

HANDBUCH **VAO** REST API

Version 2026-01-28

Verkehrsauskunft Österreich VAO GmbH



Inhalt

1	ALLGEMEINES ZUM HANDBUCH	6
1.1	ÄNDERUNGSHISTORIE	6
1.2	ÜBER DAS HANDBUCH	9
2	EINFÜHRUNG & API SPEZIFIKATION	10
2.1	ALLGEMEINES ZUR VAO REST API	10
2.2	API SPEZIFIKATION	10
3	VERSIONIERUNG UND RÜCKWÄRTSKOMPATIBILITÄT	12
3.1	STRUKTURELLE UND INHALTLICHE ÄNDERUNGEN	12
3.2	RELEASE- UND VERSIONSSTRATEGIE	13
4	AUFRUF & NUTZUNG	15
4.1	AUFRUF	15
4.2	VERSIONEN	15
4.3	AUTHENTIFIZIERUNG	15
4.4	VERWENDUNG VON URI SYNTAX	15
4.5	ERGEBNISFORMAT	15
4.6	XML ZU JSON KONVERTIERUNG	16
5	WICHTIGE INFORMATIONEN ZUR NUTZUNG DER API	17
5.1	STATELESS API & ID-GÜLTIGKEIT	17
5.2	DATUM- UND UHRZEIT	17
5.3	ZEITZONEN	17
5.4	UNTERSTÜTZTE SPRACHEN	17
5.5	REQUEST VERFOLGUNG	17
5.6	GÜLTIGKEIT VON FAHRPLANDATEN & HISTORISIERUNG	17
5.7	HIERARCHISCHE HALTESTELLEN-STRUKTUR	18
5.8	METAORTE	19
5.9	ID-TYPEN FÜR HALTESTELLEN	19
5.10	ÖV-VERKEHRSMITTEL	20
5.11	ADRESSDATEN	20
5.12	GEOREFERENZIERUNG	20
5.13	ROUTING-GRAPH (KARTENGRUNDLAGE)	20
5.14	SNAPPING UND EINSCHRÄNKUNGEN BEI FREI GEWÄHLTEN KOORDINATEN	21



5.15	IV-ROUTING	21
5.16	ÖV ROUTING	22
5.17	REALTIME-DATEN (ÖV)	23
5.18	REALTIME-DATEN (IV)	23
5.19	EINSCHRÄNKUNGEN BEI ROUTING IM AUSLAND	24
5.20	WETTERINFORMATIONEN	24
5.21	DATEN ZU E-AUTO LADESTELLEN	24
5.22	POLYLINIE-ENCODING & GEOMETRIEN	24
5.23	FEHLERBEHANDLUNG	25
5.24	RATE-LIMITS & NUTZUNG	25
5.25	SESSION-MANAGEMENT	25
6	SERVICE-ÜBERSICHT	26
7	LOCATION.NAME (ORTSPUNKTAUFLÖSUNG MIT NAMEN)	28
7.1	REQUEST PARAMETER	28
7.2	VERWENDUNG ZUSÄTZLICHER KOORDINATEN (STANDORT-BASIERTE AUFLÖSUNG)	30
7.3	FILTER-MODUS (PARAMETER FILTERMODE)	31
7.4	HÖHER AUFLÖSBARE ORTE	31
8	LOCATION.NEARBYSTOPS (UMKREISSUCHE MIT KOORDINATEN)	32
8.1	REQUEST PARAMETER	32
8.2	META-FILTER FÜR BESTIMMTE INHALTE IN DER UMKREISSUCHE	33
8.3	META-FILTER FÜR POI-FILTERUNG	35
8.4	ENTRY POINTS IM DETAIL	37
8.5	SHARING-ANGEBOTE & ECHTZEIT-FAHRZEUGPOSITION	37
9	LOCATION.DETAILS (ORTSPUNKTDETAILS)	38
9.1	REQUEST PARAMETER	39
10	ADDRESSLOOKUP (ADRESS-SUCHE UM KOORDINATE)	42
10.1	REQUEST PARAMETER	42
11	TRIP SERVICE	44
11.1	REQUEST PARAMETER	45
11.2	MODALITÄTEN UND KOMBINATIONEN (GROUPFILTER)	53
11.3	META-PROFILE FÜR IV ROUTING (AUTO, RAD, E-AUTO)	56
11.4	META-PROFILE FÜR E-AUTO ROUTING	58



11.5	BESONDERHEITEN FÜR ÖV-ROUTENSUCHE	60
11.6	BESONDERHEITEN FÜR IV-ROUTENSUCHE	62
12	GIS ROUTEN SERVICE	65
12.1	REQUEST PARAMETER	65
13	DEPARTUREBOARD SERVICE	66
13.1	REQUEST PARAMETER	67
13.2	SCROLLING IN DEN TAFELN	69
14	ARRIVALBOARD (ANKUNFTSTAFEL)	70
14.1	REQUEST PARAMETER	70
14.2	SCROLLING	72
15	JOURNEY DETAIL SERVICE	73
15.1	REQUEST PARAMETER	73
16	RECONSTRUCTION SERVICE	76
16.1	REQUEST PARAMETER	76
17	TIMETABLE INFO	78
17.1	REQUEST PARAMETER	78
18	DATA INFORMATION (DATAINFO)	79
18.1	REQUEST PARAMETER	79
19	(LINE INFORMATION (LINEINFO) – BETA VERSION)	81
19.1	REQUEST PARAMETER	81
20	FEHLERCODES UND WARNUNGEN	83
20.1	REST REQUEST-FEHLER	83
20.2	WARNUNGEN	85
21	WEITERFÜHRENDE INFORMATIONEN ZU ÖV	86
21.1	FAHRPLANDATEN (SOLL-DATEN)	86
21.2	VERARBEITUNG VON REALTIME-DATEN (ÖV)	87
21.3	ÖV-STÖRUNGSMELDUNGEN (IMS MELDUNGEN)	92
21.4	HALTESTELLEN-TOPOLOGIE (METAORT-HAUPTMAST-MAST)	92
21.5	ÖV-VERKEHRSMITTEL & PRODUKTKLASSEN	93
22	WEITERFÜHRENDE INFORMATIONEN ZU IV	97
22.1	ALTERNATIVE ROUTE IN DER RESPONSE	97
22.2	IV-VERKEHRSMELDUNGEN IN DER TRIP RESPONSE	97



22.3	MAUTINFORMATIONEN IN DER TRIP-RESPONSE	98
22.4	INTERPRETATION DER HINWEISE IN DEN SEGMENTEN EINER IV-ROUTE	99
22.5	VIA-ORTE IN DER TRIP RESPONSE EINER IV -ROUTE	100
22.6	PRAXIS ZUM DECODIEREN DER ROUTEN-POLYLINIE	100
23	ORTSPUNKTE UND VERKEHRSNETZ (GIP)	101
24	EINSCHRÄNKUNGEN BEI ROUTEN IM AUSLAND	102

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 Paramater (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	10.05.2022	Bettina Neuhäuser



	Stationen mit Entry Points (E) im Parameter Type		
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, metaFilter in Location.nearby ergänzt, Fehlercodes ergänzt Geo-basierte Orspunktauflösung in location name	20.07.2022	Bettina Neuhäuser
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	08.11.2025	Bettina Neuhäuser
2026-01-28	Überarbeitung der Kapitel. Einfügen des Parameters „products“ bei Location.nearbystops Einfügen der Information zur Auslastungsinformation cdyn=0 hinzugefügt (Deaktivieren der Dynamik im IV Router)	28.01.2026	Bettina Neuhäuser



1.2 Über das Handbuch

Dieses Handbuch beschreibt die VAO REST API mit den von der VAO-Routenplanung unterstützten Endpoints und Parametern.

Das Dokument basiert auf der Beschreibung der HAFAS,api der HaCon Ingenieurgesellschaft mbH und wurde um VAO-spezifische Erweiterungen und Einschränkungen für die Verkehrs Auskunft Österreich ergänzt.

Das Handbuch zur VAO REST API bietet einen Überblick über die verfügbaren Funktionalitäten der Programmierschnittstelle der Verkehrs Auskunft Österreich (VAO). Ergänzend stehen technische Beschreibungen der Schnittstelle in Form von XSD-Schemata, JSON-Strukturen sowie einer Swagger-/OpenAPI-Dokumentation zur Verfügung.

