requests. Sie beinhaltet Informationen zur Abfahrthaltestelle, die Koordinate, die interne ID und die externe, globale ID.

Beispiel für `Id` der Haltestelle Westbahnhof:

```
id="A=1@O=Wien Westbahnhof@X=16337652@Y=48196654@U=81@L=490146800@B=1@p=1641949238@i=A  
xat:49:146800"
```

#### extId (interne ID)

Die `extId` entspricht der internen `id`, allerdings ohne Versionierung. Sie kann in den Services genutzt werden, allerdings wird empfohlen die `id` einzusetzen. Sie ist ebenfalls eine interne API ID.

Beispiel für die externe `Id` (`extId`) der Haltestelle Westbahnhof:

```
extId="490146800"
```

Die externe `Id` ist eine 9-stellige Id. Für die Unterscheidung zwischen Haltestelle (Hauptmast) und Steigen, sind die letzten beiden Stellen (Stelle 8 und 9), also die Endung der ID, relevant.

- Endung 00 = Hauptmast der Haltestelle
- Endung 01 bis 59 = Steige der Haltestelle

#### altId (globale ID)

Zusätzlich zur `id` ist es auch möglich die globale `Id` (`altId`) im Request zu verwenden. Die persistente Vergabe der globalen ID erfolgt in Österreich nach dem IFOPT Standard vergeben und wird durch den zuständigen Verkehrsverbund vergeben.

Beispiel für alternative ID (`altId`) der Haltestelle Westbahnhof:

```
<altId>at:49:1468</altId>
```

## 5.7 Fahrplandaten (SOLL-Daten)

Die Mobilitätsverbünde Österreich betreiben die österreichweite Datensysteme für den Öffentlichen Verkehr zur Sammlung, Integration und Verteilung von Sollfahrplänen, Echtzeitdaten und Störungsinformationen. Die verarbeiteten Daten beinhalten den gesamten Öffentlichen Verkehr in Österreich und werden von Partnern wie den österreichischen Verkehrsverbünden, der ÖBB-Personenverkehr AG oder Städtischen Verkehrsunternehmen zugeliefert und der VAO zur Verarbeitung bereit gestellt.

### 5.7.1 Prozesskette und Aktualisierung der Solldaten

Nach Bereitstellung der Sollfahrplandaten durch die Lieferanten erfolgt eine stufenweise Verarbeitung der Inhalte. Am Ende des Prozesses findet eine Verteilung der Rohdaten an abnehmende VAO Routenplaner statt. Die Solldaten (Fahrplandaten inklusive Haltestellen) werden im übergreifenden, zentralen Datensammelsystem (DASS) der Mobilitätsverbünde Österreich OG gesammelt und VAO in einem konsolidierten und 2 mal täglich aktualisierten Datenpool bereitgestellt. Die Daten können 2 mal Tag aktualisiert werden. Die Daten der VAO werden täglich über Nacht mit neuen Daten versorgt.

### 5.7.2 Gültigkeit der Fahrplandaten und Fahrplanwechsel

Fahrplandaten (Solldaten) sind grundsätzlich 1 Jahr beginnend mit Dezember gültig. Die Eckdaten bestimmen den Beginn und Ende der Fahrplanperiode und somit die Abfragemöglichkeiten in die Zukunft und Vergangenheit.

Der Fahrplanwechsel erfolgt mit einer Übergangsphase und beginnt im November, um Abfragen über Dezember hinaus zu ermöglichen. Hier werden für 3 Monate im Voraus die neuen Fahrpläne eingebettet. Nach dem Ende der Fahrplanperiode, werden die neuen Fahrpläne gänzlich für das ganze Jahr eingestellt.

### 5.7.3 Abfragen in die Zukunft und Vergangenheit

An der Schnittstelle können technisch grundsätzlich Routen bis zu 4 Monate im Voraus und 62 Tage in die Vergangenheit abgefragt werden, ohne dass es zu der Fehlermeldung käme, dass die Abfrage außerhalb der Fahrplanperiode (bzw. außerhalb der Fahrplan-Eckdaten) liegen. Die definierten Eckdaten der Fahrplanperiode können im `Timetable Info` Service abgerufen werden.

**ABER**, neben dem Grundfahrplan und dem regulären Fahrplanwechsel, gibt es unterjährige Änderungen in den Fahrplänen (Sommerfahrpläne, Winterfahrpläne, kleiner Fahrplanwechsel ÖBB, sowie laufende kleine Anpassungen), die tagesaktuell ins Auskunftssystem eingespielt werden. Daher kann nicht garantiert werden, dass eine Route, die man im Voraus abgefragt hat, dann tatsächlich so besteht, da sich zwischenzeitlich Änderungen im Netz ergeben haben.

Grundsätzlich gelten bei Abfragen in die Vergangenheit die gleichen Rahmenbedingungen: technisch ist es möglich bis 62 Tage in die Vergangenheit abzufragen. Da aber die Verkehrsverbünde, anders als die ÖBB, die Datensets nicht historisiert, können sich Änderungen in den Verbindungen ergeben. Das bedeutet, bei einer Abfrage in die Vergangenheit kann es sein, dass eine Verbindung ausgegeben wird, die es tatsächlich so an dem Datum in der Vergangenheit nicht gab, weil der aktuelle (und nicht der damalige) Fahrplan verwendet wird. Das ÖBB Netz wird historisiert und damit sind die Zugverbindungen auch in die Vergangenheit korrekt. Diese Art der Fahrplanänderungen sind meist minimale Änderungen und die Chance, dass eine Verbindung sich grundlegend ändert, ist gering, dennoch besteht diese Möglichkeit.

Die Empfehlung ist daher maximal 6 bis 8 Wochen im Voraus oder in die Vergangenheit Routen abzufragen, oder bewusst mit dem Umstand umzugehen, dass keine Garantie abgegeben werden kann, dass die Route exakt so stattfindet oder stattgefunden hat. Nutzt man das `Reconstruction-Service` zur Rekonstruktion einer Fahrt, so kann es Einzelfälle geben, wo durch eine Änderung in einer Fahrplandaten, die Fahrt nicht





rekonstruierbar ist, obwohl sie innerhalb der Fahrplan-Eckdaten liegt. In einem Zeitfenster von 6 bis 8 Wochen ist die Chance sehr gering, dass Änderungen stattfinden.

#### Wichtiger Hinweis:

Bei der Nutzung des `Reconstructions-Service` ist zu beachten, dass Rekonstruktionskontexte nicht langfristig gespeichert werden sollen, um Fahrten zu rekonstruieren. Diese dienen zur sofortigen Rekonstruktion, z.B. für die Übergabe einer Route in eine andere Applikation, oder zum Teilen der Route.

Die Empfehlung ist maximal 6 bis 8 Wochen im Voraus oder in die Vergangenheit Routen abzufragen, oder bewusst mit dem Umstand umzugehen, dass keine Garantie abgegeben werden kann, dass die Route exakt so stattfindet oder stattgefunden hat.

## 5.8 ÖV Produktklassen (Verkehrsmittel-Kategorien)

Aktuell verwendete Produktklassen (Verkehrsmittel-Kategorien) und die dazu gehörigen Produkte (Verkehrsmittel) können im `datainfo` Service abgerufen werden.

### 5.8.1 Filterung von Produktklassen

Bei Requests wie `Trip`, `Departure-/Arrivalboard` oder `Location` Requests können die zu berücksichtigenden ÖV Produktklassen über den Parameter `products` gefiltert werden. Diese Produktklassen werden dabei als Bit-Maske ( $2^{\text{Produktklasse}}$ ) eingegeben. Beispiel `cls=64` entspricht  $2^6$  und damit Produktklasse 6, also Regionalbus.

Bei der Kombinationen von Produktklassen ist die Summe zu bilden. Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bit-Maske die Bits 2 und 4 gesetzt. Dadurch ergibt sich:  $2^2 + 2^4 = 20$

Zu beachten ist, dass die Filterung der Fahrten in `Departure-/Arrivalboard` mit der Produktklasse 10 (Anrufsammeltaxi/Rufbus AST) nicht wirksam ist, da bedarfsorientierte Angebote keinen Fahrplan haben und daher diese Produktklasse nicht sinnvoll bei den Abfahrten- und Ankünften dargestellt werden kann. Fragt man nur mit dem AST Produktfilter an wird dieser ignoriert. Als Ergebnis werden alle Fahrten (ÖV und AST) angeboten.

Belegung der Bit-Maske:

Bit	Produktklasse
0	Zug
1	S-Bahn
2	U-Bahn
3	Stadtbahn
4	Straßenbahn
5	Schnellbus
6	Regionalbus
7	Stadtbus
8	Seil-/Zahnradbahn
9	Schiff
10	Anrufsammeltaxi/Rufbus (AST)
11	Sonstige
12	Autoreisezug
13-15	Nicht in Verwendung

### 5.8.2 Interpretation der Produktklasse in der Response

Diese Produktklassen werden dabei als Bitmaske (2<sup>Produktklasse</sup>) verarbeitet und als Dezimalwert in der Response im Attribut `cls` ausgegeben.

#### Auswertung des Produktnamens und Produktklasse:

Aktuell gibt es 12 ÖV Produktklassen (siehe obige Liste), welche in der Verarbeitungskette berücksichtigt werden, allerdings müssen nicht zwingend alle Produktklassen in Verwendung bei den Verkehrsverbünden sein. Zur Auswertung eignet sich das Attribut `name`. Dieses enthält bei der Produktklasse Zug das Gattungskürzel und die Zugnummer. Bei anderen Produktklassen enthält `name` Umsteigeverkehrsmittel oder Verkehrsmitteltext (z.B. „Stadtbus“). Um die Produktklasse auszuwerten (Art des Verkehrsmittels) eignet sich das Attribut `catOutL` (z.B. Regionalbus, Bus, oder Railjet) welches den Verkehrsmitteltext „Name“ enthält.

#### Zur Auswertung des Attributs `line`:

Das Attribut `lines` kann bei Produktklasse 1 bis 11 (alles nur nicht die Produktklasse Zug und Autoreisezug) ausgewertet werden. Dieses entspricht der veröffentlichten Liniennummer (EFA Linienbezeichnung). Bei der Produktklasse Zug und Autoreisezug kann dieses Attribut nicht genutzt werden, da hier keine Liniennummer und ID in den Grunddaten gepflegt ist. In Ausnahmefällen, wie z.B. der Pinzgauer Lokalbahn, können auch bei Produktklasse Zug Liniennummern gepflegt sein, dies ist aber nicht der Regelfall und grundsätzlich wird in den Grunddaten der Linienplatzhalter „xxx“ verwendet, der nicht für die Anzeige auf Displays geeignet ist und daher entfällt das Attribut `line` in der API, wenn keine Liniennummer vorhanden ist. Für die Anzeige von Gattung und Zugnummer auf einem Display, falls das Attribut `line` nicht vorhanden ist, kann das Attribut `catOutL` und ggf. das Attribut `num` verarbeitet werden.

In der Response werden Produktklassen mit folgenden Attributen beschrieben (Beispiel Railjet 649):

```
<Product
name="RJ 649" (Gattungskürzel und Zugnummer oder)
displayNumber="649" (Linienbezeichnung)
num="649" (Zugnummer)
line="xxx" (Liniennummer, dieses Attribut ist bei Zügen nicht vorhanden)
lineId="obb-10-20B-X-j21-1"
(Teilnetz+Betriebszweig+Speichernamen+Ergänzung+Projekt+Linienversion)
catOut="RJ" (Zuggattungen "Kurzbezeichnung, nur bei der Gattung Zug)
catIn="RJ" (Zuggattungen "Kurzbezeichnung oder
Teilnetz-kürzel+Verkehrsmitteltext "Nummer")
catCode="0"
cls="1" (Produktklasse als 2Produktklasse)
catOutS="RJ" (Zuggattungskürzel, oder Teilnetz-kürzel+Verkehrsmitteltext
"Nummer")
catOutL="Railjet" (Verkehrsmitteltext „Name“- nicht persistent)
operatorCode="01" (Unternehmer Code im DIVA)
operator="OEBB" (Unternehmer „Name“)
admin="001AA_" (Teilnetz + Unternehmer Code + Kurzbezeichnung Zweigstelle (6-
stellig)
matchId="xxx"> (EFA Linienbezeichnung)
</Product>
```

(In dem oben angeführten Beispiel sind Verkehrsmitteltext und Produktklasse ident, aber das ist nicht der Regelfall, sondern nur bei der Gattung Zug).

#### Beispiel der Response für Liniennummer gepflegt und vorhanden:

```
<Product name="Regionalbus 645" displayNumber="645" num="10294" line="645" lineId="vor-90-645-j21-
```



```
1" catIn="v14" catCode="6" cls="64" catOutS="v14" catOutL="Regionalbus" operatorCode="
12" operator="Österreichische Postbus Aktiengesellschaft" admin="v12Pos" matchId="6451
42">
    <icon res="prod_ic_bus" txtS="645">
        <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
        <backgroundcolor r="0" g="121" b="58" hex="#00793A"/>
    </icon>
</Product>
```

Ist die Liniennummer nicht gepflegt, muss das Attribut `catOutS` (Gattungsname, Kurzform) und `catOutL` (Gattungsname, Langform) ggf. die Zugnummer (`num`) verwendet werden, um eine entsprechende Bezeichnung (z.B. RJX 760) zu ermöglichen. Gattungsnamen sind z.B. REX, ARZ, RJX etc. Das Attribut `name` setzt sich ebenfalls immer zusammen aus der Gattung/Kategorie `catOutS` und der `displayNumber`.

#### Beispiele für ÖV Produktklassen ohne Liniennummer in der Response:

```
<Product name="RJX 760" displayNumber="760" num="760" lineId="obb-10-A3-j20-1"
catOut="RJX" catIn="RJX" catCode="0" cls="1" catOutS="RJX" catOutL="Railjet Express"
operatorCode="01" operator="OEBB" admin="001AA_" matchId="xxx">
    <icon res="prod_ic" txtS="RJX">
<Product name="EN 40462" displayNumber="40462" num="40462" lineId="obb-12-A3-j20-1"
catOut="EN" catIn="EN" catCode="0" cls="1" catOutS="EN"
catOutL="EuroNight" operatorCode="01" operator="OEBB" admin="001AA_" matchId="xxx">
```

## 5.9 Echtzeitinformationen im ÖV (IST-DATEN)

Die Verkehrsunternehmen übermitteln aktuelle Daten über das tatsächliche Verkehrsgeschehen, vor allem über Abweichungen (Verspätungen, Gleisänderungen etc.) an den zentralen Sammelpool der Mobilitätsverbünde Österreich OG. Der HAFAS Information Manager (HIM) der Mobilitätsverbünde Österreich OG ist die zentrale Plattform für die österreichweit flächendeckende Störungskommunikation in Richtung Verkehrsauskunft Österreich (VAO). Österreichweit gesammelte Echtzeitdaten (Fahrplanabweichungen) werden laufend im Auskunftssystem berücksichtigt und die Fahrten aktualisiert, sowie Störungen übermittelt.

Die Berechnung der Verbindung erfolgt auf Basis der Fahrplandaten (SOLL-Daten), aber unter Berücksichtigung der Echtzeitdaten. Solange die Verbindung fahrbar ist, wird diese auf Basis der Fahrplandaten berechnet und die Abweichungen durch die Echtzeitdaten (wie Verspätung oder Gleiswechsel) (Echtzeitdaten) parallel zu den Soll-Daten ausgegeben (siehe „rt“-Attribute). Ist eine Verdingung allerdings aufgrund der Echtzeit nicht möglich, wird die Verbindung auf Basis der Echtzeitdaten berechnet und ausgegeben. Im Detail erfolgt die ÖV Verbindungssuche in folgenden Schritten:

- 1. Schritt:** Verbindungssuche/-berechnung auf Basis der Plandaten (SOLL-Daten)
- 2. Schritt:** Echtzeiten werden in die Berechnung dazu gestellt (berücksichtigt)
- 3. Schritt:** Sollte die Echtzeitberechnung ergeben, dass eine Verbindung gebrochen wird (ein Umstieg kann nicht rechtzeitig erreicht werden), erfolgt die Berechnung der Route auf Basis der Echtzeitdaten (IST-Daten).

### 5.9.1 Fahrplanabweichungen

Sofern vorhanden, werden in den angebotenen Services Echtzeitinformationen (zusätzlich zu den Fahrplandaten) ausgegeben. Echtzeitdaten werden in der Verbindungssuche (`trip`), Liniendetails (`journey details`) sowie in den Abfahrts- und Ankunftsübersichten (`station board`) verwendet.

**Fahrplanabweichungen:** In der ÖV-Verbindungssuche (`trip`) werden standardmäßig (per Default) Echtzeitinformationen (z.B. Verspätungen oder Gleiswechsel) einbezogen. Die aktualisierte Ankunfts-, Abfahrtszeit wird in eigenen Feldern ausgegeben. Die Echtzeitinformation ist mit dem Präfix „rt“ gekennzeichnet.



„rt“-Attribut	Echtzeitinformation	Service
rtDate	Datum der Fahrt	Trip, station board
rtTime	Uhrzeit der Fahrt	Trip, station board
rtTrack	Gleis	Trip, station board
rtDepTime	Abfahrtszeit	joureny details
rtDepDate	Abfahrtsdatum	joureny details
rtArrTime	Ankunftszeit	joureny details
rtArrDate	Ankunftsdatum	joureny details
rtArrTrack	Ankunftsgleis	joureny details
rtDepTrack	Abfahrtsgleis	joureny details

**Verbindungsausfälle:** Verbindungen, die komplett ausfallen, werden ausgegeben und mit dem Attribut `cancelled=true` gekennzeichnet. Dieses Attribut wird nur im Fall eines Ausfalles ausgegeben. Es ist im Element `<Trip><Origin>` also in der Verbindungsübersicht, sowie im Bereich `<LegList><Leg>` also in den Verbindungsdetails (mit allen Abschnitten) ausgegeben. Solche Fahrten können als „gestrichen“ angezeigt werden. Zusätzlich haben solche Fahrten auch Informationstexte in sogenannten `<Notes>` (siehe folgenden Abschnitt 5.7.2).

### 5.9.2 Informationstexte bei Störungen

Störungen werden in Form von Störungsinfortexten in Form von Infotexten ausgegeben. In der Response sind Infotexte im Bereich `<Notes>` erkennbar. Infotexte sind mit den Attributen `key` und `type` kategorisiert. Konkret werden die Infotexte immer am jeweiligen Root-Element des Services angehängt (für `trip service` im Root-Element `<trip>`, im `station board` z.B. im Root-Element `<departure>`).

Die Infotexte mit dem Type `type=R` (Realtime) enthalten im Attribut `txtN` den Informationstext, der aufgrund der Echtzeitdaten generiert wird. Dies sind Standardtexte, die bestimmte Fälle beschreiben. In einigen Fällen wird im `txtN` auch angegeben welches Verkehrsmittel (Produkt) betroffen ist.

Die Infotexte mit dem Type `type=A` (Attribut) dienen hauptsächlich der Übermittlung von Fahrzeugausstattung und anderen Attributen (keine Echtzeitdaten). Dieser Typ wird auch dazu verwendet um standardisierte Informationstexte zur Ursache der Störungen (die über die Echtzeitdaten übermittelt werden) auszugeben. Hier wird ein zweistelliger Attributcode im Attribut `key` und in `txtN` der standardisierte Ursachentext für die Störung ausgegeben.

Note type	Note key	txtN
Realtime (type=R)	text.realtime.connection. cancelled	Verbindung nicht möglich
	text.realtime.connection. attribute.change	Bei Ihrer Verbindung hat sich die Fahrzeugausstattung geändert
	text.realtime.joureny. cancelled	{PRODUCT_NAME}: Fällt aus
	text.realtime.journey.partially. cancelled.between	{PRODUCT_NAME}: Fahrt fällt aus zwischen {STOP_NAME} und {STOP_NAME} Bitte benutzen Sie eine andere Verbindung.
	text.realtime.journey.partially. cancelled.between.in	{PRODUCT_NAME}: Fahrt fällt aus an {STOP_NAME} Bitte benutzen Sie eine andere Verbindung.
	text.realtime.journey.partially.cancelled.unspecified	Ausfall zwischen Haltestellen



text.realtime.connection.critical	/
text.realtime.journey.missed.connection	Der Anschluss kann voraussichtlich nicht erreicht werden.
text.realtime.journey.additional.service	Zusatzfahrt
text.realtime.journey.special.service	Sonderfahrt
text.realtime.journey.attribute.change	Bei Ihrer Verbindung hat sich die Zugausstattung geändert.
text.realtime.connection.cancelled	Verbindung fällt aus
text.realtime.connection.brokentrip	Der Anschluss kann voraussichtlich nicht erreicht werden.
text.realtime.connection.attribute.change	Verbindung mit geänderter Ausstattung
text.realtime.connection.not.ridable	Verbindung gemäß aktueller Echtzeitinformationen nicht fahrbar
text.realtime.connection.alternative.summary	Fahrtmöglichkeit gemäß aktueller Verkehrslage
text.realtime.connection.alternative.detail	Fahrtmöglichkeit gemäß aktueller Verkehrslage. Bitte beachten Sie die aktuellen Echtzeitinformationen.
text.realtime.connection.nodelayinfo	Bei fehlenden Prognosen liegen zu der betroffenen Fahrt keine Echtzeitinformationen vor.
text.realtime.connection.delayed	Bei dieser Verbindung erwarten wir Verspätungen. Nach aktueller Lage sollte die Verbindung jedoch möglich sein. Bitte beachten Sie die Information am Bahnhof/im Zug.
text.realtime.connection.notdelayed	Verbindung pünktlich erwartet.
text.realtime.connection.platform.change	{PRODUCT_NAME} fährt abweichend von {FROM_STOP_NAME} ab {PLATFORM_TYPE_STRING} \${PLATFORM}
text.realtime.stop.cancelled	Halt entfällt
text.realtime.stop.additional	Zusatzhalt
text.realtime.stop.entry.disabled	Hält nur zum Aussteigen
text.realtime.stop.exit.disabled	Hält nur zum Einsteigen
text.realtime.stop.entry.exit.disabled	Kein Halt zum Ein- oder Aussteigen
text.realtime.stop.scheduled.dep.arr.time.changed	Achtung: Änderung der Fahrplanzeiten möglich
text.realtime.stop.scheduled.dep.time.changed	Achtung: Änderung der Fahrplanzeiten möglich
text.realtime.stop.scheduled.arr.time.changed	Achtung: Änderung der Fahrplanzeiten möglich



	text.realtime.stop.platformChange	Ankunft und Abfahrt heute an {PLATFORM_TYPE_STRING} {PLATFORM}
	text.realtime.stop.arr.platformChange	Ankunft heute an {PLATFORM_TYPE_STRING} {PLATFORM}
	text.realtime.stop.dep.platformChange	Abfahrt heute von {PLATFORM_TYPE_STRING} {PLATFORM}
	text.realtime.walk.cancelled	Fußweg nicht möglich
	text.realtime.walk.out.of.order	Außer Betrieb
	text.realtime.line.statusicon.norealtime	keine Echtzeitinformationen
	text.realtime.line.statusicon.nodelay	pünktlich
	text.realtime.line.statusicon.somedelay	einige Verspätungen
	text.realtime.line.statusicon.muchdelay	viele Verspätungen
	text.realtime.connection.alternative.summary"	Alternativfahrt
	text.realtime.connection.alternative.detail	Alternativfahrt. Bitte beachten Sie die aktuellen Echtzeitinformationen
Attribut (type=A)	zweistelliger Attributcode [01]	Verkehrsüberlastung: Echtzeitprognose ungenau
	zweistelliger Attributcode [02]	Technische Störung: Echtzeitprognose ungenau
	zweistelliger Attributcode [03]	Witterung: Echtzeitprognose ungenau
	zweistelliger Attributcode [04]	Polizeieinsatz: Echtzeitprognose ungenau
	zweistelliger Attributcode [05]	Verspätung wegen erhöhtem Fahrgastaufkommen
	zweistelliger Attributcode [06]	(Ausfall) Fahrtausfall
	bc	Freitext
Information (type=I)	DI	Ausfall

### 5.9.1 ÖV Störungsmeldungen (IMS Meldungen)

Die österreichweit gesammelten ÖV Störungsmeldungen (Incident Management System) werden in den Services `Trip`, `StationBoard` und `JourneyDetails` ausgegeben.