Das Handbuch wird kontinuierlich aktualisiert und im Rahmen von Release-Informationen an den Kundenkreis verteilt. Die jeweils aktuelle Version des Handbuchs ist unter folgendem Link verfügbar:

<https://cloud.verkehrsauskunft.at/index.php/s/tM2LGbysT4rGjr4>

(Zugriff: Passwort wird über den VAO Support bereit gestellt)

2 EINFÜHRUNG & API SPEZIFIKATION

2.1 Allgemeines zur VAO REST API

Die VAO REST API ist eine spezialisierte Implementierung auf Basis der HAFAS.api der HaCon Ingenieurgesellschaft mbH und stellt die zentrale Programmierschnittstelle für die Routenplanung in Österreich dar.

Die API bildet funktional die VAO-Routenauskunft ab, wie sie in den VAO WebApps und mobilen Anwendungen eingesetzt wird. Funktionsumfang und Konfigurationsmöglichkeiten entsprechen mindestens jenen der VAO WebApp.

Im Vergleich zur HAFAS Standard HAFAS.api erweitert die VAO REST API den Funktionsumfang um VAO-spezifische Anpassungen und Einschränkungen zur Abbildung österreichischer Mobilitätsangebote. Unterstützt werden multimodale Routenberechnungen über öffentliche Verkehrsmittel, Individualverkehr (PKW, Fahrrad) sowie E-Mobilität.

Technische Grundlagen:

- API-Basis: HaCon HAFAS.api
- Datenbasis: Konsolidierte Soll- und Echtzeitdaten aus der österreichweiten Datendrehscheibe der Mobilitätsverbünde Österreich (MVO)
- Externe Datenquellen: GeoSphere, E-Control, EVIS Verkehrsdaten, Sharing-Angebote und andere
- Routing-Grundlage: Graphen-Integrations-Plattform (GIP)
- Datenaktualisierung: Hohe Aktualisierungsfrequenz mit dynamischen IDs und Referenzen

Die VAO REST API verwendet ausschließlich eine definierte Teilmenge der in der HAFAS.api verfügbaren Request-Parameter und API Schemas. Entsprechend kann die maschinenlesbare OpenAPI-Dokumentation Parameter und Endpunkte enthalten, die von der VAO REST API nicht unterstützt werden. Dies ist insbesondere bei der automatisierten Client-Generierung zu berücksichtigen.

Die Weiterentwicklung der VAO-Produkte erfolgt parallel zur Weiterentwicklung der VAO REST API. Neue Funktionen werden versioniert bereitgestellt und in der Regel zeitnah in der API nachgezogen. Zur Sicherstellung der Backend-Kompatibilität unterliegt die API einer kontinuierlichen Wartung und Integration.

2.2 API Spezifikation

2.2.1 WADL (Web Application Description Language)

XML-basiertes Format zur Beschreibung der Services und Request-Parameter.

Aufruf:

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

Beispiel:

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

2.2.2 OpenAPI 3.0 (Swagger)

Die Swagger Dokumentation

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



Eine interaktive Swagger-Dokumentation (Swagger UI)

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

2.2.3 XML-Schemadefinition (XSD)

Liefert die Definitionen der Response-Elemente einer bestimmten Version

Ausgabe: HTML-Liste aller verfügbaren XSD-Dateien

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

WADL und OpenAPI enthalten Parameter, die **nicht von VAO unterstützt** werden. Für die Entwicklung ist ausschließlich der tatsächliche im Handbuch dokumentierte Funktionsumfang der VAO REST API relevant.



3 VERSIONIERUNG UND RÜCKWÄRTSKOMPATIBILITÄT

Die VAO-Services und die darauf aufbauende VAO REST API werden kontinuierlich weiterentwickelt. Änderungen können sowohl die Struktur der Schnittstelle als auch die gelieferten Inhalte betreffen.

3.1 Strukturelle und inhaltliche Änderungen

3.1.1 Strukturelle Änderungen

Strukturelle Änderungen insbesondere der Wegfall oder die semantische Änderung von Attributen sowie Anpassungen an Request- oder Response-Schemata, können zu Breaking Changes führen. Solche Änderungen werden ausschließlich über neue Versionen der VAO REST API eingeführt. Bestehende Versionen bleiben parallel verfügbar, um Abnehmern ausreichend Zeit für Anpassungen zu geben.

Die VAO behält sich vor, ältere API-Versionen nach vorheriger Ankündigung und mit angemessener Vorlaufzeit außer Betrieb zu nehmen.

3.1.2 Inhaltliche Änderungen

wie etwa Aktualisierungen von Fahrplan- oder Kartendaten, betriebliche Änderungen von Verkehrsmitteln oder die Einführung neuer Mobilitätsmodalitäten, erfolgen laufend und können **ohne Einführung einer neuen API-Version** vorgenommen werden, sofern sie sich innerhalb des bestehenden Schemas bewegen.

Geringfügige Abweichungen in API-Antworten können auftreten, beispielsweise:

- zusätzliche oder optionale Felder bei bestehenden Elementen
- neue Elemente in Listen
- angepasste oder erweiterte Wertebereiche

3.1.3 Anforderungen an Abnehmer-Implementierungen

API-Abnehmer sind daher verpflichtet, ihre Integrationen **robust und tolerant** zu implementieren. Insbesondere ist beim Parsen der API-Antworten sicherzustellen, dass:

- unbekannte Felder ignoriert werden
- nicht von festen oder vollständigen Strukturen ausgegangen wird
- optionale Felder sowohl als leerer Wert als auch als null korrekt verarbeitet werden
- Wertebereiche validiert und mit geeigneten Default- oder Fallback-Werten behandelt werden

Diese Robustheit ist notwendig, um einen stabilen Betrieb auch bei laufenden Daten- oder Backend-Updates sicherzustellen.

3.2 Release- und Versionsstrategie

3.2.1 Release-Frequenz und Versionsnummerierung

Die Veröffentlichung neuer Versionen erfolgt abhängig von **Kundenanforderungen und VAO-Weiterentwicklungen**. Grundsätzlich sind bis zu drei API-Versionen pro Jahr vorgesehen, die tatsächliche Anzahl kann variieren.

Die VAO REST API verwendet **Semantic Versioning 2.0.0** (MAJOR.MINOR.PATCH) gemäß <https://semver.org>:

- MAJOR: Inkompatible Änderungen (Breaking Changes)
- MINOR: Abwärtskompatible funktionale Erweiterungen
- PATCH: Abwärtskompatible Fehlerkorrekturen

3.2.2 LTS- und Laufzeiten

Aktuelle Informationen zu verfügbaren Versionen, Laufzeiten und Endpunkten werden über die API bereitgestellt: <https://routenplaner.verkehrsauskunft.at/vao/restproxy>

Somit haben Abnehmer:innen jederzeit Zugriff auf die **aktuelle Liste der Versionen**, Endpunkte und geplanten Außerbetriebnahmen, ohne dass das Handbuch verfügbar sein muss.

- Standard-Versionen werden **mindestens 1 Jahr** betrieben
- LTS-Versionen (Long-Term Service) werden **mindestens 1,5 Jahre** betrieben
- Nach Ablauf dieser Zeit behält sich die VAO das Recht vor, eine Version außer Betrieb zu nehmen
- Abnehmer sollten Upgrades auf die aktuelle Version in ihre **Release- und Wartungszyklen** einplanen

Tabelle: Versionen, Inbetriebnahme und EOL

Status	Version	Inbetriebnahme	Außerbetriebnahme (End Of Life)
Geplant	V.X.X LTS *	Q3/2026	Q3/2028
In Betrieb	V1.12.0	17.12.2025	30.08.2027
In Betrieb	V1.11.0	23.05.2024	31.05.2027
Außer Betrieb	V1.10.0	3.12.2024	31.12.2025
In Betrieb	V1.9.0 LTS*	5.08.2024	31.08.2026
Außer Betrieb	V1.8.0	4.10.2023	31.01.2025 (verlängert)
Außer Betrieb	V1.7.0 LTS*	21.04.2023	30.04.2025 (verlängert)
Außer Betrieb	V1.6.0	14.12.2022	30.04.2024 (verlängert)
Außer Betrieb	V1.5.0	19.07.2022	31.07.2023
Außer Betrieb	v1.4.0	11.03.2022	30.06.2023 (verlängert)



Außer Betrieb	v1.3.2 LTS*	21.10.2021	30.06.2023 (verlängert)
Außer Betrieb	v1.3.1	28.02.2021	30.06.2022
Außer Betrieb	v1.3.0*	24.11.2020	30.06.2022
Außer Betrieb	v1.2.0	27.07.2020	31.12.2021
Außer Betrieb	v1.1.0	11.03.2020	31.12.2021
Außer Betrieb	V1.0.0	08.04.2019	31.12.2021

* Long-term Service

Es wird empfohlen, regelmäßig auf die **aktuelle Version** der REST API umzusteigen.