Meldungen sind als eigenes `<Messages>` Element in der `Trip-`, `StationBoard-Response` angehängt und enthalten neben der eigentlichen Meldungsüberschrift im Attribut `head`, und den Meldungstext im Attribut `text`, weitere Informationen zur Meldung. Im Attribut `products` wird die betroffene Verkehrsmittelkategorie angegeben. Das Attribut `company` enthält den Verweis auf den Urheber bzw. Einmelder der Meldung. Die Attribute `sTime` (Beginn-Zeit), `sDate` (Beginn-Datum), `eTime` (End-Zeit), `eDate` (End-Datum) geben Zeitpunkt und Gültigkeit der Meldung an.

Das Attribut `category` gibt die sogenannte „HIM“-Meldungskategorie an. In der API werden die IMS-Meldungen als sogenannte HIM-Meldungen in der HIM-Kategorisierung ausgegeben (siehe Tabelle).

Category	Bedeutung
0	Ersatzverkehr
1	Baustelle
2	Ausfall
4	Information



5	Notfall
6	Endmeldung
7	Verladezeit
9	Vorankündigung

Ein entsprechendes API Service zur Auflistung und Filterung aller Meldungen ist über die Mobilitätsverbünde Österreich (MVO) auf Anfrage zu beziehen. Die MeldungsID (messageID) ist in der VAO ReST API und der MVO IMS API übergreifend und daher konsistent.

Im Attribut `affectedStops` werden die betroffenen Haltestellen gelistet. Im Attribut `DailyStartingDate` und `DailyDuration` wird die tägliche Beginnzeit und die tägliche Gültigkeitsdauer ausgegeben.

## 5.10 Echtzeitinformationen im IV

### 5.10.1 Verkehrslage (Dynamiken)

Die Berücksichtigung der Verkehrslage (Dynamiken) in die Berechnung einer Route wird **standardmäßig einbezogen (default=aktiv)**. Möchte man diese nicht berücksichtigen, muss sie per Parameter deaktiviert werden. Auch Parksuchzeiten sind standardmäßig enthalten. Siehe dazu [Abschnitt 11.3 Meta-Profil für IV Routing](#).

### 5.10.2 Verkehrsmeldungen im IV

Verkehrsmeldungen werden **standardmäßig nicht ausgegeben (default=inaktiv)**, da diese den Umfang der Response sehr vergrößern. Daher muss man die Verkehrsmeldungen per Parameter anschalten (`totalMeta=event`), damit sie ausgegeben werden. Siehe dazu [Abschnitt 11.3 Meta-Profil für IV Routing](#).

Dies gilt auch für Kurzparkzoneninformationen und Mautpflicht. Die Ausgabe erfolgt im betroffenen Routenabschnitt (`Leg/GisRef/GisRoute`) als Meldung (`Note`) mit dem der Kennzeichnung (`key`) `EVENTS`, `PARKINZONE` oder `TOLL_HIGHWAY` (z.B. `Note key="EVENTS"`). Der Inhalt (`txtN`) der Nachricht enthält eine XML1.0 UTF8. Bei Verkehrsmeldungen wird für die Visualisierung des betroffenen Streckenabschnittes bereitgestellte Polyline eine GML Nachricht mit der Koordinatenabfolge für eine gekurvte Linie ausgegeben. Weiters wird wir auch die Art der Meldung (`Message type`) ausgegeben.

#### Auszug aus der Response (auswertbare Informationen)

Polylinie mit Koordinaten: `MultiCurve & posList`

VAO Message ID: `vao:messageID`; z.B. `RW_S_2197383_1`

Message type: `vao:messageType`; z.B. `Roadworks`

SituationID: `vao:situationID`; z.B. `RW_S_2197383`

Name der Meldung: `vao:name_de`; z.B. `A1 West Autobahn: St.Pölten Richtung Wien`

Beschreibung: `vao:description_de`; z.B. `In Fahrtrichtung von Walserberg nach`

`Wien zwischen Anschlussstelle St.Pölten Süd und Anschlussstelle St.Pölten`

`veränderte Verkehrsführung im Baustellenbereich, 17.05.2021 05:00 - 27.08.2021`

`20:00, Generalsanierung.`

#### Auszug aus der Response

```
<Note key="PARKINGZONE" type="A" txtN="<?xml version="1.0" encoding="UTF-8"?><wfs:FeatureCollection xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:vao="vao" xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" numberMatched="1" numberReturned="1" timeStamp="2021-02-16T11:32:19.848Z"
xsi:schemaLocation="http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0/wfs.xsd vao
```





```
http://10.1.32.199:80/geoserver/vao/wfs?service=WFS&version=2.0.0&request=DescribeFeatureType&typeName=vao%3Apoi_original http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd"><wfs:member><vao:poi_original
gml:id="poi_original.fid--
176e0ff5_177aa84822c_2415"><vao:id>417636863</vao:id><vao:ownerID>oeamtc</vao:ownerID><vao:category>ParkraumbewirtschaftungStraße</vao:category><vao:name_de>Gebührenpflichtige
Blaue Kurzparkzone Graz Europaplatz-Bahnhofvorplatz</vao:name_de><vao:name_en>blue short
time parking zone Graz Europaplatz-
Bahnhofvorplatz</vao:name_en><vao:searchable>false</vao:searchable><vao:routable>false</vao:routable><vao:link_url>http://www.graztourismus.at/de/anreise-und-
verkehr/parken/kurzparkzonen</vao:link_url></vao:poi_original></wfs:member></wfs:FeatureCollection>"><?xml version="1.0" encoding="UTF-8"?><wfs:FeatureCollection
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:vao="vao"
xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" numberMatched="1"
numberReturned="1" timeStamp="2021-02-16T11:32:19.848Z"
xsi:schemaLocation="http://www.opengis.net/wfs/2.0
http://schemas.opengis.net/wfs/2.0/wfs.xsd vao
http://10.1.32.199:80/geoserver/vao/wfs?service=WFS&version=2.0.0&request=DescribeFeatureType&typeName=vao%3Apoi_original http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd"><wfs:member><vao:poi_original
gml:id="poi_original.fid--
176e0ff5_177aa84822c_2415"><vao:id>417636863</vao:id><vao:ownerID>oeamtc</vao:ownerID><vao:category>ParkraumbewirtschaftungStraße</vao:category><vao:name_de>Gebührenpflichtige
Blaue Kurzparkzone Graz Europaplatz-Bahnhofvorplatz</vao:name_de><vao:name_en>blue short
time parking zone Graz Europaplatz-
Bahnhofvorplatz</vao:name_en><vao:searchable>false</vao:searchable><vao:routable>false</vao:routable><vao:link_url>http://www.graztourismus.at/de/anreise-und-
verkehr/parken/kurzparkzonen</vao:link_url></vao:poi_original></wfs:member></wfs:FeatureCollection></Note>
```

## 5.11 Mautinformationen

Bei IV Routen werden den einzelnen Streckenabschnitten die Mautinformationen beigelegt, die in der GIP (siehe Abschnitt 5.10) eingetragen sind. Diese Informationen können genutzt werden um Routenabschnitte mit/ohne Vignette bzw. Sondermaut entsprechend auszuwerten. In den einzelnen Abschnitten einer IV Route (Element <GisRoute><Seg> wird ein Hinweis <Note> beigelegt mit den folgenden Keys. Der Key muss zusammen mit dem Feld txtN ausgewertet werden.

Erläuterungen zur Auswertung der Mautinformation im <GisRoute><Seg><Note>		
key	txtN	Bedeutung
Autobahn (tollhighw)		
tollhighw	-1	Vignettenpflicht
tollhighw	0	Vignettenbefreiung
tollhighw	1	Sondermaut
tollhighw	2	keine Maut (=niederrangiges Netz)
Niederrangiges Netz (z.B. Passstraßen) (tollother)		
tollother	-1	Keine Maut
tollother	1	Mautstraße am Straßennetz außerhalb von ASFINAG Netzen
Mautinfos für LKW (tolltruck)		
tolltruck	1	Sondermaut
tolltruck	2	Standard streckenabhängiger Maut
tolltruck	-1	keine LKW Maut





## 5.12 Ortspunkte und Verkehrsnetz

Ortspunkte in der VAO sind jegliche geografischen Orte, wie Stationen/Haltestellen, Adressen, Points of Interest (kurz POIs), die in der Routenabfrage verwendet werden können. Haltestellen werden von den Verkehrsverbünden zugeliefert. Für Adressen werden die gültigen Adressen des Bundesamt für Eich- und Vermessungswesen eingebettet. POIs stammen aus unterschiedlichen, qualitätsgeprüften Quellen wie beispielsweise Asfinag, ÖBB Infrastruktur, ÖAMTC, Österreich Werbung, E-Control. Das zugrundeliegende Wegenetz auf dem geroutet wird ist die österreichische Graphenintegrationsplattform (GIP), die permanent weiterentwickelt wird.

Die Konsistenz der Daten ist daher grundsätzlich gegeben. Im Regelfall sind die Ortspunkte an das Wegenetz angebunden. Eine Anbindung an das Netz kann aber nicht für alle Modalitäten garantiert werden. Nicht jeder Ortspunkt ist mit jeder Modalität (siehe dazu `groupFilter` im [Abschnitt 11.2](#)) erreichbar. Ein Berggipfel ist beispielsweise oft nicht mit der Modalität PKW erreichbar, sondern nur zu Fuß.

### Wichtiger Hinweis:

Es ist vor allem bei automatisierten Tests zu beachten, dass eine Trip-Abfrage mit gültigen Ortspunkten auch zu einer Fehlermeldung führen kann, wenn nicht die Modalität gewählt wurde, mit welcher dieser Ort erreichbar ist. Da besonders auch Wert auf die Fahrerlaubnisse beim Routing gelegt wird, kann es zu Unterschieden mit anderen Routenplanern kommen, die diese Erlaubnisse oder Verbote nicht beachten. Eine Fehlermeldung eines Trip Requests (es kann keine Route gefunden werden), kann daher fachlich korrekt sein.

Da das zugrundeliegende Netz permanent gewartet und verändert wird und auch Ortspunkte zugeliefert werden, können Fehler bei der Erfassung nicht ausgeschlossen werden, sodass eine Netzanbindung und damit ein Routing auch fehlschlagen kann. Fehler im Netz werden dem GIP Betrieb gemeldet und müssen bei den Erfassern behoben werden. Der 2-Monatige Update-Zyklus der GIP bestimmt die Geschwindigkeit der Fehlerbehebung. VAO ist nicht ermächtigt GIP-Fehler selbst zu korrigieren.

## 5.13 Einschränkungen bei Routen im Ausland

Wie unter [Abschnitt 5.09](#) beschrieben umfasst die VAO die Daten vollumfänglich für Österreich. Routen sind aber – mit bestimmten Einschränkungen – im Ausland möglich. Dies ist vor allem in den Grenzregionen wichtig. Dabei sind aber folgende Einschränkungen zu beachten:

Es steht **kein intermodales Routing im Ausland** zur Verfügung, sondern nur ein **reines PKW** und **reines ÖV Routing** (siehe dazu die relevanten `groupFilter`). Im ÖV können daher auch keine Vor- und Nachläufe von oder zu einer Haltestelle zu einer Adresse oder einem anderen Ortspunkt berechnet werden. Es können daher nur Routen von Haltestelle zu Haltestelle berechnet werden (Koordinaten-Inputs oder Adress-Inputs würden einen Fußwegvorlauf benötigen und können daher nicht berechnet werden).

PKW Routing ist im Ausland auf Basis von OpenStreetMap (OSM) möglich, aber nur für die Nachbarländer Deutschland, Schweiz, Niederlande, Ungarn, Tschechien, Slowakei, Lichtenstein, Italien, Slowenien und Kroatien. Das Routing erfolgt auf Basis des OSM Graphen.

ÖV Routing ist in jene Länder/Regionen möglich, wo es eine Datenaustauschvereinbarung mit den Mobilitätsverbünden Österreich gibt. Dies ist für Südtirol, Tschechien, Südbayern (Allgäu und Bodenseekreis) möglich. Zu beachten ist, dass nur ein monomodales ÖV-Routing von Haltestelle zu Haltestelle zur Verfügung steht. Daher können im Trip Service keine Adressen oder POIs als Start oder Ziel verwendet werden, sondern nur Haltestellen. Die ReSt-API verhindert die Eingabe nicht, aber es wird keine Route gefunden.

## 5.14 Wetterinformationen

Die Wetterprognosedaten (Wetter in der Route und bei Haltestellen) werden über eine Schnittstelle der GeoSphere Austria eingebunden- Prognosedaten liegen für die nächsten 2 Stunden im Voraus vor.



### 5.15 Daten zu E-Auto Ladestellen

Die in der VAO verfügbaren Ladestellen (siehe dazu Meta-Filter in [Kapitel 8.1](#)) werden über eine Schnittstelle der E-Control bezogen. Diese führt ein Ladestellenverzeichnis aller öffentlichen Ladestellen, das auf der gesetzlichen Meldepflicht für Ladestellenbetreiber beruht. Daher sind Ladestellen nur für Österreich verfügbar. Da das E-Auto Routing die Ladestellen miteinbezieht, können aktuell aufgrund der Datenlage nur Routen in Österreich korrekt berechnet werden. Entsprechende Hinweise zum E-Auto Routing finden sich in [Kapitel 11.6](#). Weiters gibt es bei der Erfassung von Zahlungsmittel und Ökostrom noch keine Verpflichtung, weshalb Vollständigkeit nicht zugesichert werden kann.

### 5.16 Polylinie der Fahrt

Mit dem Parameter `poly=1` kann die Ausgabe der Polylinie in jedem Abschnitt der Route aktiviert werden. Die Polylinie wird entweder in einer Koordinatenabfolge ausgegeben, oder (durch Aktivieren des Parameters `polyEnc=GPA`) kann die Ausgabe im Google Encoded Polyline Format aktiviert werden.

Zum Decodieren können zum Beispiel folgende Tools verwendet werden: <https://open-polyline-decoder.60devs.com/>

Weiters ist eine häufige Fehlerquelle bei der Verwertung der Polylinie, dass für JSON reservierte Zeichen nicht decodiert werden und es daher zu einer Fehlinterpretation im JSON Response-Format kommt

Es ist daher wichtig, den Code zu „unescape“ um ihn im String verwendet zu können. Dabei wird eine Escape-Sequenz in einem String in ihre entsprechenden Zeichen zurückverwandelt. Escape-Sequenzen sind spezielle Zeichenkombinationen, die in Strings verwendet werden, um Zeichen darzustellen, die sonst als Steuerzeichen oder reservierte Zeichen interpretiert werden könnten.

Das bedeutet, dass zum Beispiel in JSON reservierte Zeichen decodiert/unescape werden, um im String verwendet zu werden. Ein Tool zum unescape <https://www.freeformatter.com/json-escape.html>.

## 6 SERVICE-ÜBERSICHT

Dieses Kapitel listet die verfügbaren Services der VAO Schnittstelle auf. In den jeweiligen Unterkapiteln für die Services sind auch die wesentlichen Request-Parameter angeführt. Ausgegraute Parameter und deren „Funktionen“ stehen nicht oder noch nicht zur Verfügung.

Während Location, Trip und StationBoard Services direkt aufgerufen werden können, müssen Journey Detail Service per Referenz aus einem Ergebnis eines Trip, Departure oder ArrivalBoard Requests angefordert werden. Der GisRoute Request, der in der VAO für die Berechnung von Vor- und Nachläufen vorgesehen ist, wird ebenfalls mit einer vorab ermittelten Referenz über den Trip-Request ermittelt. Der TimetableInfo und DataInfo Service enthält wichtige Metadaten zu den ÖV Daten.

Im Folgenden werden alle Services aufgelistet, die bei der ReST API zur Verfügung stehen. Die Liste der verfügbaren Services sowie deren geplante Lebensdauer (**decomission**) kann auch über folgenden Aufruf abgefragt werden.

<https://routenplaner.verkehrsauskunft.at/vao/restproxy>

Service Group	Service /Endpoints	Service name	In Version
Location Services	location.name	Ortspunktauflösung mit Namen (Location search by name)	ab V1.0.0
	location.nearbystops	Ortspunktauflösung mit Koordinate (Location search by coordinate)	ab V1.0.0
	location.details	Details zu Ortspunkten	ab V1.3.2
	addressLookup	Adress-Suche (Address lookup)	ab V1.1.0
Trip Service	trip	Verbindungssuche (Trip search)	ab V1.0.0
Journey Detail Service	journeyDetail	Verbindungsdetails (Journey detail)	ab V1.1.0
Station Board Service	DepartureBoard	Abfahrtstafel (Arrival board)	ab V1.0.0
	ArrivalBoard	Ankunftstafel (DepartureBoard)	ab V1.0.0
GIS Route	gisroute	Gis Route durch Kontext (Gis Route by Context)	ab V.1.0.0
Reconstruction Service	recon	Rekonstruktion (Reconstruction)	ab V1.1.0
Timetable Info Service	tti	Fahrplandaten Information (Timetable Info)	ab V1.1.0
Data Information	datainfo	Dateninformation	ab v1.4.0
Line Info Service (BETA Version)	lineinfo	Linieninformation Änderungen vorbehalten. Der Service dient ausschließlich zu Test- und Explorationszwecken bis zum finalen Launch.	Ab 2025, alle Versionen

## 7 LOCATION.NAME (ORTSPUNKTAUFLÖSUNG MIT NAMEN)

Die Ortspunktsuche, das Location.name Service, ist der vorgelagerte Schritt zu den anderen Services der Schnittstelle, insbesondere der Verbindungsabfrage (Trip). Sie dient dazu, die verfügbaren Ortspunkte und ihre IDs zu identifizieren, um sie weiter in den anderen Endpoints zu nutzen. Hinweise zur den Ortspunkten und deren Erreichbarkeit (**Abschnitt 5.9 Ortspunkte und Verkehrsnetz**) sind zu beachten.

Im Location.name Service wird über die Eingabe eines Ortsnamens eine Ortspunktsuche über alle in der VAO verfügbaren Ortspunkte (Adressen, ÖV Haltestellen und POIs) angestoßen. Ein Pattern-Matching Algorithmus sucht nach den besten Treffern, also der besten Übereinstimmung innerhalb der Ortspunkte-Pools. Zurück gegeben wird eine Liste von Ortspunkten, beginnend mit der höchsten Übereinstimmung. Aus dem Ergebnis der Ortspunktsuche wird ein Eintrag für einen Trip-Request als Start- oder Zielpunkt, oder als Input für Departure-/ArrivalBoards (nur Haltestellen/Stationen) gewählt.



Die Standardeinstellung liefert 10 Ergebnisse zurück. Die Anzahl der ausgegebenen Ortspunkte kann über den Parameter `maxNo` herauf- oder herabgesetzt werden. Über den Parameter `type` kann die Ausgabe nach Ortstypen Adresse (A), Haltestellen/Stationen (S) und POIs (P) gefiltert werden. Auch Kombinationen der Ortspunkttypen sind möglich. Das Root Element für diese Ausgabe ist eine `LocationList`.

## 7.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>input</b>	Erforderlich	Text		Eingabefeld des zu suchenden Namens (Ortes)
<b>maxNo</b>	Optional	1-1000	10	Maximalanzahl der ausgegebenen Halte
<b>type</b>	Optional	ALL, S, A, P, SA, SP, AP	ALL	Typenfilter für Ortstypen: ALL: Suche in allen verfügbaren Datenpools S: nur Haltestellen A: nur Adressen P: nur POIs SA: Haltestellen und Adressen SP: Haltestellen und POIs AP: Adressen und POIs
<b>products</b>	Optional	Dezimal- wert $2^{\text{Produktklasse}}$	ALL	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (siehe Abschnitt 6) in der Suche. Der Dezimalwert entspricht $2^{\text{Produktklasse}}$ . Für Kombinationen von Produktklassen ist die Summe zu bilden, wie folgt:  Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bitmaske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$
<b>coordLong</b>	Optional	Dezimal- grad	-	Längengrad der Koordinate
<b>coordLat</b>	Optional	Dezimal- grad	-	Breitengrad der Koordinate
<b>r</b>	Optional	1-10000	1000	Suchradius um die gegebene Koordinate in Metern
<b>filterMode</b>	Optional	DIST_PERI, EXCL_PERI, SLCT_PERI,	DIST_PERI	ab Version 1.1.0 Filterung der Ergebnisse der Ortspunktauflösung in Bezug auf den Radius. Für nähere Beschreibung siehe Abschnitt 7.3.
<b>refined</b>	Optional	-	-	Im Falle eines weiter aufzulösenden Ortspunktes enthält dieser Wert die id aus einem früher abgesetzten Request



<b>poolId</b>	Optional	103, 104, !103, !104	103	Filterung von inländischen und ausländischen Adressen <ul style="list-style-type: none"><li>• Inländischer Adresspool: poolId=103</li><li>• Ausländischer Adresspool: poolId=104</li><li>• Ausschluss eines Adresspools mit vornagestellten „!“ Zeichen: poolId=!103</li><li>• Kombination der Adresspools mit „ , “ : 103, 104</li></ul>
<b>stations</b>	Optional	41-49	-	Filterung der Response nach ÖV-Teilnetzen (Bundesländer) (es werden nur passende Haltestellen in angegeben Teilnetz ausgegeben) <ul style="list-style-type: none"><li>• 41 Burgenland</li><li>• 42 Kärnten</li><li>• 43 NÖ</li><li>• 44 OÖ</li><li>• 45 Salzburg</li><li>• 46 Steiermark</li><li>• 47 Tirol</li><li>• 48 Vorarlberg</li><li>• 49 Wien</li></ul>
<b>meta</b>	Optional	meta-Filter name		Metafilter nur in Kombination mit type=P für die Filterung nach POI Kategorien (siehe Abschnitt 8.3)

#### Beispiel Request

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/location.name?accessId=yourId&input=Alserstraße&type=A>

## 7.2 Verwendung zusätzlicher Koordinaten (Standort-basierte Auflösung)

Ab Version V1.5.0 steht ein Geo-based Pattern Matching, also die Standort-basierte Auflösung der Ortspunkte zur Verfügung. Das bedeutet, dass zusätzlich zu einem Input eines Namens auch die Koordinate (z.B. der User Standort) mitgegeben werden kann mit dem Parameter `coordLat` und `coordLong`. Die Ortspunkte in einem Radius von 1000m um diese Koordinate werden priorisiert gesucht und somit werden die geografisch passenden Ortspunkte gegenüber einer der reinen Namens-Übereinstimmung vorgereiht.

Wird zum Beispiel der Input „Hauptbahnhof“ angegeben und Koordinaten in Linz, wird das erste Ergebnis in der Liste der Linzer Hauptbahnhof sein. An zweiter Stelle steht auf Grund seiner hoch gewichteten Relevanz meist der Wiener Hauptbahnhof, anschließend folgen weitere Einträge in geografischer Nähe der Koordinaten. Der Radius, in welchen man sich bei dieser Suche rund um die Koordinaten bewegt, wird über den Parameter `r` angegeben. Der Umkreis kann mit dem Parameter `r` noch adaptiert werden (dieser kostet allerdings Rechenzeit).

Wenn `coordLong` und `coordLat` verwendet werden, wird das Pattern-Matching (Ortspunktauflösung) in diesem Bereich durchgeführt. Im Falle eines selektiven Filters (siehe Abschnitt 7.3 Filter-Modus) werden nur Orte im festgelegten Bereich geliefert. Der Filter erlaubt im gesuchten Bereich eine Priorisierung und Vorreihung der Ergebnisse.

Die Verwendung von Koordinaten beim `location.name` Service ist rechenaufwändig und sollte daher nur in Ausnahmefällen genutzt werden. Wenn alle Halte oder POIs rund um eine Koordinate gefunden werden sollen, sollte das `location.nearbystops` Service verwendet werden (siehe Abschnitt 4.2.2).



### 7.3 Filter-Modus (Parameter filterMode)

Wenn `coordLong` und `coordLat` verwendet werden, ist es möglich über den Parameter `filterMode` die Ergebnisse zu steuern. Dadurch können die Ortspunkte gezielt ausgewählt und gereiht werden. Dabei sind folgende 3 Einstellungen möglich:

Filter Modus	Beschreibung
DIST_PERI (Default)	Ergebnisse innerhalb des Radius um die eingegebene Koordinate werden <b>vorgereiht</b> , allerdings werden <b>auch Ergebnisse ohne geografische Nähe</b> angezeigt
EXCL_PERI	Ergebnisse innerhalb des Radius werden <b>ausgeschlossen</b> . Das heißt, alle Ergebnisse im Radius der eingegebenen Koordinate werden in der Ausgabe nicht berücksichtigt
SLCT_PERI	Nur <b>Ergebnisse innerhalb des Radius</b> werden in der Ausgabe <b>berücksichtigt</b>

### 7.4 Höher auflösbare Orte

Manche Orte können im ersten Ergebnis nicht eindeutig aufgelöst werden. Dies wird mit dem Attribut `refinable="true"` in der Response angezeigt. Bei diesen nicht eindeutig aufgelösten Ortspunkten befindet sich im Response ein angehängter Service Request mit dem das Ergebnis noch präzisiert werden kann.

```
<CoordLocation name="1716 Schwarzsee, Ättenbergstrasse" type="ADR" id="A=2@O=1716  
Schwarzsee, Ättenbergstras-  
se@X=7307502@Y=46683563@U=103@b=990017935@B=1@p=1413523432@" lon="7.307502"  
lat="46.683563" refinable="true">  
  <links>  
    <link rel="refine"  
      href="https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}  
/location.name?input=1716 Schwarzsee,  
Ättenbergstrasse&refineId=A=2@O=1716 Schwarzsee, Ätten-  
bergstrasse@X=7307502@Y=46683563@U=103@b=990017935@B=1@p=1413523432@&typ  
e=A"/>  
  </links>  
</CoordLocation>
```

## 8 LOCATION.NEARBYSTOPS (UMKREISSUCHE MIT KOORDINATEN)

Das `location.nearbystops` Service ist eine Suche nach Ortspunkten mit Hilfe einer Umkreissuche um einen Koordinaten-Mittelpunkt. Das Service gibt eine Liste von Ortspunkten (ÖV Haltestellen, POIs, Echtzeit-Fahrzeugstandorte) rund um eine vom Benutzer definierte Koordinate und Suchradius zurück. Die Ergebnisliste ist nach Distanz (geringste Distanz zuerst) zur definierten Koordinate gereiht. Das Service umfasst nicht nur die gängigen Ortspunkte, sondern ist auch um sogenannte EntryPoints erweitert. Dies sind Ortspunkte, die nicht in der Verbindungssuche verwendet werden können, aber relevante Karteninhalte darstellen, wie WebCams oder auch Haltestellen-Zugangspunkte.

Mit dem Parameter `r` kann der Suchradius (default = 1000m) verkleinert oder vergrößert werden (im Bereich 1-10.000m). Mit dem Parameter `maxNo` kann die Anzahl der zurückgegebenen Ergebnisse gesteuert werden. Standardmäßig werden 10 Treffer geliefert. Die Ergebnisliste kann mit dem Parameter `type` auf Ortstypen und mit dem Parameter `products` auf die ÖV Produktklassen eingeschränkt/fokussiert werden.

#### Ortstypen (relevant für Parameter Type)

Folgende Ortspunkttypen können gefiltert werden: Stationen (S), Points of Interest/POIs (P) und EntryPoints (PE und SE). Auch Kombinationen der Typen sind möglich, z.B. Stationen und POIs (SP).

### Wichtiger Hinweis:

Im Service `location.nearbystops` sollen daher Stationen *und* `EntryPoints` (`type=SE`) und POIs *und* `EntryPoints` (`type=PE`) genutzt werden um alle Ortspunkte in der VAO zu erhalten, auch jene die nicht für die Verbindungssuche zur Verfügung stehen, aber als Karteninhalte angeboten werden.

### Unterschied Station (S) und EntryPoint (SE)

Stationen (S) sind „normale“ Stationen, die als Haltestelle mit An-/Abfahrtsmasten modelliert sind. `EntryPoints` (SE) sind komplexe Stationen, die als `EntryPoints` modelliert sind. Diese stehen nur im `location.nearbystops` Service zusätzlich zur Verfügung.

`EntryPoints` sind Zugangspunkte von Haltestellen (SE), von welchen Verbindungswege (Fußwege in der Station) zu den eigentlichen An-/Abfahrtsmasten führen. Für Stationen, die einen Zugangspunkt modelliert haben (Beispiel der Zugangspunkt Bahnhofsvorplatz der Station Bahnhof XY), können daher Fußwegvorläufe zu den in den Daten definierten Zugangspunkt (`EntryPoint`) geführt werden (zum Beispiel zum Bahnhofsvorplatz). Von diesem Zugangspunkt wird dann automatisch der Verbindungsweg zum entsprechenden An-/Abfahrtsmast im Verbindungsergebnis ausgegeben. Daher benötigt es keine explizite Nutzung von `EntryPoints` in der Verbindungssuche. `EntryPoints` können auch POIs sein (PE), welche in der Umkreissuche auffindbar sind (also Karteninhalte), aber für die Verbindungssuche gesperrt sind, wie zum Beispiel WebCams.

### Wichtiger Hinweis:

Entrypoints (SE und PE) können *nicht als Input in einer Trip-Abfrage* genutzt werden (d.h. es kann nicht explizit von oder zum Bahnhofsvorplatz oder von einer POI WebCam geroutet werden), daher ist dieser Ortspunkt-Typ für Trip Abfragen nicht relevant. `EntryPoints` werden in der Response mit dem Attribut `entry="true"` gekennzeichnet. Das Service `location.nearbystops` bietet den zusätzlichen Einbezug von `EntryPoints` an, damit diese in der Umkreissuche auffindbar und damit als Karteninhalte genutzt werden können.

### Stationen und Echtzeit-Fahrzeugstandorte von Sharing-Angeboten

Um die Ortspunktsuche auf Stationen und Fahrzeuge von Sharing-Anbietern, wie Carsharing-PKWs, Leihfahrräder oder E-Scooter sowie Fahrradboxen auszudehnen, muss in Parameter `meta` das gewünschte Verkehrsmittel (Betreiber) als Meta-Filter ausgewählt werden (siehe Abschnitt 8.2 Meta-Filter für bestimmte Inhalte in der Umkreissuche).

In der VAO sind überwiegend stationsbasierte Sharing-Provider und ein Free-floater Provider integriert. Grundsätzlich werden für stationsbasierte Provider in der `location.nearbystops` Response die Stationen (`coordlocation`) und die - an dieser Station verfügbaren Fahrzeuge - als zugeordnetes Element (`childLocation`) ausgegeben.

Die in der Response ausgegebene `id` des Fahrzeuges (`childLocation id`) kann dazu verwendet werden im `location.detail` Service die Detailinformationen zu dem Fahrzeug abzurufen. Dort werden alle Informationen ausgegeben, die über die Sharing-Provider-Schnittstelle zur Verfügung gestellt werden.

## 8.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>originCoordLat</b>	Erforderlich	Dezimal-grad	-	Breitengrad der Koordinate (Mittelpunktes)
<b>originCoordLong</b>	Erforderlich	Dezimal-grad	-	Längengrad der Koordinate (Mittelpunktes)





<b>r</b>	Optional	1-10000	1000	Suchradius um die Koordinate in Meter
<b>maxNo</b>	Optional	1-1000	10	Maximalanzahl der zurückgelieferten Halte
<b>type</b>	Optional	S, P, SP, SE, SPE  SE, SPE (ab V1.4.0) PE	S  SE (ab V1.4.0)	Typenfilter für Ortstypen:  S: nur Stationen P: nur POIs SE: Stationen und -EntryPoints PE: POIs und EntryPoints  Kombinationen sind ebenfalls möglich: SP: Haltestellen und POIs  Ab V1.4.0 werden zwischen Stationen (S) und EntryPoints (SE und PE) unterschieden. Der Type=SE in V1.4.0 entspricht daher Type=S in Vorversionen.
<b>products</b>	Optional	Dezimalwert  $2^{\text{Produktklasse}}$	ALL	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (siehe Abschnitt 0) in der Suche. Der Dezimalwert entspricht $2^{\text{Produktklasse}}$ . Für Kombinationen von Produktklassen ist die Summe zu bilden, wie folgt:  Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bit-Maske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$
<b>meta</b>	Optional	meta Filter Name		Metafilter, der die Suche und Ausgabe Sharing Stationen und Fahrzeugen, sowie Infrastruktur (Ladesäulen und VMOBIL Rad-Boxen) aktiviert (siehe Abschnitt 8.2)  Metafilter in Kombination mit type=P auch für die Filterung von POI Kategorien verwendbar (siehe Abschnitt 8.3)

#### Beispiel für eine Suche nach Halten/Stationen rund um eine gegebene Koordinate

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/location.nearbystops?&accessId=yourId&originCoordLong=16.362264&originCoordLat=48.21394&products=1&r=10000>

## 8.2 Meta-Filter für bestimmte Inhalte in der Umkreissuche

MCP Kategorie	API Meta-Filter	Verfügbarkeit	Beschreibung
<b>Car-Sharing</b>			
<b>floMobil</b> stationsbasiert	API_FLOMOBIL	regional (siehe Betreiberinformationen)	floMobil-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle Cantamen/Stadtwerke Wörgl).
<b>Caruso</b> stationsbasiert	API_CARUSO	regional (siehe Betreiberinformationen)	CARUSO-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle CARUSO).





<b>WienMobil Auto</b> stationsbasiert	API_WIENMOBIL AUTO	regional (Wien)	WienMobil-Auto-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle MOQO).
<b>ShareToo</b> stationsbasiert	API_SHARETOO	regional (Wien)	ShareToo-Auto-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt (Quelle: Betreiberschnittstelle MOQO).
<b>Bike-Sharing</b>			
<b>NEXTBIKE</b> stationsbasiert	API_NEXTBIKE	regional	NEXTBIKE-Stationen werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle Nextbike). <b>(inklusive RegioRad Tirol, Nextbikes)</b>
<b>CITYBIKE</b> stationsbasiert	API_CITYBIKE	regional	CITYBIKE Stationen werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle Nextbike). <b>WienMobil Rad, CityBike Linz</b>
<b>*LIME Fahrräder</b> free-Floater	API_LIME_BIKE	regional (siehe Betreiberinforma- tionen)	LIME Leihfahrrad-Standorte werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle LIME).
<b>Scooter-Sharing</b>			
<b>TIER Roller</b> free-Floater	API_TIER	regional	Service wurde vom Betreiber eingestellt. (Service in Wien vom Betreiber eingestellt).
<b>*LIME Scooter</b> free-Floater	API_LIME_SCOO TER	regional (siehe Betreiberinforma- tionen)	LIME Scooter-Fahrzeug-Standorte werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle LIME).
<del>EASYWAY Roller</del>	<del>API_EASYWAY</del>		Service vom Betreiber eingestellt
<b>Sonstige</b>			
<b>VMOBIL Fahrrad Boxen</b>	API_VMOBIL	regional (Vorarlberg)	Standorte der VMOBIL Fahrrad-Boxen werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle).
<b>E-Auto Ladesäulen (E-Control)</b>	API_ECONTROL	Österreich	Ladesäulen (e-Auto) werden in der Umkreissuche berücksichtigt. (Quelle: E-Control Schnittstelle/ Ladestellenverzeichnis).
<b>Park&amp;Ride Anlagen</b>	API_PARK_RIDE	Österreich	Park&Ride Anlagen werden in der Umkreissuche berücksichtigt. (Quelle: ÖAMTC)

\* **LIME:** Da es auf Basis von vertraglichen Vorgaben in den Nutzungsbedingungen der Sharing-Anbieter zu Einschränkungen bei der Bereitstellung von Sharing-Diensten kommen kann steht LIME *derzeit* nur einem eingeschränkten Kundenkreis zur Verfügung.

#### Beispiel für eine Suche nach TIER Roller Standorten:

[https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/location.nearbystops?accessId={{accessId}}&originCoordLong=16.386663&originCoordLat=48.231137&meta=API\\_TIER](https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/location.nearbystops?accessId={{accessId}}&originCoordLong=16.386663&originCoordLat=48.231137&meta=API_TIER)

#### Auszug aus der Response -> TIER Roller im Element CoordLocation im gewählten Umkreis mit Standort, Roller Nummer und Distanz, sowie weiteren Informationen im Bereich LocationNotes

```
<CoordLocation id="A=2048@O=TIER No 159741@X=16385484@Y=48231406@u=92@L=%23mcp%3AV%3Atier%3A62c6cddb-cdae-4a16-889d-9114ba7bc77b" extId="%23mcp%3AV%3Atier%3A62c6cddb-cdae-4a16-889d-9114ba7bc77b" name="TIER No 159741" type="MCP" lon="16.385484" lat="48.231406" dist="92">
  <LocationNotes>
    <LocationNote key="provider" type="U">tier</LocationNote>
  </LocationNotes>
</CoordLocation>
```



```
<LocationNote key="providerName" type="U">TIER</LocationNote>
<LocationNote key="availability" type="U">A</LocationNote>
<LocationNote key="mcpId" type="U">tier:62c6cdbb-cdae-4a16-889d-
9114ba7bc77b</LocationNote>
<LocationNote key="externalId" type="U">62c6cdbb-cdae-4a16-889d-
9114ba7bc77b</LocationNote>
<LocationNote key="mcpType" type="U">V</LocationNote>
</LocationNotes>
<icon res="prod_tier">
  <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
  <backgroundcolor r="83" g="186" b="170" hex="#53BAAA"/>
</icon>
</CoordLocation>
```

Wenn Detailinformationen (Echtzeit) zum Fahrzeug abgerufen werden sollen, kann mit der Id des Fahrzeuges ein `location.details` Request angestoßen werden (siehe Abschnitt 9 `Location.details`).

### 8.3 Meta-Filter für POI-Filterung

In der VAO werden Points of Interest (POIs) als Ortspunkte ausgegeben. Diese POIs werden in Kategorien zusammengefasst. Derzeit wird die POI Kategorie in einem Hinweistext (`location Note`) bei jedem POI ausgegeben:

```
LocationNote key="Pv" type="A" txtN="Bank/Geld">Bank/Geld</LocationNote>.
```

Diese Kategorien können mit Hilfe des passenden Meta-Filters (siehe unten stehende Tabelle) als Input im Parameter `meta` verwendet werden, um die Response auf eine bestimmte POI Kategorie einzuschränken. Diese Funktionalität ist ab Version 1.5.0 verfügbar.

#### Wichtige Hinweise:

Zu beachten ist, dass diese POI-Kategorien (Meta-Filter für POI-Filterung) nur mit dem Ortspunkttyp POIs verwendet werden können, also bei gesetztem Parameter **type=PE**. Eine Kombination (Mehrfachauswahl) von Kategorien ist ebenfalls möglich.

Zu beachten ist, dass *keine* Kombination mit den Meta-Filtern zum Abfragen von anderen Karteninhalten (wie Sharing Providern) möglich ist. Meta-Filter für POI und Meta-Filter für andere Karteninhalte müssen in separaten Requests verwendet werden. Denn Meta-Filter für andere Karteninhalte werden ohne `type=PE` angewendet und Meta-Filter für POI müssen mit `type=PE` angewendet werden.

Es gibt POIs die reine Karteninhalte darstellen, die in den VAO Frontends daher für die Kartenauswahl im Routing gesperrt sind. Diese POIs sind teilweise nicht an das Verkehrsnetz angebunden, wie zu Beispiel Almen, Gipfel, Asfiang Webcams. Daher gibt eine Routenberechnung mit POIs, die reine Karteninhalte sind, einen Fehler aus. Daher sollten keine POIs im Routing verwendet werden, die folgenden Hinweis in der Response haben:

```
<Note key="NS" type="A" priority="101" txtN="Gesperrt fuer Kartenauswahl">Gesperrt fuer Kartenauswahl</Note>
```

#### Beispielabfragen:

##### Filter einzelner Kategorien: Beispiel Fahrradabstellanlage:

```
{{baseURL}}/{{version}}/location.nearbystops?accessId={{key}}&originCoordLong=16.31195
0683593754&originCoordLat=48.202393276608746&r=33608&meta=API_PCAT_FAHRRADABSTELL&type
=P&maxNo=100
```

##### Filterung mit mehreren Kategorien:

```
{{baseURL}}/{{version}}/location.nearbystops?accessId={{key}}&originCoordLong=16.31195
0683593754&originCoordLat=48.202393276608746&r=33608&meta=API_PCAT_CASINO,API_PCAT_KIN
O&type=P&maxNo=100
```



Kategorie	Meta-Filter Name	Anmerkung
<b>Go-Vertriebsstelle</b>	<del>API_PCAT_COVERTRIEB</del>	<del>ASFINAG Vertriebsstellen (Vignette, GO-Maut) (Quelle: ASFINAG)</del>
<b>Rastplatz (PKW/LKW)</b>	API_PCAT_RASTPLATZ	ASFINAG Rastplatz (Quelle: ASFINAG)
<b>Raststation (PKW/LKW)</b>	API_PCAT_RASTSTATION	ASFINAG Raststation (Quelle: ASFINAG)
<b>Webcam (Autobahnen und Wien)</b>	API_PCAT_WEBCAM	WebCams (Quelle: ASFINAG, ITS Vienna Region)
<b>Parkgarage oder Parkfläche</b>	API_PCAT_PARKGARAGE	Parkgaragen, Parkflächen (Quelle: ÖAMTC)
<b>Berg-/ Passstraße</b>	<del>API_PCAT_BERGSTRASSE</del>	<del>Berg- und Passstraße (Quelle: ÖAMTC)</del>
<b>Mautstraße</b>	<del>API_PCAT_MAUTSTRASSE</del>	<del>Mautstraßen (Quelle: ÖAMTC)</del>
<b>Tunnel</b>	API_PCAT_TUNNEL	Tunnel (Quelle: ÖAMTC)
<b>ÖAMTC Standort</b>	API_PCAT_OEAMTC	ÖAMTC Standorte (Quelle: ÖAMTC)
<b>Bundesministerien</b>	API_PCAT_BUNDESMINISTERIUM	Bundesministerien (Quelle: VAO)
<b>Krankenanstalt</b>	API_PCAT_KRANKENANSTALT	Krankenanstalt (Quelle: VAO)
<b>Diplomatische Vertretung</b>	API_PCAT_DIPLVERTRETUNG	Diplomatische Vertretungen (Quelle: VAO)
<b>Anschlussstelle</b>	API_PCAT_ANSCHLUSSSTELLE	Autobahn-Anschluss-stelle/Knoten ASFINAG Raststation (Quelle: ASFINAG)
<b>Fahrradabstellanlage</b>	API_PCAT_FAHRRADABSTELL	Fahrradabstellanlagen und Radboxanlage (Quelle: ÖVDAT)
<b>ARBÖ-Standort</b>	API_PCAT_ARBOE	ARBÖ Standorte (Quelle: VAO)
<b>ASFINAG Standort</b>	API_PCAT_ASFINAG	ASFINAG Standorte (Quelle: ASFINAG)
<b>Vertriebsstelle GO Box</b>	API_PCAT_GOBX	ASFINAG Vertriebsstellen GO-BOX (Quelle: ASFINAG)
<b>ASFINAG Vignette</b>	API_PCAT_VIGNETTE	ASFINAG Vignette (Quelle: ASFINAG)
<b>VideoMaut</b>	API_PCAT_VIDEMAUT	ASFINAG digitale Streckenmaut (Quelle: ASFINAG)
<b>Bank/Geld</b>	API_PCAT_BANK	Quelle: ÖAMTC
<b>Camping</b>	API_PCAT_CAMPING	Quelle: ÖAMTC
<b>Casino</b>	API_PCAT_CASINO	Quelle: ÖAMTC
<b>Einkaufen</b>	API_PCAT_EINKAUFEN	Quelle: ÖAMTC
<b>Essen</b>	API_PCAT_ESSEN	Quelle: ÖAMTC
<b>Freizeitpark</b>	API_PCAT_FREIZEITPARK	Quelle: ÖAMTC
<b>Apotheke</b>	API_PCAT_APOTHEKE	Quelle: ÖAMTC
<b>Kino</b>	API_PCAT_KINO	Quelle: ÖAMTC
<b>Theater</b>	API_PCAT_THEATER	Quelle: ÖAMTC
<b>Touristeninformation</b>	API_PCAT_TOURISTINFO	Quelle: ÖAMTC
<b>Unterkunft</b>	API_PCAT_UNTERKUNFT	Quelle: ÖAMTC
<b>Markt</b>	API_PCAT_MARKT	Quelle: ÖAMTC
<b>Veranstaltungen/Festivals</b>	API_PCAT_KULTUR	Quelle: ÖAMTC
<b>Wassersport</b>	API_PCAT_WASSERSPORT	Quelle: ÖAMTC
<b>Weihnachtsmarkt</b>	API_PCAT_WEIHNACHTSMARKT	Quelle: ÖAMTC
<b>Wintersport</b>	API_PCAT_WINTERSPORT	Quelle: ÖAMTC
<b>Tankstellen</b>	API_PCAT_TANKEN	Quelle: ÖAMTC
<b>Tanken &amp; Spritpreis</b>	<del>API_PCAT_TANKENPREIS</del>	<del>Quelle: ÖAMTC</del>
<b>Taxistand</b>	API_PCAT_TAXISTAND	Quelle: WKO
<b>Park &amp; Ride Anlage</b>	<del>API_PCAT_PR</del>	<del>Quelle: ÖAMTC</del>
<b>Sehenswürdigkeit</b>	API_PCAT_SEHENSWERDIGKEIT	Quelle: ÖAMTC
<b>Bildung</b>	API_PCAT_BILDUNG	Quelle: ÖAMTC



<b>Museum</b>	API_PCAT_MUSEUM	Quelle: ÖAMTC
<b>Park</b>	API_PCAT_PARK	Quelle: ÖAMTC
<b>Coming soon: Tiergarten</b>	API_PCAT_ZOO	Quelle: ÖAMTC
<b>Almen</b>	API_PCAT_ALMEN	Quelle: Land Tirol
<b>Gipfel</b>	API_PCAT_GIPFEL	Quelle: Land Tirol
<b>Flughafen</b>	API_PCAT_FLUGHAFEN	Quelle VAO
<b>Friedhof</b>	API_PCAT_FRIEDHOF	Quelle: VAO
<b>TIM Standort</b>	API_PCAT_TIM	Grazer und Linzer Mobilitätsdienst
<b>WienMobil</b>	<b>API_PCAT_WIENMOBIL</b>	<b>Wien-Mobil-Standorte</b>
<b>Park+Drive</b>	API_PCAT_PARKANDDRIVE	Quelle Asfinag
<b>Paketboxen</b>	API_PCAT_PAKETBOXEN	Quelle BMK
<b>Rad servicestationen</b>	API_PCAT_RADSERVICESTATION	Quelle: ÖVDAT
<b>Sonstiges</b>	API_PCAT_SONSTIGES	Sammelkategorie



## 9 LOCATION.DETAILS (ORTSPUNKTDETAILS)

Das `location.details` Service liefert alle **Detailinformationen und Echtzeitinformationen** zu den Ortschaften, die in der ReST API auffindbar sind. Dies sind neben Haltestellen und POIs, welche rein statische Informationen enthalten, auch Fahrzeugstandorte von Sharing-Angeboten, E-Auto Ladestationen und VMOBIL Radboxen. Das Service muss mit der `id` eines Ortschaftes das man durch einen vorangegangenen `location.name` oder `location.nearbystops` Request erhält, angefragt werden.

Über den Parameter `weather` können die **Wetterprognoseinformationen** zu diesem Ortschaft abgefragt werden. Durch das Setzen des Parameters `weather=true` kann die Wetterprognose für diese Orte abgefragt werden (Prognosezeitraum 2 Stunden). Die Wetterdaten sind in der `location.details` Response als Zusammenfassung (Summary) und als einzelne Attribute inklusive Kategorisierung (`type`) und Icon für die Anzeige vorhanden.

### Haltestellendetails:

Bei einer Anfrage mit einer `id` einer Haltestelle werden zu dieser Haltestelle bzw. `StopLocation`, alle Produkte, also Linien/Verkehrsmittel gelistet, die an dieser Haltestelle in einer Fahrplanperiode abfahren (im Element `productAtStop`).

### POI-Details:

Bei einer Anfrage mit einer `id` eines POIs werden die POI Informationen im Bereich `Location Notes` ausgegeben. Derzeit sind die Informationen aber analog zu den Informationen in `Location.nearbystops`. Eine Ausnahme stellen die Paketboxen dar, hier werden auch weitere Informationen ausgegeben.

### Fahrzeugdetails:

Bei der Anfrage mit einer `id` eines Fahrzeuges (`childLocation id` aus `location.nearbystops`) werden zusätzlich zur Fahrzeugnummer, Standort des Fahrzeuges (Koordinaten) im Bereich `LocationNotes` alle vom Betreiber zur Verfügung gestellten Detail- und Echtzeitinformationen ausgegeben. Neben dem Fahrzeugtyp, Treibstofftyp, sind dies auch aktuelle Ladestand- und Verfügbarkeits-Informationen, sowie Preisinformation und je nach Betreiber auch weitere Informationen.