4 AUFRUF & NUTZUNG

4.1 Aufruf

Zugriff auf die **VAO REST API** erfolgt über den zentralen Endpunkt. Die Authentifizierung erfolgt über einen **Access Key**, der nach Vertragsabschluss von der VAO bereitgestellt wird.

Basis-Endpoint: <https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/>

Die API unterstützt **read-only GET-Requests**, die über Service-URLs und GET-Parameter aufgerufen werden.

- Alle Parameterwerte müssen **UTF-8** kodiert sein.
- Das Ergebnis wird entweder als **XML** oder **JSON** geliefert.
- Bei inkorrektur Parameterkodierung kann es zu unvorhersehbaren Ergebnissen kommen.

4.2 Versionen

Die aktuell verfügbaren Versionen können über den folgenden Endpunkt abgefragt werden:
<https://routenplaner.verkehrsauskunft.at/vao/restproxy>

Jede Version hat einen eigenen Endpunkt:

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

Das Trip-Service der API Version v1.12.0 wird somit beispielsweise mit folgender URL aufgerufen:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/v1.12.0/trip>

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 von VAO bereit gestellt.

4.4 Verwendung von URI Syntax

- Die API verwendet das HTTP-Protokoll; Aufrufe müssen URI-konform sein.
- Browser führen oft automatisch URL-Encoding durch, aber bei komplexen Strukturen (z. B. GisRefs) müssen reservierte Zeichen korrekt percent-encoded werden.
- Alle URL-Parameter sind case-sensitive und müssen exakt übergeben werden.

4.5 Ergebnisformat

- Standardmäßig liefert die API XML.
- Für JSON-Ausgaben kann entweder der GET-Parameter `format=json` oder der HTTP Accept Header gesetzt werden: `Accept: application/json`
- Wenn beide verwendet werden, hat der Parameter `format` Vorrang.
- JSONP kann über `format=jsonp&jsonpCallback=<callback_name>` verwendet werden.



4.6 XML zu JSON Konvertierung

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

- Elementnamen werden zu Objekteigenschaften
- Textinhalt (PCDATA) wird zu "value"
- Verschachtelte Elemente werden zu verschachtelten Eigenschaften

```
<a><b>foo</b><c>foo</c></a>
```

wird zu

```
{ "a": { "b" : { "$" : "foo" }, "c": { "$": "foo"} } }
```
- Mehrere gleichnamige Elemente werden zu Arrays

```
<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" } }
```

Beispiel:

- XML- Response

```
<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-Response

```
"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": "490019508", "routeIdx": "12", "time": "11:15",
      "date": "2019-12-18" }
  }
}
```




5 WICHTIGE INFORMATIONEN ZUR NUTZUNG DER API

5.1 Stateless API & ID-Gültigkeit

Alle Services der VAO ReST-API sind zustandslos (stateless), wie für REST üblich.

Einschränkung: Fahrplandaten oder Adressen können sich ändern (z. B. Fahrplanwechsel oder Adressupdates). Dadurch verlieren IDs und Referenzen (z. B. `Id`, `GisRef`, `ctxRecon`, `JourneyDetailRef`) ihre Gültigkeit.

IDs nicht langfristig speichern, Immer aktuell in einer Session abfragen.

5.2 Datum- und Uhrzeit

- Datum: `YYYY-MM-DD`
- Zeit: `hh:mm[:ss]` (24h-Format angegeben. Sekunden sind optional)
 - ÖV Requests ignorieren Sekunden (z.B. `14:37:52` → `14:37:00`)

5.3 Zeitzonen

- Datum/Uhrzeit immer in lokaler **Zeitzone Österreich (CET/CEST)** interpretieren.

5.4 Unterstützte Sprachen

Aktuell: **Deutsch** (de) und **Englisch** (en)

- Parameter: `lang=<code>`
- Nicht alle Inhalte sind in beiden Sprachen verfügbar.
- Die Sprachauswahl beeinflusst die Ausgabe von Notes / Verkehrsmitteltexte und Verkehrsmeldungen

5.5 Request Verfolgung

- Im Parameter `requestId` kann eine **frei gewählte Kennung** des API-Abnehmers transportiert werden
- Der Parameter ist auch in der Response enthalten
- Kann für Log-Tracking und **Fehleranalyse** seitens VAO genutzt werden, eine Verwendung des Parameters ist daher empfohlen

5.6 Gültigkeit von Fahrplandaten & Historisierung

Die API nutzt die vollständigen österreichische ÖV-Fahrplandaten, bereitgestellt von den Mobilitätsverbünden Österreich.

- Gültigkeit: **Fahrplanjahr**, Beginn Dezember.
- Fahrplanwechsel: Übergangsphase ab November, neue Fahrpläne eingebettet für 3 Monate.
- Aktualisierung: **zweimal täglich** im zentralen DASS-System (Daten-Sammelsystem) für unterjährige Änderungen



Die verwendete Fahrplanversion und die Gültigkeitsdauer ist in API Service `TableTableInfo` abrufbar.

- Abfragen in die Zukunft:
 - ist bis zu 4 Monate möglich
 - **Achtung:** Änderungen im Fahrplan sind durch tägliche Updates möglich
- Abfragen in der Vergangenheit: bis 62 Tage technisch möglich.
 - ÖBB Daten sind historisiert
 - Daten der Verkehrsverbünde nicht historisiert → kleine Abweichungen bei der Rekonstruktion einer Fahrt sind möglich.

Weiterführende Information → Abschnitt 21.1.2

Praxis-Tipp: maximal 6–8 Wochen im Voraus/Retrospektiv abfragen, oder bewusst mit dem Umstand umzugehen, dass keine Garantie abgegeben werden kann, dass die Route exakt so stattfinden wird oder stattgefunden hat.

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

5.7 Hierarchische Haltestellen-Struktur

Haltestellen werden **hierarchisch modelliert**, um komplexe Verkehrsknotenpunkte sauber abzubilden:

- Hauptmast (Haltestelle): Repräsentiert die zentrale Haltestelle, z. B. „Wien Hauptbahnhof“. Kennzeichnung durch Element `mainMast` und Attribut `isMainMast="true"`
- Masten (Steige): Untergeordnete Einheiten eines Hauptmastes, die konkrete Fahrten oder Zugänge repräsentieren.
- Bereiche: In großen Haltestellen (Bahnhöfe, Busknotenpunkte) können Masten zu Bereichen gruppiert werden. Wenn keine Bereiche definiert sind, gilt der Bereich 0 und wird als HoB („Haltestelle ohne Bereich“) bezeichnet.

Beispiel eines Hauptmasten