### Fahrradboxdetails:

Bei der Anfrage mit einer `id` einer VMBOL Fahrradbox werden neben der Anzahl der Slots an diesem Standort auch die aktuell verfügbaren Slots angegeben.

Bei einer Anfrage mit einer `id` einer Ladestation werden Steckertypen und alle sonstigen Informationen zur Ladestation, die von E-Control erfasst werden, ausgegeben.

### Buchungsaussprünge

In den `LocationNotes` zu dem Fahrzeug und zur VMOBIL Radbox sind neben den oben genannten Informationen auch Buchungs-Aussprünge als Deep-Links verfügbar, zumindest für WebApps und mehrfach auch für Android und iOS (Buchungsaussprünge sind verfügbar für WienMobil Auto, Sharetoo, Flomobil, Caruso, TIER, LIME).

### Ladestellendetails:

In den `LocationNotes` werden alle über die E-Control Schnittstelle zur Verfügung gestellten Informationen über die Ladestelle ausgegeben (siehe dazu [Kapitel 5.11](#)).



## 9.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>id</b>	Erforderlich		-	Id des Ortpunktes (Haltestelle, Fahrzeuges, Ladestation, VMOBIL Box) über den weitere Details abgefragt werden sollen (muss durch ein location services vorher abgefragt werden).
<b>weather</b>	Optional	true/ false	false	Aktiviert die Ausgabe der Wetterprognoseinformationen für den Ortpunkt

### Beispiel Abfrage für eine Haltestelle mit Datum und Startzeitpunkt:

[https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/location.details?accessId={{accessId}}&id=A=1@O=Graz Hauptbahnhof@X=15417507@Y=47072481@U=81@L=460304000@i=A\\*at:46:3040@&date=2021-09-22&time=23:00:00](https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/location.details?accessId={{accessId}}&id=A=1@O=Graz Hauptbahnhof@X=15417507@Y=47072481@U=81@L=460304000@i=A*at:46:3040@&date=2021-09-22&time=23:00:00)

### Auszug aus einer Response für eine Haltestelle (Graz Hauptbahnhof) -> Ausgabe von Elementen productsAtStop in StopLocation Element

```
<StopLocation id="A=1@O=Graz Hauptbahnhof@X=15417507@Y=47072481@U=81@L=460304000@i=A*at:46:3040@
" extId="460304000" name="Graz Hauptbahnhof" lon="15.417507" lat="47.072481" products="4339" met
a="true">
  <productAtStop name="D" internalName="D" catOut="D" cls="1" catOutS="D" catOutL="Schnell
zug">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
  </productAtStop>
  <productAtStop name="EC" internalName="EC" catOut="EC" cls="1" catOutS="EC" catOutL="Eur
ocity">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
  </productAtStop>
  <productAtStop name="IC" internalName="IC" catOut="IC" cls="1" catOutS="IC" catOutL="Int
ercity">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
  </productAtStop>
</StopLocation>
```

### Beispiel Abfrage TIER Roller:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/location.details?accessId={{accessId}}&id=A=2048@O=TIERNö 128742@X=16385380@Y=48232084@u=141@L=%23mcp%3AV%3Atier%3Ad0fa164d-a3a1-4f78-96c9-e488c4a9e741@>

Da es sich um Echtzeit-Daten handelt, dient da Beispiel nur dazu, die Verwendung der id des Fahrzeuges zu zeigen.

### Auszug aus einer Response für einen TIER Roller -> Ausgabe von Elementen LocationNotes mit Fahrzeuginformationen in CoordLocation Element



```
<CoordLocation id="A=2048@O=TIER No 128742@X=16378864@Y=48236651@L=%23mcp%3AV%3Atier%3Ad0fa164d-a3a1-4f78-96c9-e488c4a9e741@" extId="%23mcp%3AV%3Atier%3Ad0fa164d-a3a1-4f78-96c9-e488c4a9e741" name="TIER No 128742" type="MCP" lon="16.378864" lat="48.236651">
  <LocationNotes>
    <LocationNote key="FUEL_TYPE" type="W" txtN="ELECTRIC"/>
    <LocationNote key="FUEL_LEVEL" type="W" txtN="72"/>
    <LocationNote key="CHARGE_LEVEL" type="W" txtN="72"/>
    <LocationNote key="IS_BOOKABLE" type="W" txtN="true"/>
    <LocationNote key="BRAND" type="W" txtN="okai"/>
    <LocationNote key="REGION" type="W" txtN="VIENNA"/>
    <LocationNote key="TIER_VEHICLE_CODE" type="W" txtN="128742"/>
    <LocationNote key="VEHICLE_TYPE" type="W" txtN="escooter"/>
    <LocationNote key="BOOKEE_NAME" type="W" txtN="TIER No 128742"/>
    <LocationNote key="UPDATE_INFO_STALE_TIME" type="W" txtN="2100-01-01T00:00:00Z"/>
    <LocationNote key="UPDATE_INFO_EXPIRATION_TIME" type="W" txtN="2100-01-01T00:00:00Z"/>
    <LocationNote key="LINK_WEB" type="W" txtN="https://www.tier.app/de/">
    <LocationNote key="PROVIDER_NAME" type="W" txtN="TIER"/>
    <LocationNote key="provider" type="U">tier</LocationNote>
    <LocationNote key="providerName" type="U">TIER</LocationNote>
    <LocationNote key="availability" type="U">A</LocationNote>
    <LocationNote key="mcpId" type="U">tier:d0fa164d...e488c4a9e741</LocationNote>
    <LocationNote key="externalId" type="U">..e488c4a9e741</LocationNote>
    <LocationNote key="mcpType" type="U">V</LocationNote>
  </LocationNotes>
```

Die Informationen sind abhängig von der angebundenen Provider-Schnittstelle und deren Inhalte.

#### Beispiel Abfrage für eine Parkgarage (POI):

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/location.details?accessId={accessId}&id=960081747>

#### Auszug aus einer Response für eine Parkgarage (POI) -> Ausgabe von Elementen LocationNotes mit Fahrzeuginformationen in CoordLocation Element

```
<CoordLocation id="A=4@O=Dresdner Straße Parkplatz, 1200 Wien (Parkgarage / Parkfläche)@X=16385744@Y=48231793@U=130@L=960081747@" extId="960081747" name="Dresdner Straße Parkplatz, 1200 Wien (Parkgarage / Parkfläche)" type="POI" lon="16.385744" lat="48.231793">
  <LocationNotes>
    <LocationNote key="Pe" type="A" txtN="Parkgarage / Parkfläche"/>
    <LocationNote key="QE" type="A" txtN=""/>
    <LocationNote key="availability" type="U">N</LocationNote>
    <LocationNote key="mcpId" type="U">0</LocationNote>
    <LocationNote key="externalId" type="U">oeamtc_pg-008316006076AD</LocationNote>
    <LocationNote key="mcpType" type="U">P</LocationNote>
    <LocationNote key="WV_URL" type="W" txtN="WEBVIEW">https://demo.hafas.de/webapp/vao/lyra/webviews/poi_loc_based/?extId=960081747&type=P&language=deu&clientType=API</LocationNote>
  </LocationNotes>
```



## 10 ADDRESSLOOKUP (ADRESS-SUCHE UM KOORDINATE)

Das AddressLookup Service ermöglicht das Identifizieren der nächstgelegenen Adresse um eine angegebene Koordinate. Das Service entspricht im Frontend der Funktion „Klicken in die Karte“, bei dem die nächstliegende Adresse identifiziert und ausgegeben wird.

Der AddressLookup Request gibt die nächstgelegenen Ortspunkt/Location zurück (kann Adresse, Haltestelle oder POI sein). Der Request liefert immer nur ein Ergebnis zurück (nächstgelegene Haltestelle, POI oder Adresse). Die Response liefert für die nächstgelegenen Straßen die zur Koordinate nächstgelegene Hausnummer (Adresse). Der Radius für die Suche ist fest hinterlegt.

Der Parameter `type` ist ab Release V1.3.2 verfügbar. Dieser ermöglicht die Auswahl des zurückgebenden Ortstyps (Adresse, POI oder Haltestelle).

### 10.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>originCoordLat</b>	Erforderlich	siehe 5.1	-	Breitengrad der Koordinate
<b>originCoordLong</b>	Erforderlich	siehe 5.1	-	Längengrad der Koordinate
<b>maxNo</b>	Optional	<del>1-1000</del>	<del>10</del>	Veraltet/aufgelassen
<b>type</b>	Optional	<b>A, S, P, AP, SA, SAP, SP, C</b>	<b>C</b>	Ermöglicht die Filterung nach Ortstyp Adresse: type=A POI: type=P Haltestelle/Station: type=S Koordinaten-basierte suche: type=C Kombinationen sind ebenfalls möglich.

#### Beispiel für Abfrage mit `type=S` (Haltestelle/Station)

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/addresslookup?accessId={{accessId}}&format=xml&originCoordLat=48.196803&originCoordLong=16.338169&type=S>

#### Auszug aus Response mit `type=S` (Haltestelle/Station) → Ausgabe eines Elementes StopLocation (Wien Westbahnhof)

```
<StopLocation id="A=1@O=Wien Westbahnhof@X=16338281@Y=48196762@u=9@U=81@L=490146869@" extId="490146869" hasMainMast="true" mainMastId="A=1@O=Wien Westbahnhof@X=16337652@Y=48196654@U=81@L=490146800@" mainMastExtId="490146800" name="Wien Westbahnhof" lon="16.338281" lat="48.196762" dist="9" products="2135">
  <LocationNotes>
    <LocationNote key="FB" type="A" txtN="Fahrradbeförderung"/>
    <LocationNote key="TZ" type="I" txtN="stv$1"/>
    <LocationNote key="TZ" type="I" txtN="vor$100"/>
  </LocationNotes>
  <altId>at:49:1468:9</altId>
  <mainMastAltId>at:49:1468</mainMastAltId>
  <timezoneOffset>60</timezoneOffset>
</StopLocation>
```





### Beispiel für Abfrage mit **type=P** (POI)

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/addresslookup?accessId={{accessId}}&format=XML&originCoordLat=48.196803&originCoordLong=16.338169&type=P>

Auszug aus Response mit **type=P** (POI) → Ausgabe eines Elementes CoordLocation mit **type="POI"** (Westbahnhof Europaplatz (Citybike-Station))

```
<CoordLocation id="A=4@O=Westbahnhof Europaplatz@X=16338569@Y=48196825@u=29@U=130@I=960003503@"  
extId="960003503" name="Westbahnhof Europaplatz" type="POI" lon="16.338569" lat="48.196825" dist  
="29">  
  <LocationNotes>  
    <LocationNote key="Pj" type="A" txtN="Citybike"/>  
    <LocationNote key="Q1" type="A" txtN=""/>  
    <LocationNote key="availability" type="U">N</LocationNote>  
    <LocationNote key="mcpId" type="U">0</LocationNote>  
    <LocationNote key="externalId" type="U">ITSVR_1103</LocationNote>  
    <LocationNote key="mcpType" type="U">P</LocationNote>  
  </LocationNotes>  
  <icon res="PCAT_CITYBIKE"/>  
</CoordLocation>
```

Auszug aus Response mit **type=C** (Koordinate) → Ausgabe eines Elementes CoordLocation oder StopLocation (Es wird die nächstgelegene Location zurückgegeben (kann Adresse, Haltestelle oder POI sein, hier ist es die Haltestelle Wien Westbahnhof).

```
<StopLocation id="A=1@O=Wien Westbahnhof@X=16338281@Y=48196762@u=9@U=81@L=490146869@" extId="490  
146869" hasMainMast="true" mainMastId="A=1@O=Wien Westbahnhof@X=16337652@Y=48196654@U=81@L=49014  
6800@" mainMastExtId="490146800" name="Wien Westbahnhof" lon="16.338281" lat="48.196762" dist="9  
" products="2135">  
  <LocationNotes>  
    <LocationNote key="FB" type="A" txtN="Fahrradbeförderung"/>  
    <LocationNote key="TZ" type="I" txtN="stv$1"/>  
    <LocationNote key="TZ" type="I" txtN="vor$100"/>  
  </LocationNotes>  
  <altId>at:49:1468:9</altId>  
  <mainMastAltId>at:49:1468</mainMastAltId>  
  <timezoneOffset>60</timezoneOffset>  
</StopLocation>
```

### Abfrage ohne Parameter **type**

Ausgabe wie unter Verwendung von **type=C**, es kommt zur Ausgabe eines Elementes CoordLocation oder StopLocation (Es wird die nächst gelegene Location zurückgegeben (kann Adresse, Haltestelle oder POI sein, hier ist es die Haltestelle Wien Westbahnhof)

## 11 TRIP SERVICE

Das Trip Service besteht aus dem Trip Service dieser führt eine Routenberechnung von einem bestimmten Start- zu einem bestimmten Zielpunkt durch. Über das Trip Service können Routenberechnungen für alle in der VAO verfügbaren Modalitäten und Betreiber (mit Hilfe von Group-Filtern) durchgeführt werden.

Als Ergebnis liefert der Dienst die berechneten Fahrten mit Basisinformationen für jeden Abschnitt (Legs) der gefundenen Fahrten. Diese beinhaltet Ankunfts- und AbfahrtsHaltestelle/Bahnhof, Ankunfts- und Abfahrtszeit, Distanz und Dauer der Fahrt (inkl. Echtzeit, falls verfügbar).

Ab V1.3.0 erfolgt auch die Ausgabe von CO<sub>2</sub>-Ausstoßes je Verbindung und Vergleichswert, sodass die CO<sub>2</sub>-Ersparnis berechnet werden kann gegenüber der PKW Nutzung. im Element `Eco co2` wird der CO<sub>2</sub>-Ausstoß in



Kilogramm für die zugrunde liegende Verbindung ausgegeben. im Element `EcoComp co2` wird der CO<sub>2</sub>-Ausstoß in Kilogramm der Referenzverbindung, die für den Vergleich herangezogen wird, ausgegeben.

Ab V1.7.0 sind die Wetterprognosedaten der GeoSphere integriert und in der Trip Response beim Start und Ziel vorhanden, wenn die Startzeit und Zielzeit innerhalb des Prognosezeitraums von 2 Stunden liegt.

#### Wichtiger Hinweis:

Abfragen des Trip Service können mit Hilfe von gültigen Ortspunkten erfolgen, die in den Parameter `originId` und `destId` eingegeben werden. Es sind die Hinweise zur Erreichbarkeit von Ortspunkten (siehe [Abschnitt 5.9 Ortspunkte und Verkehrsnetz](#)) zu beachten. Hier können sowohl die `Id` als auch die `extId` eines Ortspunktes (ermittelbar durch die Location services) verwendet werden (siehe dazu [Abschnitt 5.4 Haltestellenmodellierung und IDs](#))

Weiters ist auch die Verwendung von Koordinaten als Startpunkt (Parameter `originCoordLat` und `originCoordLong`) und Zielpunkt (Parameter `destCoordLat` und `destCoordLong`) möglich. Grundsätzlich sind Koordinaten frei wählbar, die Location der Koordinate darf aber nicht auf einer „Erreichbarkeits-Insel“ und sehr abgelegen vom Wegenetz in Bezug auf die gewählte Modalität liegen. Hier sind die Hinweise zum Snapping aufs Wegenetz unter [Abschnitt 5.9 Ortspunkte und Verkehrsnetz](#) zu beachten.

Die Standard-Modalität (default) beim Trip Request ist der ÖV Modus. Abfragen können für einzelne Modalitäten und multimodalen Kombinationen von Verkehrsmitteln über den Parameter `groupFilter` abgesetzt werden (siehe [Abschnitt 11.2 Modalitäten und Kombinationen \(groupFilter\)](#)).

#### Hinweise zu ÖV Routen

Der Standardsuchmodus der ÖV Verbindungssuche liefert die schnellsten und bequemsten Fahrten (möglichst wenige Umsteige). Mit dem „ökonomischen“ Suchmodus können mehr Suchvorgänge mit verschiedenen Auswertungsmethoden durchgeführt werden, die auch gute Direktverbindungen ausgegeben. Eine genaue Beschreibung der Routen-Suchalgorithmus ist unter [Abschnitt 11.3](#) dokumentiert.

Bei der Routenberechnung werden standardmäßig Echtzeitinformationen (Verspätungen, Gleiswechsel) einbezogen, sodass die Dauer der Reise/Fahrt die Verzögerungen beinhaltet. Eine genaue Beschreibung des Verhalten ist unter [Abschnitt 11.3.2 Berücksichtigung von Echtzeit](#) zu finden. Die Ausgabe von SOLL- und IST-Daten in der ÖV Route sind in [Abschnitt 5.5 Fahrplandaten](#) und [Abschnitt 5.7 Echtzeitinformationen im ÖV](#) genauer beschrieben.

Für ÖV Routen kann ein Filter für bestimmte Verkehrsmittel (ÖV Produktklassen) über den Parameter `products` gesetzt werden. Weiters stehen eine Reihe von Parametern zur Verfügung, die es ermöglichen die Verbindungssuche zu steuern, wie Zwischenziele (VIA), Anzahl der Umstiege, Länge des Fußweg-Vorlaufs und Nachlaufs, etc. Bei ÖV Routen können spezielle Vor- und Nachläufe über die Parameter `originWalk`, `originBike`, `originCar`, `originPark`, `originMeta` für den Vorlauf und `destWalk`, `destBike`, `destCar`, `destPark`, `destMeta` für den Nachlauf definiert werden. Standardmäßig ist ein Vor- und Nachlauf von 2000m für den ÖV eingestellt.

Bei adressgenauen ÖV Routen (starten von einer Adresse und nicht von einer Haltestelle) und bei intermodalen Routen mit einem ÖV-Anteil gibt es immer einen entsprechenden Vor- und Nachlauf vom/zur ÖV-Haltestelle der zu Fuß, mit dem Rad oder mit einem PKW zurückgelegt wird. Die Basis-Infos wie Weglänge und benötigte Zeit werden in den einzelnen Abschnitten (Legs) des Trip Responses ausgegeben. Wenn für eine detailliertere Darstellung in einer Karte die gesamte Wegbeschreibung *inklusive Vorlauf und Nachlauf mit Polylinie erforderlich ist*, muss mit der (in der Trip Response zurückgegebenen) `GisRef` ein `GisRoute` Request (siehe [Abschnitt 12 GIS Routen Service](#)) für den Vorlauf und Nachlauf angestoßen werden. Dies ist auch bei kombinierten ÖV-Routen (Bike&Ride, Park&Ride) erforderlich die einen IV-Vorlauf haben.

Tarifinformationen können mit dem Parameter `tariff=1` ausgegeben werden.



Der Abfragezeitpunkt kann durch den Parameter `date` und `time` angepasst werden. Grundsätzlich kann in die Vergangenheit 62 Tage rückwärts vom aktuellen Datum abgefragt werden. In die Zukunft kann bis zum Ende der Fahrplanperiode oder bis zum nächsten Fahrplanwechsel abgefragt werden. Es gibt jedoch Fakten zur berücksichtigen bei der Abfrage in die Zukunft und in die Vergangenheit. Siehe dazu [Abschnitt 5.5.3 Abfragen in die Zukunft und Vergangenheit](#).

#### Hinweise zu IV Routen

Bei der Routenberechnung von IV-Routen (Auto, E-Auto, Fahrrad, Fuß) werden die VAO Standardkonfigurationen (per Default) verwendet, die in [Abschnitt 11.4 Routen-Suchalgorithmus für IV](#) beschrieben sind. Bei der Routenberechnung werden standardmäßig Echtzeitinformationen (Verkehrsmeldungen, Verspätungen) einbezogen, sodass die Dauer der Reise/Fahrt die Verzögerungen beinhaltet. Diese Dynamiken können per Parameter deaktiviert werden, sodass die Fahrt ohne Echtzeitinformationen ausgegeben wird. Es werden allerdings keine Verkehrsmeldungen und Kurzparkzoneninformationen per Default ausgegeben. Sofern das gewünscht ist, muss die Ausgabe aktiviert werden. Eine genaue Beschreibung ist in [Abschnitt 11.4.2 Berücksichtigung von Echtzeit](#) zu finden.

Die VAO stellt zudem weitere vordefinierte Konfigurationen des IV-Router, sogenannte Meta-Profile, bereit. Diese Meta-Profile enthalten Anpassungen für den Router für Geschwindigkeiten (Gehen und Auto), Optimierung nach Zeit oder Distanz (schnellste oder kürzeste Route), bestimmte Anpassungen für Radfahrer, sowie oben genannte Aktivierung oder Deaktivierung von Verkehrsmeldungen und Kurzparkzoneninformation in der Auskunft. Diese Meta-Profile werden in [Abschnitt 11.5 Meta-profile für IV-Routing](#) beschrieben.

Ab V1.8.0 wird das IV Routing für E-Auto erweitert und umfasst eine Reihe von Konfigurationsmöglichkeiten, die in [Abschnitt 11.6 Meta-Profile für E-Auto Routing](#) dokumentiert sind.

#### Alternative Fußwege:

Bei ÖV-Abfragen ist konfiguriert, dass im Falle eines effizienteren Fußwegs (wenn die Luftliniendistanz zwischen Start und Ziel <1500m ist) eine alternative Fußwegverbindung ausgegeben wird. Desweiteren werden für Verbindungen mit nah beieinanderliegendem Start und Ziel Fußwegverbindungen mitberechnet (unscharfe Suche).

Bei den Modi PKW und Fahrrad hingegen, wird kein alternativer Fußweg vorgeschlagen bei einer Luftliniendistanz <1500m. Hier liefert die Abfrage *kein Ergebnis*, wenn die Distanz zu kurz ist für eine IV Route.

#### Zwischenziele (Via Orte)

Bei der Routenberechnung kann auch ein Via-Ort, also ein Zwischenort bzw. ein Zwischenhalt (Haltestelle oder Adresse) der durchfahren werden soll, mitberücksichtigt werden.

#### Wichtiger Hinweis:

Via-Orte können nur für monomodale Routen (siehe Group-Filter für monomodale Routen) verwendet werden und nicht für kombinierte Routen.

Ein Via-Ort wird über den Parameter `viaId` mitgegeben und bewirkt, dass die Routensuche über diese Haltestelle oder Adresse geführt wird. Über den Parameter `viaWaitTime` kann die Verweilzeit in der Station (in Minuten) festgelegt werden. Sollen mehrere Vias mitgegeben werden, so muss dies mit einer komplexen Struktur im Parameter `via` erfolgen. Wird dieser Parameter verwendet, so werden die Parameter `viaId` und `viaWaitTime` hinfällig. Die Via Funktionalität ist im [Abschnitt 11.7 Via Orte in der ÖV und IV Route](#) genauer beschrieben.

Die IDs für den Parameter `via` und `viaID` kann über Locations Service erhalten werden. Haltestellen (Modalität ÖV) und Adressen (Modalität Auto) haben in der Response ein Attribut `id` oder `extId`, die entsprechend hier weiterverwendet werden muss. Im Parameter `via` können für die Modalität Auto auch Koordinaten verwendet werden.



### 11.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access Id zur Authentifizierung
<b>originId</b>	Erforderlich	siehe Abschnitt 5.4.2	-	Legt den Startpunkt fest. Diese kann über das location.name- oder location.nearbystops- Service gefunden werden.
<b>destId</b>	Erforderlich	siehe Abschnitt 5.4.2		Legt den Zielpunkt fest Diese kann über das location.name- oder location.nearbystops- Service gefunden werden.
<b>originExtId</b>			-	Veraltet/aufgelassen
<b>destExtId</b>				Veraltet/aufgelassen
<b>originCoordLat</b>	Erforderlich wenn Id nicht angegeben ist	siehe Abschnitt 5.2	-	Breitengrad des Routenstartpunkts
<b>originCoordLong</b>	Optional	siehe Abschnitt 5.2	-	Längengrad des Routenstartpunkts. Dieser kann über das location.name- oder location.nearbystops- Service ermittelt oder innerhalb von Österreich frei gewählt werden.
<b>destCoordLat</b>	Erforderlich wenn Id nicht angegeben ist	siehe Abschnitt 5.2	-	Breitengrad des Routenzielpunkts. Dieser kann über das location.name- oder location.nearbystops- Service ermittelt oder innerhalb von Österreich frei gewählt werden.
<b>via</b>	Optional	viaId wait time products	-	<p>Komplexe Struktur für die Definition von mehreren VIA und VIA Parametern. Die Via werden mit ; (semicolon) getrennt.</p> <p>Siehe <b>Abschnitt 11.7</b>.</p> <p>Der Aufbau sieht wie folgt aus: viaId waittime products</p> <p>viald: → erforderlich</p> <ul style="list-style-type: none"><li>- Id oder extId des VIA; die ID kann über das location.name- oder location.nearbystops- Service erhalten werden.</li><li>- Koordinaten (nur im IV) wie folgt definiert werden: Via=geo:x,y</li></ul>



				<p>z.B. Via=geo: 14.667915,47.169457</p> <p>– Angabe von mehreren Koordinaten: Via=geo:14.397096,48.042723;geo:14.662773,47.17476;geo:16.573277,48.576044</p> <p>waittime: → optional Zeitverbleib im VIA- Halt in Minuten</p> <p>products: → optional Produktklassen (nur bei ÖV)</p> <p>Beispiel 1: Drei VIA die durchfahren werden sollen durch extID festgelegt: via=801234;801235;801236</p> <p>Beispiel 2: Zwei VIA mit 10min und 20min Wartezeit: via=801234 10;801235 20</p>
<b>viaId</b>	Optional	ID	-	<p>ID einer/s Station/Haltes genutzt als VIA- Punkt der Route. Eine VIA- Station zwingt die Routensuche nach allen Routen die über diese Station führen zu suchen (siehe Abschnitt 11.7)</p> <p>Wird via genutzt, sind viaId und viaWaitTime wirkungslos</p>
<b>viaWaitTime</b>	Optional	siehe Abschnitt 11.7	0	<p>Legt die Verweilzeit in der Station (in Minuten) fest.</p> <p>Wird via genutzt, sind viaId und viaWaitTime wirkungslos.</p>
<b>minChangeTime</b>	Optional	siehe Abschnitt 5.3	-	<p>Minimale Umstiegszeit in Minuten.</p>
<b>maxChangeTime</b>	Optional	siehe Abschnitt 5.3	-	<p>Maximale Umstiegszeit in Minuten.</p>
<b>addChangeTime</b>	Optional	siehe Abschnitt 5.3	-	<p>Zeitwert, welcher der Umstiegszeit bei jedem Halt zugerechnet wird.</p>
<b>maxChange</b>	Optional	0 – 11	-	<p>Maximale Anzahl der Umstiege</p>
<b>date</b>	Optional	siehe Abschnitt 5.3	Aktuelles Server-Datum	<p>Setzt das Datum für welches Abfahrten gesucht werden im Format YYYY-MM-DD (Abfahrtsdatum).</p>
<b>time</b>	Optional	siehe Abschnitt 5.3	Aktuelle Server-Zeit	<p>Setzt die Startzeit für welche Abfahrten gesucht werden (Abfahrtszeit)</p>



<b>searchForArrival</b>	Optional	0 / 1	0	Wenn gesetzt, definieren <code>date</code> und <code>time</code> die Ankunfts- nicht die Abfahrtszeit.
<b>numF</b>	Optional	1 – 6	5	Mindestanzahl von Fahrten nach der Suchzeit. Summe von <code>numF</code> und <code>numB</code> muss $\leq 6$ betragen (also maximal 6).  siehe Abschnitt 11.3.3
<b>numB</b>	Optional	0 - 3	0	Mindestanzahl von Fahrten vor der Suchzeit. Summe von <code>numF</code> und <code>numB</code> muss $\leq 6$ betragen (also maximal 6).  siehe Abschnitt 11.3.3
<b>products</b>	Optional	-	-	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (siehe Abschnitt 5.6) in der Suche. Der Dezimalwert entspricht $2^{\text{Produktklasse}}$ . Für Kombinationen von Produktklassen ist die Summe zu bilden, wie folgt:  Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bitmaske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$
<b>categories</b>	Optional	Alle <code>catOut</code> Kategorien, die im <code>data.info</code> service bei dem Produkt verfügbar sind	-	Filterung der Verbindungen über Kategorien von Verkehrsmitteln und Betreiber.  Verbindungen mit CAT mit <code>categories=CAT</code>  Verbindungen unter Ausschluss von CAT <code>categories!=CAT</code>
<b>context</b>	Optional	Werte von <code>scrB</code> , <code>scrF</code> aus Trip Response	-	Definiert den Startpunkt für Vor- oder Rückwärtsblättern (Scrolling siehe Abschnitt 11.3.4)  Mit dem <code>scrB</code> - Wert eines früheren Ergebnisses kann in der Zeit zurückgeblättert werden und mit <code>scrF</code> kann vorwärts geblättert werden.



<b>poly</b>	Optional	0 / 1	0	Aktiviert/Deaktiviert die Polylinie für jede Etappe (leg) der Route.
<b>polyEnc</b>	Optional	GPA	-	Aktiviert die Ausgabe der Polylinie im Format „Google Encoded Polyline Format“.
<b>passlist</b>	Optional	0 / 1	0	Aktiviert (1)/Deaktiviert (0) die Ausgabe aller Haltestellen (Stopps) zwischen Start und Ziel der gewählten Verbindung für jede Etappe (leg) der Route.
<b>showPassingPoints</b>	Optional	0 / 1	0	<p>Passlist muss aktiv (passlist = 1) gesetzt sein.</p> <p>Aktiviert (1)/Deaktiviert (0) die Ausgabe <i>aller</i> passierten Haltestellen <i>ohne Zu- und Ausstiegsmöglichkeiten</i> innerhalb der Passlist zwischen Start und Ziel der gewählten Verbindung für jede Etappe (leg) der Route.</p>
<b>originWalk</b>	Optional	0 / 1	1	<p>Aktiviert (originWalk=1) /Deaktiviert (originWalk=0) den Fußvorlauf am Beginn einer Route bei der Adresssuche (nicht von Haltestellen).</p> <p>Der Default-Wert für den Fußvorlauf beträgt 0m (Minimalwert) bis 2000m für den Maximalwert.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station, können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiele: Fußvorlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet <code>originwalk=1,0,1000</code></p>
<b>originBike</b>	Optional	0 / 1	-	<p>Aktiviert (originBike=1) /Deaktiviert (originBike=0) den Radvorlauf bei Adresssuche.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station, können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiele: Radvorlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m →</p>



				Parameter lautet <code>originbike=1,0,1000</code>
<b>originCar</b>	Optional	0 / 1	-	<p>Aktiviert (<code>originCar=1</code>) /Deaktiviert (<code>originCar=0</code>) den Vorlauf mit Auto bei einer Abfrage von Adressen.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station, können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiel: Autovorlauf aktiv, minimale Distanz 2000m, maximale Distanz 100km → Parameter lautet <code>originCar=1,2000,100000</code></p>
<b>originPark</b>	Optional	0 / 1	-	<p>Zur Kalibrierung des Park&amp;Ride-Vorlaufes bei Auto-Routen (<code>groupFilter API_PARK_AND_RIDE</code>).</p> <p>Aktiviert (<code>originPark=1</code>) /Deaktiviert (<code>originPark=0</code>) Park &amp; Ride- Vorlauf für die Parameterübergabe.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten P+R- Station, können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiel: Park &amp; Ride- Vorlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet <code>originPark=1,0,1000</code></p>
<b>originMeta</b>	Optional	siehe Abschnitt 11.5	-	<p>Aktiviert (<code>originMeta=1</code>) /Deaktiviert (<code>originMeta=0</code>) die Verwendung von vordefinierten IV-Profilen <code>&lt;metaprofil&gt;</code> <b>im Vorlauf der Strecke.</b></p> <p><code>originMeta=&lt;metaprofil&gt;</code> Mehrere Meta-Profile können mit , (Beistrich) getrennt mitgegeben werden.</p>
<b>destWalk</b>	Optional	0 / 1	1	<p>Aktiviert (<code>destWalk=1</code>) /Deaktiviert (<code>destWalk=0</code>) den Fußweg im Nachlauf bei Verbindungssuche.</p> <p>Der Default-Wert für den Fußvorlauf beträgt 2000m. Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station,</p>





				<p>können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiel:</p> <p>Fußwegnachlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet <code>destWalk=1,0,1000</code></p> <p>Soll die Default- Distanz verwendet werden, keinen Wert setzen, z.B. <code>destWalk=1,,1500</code>.</p>
<b>destBike</b>	Optional	0 / 1	-	<p>Aktiviert (<code>destBike=1</code>) /Deaktiviert (<code>destBike=0</code>) den Radnachlauf bei Adresssuche für die Parameterübergabe.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station, können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiel:</p> <p>Radnachlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet <code>destBike=1,0,1000</code></p>
<b>destCar</b>	Optional	0 / 1-	-	<p>Aktiviert (<code>destCar=1</code>) /Deaktiviert (<code>destCar=0</code>) den Autonachlauf bei Adresssuche.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station, können diese Werte (in Metern) mit Komma getrennt hinter 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiele:</p> <p>Autonachlauf aktiv, minimale Distanz 2000m, maximale Distanz 100km → Parameter lautet <code>destCar=1,2000,100000</code></p>
<b>destPark</b>	Optional	0 / 1	-	<p>Aktiviert (<code>destPark=1</code>) /Deaktiviert (<code>destPark=0</code>) den Park &amp; Ride-Nachlauf bei Adresssuche.</p> <p>Für Feintuning der minimalen- oder maximalen Entfernung zur nächsten ÖV- Station, können diese Werte (in Metern) mit Komma getrennt hinter</p>



				<p>1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiel:</p> <p>Park &amp; Ride- Nachlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet <code>destPark=1,0,1000</code></p>
<b>destMeta</b>	Optional	siehe Abschnitt 11.5	-	<p>Aktiviert (<code>destMeta=1</code>) /Deaktiviert (<code>destMeta=0</code>) die Verwendung von vordefinierten IV-Profilen <code>&lt;metaprofil&gt;</code> <b>im Nachlauf der Strecke.</b></p> <p><code>destMeta=&lt;metaprofil&gt;</code></p> <p>Mehrere Meta-Profile können mit , (Beistrich) getrennt mitgegeben werden.</p>
<b>totalMeta</b>	Optional	siehe Abschnitt 11.5	-	<p>Aktiviert (<code>totalMeta=1</code>) /Deaktiviert (<code>totalMeta=0</code>) die Verwendung von vordefinierten IV-Profilen <code>&lt;metaprofil&gt;</code> <b>im Verlauf der gesamten Strecke.</b></p> <p><code>totalMeta=&lt;metaprofil&gt;</code></p> <p>Mehrere Meta-Profile können mit , (Beistrich) getrennt mitgegeben werden.</p>
<b>totalCar</b>	Optional	siehe Abschnitt 11.5		<p>Aktiviert (<code>totalCar=1</code>)/Deaktiviert (<code>totalCar=0</code>) Autorouten für die gesamte Route und lässt die Konfiguration des IV Router zu</p>
<b>bikeCarriage</b>	Optional	0 / 1	0	<p>Aktiviert (<code>bikeCarriage=1</code>) /Deaktiviert (<code>bikeCarriage=0</code>) die Suche nach Routen, welche ausdrücklich Fahrradtransport (Fahrradmitnahme) erlauben.</p>
<b>rtMode</b>	Optional	OFF, SERVER_DEFAULT	SERVER_DEFAULT	<p>Aktiviert (1)/Deaktiviert (0) den Realtime-Modus bei der <b>ÖV Routenberechnung.</b></p> <p><code>rtMode=OFF</code></p> <p>Verbindungen werden auf Basis der Plandaten berechnet. Echtzeitinformationen werden ignoriert (RT- Info).</p> <p><code>rtMode=SERVER_DEFAULT</code></p> <p>Echtzeitinformationen werden in die Routenberechnung miteinbezogen.</p>



<b>unsharp</b>	Optional	0 / 1	0	<p>Veraltet/aufgelassen</p> <p>da nun standardmäßig eine unscharfe Suche bei der Routenberechnung angewendet wird. Auch umliegende Stationen um den Start- und Zielpunkt werden mit einbezogen.</p>
<b>economic</b>	Optional	0 / 1	0	<p>Aktiviert (economic=1) den „ökonomischen“ Suchmodus. Es werden daher auch langsame Direkt-Verbindungen ausgegeben.</p>
<b>groupFilter</b>	Optional	siehe Abschnitt 11.2	-	<p>Bestimmt das Verkehrsmittel oder die intermodale Kombination von Verkehrsmittel für welche die Routenberechnung erfolgt. Ohne setzen dieses Parameters werden <b>ÖV Routen</b> (mit allen Produktklassen) ausgegeben.</p> <p>z.B. API_OEV, API_CAR</p>
<b>mobilityProfile</b>	Optional	baimAvoid Stairs, baimAvoid Escalator baimAvoid Elevator		<p>Ermöglicht die Verwendung von vordefinierten Mobilitätsprofilen. Nur für jene Haltestellen nutzbar, für die diese stationsbasierten Informationen bei den Haltestellendaten gepflegt sind.</p> <p>Folgende Mobilitätsprofile stehen zur Verfügung:</p> <p>Vermeiden von Treppen mobilityProfile=baimAvoidStairs</p> <p>Vermeiden von Rolltreppen mobilityProfile=baimAvoidEscalator</p> <p>Vermeiden von Aufzügen mobilityProfile=baimAvoidElevator</p> <p>Eine Kombination der Mobilitätsprofile ist mit einer Komma-separierten Liste als Eingabe möglich.</p>
<b>includeDrt</b>	Optional	0/1	1	<p>Ermöglicht es liniengebundene Bedarfsverkehre aus der Routensuche zu exkludieren mit includeDrt=0.</p>



<b>tariff</b>	Optional	0/1	1	Aktiviert (tariff=1) /Deaktiviert (tariff=0) die Ausgabe von Tariffinformationen.
---------------	----------	-----	---	---

## 11.2 Modalitäten und Kombinationen (groupFilter)

Ein `groupFilter` ermöglicht die *zusätzliche Berücksichtigung von Modalitäten* in einer *intermodalen* Verbindungssuche oder die *alleinige Berücksichtigung einer Modalität* in einer *monomodeln* Verbindungssuche. Der `groupFilter API_OEV` entspricht dem Default-Anfragemodus.

Generell gilt, dass die Verwendung eines `groupFilters` allein keine Garantie dafür ist, dass die ausgegebene Verbindung den gewählten Betreiber enthält. Die Routinganfrage wird mit dieser Modalität *erweitert*, aber nicht darauf eingeschränkt. Der Optimierungsalgorithmus schließt *alle Modalitäten* mit ein und berechnet daraus die optimale Verbindung. Dies kann, muss aber nicht auf den ausgewählten Betreiber fallen, sondern eröffnet die Option dafür.

Bei einem *intermodalen* Anfragemodus wird die Verbindungssuche um die gewünschte Modalität erweitert und mit ÖV und allen anderen Modalitäten kombiniert. Bei diesem Abfragemodus steht die Modalität in einem bestimmten Lauf (Leg) (bei Bike & Ride, Autoreisezug, Park&Ride), im Falle der Sharingangebote können diese im Vorlauf, im Hauptlauf oder im Nachlauf stehen und werden mit anderen Modalitäten kombiniert.

Bei einem *monomodalen* Anfragemodus wird eine Verbindung mit der gewünschten Modalität gesucht. Fußweg-Vorlauf und -nachlauf ist in der Response enthalten, sofern nicht von Haltestelle zu Haltestelle angefragt wird.

Wird *keine Verbindung gefunden*, wird kein Ergebnis ausgegeben. Wird keine Verbindung mit einem bestimmten `groupFilter` gefunden, aber eine ÖV Verbindung, so wird die ÖV Verbindung ausgegeben.

Der Routingalgorithmus ist ein Optimierungsalgorithmus, der für jede Verbindungsanfrage eine ÖV Verbindung berechnet und prüft. Sofern diese ÖV Verbindung schneller ist, wird sie gegebenenfalls. zusätzlich zur angefragten Route mit einem bestimmten `groupFilter` ausgegeben. Die Wahrscheinlichkeit, dass eine ÖV Verbindung effizienter ist, ist besonders hoch, wenn Haltestellen als Start und/oder Ziel verwendet werden. Bei Abfragen von Adressen/Koordinaten, die nicht direkt an Haltestellen liegen, ist die Wahrscheinlichkeit, dass es eine schnellere ÖV-Verbindung gibt, geringer.

Zusätzlich ist für ÖV-Abfragen `groupFilter (API_OEV)` konfiguriert, dass im Falle eines effizienteren Fußwegs dieser als alternative Fußwegverbindung, ausgegeben wird. Dies tritt ein, wenn die Luftliniendistanz zwischen Start und Ziel weniger als 1,5 km beträgt. Es wird also ein zusätzlicher Fußweg in der Modalität ÖV ausgegeben. In den anderen Modi PKW und Fahrrad, wird kein alternativer Fußweg vorgeschlagen.

Bei den `groupFilter` für Sharing-Angebote sind Betriebszeiten der Dienste zu beachten. Es können beispielsweise saisonale Betriebszeiten bestehen. In dem Fall können keine Routen außerhalb der Betriebssaison gefunden werden.

Die `groupFilter` für die unterschiedlichen Fahrrad-Typen steht nur für die Produktvariante „Radrouter“ und in Tirol zu Verfügung: <https://radrouting.verkehrsauskunft.at/vao/restproxy/<version>>.

Hier sind bestimmte Rad-Profile (Geschwindigkeit, Berücksichtigung von Steigung, etc.) beim Routing hinterlegt, die dem gewählten Radtyp entsprechen.



Modalität	groupFilter	Anfragemodus
<b>Öffentlicher Verkehr</b> monomodal	API_OEV	monomodale Modus sucht eine direkte ÖV Route mit allen Produktklassen
<b>Auto</b> monomodal	API_CAR	monomodaler Modus sucht eine direkte Auto-Route (IV)
<b>e-Auto</b> monomodal	API_ECAR	monomodaler Modus sucht eine direkte E-Auto Route (IV)
<b>Fußweg</b> monomodal	API_WALK	monomodaler Modus sucht eine direkte Fußwegroute (IV)
<b>Fahrrad</b> monomodal	API_BIKE	monomodaler Modus sucht eine direkte Fahrrad-Route (IV) mit Alltagsrad-Profil
<b>Bike &amp; Ride</b> intermodal	API_BIKE_AND_RIDE	intermodaler Modus sucht eine Route bestehend aus einem Fahrradvorlauf sowie einer ÖV-Verbindung mit <i>Fahrradabstellplatz an der Haltestelle</i> im Hauptlauf und einem Fußwegnachlauf.
<b>Bike &amp; Ride mit VMobil Radboxen</b> intermodal	API_VMOBIL_AND_RIDE	intermodaler Modus sucht eine Route bestehend aus einem Fahrradvorlauf bis zu einer <i>VMOBIL Box</i> sowie einer ÖV-Verbindung im Hauptlauf und einem Fußwegnachlauf.
<b>Fahrrad-mitnahme</b> intermodal	API_BIKECARRIAGE	intermodaler Modus sucht eine Route bestehend aus einem Fahrradvorlauf sowie einer ÖV-Verbindung mit Fahrradbeförderung im Hauptlauf und einem Fahrradnachlauf.
<b>Park &amp; Ride</b> intermodal	API_PARK_AND_RIDE	intermodaler Modus sucht eine Route bestehend aus einer Auto-Route zu einer P&R Anlage im Vorlauf, einer ÖV-Route im Hauptlauf und einem Fußwegnachlauf.
<b>Autoreisezug</b> intermodal	API_CARTRAIN	intermodaler Modus sucht eine Route bestehend aus einem Autovorlauf, einem Autoreiszug im Hauptlauf und einem Autonachlauf.
<b>groupFilter BIKE Sharing Provider (oft lokal begrenzt Angebote, nicht immer flächendeckend)</b>		
<b>NEXTBIKE</b> monomodal	API_BIKESHR_NEXTBIKE_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem Leihfahrrad des Betreibers NEXTBIKE. <b>Enthält komplette Nextbike-Abdeckung exklusive WienMobilRad und CityBike Linz (siehe groupFilter CityBike)</b>
<b>NEXTBIKE</b> intermodal	API_BIKESHR_NEXTBIKE	intermodaler Modus sucht eine Route mit einem stationsbasierten Leihfahrrad des Betreibers NEXTBIKE und ÖV/anderer Modalität.
<b>CITYBIKE</b> monomodal	API_BIKESHR_CITYBIKE_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem Leihfahrrad des Betreibers <b>WienMobil Rad (ehemals CityBikes) und CityBike Linz.</b>
<b>CITYBIKE</b> intermodal	API_BIKESHR_CITYBIKE	intermodaler Modus sucht eine kombinierte Route mit einem stationsbasierten Leihfahrrad des Betreibers CITYBIKE und ÖV/anderer Modalität. <b>WienMobil Rad (ehemals CityBikes) und CityBike Linz.</b>
<b>*LIME Bikes</b> monomodal	API_BIKESH_LIME_ONLY	monomodaler Modus sucht eine Direktroute mit einem Leihfahrrad des Betreibers LIME.



<b>*LIME Bikes</b> intermodal	API_BIKESH_LIME	intermodaler Modus sucht eine kombinierte Route mit einem Leihfahrrad des Betreibers LIME und ÖV/anderer Modalität.
<b>groupFilter SCOOTER Sharing Provider (oft lokal begrenzt Angebote, nicht immer flächendeckend)</b>		
<del>EASYWAY Roller</del> monomodal	<del>API_SCOOTERSHR_EASYWAY_ONLY</del>	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem E-Roller des Betreibers EASYWAY.
<del>EASYWAY Roller</del> intermodal	<del>API_SCOOTERSHR_EASYWAY</del>	intermodaler Modus sucht eine kombinierte Route mit einem stationsbasiertem E-Roller des Betreibers EASYWAY und ÖV/anderer Modalität.
<b>TIER Scooter</b> monomodal	API_SCOOTERSHR_TIER_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem E-Scooter des Betreibers TIER.
<b>TIER Scooter</b> intermodal	API_SCOOTERSHR_TIER	intermodaler Modus sucht eine kombinierte Route mit einem stationsbasiertem E-Scooter des Betreibers TIER und ÖV/anderer Modalität.
<b>*LIME Scooter</b> monomodal	API_SCOOTERSH_LIME_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem E-Scooter des Betreibers LIME.
<b>*LIME Scooter</b> intermodal	API_SCOOTERSH_LIME	intermodaler Modus sucht eine kombinierte Route mit einem stationsbasiertem E-Scooter des Betreibers LIME und ÖV/anderer Modalität.
<b>groupFilter CAR Sharing Provider (oft lokal begrenzt Angebote, nicht immer flächendeckend)</b>		
<b>CARUSO</b> monomodal	API_CARUSOSHR_ONLY	monomodaler Modus sucht eine Direktroute mit einem Carsharing-Fahrzeug des Betreibers CARUSO.
<b>CARUSO</b> intermodal	API_CARUSOSHR	intermodaler Modus sucht eine kombinierte Route mit einem Carsharing-Fahrzeug des Betreibers CARUSO und ÖV/anderer Modalität.
<b>FLOMOBIL</b> monomodal	API_CARSHR_FLOMOBIL_ONLY	monomodaler Modus sucht eine Direktroute mit einem Carsharing-Fahrzeug des Betreibers FLOMOBIL.
<b>FLOMOBIL</b> intermodal	API_CARSHR_FLOMOBIL	intermodaler Modus sucht eine kombinierte Route mit einem Carsharing-Fahrzeug des Betreibers FLOMOBIL und ÖV/anderer Modalität.
<b>WienMobil Auto</b> monomodal	API_CARSHR_WIENMOBIL_ONLY	monomodaler Modus sucht eine Direktroute mit einem Carsharing-Fahrzeug des Betreibers WienMobil Auto.
<b>WienMobil Auto</b> intermodal	API_CARSHR_WIENMOBIL	intermodaler Modus sucht eine kombinierte Route mit einem Carsharing-Fahrzeug des Betreibers WienMobil Auto und ÖV/anderer Modalität.
<b>Sharetoo</b> intermodal	API_CARSHR_SHARETOO_ONLY	monomodaler Modus sucht eine Direktroute mit einem Carsharing-Fahrzeug des Betreibers Sharetoo.
<b>Sharetoo</b> intermodal	API_CARSHR_SHARETOO	intermodaler Modus sucht eine kombinierte Route mit einem Carsharing-Fahrzeug des Betreibers Sharetoo und ÖV/anderer Modalität.
<b>groupFilter Radrouter</b>		
<b>Alltagsrad</b>	API_BR_NORMALBIKE	Spezielle Rad-Typen spezifische Eigenschaften werden im Routing berücksichtigt.
<b>Mountainbike</b>	API_BR_MOUNTAINBIKE	
<b>Rennrad</b>	API_BR_RACINGBIKE	
<b>Radwandern</b>	API_BR_TOURINGBIKE	



\* **LIME:** Da es auf Basis von vertraglichen Vorgaben in den Nutzungsbedingungen der Sharing-Anbieter zu Einschränkungen bei der Bereitstellung von Sharing-Diensten kommen kann steht LIME derzeit nur einen eingeschränkten Kundenkreis zur Verfügung.