```
<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">
```

Parent-/MainMast-Zuordnung: Masten (Steige) besitzen eine Parent-Referenz (`MainMastId`), die auf den übergeordneten Hauptmast verweist.

- Diese Hierarchie ermöglicht die eindeutige Zuordnung von Masten zu Haltestellen.
- Jeder Hauptmast kann mehrere Masten besitzen
- Fahrten werden auf Mastenebene gespeichert, die API gibt die Hierarchie in der Antwort strukturiert aus

Weiterführende Information → Abschnitt 21.4



5.8 Metaorte

Metaort-Haltestellen sind spezielle Orte wie „WIEN“ oder „LINZ“. Sie dienen der Routenanbindung ohne Angabe eines konkreten Haltestellennamens. Der HAFAS Algorithmus berechnet automatisch (je nach ausgewählter Modalität und Ziel) die beste Verdingung von einer der verknüpften Haltestellen. Im Falle einer Individualverkehrs (IV)-Route wird die Koordinate des Metaortes verwendet (meist ein topografisch, zentraler Ort).

- Metaorte werden als `mainMast` geführt und mit dem Attribut `isMainMast="true"` gekennzeichnet.
- Metaorte besitzen eine Koordinate, die einen topographischen Ort und keine Haltestelle repräsentieren. Diese Koordinaten werden für das IV Routing verwendet
- Metaorte sind mit einer oder mehreren zentralen Haltestellen verknüpft.
- Metaorte selbst enthalten keine Fahrten, sie können jedoch in der Routenberechnung direkt genutzt werden

5.9 ID-Typen für Haltestellen

Haltestellen-Daten stammen von den Mobilitätsverbünden Österreich (MVO). Daten werden **täglich aktualisiert**, insbesondere bei Einführung neuer Haltestellen oder Änderungen bestehender Haltestellen.

Änderungen an Haltestellen-IDs sind daher jederzeit möglich und können sich auf Masten und Metaort-Zuordnungen auswirken.

Haltestellen verfügen über mehrere **ID-Typen**, um interne und externe Referenzen abzubilden:

- Aktuelle, eindeutige ID = `id`
 - Aktuelle, eindeutige ID
 - **Obligatorische ID** für alle API-Requests, da garantiert eindeutig
 - Interne HAFAS ID der Haltestelle inklusive Versionsinformationen (bei Datenupdate neu erzeugt)
 - komplexe Struktur, die neben der eigentlichen id der Haltestelle auch eine Versionierung der id enthält, welche die Haltestelle über die Datenupdates hinweg eindeutig macht.

Beispiel Haltestelle Westbahnhof:

```
id="A=1@O=Wien Westbahnhof@X=16337652@Y=48196654@U=81@L=490146800@B=1@p=1641949238@i=A@at:49:1468@"
```

- Interne ID ohne Versionierung = `extId`
 - vereinfachte interne Referenz
 - **Optionale Nutzung, da nicht garantiert eindeutig**
 - HAFAS ID der Haltestelle ohne Versionierung
 - 9-stellig, die letzten zwei Ziffern codieren die Hierarchie innerhalb einer Haltestelle.
 - Regel:
 - Endung 00 → Hauptmast (Haupt-Haltestelle)



- Endung 01–59 → einzelne Steige der Haltestelle

Beispiel: Haltestelle Westbahnhof: `extId="490146800"`

- Globale, externe ID (IFOPT-konform) = `altId`
 - globale, externe ID (IFOPT-konform)
 - **Optional, ermöglicht übergreifende Referenzen**
 - persistente Vergabe der globalen ID erfolgt durch den zuständigen Verkehrsverbund
 - Kann alternativ in Requests verwendet werden, besonders bei übergreifender Datenintegration

Beispiel: Haltestelle Westbahnhof: `<altId>at:49:1468</altId>`

IDs werden bei jedem Datenupdate neu generiert. **Eine langfristige Speicherung wird nicht empfohlen.** Die aktuelle ID sollte immer über den **Location.name Service** bezogen werden.

5.10 ÖV-Verkehrsmittel

- ÖV Verkehrsmittel sind in 12 Produktklassen abgebildet
- als Bit-Maske (`cls`) in Requests/Responses
- Filterung von Verkehrsmittel über Parameter `products` in den Services möglich
- Interpretation in Response über die Attribute `name`, `catOutL`, `line`, `num`.
- AST-Fahrten (Anrufsammeltaxi) sind nur in Kombination sichtbar, nicht als Einzelabfrage

Weiterführende Information → Abschnitt 21.5

5.11 Adressdaten

Adressen werden über das Bundesamt für Eich- und Vermessungswesen (BEV) bereitgestellt Adressen werden über GIP-Koordinaten am GIP-Graphen verortet.

Im Ausland werden OSM Adressdaten verwendet (bestimmte Korridore, nicht flächendeckend im Ausland).

5.12 Georeferenzierung

Alle Geodaten sind in **WGS84-Koordinaten** referenziert.

- Alle Koordinatenangaben in der VAO-Schnittstelle erfolgen in WGS84
- Dezimalwerte für Längengrad (Longitude) und Breitengrad (Latitude).

5.13 Routing-Graph (Kartengrundlage)

Die Graphenintegrationsplattform (GIP) ist das amtliche Verkehrsreferenzsystem Österreichs und bildet die Grundlage für das Routing innerhalb Österreichs: <https://www.gip.gv.at/>

- Sie enthält das gesamte Straßen-, ÖV- und Fußwegenetz, Haltestellen, Wegebeziehungen.



- Konsistenz und Aktualisierung: Der GIP-Graph wird regelmäßig gepflegt, inklusive Anpassungen durch neue Straßen, Haltestellenänderungen oder Netzoptimierungen.
- Der Standard-Updatezyklus beträgt **zwei Monate**

Im Ausland wird die Open Street Map Karte auf bestimmten Korridoren verwendet.

Nicht alle Ortspunkte sind zwingend mit jeder Modalität erreichbar; beispielsweise sind Berggipfel oft nur zu Fuß erreichbar.

Weiterführende Information → Abschnitt 23

5.14 Snapping und Einschränkungen bei frei gewählten Koordinaten

- Eingaben von frei gewählten Koordinaten als Start- oder Zielpunkt ist möglich
- Die Koordinaten werden, wenn möglich (räumliche Nähe) im Routingprozess auf einen passenden Ortspunkt aufgelöst
- Ist kein Snapping auf einen Ortspunkt möglich, wird der Koordinaten-Input für die Routensuche verwendet
- **Einschränkung:** Kann für dieses Koordinaten-Tupel kein passender Wegabschnitt mit entsprechender Nutzungserlaubnis für die gewählte Modalität gefunden werden, wird keine Route berechnet.
 - Dies kann sowohl bei abgelegenen Punkten (z. B. Berggipfel) als auch bei isolierten Netzbereichen im Siedlungsgebiet vorkommen.
 - Bei nicht angebundenen Koordinaten liefert die API einen **INT_ERR**.

Bei automatisierten Tests sollte sichergestellt werden, dass Koordinaten so gesetzt sind, dass eine Netzanbindung existiert um INT-ERR zu vermeiden. Freie Koordinaten sollten nur dann genutzt werden, wenn die Unsicherheit über die Netzanbindung akzeptiert wird.

Praxis-Tipp:

Verwendung primär von **existierende Ortspunkte** (Haltestellen, POIs, Adressen) statt frei gewählter Koordinaten. Abfragen idealerweise mit einem **Location.name Request** starten, um gültige Ortspunkt-IDs zu erhalten.

5.15 IV-Routing

Die Geometrie einer Individualverkehrs-Route (IV) wird in der VAO REST API segmentweise entlang des berechneten Straßennetzes ausgegeben und dient primär der kartografischen Darstellung sowie der fachlichen Auswertung einzelner Routenabschnitte.

- Struktur
 - Eine IV-Route besteht aus einem oder mehreren **<Leg>**-Elementen.
 - Innerhalb eines **<Leg>** ist die Streckengeometrie im Element **<GisRoute>** enthalten.
 - Die Route ist weiter in einzelne Segmente (**<Seg>**) unterteilt, die jeweils einen homogenen Straßenabschnitt repräsentieren.
- Geometrische Repräsentation
 - Die Geometrie eines Segments wird entweder



- als Koordinatenliste (WGS84, lon/lat) oder
 - als Google Encoded Polyline (bei aktivem `polyEnc`-Parameter) ausgegeben.
- Segmentbezogene Metadaten
 - Jedem Segment können zusätzliche fachliche Informationen zugeordnet sein, z. B.:
 - Mautpflicht / Vignette / Sondermaut
 - Kurzparkzonen
 - Verkehrsmeldungen oder Baustellen
 - Straßentyp oder Einschränkungen
 - Via-Ort
 - Diese Informationen werden als `<Note>` direkt am betroffenen Segment ausgegeben und gelten ausschließlich für den jeweiligen Abschnitt.

Für eine korrekte Verarbeitung müssen kodierte Polylines vor der Verwendung dekodiert bzw. unescaped werden.

Weiterführende Information → Abschnitt 11.6

5.16 ÖV Routing

Der Standardsuchmodus der ÖV Verbindungssuche liefert die **schnellsten und bequemsten Fahrten (möglichst wenige Umsteige)**. Der Suchmodus kann angepasst werden.

ÖV-Routen in der VAO REST API werden tripbasiert ausgegeben, wobei jede Fahrt (`<Trip>`) aus einer oder mehreren Teilstrecken (`<Leg>`) besteht. Jede Teilstrecke beschreibt einen durchgehenden ÖV-Linienabschnitt, z. B. Zug, Bus oder Straßenbahn.

- Struktur
 - Trip (`<Trip>`): Gesamte Verbindung von Start- zu Zielhaltestelle.
 - Leg (`<Leg>`): Einzelne Linienfahrt mit Angaben zu Linie, Fahrzeugtyp, Fahrzeiten und Distanz.
 - Origin/Destination: Start- und Endpunkt jedes Legs mit Koordinaten (WGS84), Echtzeitprognose, Plattforminformationen und ggf. MainMast (Hauptmast).
 - Verkehrsmittel (`<Product>`): Enthält Linieninformationen, Betreiber, Kategorie und ggf. Zusatzdienste (z. B. WLAN, Fahrradmitnahme, Bordservice).
- Geometrie
 - Die Route wird über die Haltestellenkoordinaten der Legs repräsentiert.
 - Einzelne Legs enthalten `RouteIdx`, um die Reihenfolge der Haltestellen und die Zuordnung von Zusatzinformationen zu markieren.
 - Die Geometrie selbst ist punktbasiert, nicht als kontinuierliche Polyline wie bei IV-Routen
- Segmentbezogene Zusatzinformationen
 - Je Leg können Hinweistexte (`<Note>`) definiert sein, z. B.:
 - Barrierefreiheit (Rollstuhlstellplatz, Niederflurfahrzeug)



- WLAN, Bordservice, Fahrradmitnahme
- Einschränkungen, Tarifinformationen oder Hinweise zum Einstieg
- Diese gelten nur für die entsprechenden Routeldx-Bereiche des Legs.
- Echtzeit- und Umweltdaten
 - Echtzeitinformationen zu Abfahrten, Ankünften und Verspätungen (`rtTime`, `rtTrack`).
 - CO₂-Emissionen und Distanz (`<Eco>`) zur Umweltbewertung.

ÖV-Routen sind haltestellenbasiert, Änderungen im Fahrplan oder Liniennetz wirken sich auf die Legs aus. Für Visualisierungen sollten Legs in der Reihenfolge von `routeIdx` genutzt werden, um die richtige Abfolge der Haltestellen zu gewährleisten.

Weiterführende Information → Abschnitt 11.5

5.17 Realtime-Daten (ÖV)

Echtzeitdaten zu Verspätungen, Gleisänderungen, Ausfällen werden **standardmäßig im Routing mitberücksichtigt**

- Deaktivieren der Echtzeit durch den Parameter `rtMode=OFF`
- Verbindungen werden auf Basis SOLL-Daten gerechnet → Echtzeit ergänzt → ggf. IST-Berechnung.
- nächste Verbindungen, die aufgrund der Verspätung nicht mehr erreicht werden können, werden ausgegeben und mit dem Attribut `reachable=false` gekennzeichnet (Zweck: Info an den Nutzer, dass eine Verspätung vorliegt)
- Echtzeit-Attribute in der Response sind mit "rt" gekennzeichnet: `rtDate`, `rtTime`, `rtTrack`, `rtDepTime`, `rtArrTime`, `rtDepTrack`, `rtArrTrack`.
- Verbindungsausfälle sind markiert mit: `cancelled=true`
- Infotexte geben Information über die Art der Störung: `<Notes>` mit `type=R` (Realtime), `type=A` (Attribut)

Weiterführende Information → Abschnitt 11.5.2

5.18 Realtime-Daten (IV)

Verkehrslage (Dynamiken) **standardmäßig aktiv** und wird im Routing berücksichtigt

- Deaktivieren der Dynamiken über den Parameter `totalCar`
- Verkehrsmeldungen, Parkinformationen, Maut: **default inaktiv**, müssen über den Parameter `totalMeta` aktiviert werden.

Weiterführende Information → Abschnitt 11.6.3



5.19 Einschränkungen bei Routing im Ausland

- **Intermodalität:** Im Ausland ist kein intermodales Routing verfügbar; nur reines PKW- oder ÖV-Routing ist möglich.
- **ÖV-Routing:**
 - Nur monomodales Routing von Haltestelle zu Haltestelle.
 - Adressen oder POIs können nicht als Start oder Ziel verwendet werden.
 - Datenaustausch besteht nur mit bestimmten Partnerregionen: Südtirol, Südbayern (Allgäu und Bodenseekreis) und Tschechien.
- **PKW-Routing:**
 - Möglich für Nachbarländer: Deutschland, Schweiz, Niederlande, Ungarn, Tschechien, Slowakei, Liechtenstein, Italien, Slowenien, Kroatien.
 - Routing basiert auf OpenStreetMap (OSM) Graphen.
 - E-Auto Routing ist aktuell nur innerhalb Österreichs verfügbar.

Weiterführende Information → Abschnitt 24

5.20 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.
- Im der Trip Response werden bei Start und Ziel die Wetterinformationen ausgegeben
 - als <Note>-Element mit `key="WEATHER"` und enthalten Wetterzustand, Temperatur und Windgeschwindigkeit
 - z. B.: `<Note key="WEATHER" type="A" txtN="bewölkt, 0.5 °C, Wind 14km/h"/>`
- In Location Details Response wird zu jedem Ortspunkt die Wetterinformation im Detail ausgegeben

5.21 Daten zu E-Auto Ladestellen

Die VAO bezieht öffentliche Ladestellen über die E-Control, basierend auf der Meldepflicht in Österreich.

Bei der Erfassung von Ladestationen über die E-Control ist die Information zu Zahlungsmittel und Ökostrom noch verpflichtend, weshalb Vollständigkeit nicht zugesichert werden kann.

Weiterführende Information → Abschnitt 11.6

5.22 Polylinie-Encoding & Geometrien

Routen-Geometrien werden in komprimierter Form übertragen.

- Die Polylinie muss aktiv mit dem Parameter `poly=1` aktiviert werden
- Ausgabe: Koordinaten oder Google Encoded Polyline (`polyEnc=GPA`).



- Verwendung von **Google Encoded Polylines** zur Reduktion der Datenmenge