#### Beispiel für eine Abfrage mit monomodalen PKW Modus

[https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/trip?accessId={{accessId}}&originExtId=490118400&destCoordLat=48.199126&destCoordLong=16.368503&groupFilter=API\\_CAR](https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/trip?accessId={{accessId}}&originExtId=490118400&destCoordLat=48.199126&destCoordLong=16.368503&groupFilter=API_CAR)

#### Beispiel für eine Abfrage mit intermodalen Modus für Autoreisezug

[https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/trip?accessId={{accessId}}&format=json&originExtId=460304000&destExtId=480013000&groupFilter=API\\_CARTRAIN](https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/trip?accessId={{accessId}}&format=json&originExtId=460304000&destExtId=480013000&groupFilter=API_CARTRAIN)

#### Beispiel für eine Anfrage mit intermodalen Modus für Bike&Ride (VMOBIL)

[https://demo.hafas.de/openapi/vao/restproxy/trip?accessId=hacon&format=json&originExtId=980066089&destExtId=480045200&groupFilter=API\\_VMOBIL\\_OEV\\_WALK](https://demo.hafas.de/openapi/vao/restproxy/trip?accessId=hacon&format=json&originExtId=980066089&destExtId=480045200&groupFilter=API_VMOBIL_OEV_WALK)

#### Beispiel für eine Abfrage an der Produktvariante Radrouter

[https://radrouting.verkehrsauskunft.at/vao/restproxy/{{version}}/trip?accessId={{accessId}}&originExtId=490118400&destCoordLat=48.199126&destCoordLong=16.368503&groupFilter=API\\_BR\\_TOURINGBIKE](https://radrouting.verkehrsauskunft.at/vao/restproxy/{{version}}/trip?accessId={{accessId}}&originExtId=490118400&destCoordLat=48.199126&destCoordLong=16.368503&groupFilter=API_BR_TOURINGBIKE)

## 11.3 Routen-Suchalgorithmus für den ÖV

### 11.3.1 Konfiguration der Routensuche

Der Standardsuchmodus der ÖV Verbindungssuche liefert die schnellsten und bequemsten Fahrten (möglichst wenige Umsteige). Für eine gegebene Abfahrtszeit, wird immer die schnellste Verbindung errechnet. Wenn diese keine Direktverbindung ist, werden ebenfalls sogenannte „bequeme“ Verbindungen gesucht. Diese stellen Verbindungen mit geringerer Anzahl von Umstiegen dar, die nur geringfügig langsamer als die schnellste Route sind.

Mit dem „ökonomischen“ Suchmodus können mehr Suchvorgänge mit verschiedenen Auswertungsmethoden durchgeführt werden. Wird der Parameter `economic` aktiviert, so werden auch überdurchschnittlich gute Direktverbindungen (die langsamer sind als die Verbindungen aus der Standard-Suche) ausgegeben.

Der Algorithmus bezieht auch umliegende Stationen um den Start- und Zielpunkt mit ein (=unscharfe Suche). Für diese zusätzlichen Stationen werden die Fußvorläufe miteinbezogen.

### 11.3.2 Berücksichtigung von Echtzeit

Bei der Routenberechnung werden standardmäßig Echtzeitinformationen (Verspätungen, Gleiswechsel) einbezogen, sodass die Dauer der Reise/Fahrt die Verzögerungen beinhaltet. Eine genaue Beschreibung der Berücksichtigung der Echtzeit ist unter und Abschnitt **5.7 Echtzeitinformationen im ÖV** dokumentiert. Möchte man die Routen Auskunft für ÖV auf Basis von Plandaten erhalten, so kann man den Realtime-Modus bei der Routenberechnung über den Parameter `rtMode=OFF` deaktivieren.

### 11.3.3 Vorwärts- und Rückwärts-Suche (Parameter `numB` und `numF`)

Standardmäßig werden maximal 5 Routingvorschlägen ab dem festgelegten Suchzeitpunkt ausgegeben. Mit dem Parameter `numF` (Vorwärtsuche) kann die Anzahl der Suchergebnisse auf maximal 6 (`numF=6`) nach dem Suchzeitpunkt erhöht werden. Mit dem Parameter `numB` (Rückwärtssuche) kann die Anzahl der Suchergebnisse auf maximal 3 Ergebnisse vor dem Suchzeitpunkt (`numB=3`) erhöht werden. Die Kombination von `numB` und `numF` darf allerdings zusammen nicht größer als 6 sein. Diese Parameter dienen also dazu das zeitliche Suchfenster zu verschieben. So kann man beispielsweise mit `numB=3` und `numF=3` jeweils 3 Ergebnisse vor und nach dem Suchzeitpunkt berechnen lassen.





Zu beachten ist auch der Einfluss von `searchForArrival` welcher die Suchrichtung bestimmt. Die Wirkung der Parameter `numF` und `numB` hängt von der Suchrichtung ab. Wenn die Suchrichtung in der Vergangenheit liegt (Rückwärtssuche mit Parameter `searchForArrival=1`), wirkt der Parameter `numF` auf diese Suchrichtung ein. Das heißt, der Server gibt nicht weniger als die letzten  $n$  (`numF=n`) Verbindungen vor der angeforderten Ankunftszeit. Liegt die Suchrichtung in der Zukunft (Vorwärtssuche mit Parameter: `searchForArrival=0`), so liefert der Server mit dem gleichen Parameter (`numF=n`) nicht weniger als die nächsten  $n$  möglichen Verbindungen in der Zukunft. Entsprechendes gilt für den Parameter `numB`. Wenn `numB` größer ist als 0 (`numB=m`), gibt der Server nicht weniger als die mindestens  $m$  möglichen Verbindungen in der Vergangenheit zurück. Die Rückwärts-Suche mit den Parametern `numB` und `searchForArrival` ist generell komplexer und somit rechenintensiver und sollte somit eher vermieden werden.

#### 11.3.4 Scrolling bei der ÖV Routenberechnung (Parameter context)

Basierend auf einem bereits durchgeführten Trip Request können weitere Routingergebnisse für dieselbe Fahrt durch vorwärts-/rückwärtsblättern gesucht werden. Auf diese Weise kann ein Blättern in der Zeit vor und zurück implementiert werden. Das wird erreicht, durch Beibehaltung der ursprünglichen Request- Parameter und Festlegung eines Startpunktes für das Blättern mit Hilfe eines zusätzlichen Kontextparameter.

Jeder Trip-Response enthält zwei Attribute `scrB` (Startpunkt rückwärts) und `scrF` (Startpunkt vorwärts) im `TripList`-Element welches den Startpunkt für die Scroll-Operation („Vorwärts- und Rückwärtsblättern“) darstellt. Übernahme eines dieser Werte als Parameter `context` (Parameter in einer neuen Routensuche führt zur Ausgabe früherer oder späterer Verbindungen für diese Route.

### 11.4 Routen-Suchalgorithmus für den IV

#### 11.4.1 Konfiguration der Routen-Suche

Bei der Routenberechnung von IV-Routen (Auto, E-Auto, Fahrrad, Fuß) werden die VAO Standardkonfigurationen (per Default) verwendet. Es wird die schnellste Route gesucht.

Die VAO stellt zudem weitere vordefinierte Konfigurationen des IV-Router, sogenannte Meta-Profile, bereit. Diese Meta-Profile enthalten Anpassungen für den Router für Geschwindigkeiten (Gehen und Auto), Optimierung nach Zeit oder Distanz (schnellste oder kürzeste Route), bestimmte Anpassungen für Radfahrer, sowie oben genannte Aktivierung oder Deaktivierung von Verkehrsmeldungen und Kurzparkzoneninformation in der Auskunft. Diese Meta-Profile werden in [Abschnitt 11.4](#) beschrieben.

#### 11.4.2 Berücksichtigung von Echtzeit

Bei der Routenberechnung werden standardmäßig Echtzeitinformationen (Verkehrsmeldungen, Verspätungen) einbezogen, sodass die Dauer der Reise/Fahrt die Verzögerungen beinhaltet. Diese Dynamiken können per Parameter deaktiviert werden, sodass die Fahrt ohne Echtzeitinformationen ausgegeben wird. Um die Echtzeitinformationen bei einer IV Route zu *deaktivieren*, muss der Parameter `totalCar` verwendet werden, wie in [Abschnitt 11.3 Meta-Profile für IV Routing](#) beschrieben.

Es werden allerdings keine Verkehrsmeldungen und Kurzparkzoneninformationen per Default ausgegeben. Sofern das gewünscht ist, muss die Ausgabe aktiviert werden. Damit Verkehrsmeldungen und Kurzparkzoneninformationen ausgegeben werden, müssen diese im Parameter `totalCar` mit entsprechender Konfiguration *aktiviert* werden.

#### 11.4.3 Alternative Route

Im PKW Modus (`groupFilter=API_CAR`) wird eine oder maximal zwei Routen ausgegeben. Bei passenden Fällen wird eine Alternativroute berechnet.



Diese Route wird im Bereich GisRoute in den Notes mit 0 (Route) oder 1 (Alternativroute) markiert. Die Note ist mit dem Key „ALTERNATIVE\_ROUTE\_INDEX“ auffindbar und enthält im Attribut txtN den Wert 0 oder 1.

Beispiel für die Markierung einer Alternativroute:

```
<Note key="ALTERNATIVE_ROUTE_INDEX" type="A" txtN="1">1</Note>
```

.

Da die Berechnung der Alternativroute auch Rechenzeit benötigt, kann diese deaktiviert werden (totalMeta=disableAlternativeRoutes), sodass immer nur eine Route ausgegeben wird (siehe Abschnitt 11.3 Meta-Profil für IV Routing). Das Deaktivieren der Alternativroute kann daher als Maßnahme gesetzt werden um die Antwortzeiten zu verkürzen.

## 11.5 Meta-Profil für IV Routing (Auto, Rad, E-Auto)

Dem IV-Router können bestimmte Konfigurationen für die IV Modi Gehen (groupFilter=API\_WALK), Auto (groupFilter=API\_CAR), Radfahren (groupFilter=API\_BIKE) mitgegeben werden um das Routing zu beeinflussen. Dazu werden die Parameter totalMeta, totalCar, totalWalk, totalBike verwendet. Soll das Routing im Vor- oder Nachlauf beeinflusst werden kann originWalk und destWalk verwendet werden.

### Deaktivieren aller Dynamiken (Echtzeitinformationen) für die Routenberechnung:

Per Default sind alle Dynamiken (Echtzeitinformationen) aktiviert, das heißt Verkehrsmeldungen (evnt), Umfahren von Sperren (aevnt) sowie die Verkehrslage (Echtzeitinformationen & Statistiken, wie Tagesganglinien) (tasta, htsta) werden per Default in das Routenergebnis einbezogen. Mit folgender Parameter-Pipe werden alle Dynamiken (Echtzeitinformationen) deaktiviert und damit nicht in das Routing einbezogen.

Beispiel:

```
totalCar=1|evnt=0,aevnt=0,tsta=0,htsta=0
```

### Aktivieren der Anbindung auf Autobahnen

Per Default ist das Setzen eines Start/Zielpunktes direkt auf die Autobahn nicht möglich, da bei Autobahnen nur an den Anschlussstellen auf-/abgefahren werden kann. Mit dieser Parameter-Pipe kann das Setzen eines Start-/Zielpunktes auf die Autobahn aktiviert werden.

Beispiel:

```
totalCar=1|allowFirstLastSarcHighwayConnection=1
```

### Deaktivieren der Parksuchzeiten

Per Default werden in der Berechnung von IV Routen Parksuchzeiten (derzeit im Raum Wien) mitberücksichtigt, die sich auf die Dauer einer Fahrt auswirkt. Sollen die Parksuchzeiten nicht berücksichtigt werden, müssen diese wie folgt deaktiviert werden:

Beispiel:

```
totalCar=1|getInitEndTimes=0
```

Es stehen noch eine Reihe von weiteren Meta-Profilen zur Verfügung, die eingesetzt werden können, um das Routing zu beeinflussen (siehe Tabelle).

Ab V1.8.0 stehen auch für den Modus E-Auto (groupFilter=API\_ECAR) weitere Konfigurationsmöglichkeiten zur Verfügung. Das E-Auto Routing befindet sich in einer österreichweiten Testphase und wird evaluiert.

### Verkehrsmeldungen aktivieren



Standardmäßig werden bei allen IV Modalitäten die IV Verkehrsmeldungen nicht ausgegeben werden. Diese müssen mit dem Parameter `totalMeta=event` aktiviert werden. Ist ein IV-Anteil in einer kombinierten Route vorhanden, wie bei Park&Ride, so müssen die Meldungen im Vorlauf also mit dem Parameter `originMeta=event` aktiviert werden.

#### Berechnung der Alternativroute deaktivieren

Bei passenden Fällen wird automatisch (per Default) eine Alternativroute berechnet. Da die Berechnung der Alternativroute auch Rechenzeit benötigt, kann diese deaktiviert werden.

Beispiel:

`totalMeta=disableAlternativeRoutes`

Meta-Profil für Modus Auto und Rad (groupFilter API_CAR und API_BIKE)					
Meta-Profil	Parameter	Single/ Multiple Choice	Meta-Profil / Konfiguration	Übergabewerte	Default
Meta-Profil für den Modus Auto (groupFilter=API_CAR)					
Kürzeste Route	totalMeta	single	car_shortest	--	Schnellste Route
Autogeschwindigkeit	totalMeta	single	car_speed_slow	--	normal
			car_speed_normal	--	
			car_speed_fast	--	
Autobahnen vermeiden	totalMeta	single	car_nomotorway		deaktiviert
Verkehrsmeldungen	totalMeta originMeta (bei Park&Ride)	single	event		deaktiviert
Kurzparkzonen-information	totalMeta	single	parkingzones		deaktiviert
Alternative Routen vermeiden	totalMeta	single	disableAlternativeRoutes	--	Alternative Routen werden berechnet
Berücksichtigung von Echtzeitinformationen in die Routenberechnung aktivieren/deaktivieren:					



<b>Berücksichtigung von Verkehrsmeldungen in der Routenberechnung</b>	totalCar	single	evnt (z.B. evnt=0)	0/1	aktiviert (1)
<b>Berücksichtigung von Sperren (Umfahren) in der Routenberechnung</b>	totalCar	single	aevnt (z.B. aevnt=0)	0/1	aktiviert (1)
<b>Berücksichtigung von Verkehrslage (Echtzeit &amp; Statistik) in der Routenberechnung (Dynamik 3)</b>	totalCar	single	htsta	0/1	aktiviert (1)
		single	(z.B. htsta=0)	0/1	aktiviert (1)
<b>Berücksichtigung aller Dynamiken (1-3) in der Routenberechnung</b>	totalCar	multiple	1 evnt=0,aevnt=0,tsta=0,htsta=0 siehe 4.4.1.5	0/1	aktiviert (1)
<b>Parksuchzeiten (in Wien)</b>	totalCar	single	1 getInitEndTimes=0	0/1	aktiviert (1)
<b>Anbindung auf Autobahnen in der Routenberechnung</b>	totalCar	single	1 allowFirstLastSarcHighwayConnection=1 siehe 4.4.1.5	0/1	deaktiviert (0)

Meta-Profil für den Modus Rad (groupFilter=API_WALK)					
Gehgeschwindigkeit	totalMeta originWalk destWalk	single	foot_speed_slow	--	normal
			foot_speed_normal	--	normal
			foot_speed_fast	--	normal
Meta-Profil für den Modus Rad (groupFilter=API_BIKE)					
Radgeschwindigkeit	totalMeta	single	bike_speed_slow (10km/h)	--	normal
			bike_speed_normal (15km/h)	--	normal
			bike_speed_fast (20km/h)	--	normal
			bike_speed_ebike (25km/h)	--	normal



<b>Steigungen vermeiden (Fahrrad)</b>	totalMeta	single	bike_noascents	--	deaktiviert
<b>Radinfrastruktur bevorzugen</b>	totalMeta	single	bike_preferbikeroute	--	deaktiviert
<b>Radschieben vermeiden</b>	totalMeta	single	bike_nopush	--	deaktiviert

## 11.6 Meta-Profil für E-Auto Routing

Die zu übergebenden Parameter für den IV Router werden als totalMeta Aufzählung übergeben. Die Werte sind alle nach dem Schema `ecar_Meta-Profil_Zahlenwert` vergeben.

Einfach-Auswahl (Single choice)-Parameter dürfen nur einmal übergeben werden, Mehrfach-Auswahl (multiple choice) Parameter müssen jeweils mit einem eigenen `ecar_XXX_XY` übergeben werden. Die Übergabe von Parametern, die Mehrfachauswahl (multiple choice) erlauben, lautet daher mit Beistrich ohne Leerzeichen in folgender Abfolge: z.B. `ecar_pam_cash,ecar_pam_app`

Es können nur definierte Zahlenwerte (siehe Übergabewerte) übergeben werden können. Wenn die Parameter nicht geschickt werden, werden die angeführten Defaultwerte verwendet.

### Wichtiger Hinweis:

Aktuell können bei der Berücksichtigung der Ladestellen-bezogenen Parametereingaben (PLT, PAM, GEN) *Routensuche gehäuft fehlschlagen*. Aufgrund der aktuell nicht vollständigen Datenverfügbarkeit (besonders PAM und GEN) an den Ladestellen, wird nicht empfohlen diese in Endkundenanwendungen zu integrieren (siehe Kapitel 5.11).

Meta-Profil für Modus E-Auto (groupFilter API_ECAR)					
Meta-Profil	Parameter	Single/ Multiple Choice	Meta-Profil / Konfiguration	Übergabewerte	Default
<b>Meta-Profil für den Modus E-Auto (groupFilter=API_ECAR)</b>					
<b>Fahrzeugtechnische Parameter</b>					
<b>Verbrauch in kWh/100km (EC)</b>	totalMeta	single	ecar_ec_x	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 30, 35	20
<b>Nutzbare Batterie-Kapazität in kWh (MC)</b>	totalMeta	single	ecar_mc_x	20, 25, 30, 35, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110	35



<b>Batterie-Ladestand bei Start in % (AC)</b>	totalMeta	single	ecar_ac_x	100, 90, 80, 70, 60, 50, 40, 30, 20	80
<b>Maximale Ladeleistung in kW (PC)</b>	totalMeta	single	ecar_pc_x	25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350	50
<b>Ladestellen-bezogene Parameter (nur eingeschränkt nutzbar *siehe wichtigen Hinweis)</b>					
<b>Steckertyp (PLT)</b>  (siehe Erläuterungstabelle)	totalMeta	multiple	ecar_plt_x	cccs1,cccs2,cg105,ctesla,ctype1,ctype2,s309-1p-16a,s309-1p-32a,s309-3p-16a,s309-3p-32a,scee-7-8,stype2,stype3	keiner
<b>Zahlungsmittel (PAM)</b>	totalMeta	multiple	ecar_pam_x	cash, rfid, credit_card, debit_card, app, sms, nfc, website	keiner
<b>Ökostrom nutzen (GEN)</b>	totalMeta	single	ecar_gen_true	true	keiner

Erläuterungen zu Metaprofil Steckertyp ecar_plt_x		
Übergabewert	Gängige Bezeichnung	Hinweis
CCCS2	CCS Typ2	Häufigste Steckertypen
CTYPE2	Typ2 (Kabel bei Ladestation verfügbar)	
STYPE2	Typ2 (kein Kabel bei Ladestation verfügbar)	
CG105	CHAdeMo	
SCEE-7-8	Schuko	
CTESLA	Tesla	
CCCS1	CCS Typ1	
CTYPE1	Typ1	
S309-1P-16A	CEE 3-polig (S309-1P-16A)	
S309-1P-32A	CEE 3-polig (S309-1P-32A)	
S309-3P-16A	CEE 5-polig (S309-3P-16A)	
S309-3P-32A	CEE 5-polig (S309-3P-32A)	
STYPE3	Typ3	

### 11.7 Via-Orte in der ÖV und IV Route

Bei der Routenberechnung kann für ÖV und IV auch ein oder mehrere Via Orte, also Zwischenorte oder Zwischenhalte (Haltestelle oder Adresse) die durchfahren werden sollen, mitberücksichtigt werden. Dabei ist



ein unterschiedliches Vorgehen, bei IV und ÖV Route als auch bei einem einigen und mehreren Via-Orten notwendig.

Die IDs für den Parameter `VIA` und `viaID` kann über Locations Service erhalten werden. Alle Ortspunkte haben in der Response ein Attribut `ID` oder `extID`, das entsprechend hier weiterverwendet werden muss.

### Via Orte in ÖV-Routen

Bei ÖV Routen wird die Haltestellen/Stations ID verwendet, die über das `location.name` oder `location.nearbystops` Service abgefragt wird. Sollten Koordinaten, Adressen oder POIs bei einer ÖV Route (`groupFilter=API_OEV`) definiert werden, so werden diese ignoriert.

Ein *einzig*er Via Ort wird über den Parameter `viaID` abgesetzt und zwingt die Routensuche nach allen Routen, die über diese Station oder Adresse führen, zu suchen. Über den Parameter `viaWaitTime` kann die Verweilzeit in dem Via (in Minuten) festgelegt werden.

Sollen *mehrere* Via-Orte mitgeben werden, so muss dies mit einer komplexen Struktur im Parameter `via` erfolgen. Der Aufbau der komplexen Struktur ist wie folgt: `viaId|waittime|viastatus|products`. Allerdings ist `viastatus` bei VAO nicht relevant, daher ist die Struktur folgendermaßen (ohne `viastatus`) mitzugeben: `viaId|waittime|products`

Über die Option `viaWaitTime` innerhalb der komplexen Struktur kann die Verweilzeit in der Station (in Minuten) festgelegt werden, sowie ein Verkehrsmittel, also die ÖV Produktklasse in der Option `products` festgelegt werden. Dies ist nur bei ÖV Routen wirksam.

Wird dieser Parameter `Via` verwendet, so werden die Parameter `viaID` und `viaWaitTime` hinfällig. Bei der Eingabe werden mehrere Vias mit ; (semicolon) getrennt.

Die Ausgabe der Via-Orte in der Response erfolgt in der "gewohnten" Struktur einer Trip-Response, also im Root Element `Trip/LegList/Leg/Joureny/Stops`. Der Via-Ort wird also als ein Stop bzw. Zwischenhalt in der Route ausgewiesen. In den Location Notes wird der Hinweis „Gewünschter Via-Ort erreicht“ ausgegeben.

```
<Note key="VIA_LOCATION" type="H" txtN="Gewünschter Via-Ort erreicht."/>
```

Es ist zu beachten ist, dass die Stops (Zwischenhalte) nur ausgegeben werden, wenn `passlist=true` gesetzt wurde. Ein Via kann auch ein Zwischenhalt sein, der durchfahren wird, an dem also kein Ein-/Aus-/Umstieg erfolgt. Wenn ohne `passlist=true` angefragt wird und der Via-Ort lediglich ein Zwischenhalt ist (Ein-/Aus-/Umstieg), taucht er in der Response gar nicht auf.

Wenn mit `poly=1` abgefragt wird (also die Polylinie aktiviert ist) wird für jedes Leg die zugehörige Polylinie geliefert. In einem (ggf. zwei) der Legs ist dann auch der Via-Halt enthalten.

### Via Orte in IV-Routen

Bei IV Routen können neben Adressen und POIs (die IDs sind ebenfalls über das `location.name`- oder `location.nearbystops`- Service abgefragt) zusätzlich auch Koordinaten als Via Ort verwendet werden. Diese werden in Form von `Via=geo:x,y` (z.B. `Via=geo: 14.667915,47.169457`) im Parameter `viaID` mitgegeben. Bei der Eingabe werden mehrere Koordinaten mit ; (semicolon) getrennt.

Wird ein Via gesetzt so enthält die Route den Hinweis in den Notes im Bereich `Trip` „Ihre Vias wurden bei der Routensuche berücksichtigt“.

```
<Note key="VIA_SUPPORTED" type="H" txtN="Ihre Vias wurden bei der Routensuche berücksichtigt."/>
```

Die erreichten Zwischenziele (Via Orte) werden im Bereich der `GisRoute/Seg` mit einem Hinweis „Gewünschter Via-Ort erreicht“ gekennzeichnet:

```
<Note key="VIA_LOCATION" type="H" txtN="Gewünschter Via-Ort erreicht."/>
```



Das Segment enthält am Via Ort den Manövertext „Ziel“ (`manTx=Ziel`) zu Kennzeichnung des Zwischenziels.