- Alternativ strukturierte Koordinatenlisten je nach Service
- Häufige Fehlerquelle: JSON-reservierte Zeichen nicht decodiert.

Weiterführende Information → Abschnitt 22.6

5.23 Fehlerbehandlung

- HTTP-Statuscodes: 200 = OK, 400 = Bad Request, 404 = Not Found, 500 = Server Error.
- API-spezifische Fehlercodes finden sich im Response-Body.
- Empfehlungen: Request-IDs verwenden, Retry-Logik implementieren.

Weiterführende Information → Abschnitt 20

5.24 Rate-Limits & Nutzung

- Maximale Requests pro Minute/Stunde sind abhängig vom API-Key (Vertragsmodell)
- Bei Überschreitung der Quotas → Error HTTP status code 429

5.25 Session-Management

- Rekonstruktions-Kontexte (z. B. Reconstruction-Service) nicht langfristig speichern.
 - Diese dienen zur sofortigen Rekonstruktion, z.B. für die Übergabe einer Route in eine andere Applikation, oder zum Teilen der Route
- Logging: keine langfristige Speicherung von IDs oder Fahrtdaten
 - immer aktuelle Abfragen verwenden.
- Haltestellen-Hierarchie beachten
 - Fahrten sind mastscharf zugeordnet
 - Requests starten idealerweise mit Hauptmasten.
- Metaort-Haltestellen nutzen nur für generische Routen ohne konkrete Haltestelle.

6 SERVICE-ÜBERSICHT

Dieses Kapitel gibt einen Überblick über die bei der VAO REST API verfügbaren Services. Für jeden Service werden die wesentlichen Request-Parameter beschrieben. Parameter, die ausgegraut dargestellt werden, stehen aktuell nicht oder noch nicht zur Verfügung.

Hinweise zur Nutzung der Services:

- Die Services Location, Trip und StationBoard können direkt aufgerufen werden.
- Der Service JourneyDetail und LocationDetail muss über eine Referenz aus einem Ergebnis eines Trip, Departure oder ArrivalBoard Requests angefordert werden.
- Der GisRoute Request, der für die Berechnung von Vor- und Nachläufen vorgesehen ist, wird ebenfalls über eine Referenz aus einem Trip-Request ermittelt.
- Die Services TimetableInfo und DataInfo liefern wichtige Metadaten zu den ÖV-Daten.

Tabelle: Übersicht über die API Services

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)

Der `Location.name` Service stellt die vorgelagerte Abfragekomponente dar und bildet die Grundlage für die Nutzung weiterer Services, insbesondere der **Verbindungsabfrage** (`Trip`). Über diesen Service können die **verfügbaren Ortspunkte und deren eindeutige IDs identifiziert** werden, um sie in nachfolgenden Endpunkten zu verwenden.

Durch die Eingabe eines Ortsnamens wird eine Ortspunktsuche über alle in VAO verfügbaren Ortspunkte (Adressen, ÖV-Haltestellen und Points of Interest – POIs) durchgeführt. Ein Pattern-Matching-Algorithmus ermittelt die besten Treffer und liefert eine Liste von Ortspunkten, geordnet nach Übereinstimmung.

Aus dem Ergebnis der Suche kann anschließend ein Ortspunkt ausgewählt werden, um ihn als **Start- oder Zielpunkt** für eine Trip-Abfrage oder als **Input für Abfahrts- und Ankunftstafeln**, `Departure-/ArrivalBoard Services` (nur Haltestellen/Stationen) zu verwenden.

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.

Aufbau der Response:

- Root Element: `LocationList`.
- Ortspunkte: `CoordLocation`
 - Wichtige Attribute:
 - `Id`: interne ID mit Informationen zum Ortspunkt.
 - `extId`: externe ID (z. B. aus Drittquellen).
 - `name`: vollständiger Name des Ortspunkts.
 - `type`: Typ des Ortspunkts (A = Adresse, S = Station/Haltestelle, P = POI).
 - `lon / lat`: Koordinaten (Längengrad/Breitengrad).
 - Unterelemente:
 - `LocationNotes`: Enthält zusätzliche Metadaten:
 - `icon`: Darstellungssymbol für den Ortspunkt, inklusive Vorder- und Hintergrundfarbe.
 - `mcpInformation`: Informationen zu Typ, Belegung (occupancy) und ID.

7.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
input	Erforderlich	Text		Eingabefeld des zu suchenden Namens (Ortes)
maxNo	Optional	1-1000	10	Maximalanzahl der ausgegebenen Halte



type	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
products	Optional	Dezimal- wert $2^{\text{Produktklasse}}$	ALL	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (Abschnitt 21.5) 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: $2^2 + 2^4 = 20$
coordLong	Optional	Dezimal- grad	-	Längengrad der Koordinate
coordLat	Optional	Dezimal- grad	-	Breitengrad der Koordinate
r	Optional	1-10000	1000	Suchradius um die gegebene Koordinate in Metern
filterMode	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 (Abschnitt 7.3)
refined	Optional	-	-	Im Falle eines weiter aufzulösenden Ortspunktes enthält dieser Wert die id aus einem früher abgesetzten Request
poolId	Optional	103, 104, !103, !104	103	Filterung von inländischen und ausländischen Adressen <ul style="list-style-type: none">• Inländischer Adresspool: poolId=103• Ausländischer Adresspool: poolId=104• Ausschluss eines Adresspools mit vorgestellten „!“ Zeichen: poolId=!103• Kombination der Adresspools mit „ , “ : 103, 104
stations	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">• 41 Burgenland• 42 Kärnten• 43 NÖ



				<ul style="list-style-type: none">• 44 OÖ• 45 Salzburg• 46 Steiermark• 47 Tirol• 48 Vorarlberg• 49 Wien
meta	Optional	meta-Filter name		Metafilter nur in Kombination mit type=P für die Filterung nach POI Kategorien (Abschnitt 8.3)
operator	Optional	Alle operator codes oder Namen, die Data Information service gelistet werden		Filterung der Ortspunkte (Haltestellen) nach Betreiber (Operator). Mehrere Betreiber durch Komma trennen. Für eine negativ-Selektion, ein ! voranstellen.

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 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 (Abschnitt 7.3) 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.



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 vorgereiht , allerdings werden auch Ergebnisse ohne geografische Nähe angezeigt
EXCL_PERI	Ergebnisse innerhalb des Radius werden ausgeschlossen . Das heißt, alle Ergebnisse im Radius der eingegebenen Koordinate werden in der Ausgabe nicht berücksichtigt
SLCT_PERI	Nur Ergebnisse innerhalb des Radius werden in der Ausgabe berücksichtigt

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 führt eine Umkreissuche nach Ortspunkten **rund um einen Koordinaten-Mittelpunkt** durch. Ergebnis ist eine Liste von Ortspunkten (ÖV Haltestellen, POIs, Echtzeit-Fahrzeugstandorte) sortiert nach Distanz (geringste Distanz zuerst).

Besonderheit: Das Service umfasst nicht nur die gängigen Ortspunkte, sondern ist auch um sogenannte **EntryPoints** erweitert. **EntryPoints** sind Ortspunkte, die nicht für die Trip-Berechnung genutzt werden, aber als Karteninhalte relevant sind (z. B. WebCams, Bahnhofsvorplätze).

Hinweis: EntryPoints werden in der Response mit `entry="true"` gekennzeichnet. Sie dienen der Darstellung in Karten und Umkreissuchen, sind aber für Trip-Abfragen nicht nutzbar.

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.

8.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
originCoordLat	Erforderlich	Dezimal-grad	-	Breitengrad der Koordinate (Mittelpunktes)
originCoordLong	Erforderlich	Dezimal-grad	-	Längengrad der Koordinate (Mittelpunktes)
r	Optional	1-10000	1000	Suchradius um die Koordinate in Meter
maxNo	Optional	1-1000	10	Maximalanzahl der zurückgelieferten Halte
type	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.
products	Optional	Dezimal-wert 2 ^{Produktklasse}	ALL	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse (Abschnitt 21.5) in der Suche. Der Dezimalwert entspricht 2 ^{Produktklasse} . Für Kombinationen von Produktklassen ist die Summe zu bilden, wie folgt:



				<p>Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bit-Maske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$</p>
meta	Optional	meta Filter Name		<p>Metafilter, der die Suche und Ausgabe Sharing Stationen und Fahrzeugen, sowie Infrastruktur (Ladesäulen und VMOBIL Rad-Boxen) aktiviert (Abschnitt 8.2).</p> <p>Metafilter in Kombination mit type=P auch für die Filterung von POI Kategorien verwendbar (Abschnitt 8.3)</p>

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

Tabelle: Meta-Filter für bestimmte Inhalte

MCP Kategorie	API Meta-Filter	Verfügbarkeit	Beschreibung
Car-Sharing			
floMobil stationsbasiert	API_FLOMOBIL	regional (siehe Betreiberinformationen)	floMobil-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle Cantamen/Stadtwerke Wörgl).
Caruso stationsbasiert	API_CARUSO	regional (siehe Betreiberinformationen)	CARUSO-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle CARUSO).
WienMobil Auto stationsbasiert	API_WIENMOBIL AUTO	regional (Wien)	WienMobil-Auto-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle MOQO).
ShareToo stationsbasiert	API_SHARETOO	regional (Wien)	ShareToo-Auto-Stationen und Fahrzeuge werden in der Umkreissuche berücksichtigt (Quelle: Betreiberschnittstelle MOQO).
Bike-Sharing			
NEXTBIKE stationsbasiert	API_NEXTBIKE	regional	NEXTBIKE-Stationen werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle Nextbike). (inklusive RegioRad Tirol, Nextbikes)
CITYBIKE stationsbasiert	API_CITYBIKE	regional	CITYBIKE Stationen werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle Nextbike). WienMobil Rad, CityBike Linz
*LIME Fahrräder free-Floater	API_LIME_BIKE	regional (siehe	LIME Leihfahrrad-Standorte werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle LIME).



			Betreiberinforma tionen)	
Scooter-Sharing				
TIER free-Floater	Roller	API_TIER	regional	Service wurde vom Betreiber eingestellt. (Service in Wien vom Betreiber eingestellt).
*LIME free-Floater	Scooter	API_LIME_SCOO TER	regional (siehe Betreiberinforma tionen)	LIME Scooter-Fahrzeug-Standorte werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle LIME).
EASYWAY Roller		API_EASYWAY		Service vom Betreiber eingestellt
Sonstige				
VMOBIL Boxen	Fahrrad	API_VMOBIL	regional (Vorarlberg)	Standorte der VMOBIL Fahrrad-Boxen werden in der Umkreissuche berücksichtigt. (Quelle: Betreiberschnittstelle).
E-Auto (E-Control)	Ladesäulen	API_ECONTROL	Österreich	Ladesäulen (e-Auto) werden in der Umkreissuche berücksichtigt. (Quelle: E-Control Schnittstelle/ Ladestellenverzeichnis).
Park&Ride Anlagen		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

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>
    <LocationNote key="providerName" type="U">TIER</LocationNote>
    <LocationNote key="availability" type="U">A</LocationNote>
    <LocationNote key="mcpId" type="U">tier:62c6cddb-cdae-4a16-889d-9114ba7bc77b</LocationNote>
    <LocationNote key="externalId" type="U">62c6cddb-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>
```

8.3 Meta-Filter für POI-Filterung

Mit dem passenden Meta-Filter kann der Parameter `meta` genutzt werden, um die Suchergebnisse **gezielt auf eine bestimmte POI-Kategorie zu beschränken**.

- Points of Interest (POIs) sind in Kategorien zusammengefasst und werden in der Response als `LocationNote` ausgewiesen, z. B.:

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

- Meta-Filter für POIs können **nur mit Ortspunkt-Typ `type=PE` verwendet werden**. Eine Kombination mehrerer POI-Kategorien ist möglich.
- Meta-Filter für POIs **dürfen nicht kombiniert** werden mit **Meta-Filtern für andere Karteninhalte** (Abschnitt 8.4) wie z. B. Sharing-Provider.
- Einige POIs dienen nur der Kartendarstellung und sind für die Routenberechnung gesperrt, z. B. Almen, Gipfel oder Asfinag-Webcams. Verwendung solcher POIs in Trip-Abfragen führt zu Fehlern.

- In der Response gekennzeichnet durch:

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

Beispielabfragen:

Filter einzelner Kategorien: Beispiel Radservicestationen

```
{{baseURL}}/{{version}}/location.nearbystops?accessId={{key}}&format=JSON&originCoordL
ong=16.386663&originCoordLat=48.231137&type=PE&meta=API_PCAT_RADSERVICESTATION
```

Tabelle: Meta-Filter für POI-Kategorien

Kategorie	Meta-Filter Name	Anmerkung
Go-Vertriebsstelle	<code>API_PCAT_COVERTRIEB</code>	ASFINAG Vertriebsstellen (Vignette, GO-Maut) (Quelle: ASFINAG)
Rastplatz (PKW/LKW)	<code>API_PCAT_RASTPLATZ</code>	ASFINAG Rastplatz (Quelle: ASFINAG)
Raststation (PKW/LKW)	<code>API_PCAT_RASTSTATION</code>	ASFINAG Raststation (Quelle: ASFINAG)
Webcam (Autobahnen und Wien)	<code>API_PCAT_WEBCAM</code>	WebCams (Quelle: ASFINAG, ITS Vienna Region)
Parkgarage oder Parkfläche	<code>API_PCAT_PARKGARAGE</code>	Parkgaragen, Parkflächen (Quelle: ÖAMTC)
Berg-/ Passstraße	<code>API_PCAT_BERGSTRASSE</code>	Berg- und Passstraße (Quelle: ÖAMTC)
Mautstraße	<code>API_PCAT_MAUTSTRASSE</code>	Mautstraßen (Quelle: ÖAMTC)
Tunnel	<code>API_PCAT_TUNNEL</code>	Tunnel (Quelle: ÖAMTC)
ÖAMTC Standort	<code>API_PCAT_OEAMTC</code>	ÖAMTC Standorte (Quelle: ÖAMTC)
Bundesministerien	<code>API_PCAT_BUNDESMINISTERIUM</code>	Bundesministerien (Quelle: VAO)
Krankenanstalt	<code>API_PCAT_KRANKENANSTALT</code>	Krankenanstalt (Quelle: VAO)
Diplomatische Vertretung	<code>API_PCAT_DIPLVERTRETUNG</code>	Diplomatische Vertretungen (Quelle: VAO)
Anschlussstelle	<code>API_PCAT_ANSCHLUSSSTELLE</code>	Autobahn-Anschluss-stelle/Knoten ASFINAG Raststation (Quelle: ASFINAG)



Fahrradabstellanlage	API_PCAT_FAHRRADABSTELL	Fahrradabstellanlagen und Radboxanlage (Quelle: ÖVDAT)
ARBÖ-Standort	API_PCAT_ARBOE	ARBÖ Standorte (Quelle: VAO)
ASFINAG Standort	API_PCAT_ASFINAG	ASFINAG Standorte (Quelle: ASFINAG)
Vertriebsstelle GO Box	API_PCAT_GOBBOX	ASFINAG Vertriebsstellen GO-BOX (Quelle: ASFINAG)
ASFINAG Vignette	API_PCAT_VIGNETTE	ASFINAG Vignette (Quelle: ASFINAG)
VideoMaut	API_PCAT_VIDEOMAUT	ASFINAG digitale Streckenmaut (Quelle: ASFINAG)
Bank/Geld	API_PCAT_BANK	Quelle: ÖAMTC
Camping	API_PCAT_CAMPING	Quelle: ÖAMTC
Casino	API_PCAT_CASINO	Quelle: ÖAMTC
Einkaufen	API_PCAT_EINKAUFEN	Quelle: ÖAMTC
Essen	API_PCAT_ESSEN	Quelle: ÖAMTC
Freizeitpark	API_PCAT_FREIZEITPARK	Quelle: ÖAMTC
Apotheke	API_PCAT_APOTHEKE	Quelle: ÖAMTC
Kino	API_PCAT_KINO	Quelle: ÖAMTC
Theater	API_PCAT_THEATER	Quelle: ÖAMTC
Touristeninformation	API_PCAT_TOURISTINFO	Quelle: ÖAMTC
Unterkunft	API_PCAT_UNTERKUNFT	Quelle: ÖAMTC
Markt	API_PCAT_MARKT	Quelle: ÖAMTC
Veranstaltungen/Festivals	API_PCAT_KULTUR	Quelle: ÖAMTC
Wassersport	API_PCAT_WASSERSPORT	Quelle: ÖAMTC
Weihnachtsmarkt	API_PCAT_WEIHNACHTSMARKT	Quelle: ÖAMTC
Wintersport	API_PCAT_WINTERSPORT	Quelle: ÖAMTC
Tankstellen	API_PCAT_TANKEN	Quelle: ÖAMTC
Tanken & Spritpreis	API_PCAT_TANKENPREIS	Quelle: ÖAMTC
Taxistand	API_PCAT_TAXISTAND	Quelle: WKO
Park & Ride Anlage	API_PCAT_PR	Quelle: ÖAMTC
Sehenswürdigkeit	API_PCAT_SEHENSWERDIGKEIT	Quelle: ÖAMTC
Bildung	API_PCAT_BILDUNG	Quelle: ÖAMTC
Museum	API_PCAT_MUSEUM	Quelle: ÖAMTC
Park	API_PCAT_PARK	Quelle: ÖAMTC
Coming soon: Tiergarten	API_PCAT_ZOO	Quelle: ÖAMTC
Almen	API_PCAT_ALMEN	Quelle: Land Tirol
Gipfel	API_PCAT_GIPFEL	Quelle: Land Tirol
Flughafen	API_PCAT_FLUGHAFEN	Quelle: VAO
Friedhof	API_PCAT_FRIEDHOF	Quelle: VAO
TIM Standort	API_PCAT_TIM	Grazer und Linzer Mobilitätsdienst
WienMobil	API_PCAT_WIENMOBIL	Wien-Mobil-Standorte
Park+Drive	API_PCAT_PARKANDDRIVE	Quelle: Asfinag
Paketboxen	API_PCAT_PAKETBOXEN	Quelle: BMK
Radservicestationen	API_PCAT_RADSERVICESTATION	Quelle: ÖVDAT
Sonstiges	API_PCAT_SONSTIGES	Sammelkategorie



8.4 Entry Points im Detail

- Station EntryPoints (SE): Zugangspunkte zu Haltestellen, von denen Fußwege automatisch zu den An-/Abfahrtsmasten führen. Beispiel: Bahnhofsvorplatz → Fußweg → Bahnsteig.
- POI EntryPoints (PE): Karteninhalte wie WebCams oder sonstige POIs, die in der Umkreissuche gefunden werden, aber nicht für Routing verfügbar sind.
- Wichtig: Für Trip-Berechnungen müssen EntryPoints nicht explizit genutzt werden, da die Verbindung automatisch von/zur Station über die EntryPoints berechnet wird.