### 11.8 Interpretation der Hinweise in den Segmenten einer IV Route

Bei den Trip Responses für IV Routen werden in den Segment-Notes der jeweiligen Elemente zusätzliche relevante Informationen ausgegeben. Dieser Inhalt reicht von verfügbaren Kurzparkzonen beim Zielpunkt oder etwaigen Verkehrsmeldungen entlang der Route über attributive Merkmale an den Segmenten einer Route (Radschiebestrecken, Belagsinformationen, Radinfrastruktur)

Auf den einzelnen Teilabschnitten, den sogenannten Segmenten, im Element `GisRoute` können im untergeordneten Element `Notes` bestimmte Netzwerk-Attribute ausgegeben werden, die auf dem zugrundeliegenden Graphen hinterlegt sind. Derzeit können bei Radrouten Informationen über die jeweilige Radinfrastruktur und sofern verfügbar über den Straßen-Belag (derzeit nur für das Bundesland Tirol vorhanden) ausgegeben werden. Des Weiteren wird angegeben, ob es sich um eine Strecke handelt, bei der das Rad geschoben werden muss (Radschiebestrecke).

#### Radinfrastruktur:

Für Ausgaben die Radinfrastruktur betreffend gibt es derzeit folgende Werte:

BL1: Radweg  
BL2: Verkehrsberuhigte Zone  
BL3: Wohnstraße, Fußgängerzone  
BL4: Mehrzweckstreifen, Busspur, Radfahrstreifen  
BL5: Radempfehlung, Mountainbikeempfehlung, Hauptradroute

#### Straßenbelag:

Für Ausgaben den Straßenbelag betreffend gibt es derzeit folgende Werte, die mit `key="PAVEMENT"` versehen werden (diese Information ist derzeit nur für das Bundesland Tirol verfügbar):

PAVEMENT: PAVED  
PAVEMENT: UNPAVED  
PAVEMENT: COBBLESTONE  
PAVEMENT: TERRAIN  
PAVEMENT: UNKNOWN

#### Radschiebestrecken:

Abschnitte, auf denen mit dem Rad nicht gefahren werden darf und wo somit nur „Radschieben“ erlaubt ist, werden wie folgt deklariert:

```
<Note key="PUSHBIKE" type="A" txtS="Radschiebestrecke">Radschiebestrecke</Note>
```

## 12 GIS ROUTEN SERVICE

Das `gisroute` Service dient zur Abfrage der Details wie Polylines, Weg- und Distanzinformationen für Vor- und Nachlaufberechnungen.

Dem `gisroute` Request muss somit ein `Trip` Request vorangegangen sein, aus dem die `GisRef` entnommen wird und den GIS-Kontext (Parameter `ctx`) des `gisroute`-Requests eingesetzt wird.

Die Berechnung der Polylinie (Parameter `poly`) für jede Routenetappe kann deaktiviert werden. Die Response liefert standardmäßig die Polylinie in Form einer Koordinatenabfolge. Wird eine Ausgabe im Google Encoded Polyline Format (GPA) gewünscht, kann diese im Parameter `polyEnc` gesetzt werden.

## 12.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>ctx</b>	Erforderlich	-	-	Legt den GIS Routen Kontext fest. D.h. die entsprechende Definition den Vor- oder Nachlauf einer Route.
<b>poly</b>	Optional	0 / 1	0	Aktiviert/Deaktiviert die Berechnung der Polylinie für jede Routenetappe.
<b>polyEnc</b>	Optional	GPA	–	Aktiviert die Ausgabe der Polyline im Format „Google Encoded Polyline Format“.

### Beispiel:

[https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/gisroute?accessId=yourId&ctx=G|1|G@F|A=1@O=Wien%20Resselgasse%X=16367963@Y=48199000@U=81@L=490109405@|A=2@O=Resselgasse%202%20Schule,%201040%20Wieden%20\(Wien\)@l=@X=16368503@Y=48199126@u=0@U=103@L=980097798@|11072018|122800|122900|fb|ff@\\$f@\\$f@\\$f@\\$f@\\$t@2000@50000@120@-1@100@1@15000@0@@@@false@0@-1@\\$f@\\$C2%A7bt@0@2000@120@-1@50@1@1000@0@@@@false@0@-1@\\$f@\\$f@\\$f@\\$f@\\$t@2000@50000@120@-1@100@1@15000@0@@@@false@0@-1@\\$f@\\$C2%A7tf@\\$f@\\$f@\\$f@\\$f@\\$f@\\$f@\\$C2%A7|&poly=1](https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/gisroute?accessId=yourId&ctx=G|1|G@F|A=1@O=Wien%20Resselgasse%X=16367963@Y=48199000@U=81@L=490109405@|A=2@O=Resselgasse%202%20Schule,%201040%20Wieden%20(Wien)@l=@X=16368503@Y=48199126@u=0@U=103@L=980097798@|11072018|122800|122900|fb|ff@$f@$f@$f@$f@$t@2000@50000@120@-1@100@1@15000@0@@@@false@0@-1@$f@$C2%A7bt@0@2000@120@-1@50@1@1000@0@@@@false@0@-1@$f@$f@$f@$f@$t@2000@50000@120@-1@100@1@15000@0@@@@false@0@-1@$f@$C2%A7tf@$f@$f@$f@$f@$f@$f@$C2%A7|&poly=1)

## 13 DEPARTUREBOARD SERVICE

Die Abfahrtstafel gehört zu den Diensten für Stationsanzeigen. Das Service liefert eine Abfahrtstafel. Es werden die nächsten Abfahrten an einer gewählten Haltestelle ab einem bestimmten Zeitpunkt für eine bestimmte Zeitspanne ausgegeben. Die Default-Zeitspanne ist 60 Minuten.

Die Response enthält eine Liste von Abfahrten inklusive aller Informationen zu Zeiten, Gleisen (tracks), Echtzeitdaten und Fahrtverlauf (joureny). Es enthält auch die `JourenyDetailRef` um den gesamten Fahrtverlauf einer Line abzufragen.

Das Default-Verhalten des Depature Board ist so eingestellt, dass für eine Abfrage mit einer Haltestelle nicht nur die Fahrten der angefragten Haltestelle, sondern auch die Fahrten aller äquivalenten Steige und Haltestellen automatisch auch mitausgegeben werden (äquivalent bedeutet vereinfacht, dass in der Haltestellenmodellierung Verknüpfungen bzw. Zuordnungen zwischen Steigen und Haltestellen bestehen). Das Standardverhalten entspricht dem Parameter `type= DEP_EQUIVS`. Dies kommt auch zum Tragen, wenn mit einem Steig/Mast abgefragt wird: In der Response ist per Default nicht nur die Fahrten am angefragten Steig enthalten, sondern auch die Fahrten der äquivalenten Steige/Masten und Haltestellen. Möchte man dieses





Verhalten einschränken, kann man unter Nutzung des Parameter `type` auch nur mastspezifische Abfragen gestellt werden. Dies ist erst ab V.1.3.0 und höher möglich.

Standardmäßig wird der Request-Zeitpunkt, also der Zeitpunkt an dem die Abfrage gestellt wird, als Zeitpunkt herangezogen. Es wird die volle vorherige Minute herangezogen. Die Default- Anzeigedauer beträgt 60 Minuten.

Der Abfragezeitpunkt kann durch den Parameter `date` und `time` angepasst werden. Grundsätzlich kann in die Vergangenheit 62 Tage rückwärts vom aktuellen Datum abgefragt werden. In die Zukunft kann bis zum Ende der Fahrplanperiode oder bis zum nächsten Fahrplanwechsel abgefragt werden. Es gibt jedoch Fakten zur Berücksichtigen bei der Abfrage in der Zukunft und in der Vergangenheit. Siehe dazu [Abschnitt 5.5.3 Abfragen in die Zukunft und Vergangenheit](#).

Die Zeitspanne kann durch den Parameter `duration` definiert werden, wobei die maximale Spanne 24h beträgt. Da bei der Abfrage immer die volle vorherige Minute herangezogen wird, ist der Maximalwert der in `duration` eingesetzt werden kann 1439 Minuten. (Somit ist die Zeitspanne mit der vollen Minute genau 24h).

#### Wichtiger Hinweis:

Bei stark frequentierten Haltestellen ist es nicht ratsam eine lange Zeitspanne (`duration`) zu wählen, da die Responsezeiten so lange sein können, dass das API TimeOut-Schwelle überschritten wird. Um Abfragen mit einem längeren Horizont zu bewerkstelligen, ist empfohlen „Scrolling“ zu nutzen, indem die Abfragen gesplittet werden (siehe Abschnitt 13.2).

Die Response enthält eine Liste der Abfahrten im Root Element `Departure`. Dieses enthält die Abfahrts Haltestellen (`stop`), Abfahrtsdatum und -zeit (`time`) inklusive Echtzeit sowie den Richtungstext (abfahrende Richtung) über die Endhaltestelle der Fahrt (`direction`). Im Element `ProductAtStop` finden sich Informationen zur Beförderungsart (Kategorie des Verkehrsmittels) (`catOutL`), Liniennummer (`displayNumber`) und Betreiber (`operator`) vorausgesetzt diese Daten sind vom Verkehrsverbund in erfasst und geführt.

Jede Abfahrt enthält auch einen Verweis (`JourneyDetailRef`) auf den `JourneyDetailService` um weitere Details der Fahrt abzufragen.

Der `DepartureBoard` kann Haltestellen- oder Mast/Steig-spezifisch angefragt werden, je nachdem, ob mit einer Haltestelle oder einen Steig angefragt wird. Die Unterscheidungsmöglichkeit von Haltestelle und Steig ist in [Abschnitt 5.4.1 Hauptmasten und Masten \(IDs\)](#) beschrieben.

Wenn nur abfahrende Fahrten in eine bestimmte Richtung gesucht werden, ist der Parameter `direction` zu verwenden. Durch Angabe einer Haltestelle, die sich im weiteren Laufweg der Fahrt befindet, werden die abfahrenden Fahrten gefiltert. Dabei werden alle äquivalenten Haltestellen/Steige der Eingabe berücksichtigt.

### 13.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>id</b>	Erforderlich		-	Legt die Stations-/Halte ID des Startpunkts fest. Diese kann über das <code>location.name</code> - oder <code>location.nearbystops</code> - Service erhalten werden.  Ab Version 1.3.3 kann auch die <code>altId</code> für Haltestellen verwendet werden ( <a href="#">siehe Abschnitt 5.4.2</a> ).



<b>extId</b>	Erforderlich, wenn id nicht angegeben ist	-	-	Veraltet/aufgelassen da der Parameter id nun auch externe IDs unterstützt.
<b>direction</b>	Optional		-	Wenn nur abfahrende Fahrten in eine bestimmte Richtung gesucht werden. Definition der Richtung durch Angabe der Stations-/Halte ID eines Haltes in der Reisekette der Fahrt.
<b>date</b>	Optional	siehe Abschnitt 5.3	Aktuelles Serverdatum	Setzt das Abfahrtsdatum für die Suche im Format YYYY-MM-DD
<b>time</b>	Optional	siehe Abschnitt 5.3	Aktuelle Serverzeit	Setzt die Abfahrtszeit für die Suche im Format hh:mm in 24h Nomenklatur.
<b>duration</b>	Optional	0 / 1439	60	Setzt die Intervallzeit in Minuten
<b>dur</b>	Optional	<del>0 / 1439</del>	<del>60</del>	Veraltet/aufgelassen wird mit duration abgedeckt
<b>products</b>	Optional	-	-	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (siehe Abschnitt 5.6) in der Suche. Dieser Wert wird wie folgt berechnet:  Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bitmaske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$
<b>passlist</b>	Optional	0 / 1	0	Aktiviert die durchfahrenen Wegpunkte (Haltestellen), an denen nicht gehalten wird.
<b>lines</b>	Optional	-	-	Nur Fahrten mit der gewünschten Linie werden geliefert. Um mehrfache Linien zu filtern werden die Codes per Komma getrennt. Ein „!“ am Beginn exkludiert die Linie von der Reise.  Für ÖV Produktklasse 1 bis 11 anwendbar.  z.B. Filter für Linien 120 und 140 lines=120,140
<b>maxJourneys</b>	Optional	-	-	Maximale Anzahl von Fahrten im Ergebnis. Kein Wert liefert alle An-



				und Abfahrten innerhalb der gesetzten Zeitspanne.  Beachten: maxJourneys ist keine harte Grenze. Wenn der Wert erreicht wird und weitere Fahrten mit den denselben Abfahrts-/Ankunftszeiten innerhalb des Grenzwertes (z.B. 14:57) vorhanden sind, werden diese ebenfalls ausgegeben. Das stellt sicher, dass eine Zukunftssuche durch Ausführung eines weiteren Stationboard- Request mit gleicher Abfahrts-/Ankunftszeit erhöht um 1 (14:58 im Beispiel) funktioniert.
rtMode	Optional	OFF / SERVER_DEFAULT	SERVER_DEFAULT (aktiviert)	SERVER_DEFAULT= Echtzeit ist berücksichtigt (siehe Abschnitt 5.7)  OFF = Suche nur auf Basis der Plandaten (SOLL-Daten)
type	Optional	DEP_MAST, DEP_STATION, DEP_EQUIVS	DEP_EQUIVS	ab V1.3.0  DEP_MAST: DepartureBoard am Masten  DEP_STATION: DepartureBoard mit allen Fahrten an beliebigen Masten der angeforderten Station  DEP_EQUIVS: DepartureBoard mit allen Fahrten an beliebigen Masten <b>und</b> äquivalenten Haltestellen

#### Beispiel:

Request: Abfahrtstafel für Wien Schottentor am 12.02.2021 um 17:00h.

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/DepartureBoard?accessId={{accessId}}&format=xml&id=490118400&date=2021-02-12&time=17:00&duration=60>

#### Auszug aus der Response:

```
<Departure name="Straßenbahn 1" type="ST" stop="Wien Schottentor" stopid="A=1@0=Wien Schottentor@X=16362147@Y=48213725@U=81@L=490118403@" stopExtId="490118403" prognosisType="PROGNOSSED" time="16:38:00" date="2021-02-12" rtTime="17:53:00" rtDate="2021-02-12" reachable="true" direction="Wien Prater, Hauptallee" entry="true">

  <JourneyDetailRef ref="1|74130|0|81|12022021"/>
  <JourneyStatus>P</JourneyStatus>
  <ProductAtStop name="Straßenbahn 1" internalName="1" displayNumber="1" num="31313" line="1" lineId="vor-22-1-j21-40" catIn="v04" catCode="4" cls="16" catOutS="v04" catOutL="Straßenbahn" operatorCode="04" operator="Wiener Linien" admin="v04WL_" matchId="14125">
    <icon res="prod_tram" txtS="1">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundcolor r="237" g="28" b="36" hex="#ED1C24"/>
    </icon>
  </ProductAtStop>
```



```
</Product>
  <Notes>
    <Note key="LF" type="A" priority="50" routeIdxFrom="18" routeIdxTo="29" tx
tN="Niederflurfahrzeug">Niederflurfahrzeug</Note>
    <Note key="n0" type="A" priority="50" routeIdxFrom="18" routeIdxTo="29" tx
tN="barrierefrei">barrierefrei</Note>
  </Notes>
  <altId>at:49:1184:0:3</altId>
</Departure>
```

## 13.2 Scrolling

Um die Abfahrts tafeln durchzublättern, muss folgende Aktion durchgeführt werden: Die Abfahrtszeit der letzten Abfahrt in der Response wird herangezogen. Dann wird eine Minute addiert und die gleiche Abfrage mit dem neuen Zeitwert erneut abgesetzt. Wenn die Antwort über Mitternacht hinausgeht, muss auch das Datum erhöht werden.

Dies ist möglich, weil die Ergebnisliste immer die Abfahrten der letzten Minute enthält, auch wenn ein maxJourneys-Wert überschritten werden muss.

## 14 ARRIVALBOARD (ANKUNFTSTAFEL)

Die Service liefert eine Ankunfts tafel. Die Response enthält eine Liste der Ankünfte im Root Element `Arrival`. Dieses enthält die Ankunfts haltestellen (`stop`), Ankunftsdatum und -zeit (`time`) inklusive Echtzeit sowie den Richtungstext (kommend von Richtung) (`origin`) der Fahrt. Im Element `ProductAtStop` finden sich Informationen zur Beförderungsart (`catOutL`), Liniennummer (`displayNumber`), Betreiber (`operator`) falls datenseitig geführt.

Jede Ankunft enthält auch einen Verweis (`JourneyDetailRef`) auf den Journey Detail Service um weitere Details der Fahrt abzufragen.

Wenn nur ankommende Fahrten aus einer bestimmten Richtung gesucht werden, ist der Parameter `direction` zu verwenden. Durch Angabe einer Haltestelle, die sich im vorangegangenen Laufweg der Fahrt vor der angefragten Haltestelle befindet, werden die abfahrenden Fahrten gefiltert. Dabei werden alle äquivalenten Haltestellen/Steige der Eingabe berücksichtigt.

### Wichtiger Hinweis:

Bei stark frequentierten Haltestellen ist es nicht ratsam eine lange Zeitspanne (`duration`) zu wählen, da die Responsezeiten so lange sein können, dass das API TimeOut-Schwelle überschritten wird. Um Abfragen mit einem längeren Horizont zu bewerkstelligen, ist empfohlen „Scrolling“ zu nutzen, indem die Abfragen gesplittet werden (siehe Abschnitt 14.2).

### 14.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>id</b>	Erforderlich		-	Legt die Stations-/Halte ID des Startpunkts fest. Diese kann über das location.name- oder location.nearbystops- Service erhalten werden.



				Ab Version 1.3.3 kann auch die <code>altId</code> für Haltestellen verwendet werden (siehe Abschnitt 3.3).
<b>extId</b>	Erforderlich, wenn id nicht angegeben ist		-	Veraltet/aufgelassen da der Parameter <code>id</code> nun auch externe IDs unterstützt.
<b>direction</b>	Optional		-	Wenn nur ankommende Fahrten aus einer bestimmten Richtung gesucht werden. Definition der Richtung durch Angabe der Stations-/Halte ID eines Haltes in der Reisekette der Fahrt vor der abgefragten Haltestelle.
<b>date</b>	Optional	siehe Abschnitt 5.3	Aktuelles Serverdatum	Setzt das Ankunftsdatum für die Suche im Format YYYY-MM-DD
<b>time</b>	Optional	siehe Abschnitt 5.3	Aktuelle Serverzeit	Setzt die Ankunftszeit für die Suche im Format hh:mm in 24h Nomenklatur.
<b>duration</b>	Optional	0 / 1439	60	Setzt die Intervallzeit in Minuten
<b>dur</b>	Optional	0 / 1439	60	Veraltet/aufgelassen wird mit <code>duration</code> abgedeckt
<b>products</b>	Optional	-	-	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (siehe 0) in der Suche. Dieser Wert wird wie folgt berechnet:  Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bitmaske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$
<b>lines</b>	Optional	-	-	Nur Fahrten mit der gewünschten Linie werden geliefert. Um mehrfache Linien zu filtern werden die Codes per Komma getrennt. Ein „!“ am Beginn exkludiert die Linie von der Reise.  Für ÖV Produktklasse 1 bis 11 anwendbar. z.B. Filter für Linien 120 und 140 <code>lines=120,140</code>
<b>maxJourneys</b>	Optional	-	-	Maximale Anzahl von Fahrten im Ergebnis. Kein Wert liefert alle An- und Abfahrten innerhalb der gesetzten Zeitspanne.  Beachten: <code>maxJourneys</code> ist keine harte Grenze. Wenn der Wert erreicht wird und weitere Fahrten mit den denselben Abfahrts-/Ankunftszeiten innerhalb des Grenzwertes



				(z.B. 14:57) vorhanden sind, werden diese ebenfalls ausgegeben. Das stellt sicher, dass eine Zukunftssuche durch Ausführung eines weiteren Stationboard- Request mit gleicher Abfahrts-/Ankunftszeit erhöht um 1 (14:58 im Beispiel) funktioniert.
rtMode	Optional	SERVER_DEFAULT, OFF	SERVER_DEFAULT	Echtzeit wird in SERVER_DEFAULT berücksichtigt. Echtzeit wird mit OFF nicht berücksichtigt.
type	Optional	.	ARR, ARR_MAST, ARR_STATION ARR_EQUIVS	ab V1.3.0 ARR: Arrival board wie von HAFAS konfiguriert (Default) ARR_MAST: Arrival board am Masten ARR_STATION: Arrival board mit allen Fahrten an beliebigen Masten der angeforderten Station (entspricht ARR) ARR_EQUIVS: Arrival board mit allen Fahrten an beliebigen Masten und äquivalenten Haltestellen

**Beispiel:** Request: Ankunftstafel für Wien Schottentor am 12.02.2021 um 17:00h.

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/arrivalBoard?accessId={{accessId}}&format=xml&id=490118400&date=2021-02-12&time=17:00&duration=60>

#### Auszug aus der Response:

```
<Arrival name="Straßenbahn D" type="ST" stop="Wien Schottentor" stopid="A=1@O=Wien Schottentor@X=16362147@Y=48213725@U=81@L=490118403@" stopExtId="490118403" time="17:00:00" date="2021-02-12" reachable="true" origin="Wien Absberggasse" entry="true">
  <JourneyDetailRef ref="1|90717|1|81|12022021"/>
  <JourneyStatus>P</JourneyStatus>
  <ProductAtStop name="Straßenbahn D" internalName="D" displayNumber="D" num="108867" line="D" lineId="vor-22-D-j21-40" catIn="v04" catCode="4" cls="16" catOutS="v04" catOutL="Straßenbahn" operatorCode="04" operator="Wiener Linien" admin="v04WL_" matchId="00D5641">
    <icon res="prod_tram" txtS="D">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundcolor r="237" g="28" b="36" hex="#ED1C24"/>
    </icon>
  </ProductAtStop>
  <altId>at:49:1184:0:3</altId>
</Arrival>
```

## 14.2 Scrolling

Um die Ankunftstafeln durchzublätern, muss folgende Aktion durchgeführt werden: Die Ankunftszeit der letzten Abfahrt in der Response wird herangezogen. Dann wird eine Minute addiert und die gleiche Abfrage mit dem neuen Zeitwert erneut abgesetzt. Wenn die Antwort über Mitternacht hinausgeht, muss auch das Datum erhöht werden.



Dies ist möglich, weil die Ergebnisliste immer die Ankünfte der letzten Minute enthält, auch wenn ein `maxJourneys`-Wert überschritten werden muss.

## 15 JOURNEY DETAIL SERVICE

`JourneyDetail` Service ist dazu da die Verbindungsübersicht mit Start- und Zielhaltestelle aus dem `Trip` Request mit den Verbindungsdetails zu ergänzen.

Dieses Service liefert die Liniendetails zu einem bestimmten Abschnitt einer Fahrt. Die `id` zur Abfrage einer konkreten Fahrt, wird aus der Response eines `Trip`- oder `DepartureBoard` Requests erhalten. Im `Trip` Service besitzt jede Fahrt (jeder Abschnitt einer Verbindung mit einem Verkehrsmittel) eine `JourneyDetailRef`, also eine Referenz, über welche der *gesamte Linienverlauf* der Fahrt abgefragt werden kann. Der Linienverlauf enthält *alle* Haltestellen vom Start der Linie bis zur Endstation. Im Gegensatz dazu werden in der `Trip` Response nur die *Zwischenstops* in der konkreten Start-Ziel-Relation ausgegeben (mit dem Parameter `passPoints`).

Das Service liefert eine komplette Liste aller Haltestellen der Fahrt, inklusive Ankunfts- und Abfahrzeiten, Echtzeitinformationen und zusätzlichen Informationen an jedem Stopp. Mit Hilfe des Parameters `showPassingPoints` kann zusätzlich die Ausgabe von passierten Haltestellen erzwungen werden, an denen kein Ein- oder Aussteigen möglich ist.

### 15.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>id</b>	Erforderlich	-	-	<code>id</code> für die abzufragende Fahrt Ab Version 1.3.3 kann auch die <code>altId</code> für Haltestellen verwendet werden (siehe Abschnitt 3.3).
<b>date</b>	Optional	siehe Abschnitt 5.3	Aktuelles Serverdatum	Setzt das Ankunftsdatum für die Suche im Format YYYY-MM-DD
<b>poly</b>	Optional	0/1	0	Aktiviert die Berechnung einer Polylinie für jeden leg-Abschnitt der Reise
<b>polyEnc</b>	Optional	DLT,GPA,N	N	Legt die Kodierung der Polylinie fest. Werte: N (keine Kodierung), DLT (Veränderung zur vorherigen Koordinate) und GPA (Google encoded Polyline Format)
<b>showPassingPoint</b>	Optional	0/1	0	Aktiviert oder Deaktiviert die Anzeige von Halten, in welchen nicht zugestiegen oder ausgestiegen werden kann ( <code>PassingPoints</code> werden lediglich passiert, aber es wird nicht gehalten, oder ein Ein-und Aussteigen ist nicht möglich)



rtMode	Optional	OFF, SERVER- DEFAULT	SERVER- DEFAULT	Aktiviert oder Deaktiviert Echtzeitdaten für die Fahrt. Server-Default ist ON, bei OFF werden Plandaten verwendet.
--------	----------	----------------------------	--------------------	--

#### Beispiel:

Request: Details für eine Reise von Amstetten nach Wien Hauptbahnhof mit Echtzeitinformationen.

[https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/journeyDetail?accessId={accessId}&requestId=&format=xml&lang=deu&id=1|22328|0|81|12022021&rtMode=SERVER\\_DEFAULT](https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}}/journeyDetail?accessId={accessId}&requestId=&format=xml&lang=deu&id=1|22328|0|81|12022021&rtMode=SERVER_DEFAULT)

#### Beispiel Response:

Liste mit Stopps/Haltestellen mit Detailinformationen zu Stationsname, Abfahrtszeit und - datum, Gleisinformation, Richtung

<Stops>

<Stop name="Klagenfurt Hbf" id="A=1@O=Klagenfurt Hbf@X=14313542@Y=46615649@U=81@L=420364207@" extId="420364207" routeIdx="0" lon="14.313542" lat="46.615649" depPrognosisType="PROGNOSED" depTime="14:45:00" depDate="2021-02-

12" depTrack="3" rtDepTrack="3" depDir="Flughafen Wien Bahnhof" entry="true">

<Notes>

<Note key="TZ" type="I" txtN="ktn\$34555">ktn\$34555</Note>

</Notes>

<altId>at:42:3642:0:7</altId>

</Stop>

<Stop name="Pörtlach Bahnhof" id="A=1@O=Pörtlach Bahnhof@X=14146315@Y=46636244@U=81@L=420364802@" extId="420364802" routeIdx="5" lon="14.146315" lat="46.636244" arrPrognosisType="PROGNOSED" depPrognosisType="PROGNOSED" depTime="14:55:00" depDate="2021-02-

12" arrTime="14:54:00" arrDate="2021-02-

12" arrTrack="2" depTrack="2" rtArrTrack="2" rtDepTrack="2" entry="true">

<Notes>

<Note key="TZ" type="I" txtN="ktn\$34652">ktn\$34652</Note>

</Notes>

<altId>at:42:3648:0:2</altId>

</Stop>

Informationen zur Fahrt, mit dem Beförderungsmittel (ÖV Produktklasse) und Fahrtrichtung

<Names>

<Name name="RJ 797" number="797" category="RJ" routeIdxFrom="0" routeIdxTo="95">

<Product name="RJ 797" displayNumber="797" num="797" line="xxx" lineId="obb-10-A10-j21-1" catOut="RJ" catIn="RJ" catCode="0" cls="1" catOutS="RJ" catOutL="Railjet" operatorCode="01" operator="OEBB" admin="001AA\_" matchId="xxx">

<icon res="prod\_ic">

<foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>

<backgroundcolor r="1" g="1" b="1" hex="#010101"/>

</icon>

</Product>

</Name>

</Names>





```
<Directions>
  <Direction routeIdxFrom="0" routeIdxTo="95">Flughafen Wien Bahnhof</Direction>
</Directions>
<Notes>
```

#### Echtzeitinformationen (Informationen über Störungen) über die Fahrt und betroffene Stationen.

```
<Messages>
  <Message id="129260" act="true" head="Schienenersatzverkehr - Teilausfall"
text="Wegen Reparaturarbeiten nach einem Unfall kann dieser Zug von Salzburg Hbf bis Attnang-
Puchheim nicht fahren. Dadurch verlängert sich Ihre Reisezeit um bis zu 30 Minuten. Wir bitten
um Entschuldigung." company="KGÖVV" category="2" priority="0" products="65535" sTime="00:00:00"
sDate="2021-02-11" eTime="23:59:00" eDate="2021-02-17" modTime="14:29:45" modDate="2021-02-11">
    <affectedStops>
      <StopLocation id="A=1@O=Salzburg Hauptbahnhof@X=13045937@Y=47812896@U=81@L=45500
0209@" extId="455000209" name="Salzburg Hauptbahnhof" lon="13.045937" lat="47.812896" products="
1219">
        <LocationNotes>
          <LocationNote key="TZ" type="I" txtN="svv$50101">svv$50101</LocationNote>
        </LocationNotes>
        <altId>at:45:50002:0:9</altId>
      </StopLocation>
```



## 16 RECONSTRUCTION SERVICE

Mit Hilfe des `Reconstruction Services` kann eine vorherige `Trip` Abfrage durch Benutzung des Rekonstruktions-Kontextes (`ctx`) erneut abgefragt werden. Mögliche Anwendungen dafür sind, wenn eine Abfrage für den User reproduzierbar bzw. erneut abfragbar gemacht werden soll, zum Beispiel für das Teilen einer Verbindung und das Speichern von Favoriten.

Zur Rekonstruktion einer Route muss der Wert des Attributes `ctxRecon` im Bereich `Trip` aus der `Trip Response` in den Parameter `ctx` des `recon Services` eingesetzt werden.

Grundsätzlich ist zu beachten, dass dieses Service, ebenso wie die `Trip Service`, innerhalb der definierten Eckdaten der Fahrplanperiode rekonstruierbare Trips liefert (siehe dazu `Timetable Info Service` für die Eckdaten). Unabhängig von den definierten Eckdaten können max. 62 Tage in die Vergangenheit und bis zum Ende der Fahrplanperiode Abfragen über das `Reconstruction Service` erneut berechnet werden. Daher eignet sich dieses Service nicht, um Verbindungen dauerhaft (länger als 62 Tage) zu speichern. Es sind Einschränkungen bei der Abfrage in die Vergangenheit zu beachten, die unter [Abschnitt 5.5.3 Abfragen in die Zukunft und Vergangenheit](#) beschrieben sind. Eine uneingeschränkte Gültigkeit gibt es für den Rekonstruktionskontext nicht, da regelmäßig neue Fahrplandaten einspielt und damit Verbindungen aus dem Fahrplan entfernt werden könnten. Mit einem Rekonstruktionskontext, der für eine Fahrt vor dem Entfernen der Fahrt aus den Plandaten erstellt wurde, kann die Fahrt nach dem Entfernen nicht mehr gefunden werden. In der Regel ist ein Rekonstruktionskontext aber über längere Zeit funktionsfähig, da nur selten Fahrten durch ein Update entfernt werden. Eine garantierte Zeit ist aus den genannten fachlichen Gründen aber nicht gegeben.

### Wichtiger Hinweis:

Mit dem Setzen des Parameter `date` kann das ursprüngliche Datum der Fahrt zwar mit einem neuen Datum (z.B. einem aktuellen Datum) überschrieben werden und es wird eine Fahrt mit dem neuen Datum gesucht, um die Route zu aktualisieren, allerdings sollte das Datum nicht dazu verwendet werden, Routen die über einen langen Zeitraum in einem externen Service gespeichert wurden, mit einem neuen Datum zu versehen, um die Fahrt zu rekonstruieren. Das hat den Hintergrund, dass über die 62 Tage rückwirkende Gültigkeit auch die Datenebene des Service zu berücksichtigen ist: Fahrplanänderungen oder Updates am Netz sind in diesem Zeitraum wahrscheinlich, daher kann nicht garantiert werden, dass exakt diese Fahrt noch existiert, weil beispielsweise ein Verkehrsmittel außer Betrieb genommen wurde. Die Empfehlung ist daher, das `recon Service` immer nur in aktuellen Sessions zu verwenden (z.B. zum Teilen von Routen), aber nicht dafür, um Routen langfristig abzuspeichern.

### 16.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>ctx</b>	Erforderlich	-	-	Legt den Rekonstructions-Text fest
<b>poly</b>	Optional	0/1	0	Aktiviert/ Deaktiviert die Berechnung der Polylinie für jede Routenetape.
<b>polyEnc</b>	Optional	GPA	-	Aktiviert die Ausgabe der Polyline im Format „Google Encoded Polyline Format“.
<b>date</b>	Optional	-	-	Definiert das Datum, für welches der Trip wiederhergestellt werden soll. Format YYYY-MM-DD Ist der Trip für das gewählt Datum nicht vorhanden, wird der Fehlercode <code>SVC_NO_RESULT</code> zurückgegeben.



## 17 TIMETABLE INFO

Der `Timetable Info Service` informiert über die aktuellen ÖV Eckdaten der Fahrplanperiode, sowie über das letzte Update des Datenstandes. Abfragen in den `Station Board Services` und im `Trip Service` sind durch diese Eckdaten zeitlich begrenzt.

Grundsätzlich können unabhängig von den definierten Eckdaten max. 62 Tage in die Vergangenheit und bis zum Ende der Fahrplanperiode Abfragen berechnet werden.

Die Attribute der `Timetable Information` für den Typ „ST“ (Stationen) ist wie folgt zu interpretieren: Das Attribut `date` entspricht dem Datum der letzten Aufbereitung der Daten (Daten-Update).

Das Attribut `begin` entspricht dem Eckdatenbeginn, also dem Beginn der Fahrplanperiode minus der definierten 62 Tage Rückschau. Das Attribut `end` entspricht dem Eckdatenende, also dem Ende der Fahrplanperiode.

Im Bereich zwischen `begin` und `end` können Abfragen getätigt werden. Außerhalb dieser Periode werden keine Antworten zurückgegeben. Traditionell erfolgt eine Umstellung der Fahrplanperiode Mitte Jänner. Da der Fahrplanwechsel aber zu Jahresende bereits startet, sind im Spätherbst mit Anpassungen der Period zu rechnen und Änderungen bei Fahrten. Auch unterjährige Änderungen sind üblich, daher ist das Datum der letzten Aufbereitung der Daten relevant.

Beispiel:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/tti?accessId={{accessId}}>

### 17.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung

## 18 DATA INFORMATION (DATAINFO)

Dieser Service bietet detaillierte Informationen über alle Betreiber (Element und Parameter `operator`), Verwaltungen (Element `administration`), Produktklassen (siehe Abschnitt 5.1 (Element `product category`) sowie alle Verkehrsmittel/Produkte innerhalb der Produktklasse (Element `product` Attribut `catOutL`), die vom zugrunde liegenden Routenberechnungsalgorithmus geladen werden. Die meisten Werte sind in verschiedenen Diensten als Filter verwendbare Optionen.

Grundlegende Informationen zu den Fahrplandaten und den ÖV Produktklassen sind in [Abschnitt 5.5 Fahrplandaten \(SOLL-Daten\)](#) und [Abschnitt 5.6 ÖV Produktklassen](#) dokumentiert.

### 18.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung



### Beispiel Request:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/datainfo?accessId={accessId}>

### Beispiel Response:

DataInfo begin="2019-06-17" end="2019-09-15" serverVersion="2.24.2" dialectVersion="2.24" xmlns=<http://hacon.de/hafas/proxy/hafas-proxy> requestId="1483016905755">

```
<Operator name="SAD Bus" nameS="025" nameN="025" nameL="SAD Bus" id="apb+025+1">
  <administration>I02501</administration>
</Operator>
<Operator name="SASA" nameS="004" nameN="004" nameL="SASA" id="apb+004+1">
  <administration>I00401</administration>
</Operator>
<Operator name="Simobil" nameS="022" nameN="022" nameL="Simobil" id="apb+022+1">
  <administration>I02201</administration>
</Operator>
<Operator name="Unbekannt" nameS="999" nameN="999" nameL="Unbekannt" id="apb+999+1">
  <administration>I99901</administration>
</Operator>
<Operator name="Pizzinini" nameS="019" nameN="019" nameL="Pizzinini" id="apb+019+1">
  <administration>I01901</administration>
...
<Product name="Regionalbus" catOut="" catIn="I20" cls="64" catOutL="Bus">
  <icon res="prod_ic_bus">
    <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
    <backgroundColor r="0" g="121" b="58" hex="#00793A"/>
  </icon>
</Product>
<Product name="Stadtbus" catOut="" catIn="I23" cls="128" catOutL="Citybus">
  <icon res="prod_bus">
    <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
    <backgroundColor r="0" g="121" b="58" hex="#00793A"/>
  </icon>
</Product>
<Product name="Stadtbus" catOut="" catIn="I22" cls="128" catOutL="Bus">
  <icon res="prod_bus">
    <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
    <backgroundColor r="0" g="121" b="58" hex="#00793A"/>
  </icon>
</Product>
<Product name="Zuege" catOut="R" catIn="R" cls="1" catOutL="Regionalzug">
  <icon res="prod_ic">
    <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
    <backgroundColor r="1" g="1" b="1" hex="#010101"/>
  </icon>
</Product>
...
<ProductCategory name="Zuege" cls="1">
  <Product name="Zuege" catOut="R" catIn="R" cls="1" catOutL="Regionalzug">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
  </Product>
  <Product name="Zuege" catOut="RE" catIn="RE" cls="1" catOutL="Regional-Express">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
  </Product>
  <Product name="Zuege" catOut="RV" catIn="RV" cls="1" catOutL="Regionale Veloce">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
  </Product>
  <Product name="Zuege" catOut="ICN" catIn="ICN" cls="1" catOutL="Intercity Notte">
```



```
<icon res="prod_ic">
  <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
  <backgroundcolor r="1" g="1" b="1" hex="#010101"/>
</icon>
</Product>
<Product name="Zuege" catOut="FA" catIn="FA" cls="1" catOutL="Frecciargento">
  <icon res="prod_ic">
    <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
    <backgroundcolor r="1" g="1" b="1" hex="#010101"/>
  </icon>
</Product>
<Product name="Zuege" catOut="FR" catIn="FR" cls="1" catOutL="Frecciarossa">
  <icon res="prod_ic">
    <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
    <backgroundcolor r="1" g="1" b="1" hex="#010101"/>
  </icon>
</Product>
... </ProductCategory>
</DataInfo>
```

## 19 (LINE INFORMATION (LINEINFO) – BETA VERSION)

Wichtiger Hinweis:

**Achtung: Dieses Service ist aktuell für Explorationszwecke freigeschaltet wird noch überarbeitet. Änderungen im Verhalten und bei den Linien IDs sind vorgesehen. Bis zum finalen Launch des Service soll keine produktive Einbindung erfolgen.**

Dieser Service liefert alle Informationen über eine bestimmte Linie an einem spezifischen Datum. Die Linie wird über die `lineID` im Request definiert.

Die Response besteht aus einer List (Element `LineList`), mit den repräsentativen Fahrten dieser Linie am definierten Tag inklusive der verfügbaren Produktinformationen.

Die `LinienID` ist ein Attribut des Elements `Product`. Die `LineId` kann daher aus einem Trip Response oder einer DepartureBoard, eine Journey Detail Response bezogen werden.

### 19.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
<b>accessId</b>	Erforderlich	-	-	Access ID zur Authentifizierung
<b>lineId</b>				Interne LineId der Linie
<b>date</b>				Betriebstag im Format YYYY-MM-DD

**Beispiel Request:**

`{{baseURL}}/{{version}}/lineinfo?accessId={{key}}&lineId=stv-38-30-j25-16&date=2025-08-17&format=xml`

**Beispiel Response (gekürzt)**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<LineList serverVersion="2.42.1" dialectVersion="2.42" requestId="9rm47qwiw8xiwiwx"
xmlns="http://hacon.de/hafas/proxy/hafas-proxy">
  <TechnicalMessages>
```



```
<TechnicalMessage key="requestTime">2025-07-30 15:22:48</TechnicalMessage>
<TechnicalMessage key="backendInfo">ttp=16585#16784 plancode0=agr46 planid=1753822184
planid0=1753822184 planid_adr=1753273272 plancode_adr=940lm planid_adr=1662723912
plancode_adr=r779m planid_poi=1753826766 plancode_poi=agvog srvv=5.45.KGOEVV.17.3.16
(customer/hcuvao/release/2025.q2.1.0) [2025-07-02] tlibv=TRFVER: rel/vao/6.02.1 2025-02-21
14:44:41 +0100 VAO_TARIF v1.35 + Relay_EnrichmentProxy v1 jno=1</TechnicalMessage>
</TechnicalMessages>

<Line lineId="obb-10-A13-j25-1" lineName="xxx" lineNameShort="xxx">
  <Product name="xxx" internalName="xxx" line="xxx" lineId="obb-10-A13-j25-1" catOut="RJ"
cls="1" operatorCode="01" operator="OEBB Personenverkehr AG Kundenservice">
    <icon res="prod_ic">
      <foregroundColor r="255" g="255" b="255" hex="#FFFFFF"/>
      <backgroundColor r="1" g="1" b="1" hex="#010101"/>
    </icon>
    <status>
      <rtIcon txt="einige Verspätungen" txtS="einige Verspätungen" style="N" shape="R">
        <foregroundColor r="0" g="0" b="0" hex="#000000"/>
        <backgroundColor r="255" g="204" b="0" hex="#FFCC00"/>
      </rtIcon>
    </status>

    <operatorInfo name="OEBB Personenverkehr AG Kundenservice" nameS="01" nameN="01"
nameL="OEBB Personenverkehr AG Kundenservice" id="obb+01+AA"/>
  </Product>

  <Journey name="RJ 823" direction="Flughafen Wien Bahnhof" directionFlag="R"
trainNumber="823" trainCategory="RJ">
    <Stops>
      <Stop name="Linz/Donau Hauptbahnhof" id="A=1@O=Linz/Donau
Hauptbahnhof@L=Linz/Donau Hauptbahnhof (Gleis/Steig
10)@X=14292255@Y=48290106@U=81@L=444116410@i=A@at:44:41164:0:10@" extId="444116410" routeIdx="0"
lon="14.292255" lat="48.290106" depTime="05:32:00" depDate="2025-08-17" depTrack="10A-B"
depDir="Flughafen Wien Bahnhof" minimumChangeDuration="PT4M">
        <Notes>
          <Note key="TZ" type="I" txtN="oov$42500">oov$42500</Note>
          <Note key="hi" type="I" txtN="obb:8100013">obb:8100013</Note>
        </Notes>
        <altId>at:44:41164:0:10</altId>
        <depPlatform type="X" text="10A-B"/>
      </Stop>
    </Stops>
  </Journey>
</Line>
...
</LineList>
```

## 20 FEHLERCODES UND WARNUNGEN

Im Falle eines fehlerhaft beendeten Request wird ein Fehlercode und dessen textuelle Beschreibung zurückgeliefert. Die Fehler können in verschiedene Kategorien eingeteilt werden. Die ReST API und Backend-Server- Fehler sind unabhängig vom aufgerufenen Service. Andere Fehler stehen in Bezug zu diesen.

### 20.1 ReST Request-Fehler

Code	HTTP status code (legacy)	HTTP status code Ab V1.9.0 (default)	Text
API_AUTH	403	403	Zugriff verweigert für „Schlüssel“ zum „Service“



<b>API_QUOTA</b>	400	400	Quotenüberlauf für „Schlüssel“ zum „Service“
<b>API_TOO_MANY_REQUESTS</b>	429	403	too many requests
<b>API_PARAM</b>	400	400	Erforderlicher Parameter <<name>> fehlt
<b>API_PARAM</b>	400	400	numB falsch, nur Werte im Bereich [0,6] erlaubt
<b>API_PARAM</b>	400	400	numF falsch, nur Werte im Bereich [0,6] erlaubt
<b>API_PARAM</b>	400	400	numF + numB dürfen 6 nicht überschreiten
<b>API_FORMAT</b>	400	400	Ausgabeformat nicht unterstützt
<b>SVC_PARAM</b>	400	400	Request Parameter falsch oder fehlend
<b>SVC_LOC</b>	400	400	Ortsangabe falsch oder fehlend/ Es konnte keine Haltestelle in der Umgebung der Adresse gefunden werden
<b>SVC_LOC_ARR</b>	400	400	Ankunftsort falsch oder fehlend
<b>SVC_LOC_DEP</b>	400	400	Abfahrtstort falsch oder fehlend
<b>SVC_LOC_VIA</b>	400	400	Unbekannter Umstiegshalt
<b>SVC_LOC_EQUAL</b>	400	400	Start/Ziel oder Vias identisch
<b>SVC_LOC_NEAR</b>	400	400	Start und Ziel liegen zu nahe beieinander
<b>SVC_DATETIME</b>	400	400	Datum/Zeit ungültig oder fehlend
<b>SVC_DATETIME_PERIOD</b>	400	400	Datum/Zeit außerhalb der gültigen Periode oder nicht im Fahrplan
<b>SVC_PROD</b>	400	400	Produkt (Transportmittel) Feld fehlt oder ungültig
<b>SVC_CTX</b>	400	400	Kontext ungültig
<b>SVC_MAIL_ADR</b>	400	400	Absender/Empfänger- Mailadresse ungültig oder fehlend
<b>SVC_NO_RESULT</b>	400	200	Kein Ergebnis gefunden
<b>SVC_SMS_NUM</b>	400	400	Empfängertelefonnummer für SMS ungültig oder fehlend
<b>Server Errors</b>			
<b>SVC_MAIL</b>	500	500	Fehler beim Mailversand
<b>BAIM_ERROR</b>	500	500	Konfiguration oder Routing Fehler in BAIM
<b>SVC_SMS</b>	500	500	Fehler beim SMS- Versand



<b>SVC_FAILED_SEARCH</b>	500	500	Nicht erfolgreiche Suche
<b>SVC_NO_MATCH</b>	422	422	Keine Übereinstimmung gefunden/ kein Treffer
<b>SVC_TOO_MANY</b>	400	<b>422</b>	Zu viele Suchergebnisse
<b>INT_ERR</b>	500	500	Interner Fehler
<b>INT_HAFAS_CONNECTION_ERROR</b>	503	503	Verbindung zu [HOST]:[PORT] ist nicht möglich; Time-Out der Verbindung (ConnectionException)
<b>INT_HAFAS_CONNECTION_ERROR</b>	503	503	Verbindung zu [HOST]:[PORT] durch HAFAS Server geschlossen (ConnectionException)
<b>INT_HAFAS_CONNECTION_ERROR</b>	503	503	Verbindung zu [HOST]:[PORT] durch HAFAS Server geschlossen (ConnectionException) (HAFAS Server Timeout-Limit überschritten)
<b>INT_HAFAS_CONNECTION_OPENING_ERROR</b>	503	503	Fehlermeldung vom HAFAS server [HOST]:[PORT]: error=1&[CONNECTION_STRING] in den meisten Fällen tritt dieser Fehler auf, wenn die Queue am HAFAS Server voll ist
<b>INT_TIMEOUT</b>	503	503	Timeout während der Verarbeitung (API Timeout-Limit überschritten)
<b>INT_TIMEOUT</b>	503	503	Timeout während der Kommunikation mit dem Backend (Backend-Timeout-Limits überschritten)
<b>Spezielle Errors</b>			
<b>SOT_AT_DEST</b>	400	400	Trip bereits erreicht
<b>SOT_BEFORE_START</b>	400	400	Trip nicht gestartet
<b>SOT_CANCELLED</b>	400	400	Trip gelöscht
<b>SOT_ALL_TRAINS_FILTERED</b>	400	400	Alle Trips gefiltert
<b>SOT_STAY_IN_CURRENT_CONNECTION</b>	400	200	Keine Änderung. In dem Trip verbleibend.
<b>PARTIALSEARCH_INCORRECT_PARAM</b>	400	400	Mit einer invaliden Parameterkombination wurde angefragt, z.B. wurde versucht rückwärts/vorwärts vom Beginn/Ende einer Verbindung zu suchen.
<b>SOT_TRANSIT_LOCATION_MISSING_FOR_JOURNEY</b>	400	400	Angefragte Umstiegsorte sind nicht Teil der Fahrt

## 20.2 Warnungen





In einigen Fällen kann eine Response mit Warnungen zurückgegeben werden. Diese sind im Element <Warnings> enthalten.

Code	Beschreibung
<b>TOO_MANY</b>	Das Ergebnis enthält zu viele Elemente. Einige Elemente wurden verworfen.
<b>TOO_MANY_SCROLL_CTX</b>	Das Ergebnis enthält zu viele Scroll Kontext Elemente. Einige Elemente wurden verworfen.
<b>H887</b>	Timeout überschritten. Das Ergebnis kann unvollständig sein.
<b>H889</b>	Zu viele Routen gefunden. Einige Elemente wurden verworfen.