Praxistipp: Für eine **komplette Umkreissuche** aller Inhalte in der VAO: Stationen + Station EntryPoints (**type=SE**) verwenden. Für POIs + POI EntryPoints (**type=PE**) verwenden. Für Trip-Berechnungen nur S und P verwenden.

8.5 Sharing-Angebote & Echtzeit-Fahrzeugposition

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 8.4).

Stationsbasierte Provider: Response enthält `coordLocation` für die Station sowie die `childLocation` für verfügbare Fahrzeuge.

Free-Floater Provider: Fahrzeuge erscheinen direkt als `childLocation`.

Detailabfrage: Mit `childLocation id` können weitere Fahrzeugdetails über `location.details` abgerufen werden (Detailinformationen zum Fahrzeug, etc.). Dort werden alle Informationen ausgegeben, die über die Sharing-Provider-Schnittstelle zur Verfügung gestellt werden.



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 VMOBIL 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.



9.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
id	Erforderlich		-	Id des Ortpunktes (Haltestelle, Fahrzeuges, Ladestation, VMOBIL Box) über den weitere Details abgefragt werden sollen (muss durch ein location services vorher abgefragt werden).
weather	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

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>
```

Beispiel Abfrage TIER Roller:

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{version}/location.details?accessId={{accessId}}&id=A=2048@O=TIERNö128742@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>
  </LocationNotes>
```




```
<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</Loc  
ationNote></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
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
originCoordLat	Erforderlich	siehe 5.1	-	Breitengrad der Koordinate
originCoordLong	Erforderlich	siehe 5.1	-	Längengrad der Koordinate
maxNo	Optional	1-1000	10	Veraltet/aufgelassen
type	Optional	A, S, P, AP, SA, SAP, SP, C	C	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@L=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 berechnet Routen zwischen einem definierten Start- und Zielpunkt (Origin / Destination). Es unterstützt sämtliche in der VAO verfügbaren Modalitäten (`groupFilter`), einschließlich öffentlichem Verkehr (ÖV), Individualverkehr (PKW, Fahrrad, Fuß, E-Auto) sowie intermodale Kombinationen dieser Verkehrsmittel.

Das Ergebnis einer Anfrage ist eine Liste von möglichen Verbindungen (`Trips`) . Jede Verbindung besteht aus einzelnen Abschnitten (`Legs`), für die Basisinformationen wie Abfahrts- und Ankunftszeit, Dauer, Distanz sowie – sofern verfügbar – Echtzeitinformationen ausgegeben werden.

Die Trip-Response enthält zusätzlich Informationen zum CO₂-Ausstoß der jeweiligen Verbindung. Der Wert wird im Element `Eco/co2` in Kilogramm angegeben und ermöglicht einen Vergleich mit einer PKW-Referenzverbindung zur Bewertung der CO₂-Ersparnis.

Start und Zieldefinition

- Verwendung von Ortspunkten (Haltestellen, Adressen, POIs)
 - Abfragen des Trip Service können mit gültigen Ortspunkten erfolgen, die über die Parameter `originId` und `destId` übergeben werden.
 - Dabei sind die Hinweise zur Erreichbarkeit von Ortspunkten und deren Anbindung an das Verkehrsnetz zu beachten (siehe Abschnitt 22.5).
 - Für Ortspunkte können sowohl die `id` (empfohlen) als auch die `extId` (nur für bestimmte Zwecke) verwendet werden (siehe Abschnitt 5.9) . Diese IDs können über die Location Services ermittelt werden (vorrangig `Location.name`)
- Verwendung von Koordinaten
- Startpunkt: `originCoordLat`, `originCoordLong`
- Zielpunkt: `destCoordLat`, `destCoordLong`
- Koordinaten sind grundsätzlich frei wählbar
- Koordinaten werden – soweit möglich – auf Ortspunkte gesnappt
- Es gibt Einschränkungen bei der Verwendung von Koordinaten bezüglich Netzanbindung (→ siehe Abschnitt 5.14)

Modalitäten und Kombinationslogik

- Die Standard-Modalität eines Trip Requests ist der öffentliche Verkehr (ÖV).
- Über den Parameter `groupFilter` können Abfragen jedoch gezielt für einzelne Modalitäten oder für multimodale Kombinationen von Verkehrsmitteln durchgeführt werden.

Weiterführende Information → Abschnitt 11,2

Fußwege zu und von der Haltestelle

- Für die Darstellung der gesamte Wegbeschreibung inklusive Vorlauf und Nachlauf des **ÖV-Anteiles ist der Abruf der Polylinie** erforderlich.
- Diese muss mit der in der Trip Response zurückgegebenen `GisRef` ein `GisRoute Request` angestoßen werden.
- auch bei kombinierten ÖV-Routen (Bike&Ride, Park&Ride) erforderlich die einen IV-Vorlauf haben.



Tarifinformationen

- sind standardmäßig ausgeschaltet
- mit dem Parameter `tariff=1` zu aktivieren

Der Abfragezeitpunkt

kann durch Parameter `date` und `time` angepasst werden.

bei der Abfrage in die Zukunft und in die Vergangenheit gibt es bestimmte Faktoren zu berücksichtigen:

Weiterführende Information → Abschnitt 21.1.3

11.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	Erforderlich	-	-	Access Id zur Authentifizierung
originId	Erforderlich	siehe Abschnitt 5.4.2	-	Legt den Startpunkt fest. Diese kann über das <code>location.name-</code> oder <code>location.nearbystops-</code> Service gefunden werden.
destId	Erforderlich	siehe Abschnitt 5.4.2		Legt den Zielpunkt fest. Diese kann über das <code>location.name-</code> oder <code>location.nearbystops-</code> Service gefunden werden.
originExtId			-	Veraltet/aufgelassen
destExtId				Veraltet/aufgelassen
originCoordLat	Erforderlich wenn Id nicht angegeben ist	siehe Abschnitt 5.2	-	Breitengrad des Routenstartpunkts
originCoordLong	Optional	siehe Abschnitt 5.2	-	Längengrad des Routenstartpunkts. Dieser kann über das <code>location.name-</code> oder <code>location.nearbystops-</code> Service ermittelt oder innerhalb von Österreich frei gewählt werden.
destCoordLat	Erforderlich wenn Id nicht angegeben ist	siehe Abschnitt 5.2	-	Breitengrad des Routenzielpunkts. Dieser kann über das <code>location.name-</code> oder <code>location.nearbystops-</code> Service ermittelt oder innerhalb von Österreich frei gewählt werden.
via	Optional	<code>viaId wait time prod ucts</code>	-	Komplexe Struktur für die Definition von mehreren VIA und VIA Parametern. Die Via werden mit ; (semicolon) getrennt.



				<p>Der Aufbau sieht wie folgt aus: <code>viaId waittime products</code></p> <p><code>viaId</code>: → erforderlich</p> <ul style="list-style-type: none">- <code>Id</code> oder <code>extId</code> des VIA; die ID kann über das <code>location.name</code>- oder <code>location.nearbystops</code>-Service erhalten werden.- Koordinaten (nur im IV) wie folgt definiert werden: <code>Via=geo:x,y</code> z.B. <code>Via=geo:14.667915,47.169457</code>- Angabe von mehreren Koordinaten: <code>Via=geo:14.397096,48.042723;geo:14.662773,47.17476;geo:16.573277,48.576044</code> <p><code>waittime</code>: → optional Zeitverbleib im VIA- Halt in Minuten</p> <p><code>products</code>: → optional Produktklassen (nur bei ÖV)</p> <p>Beispiel 1: Drei VIA die durchfahren werden sollen durch <code>extID</code> festgelegt: <code>via=801234;801235;801236</code></p> <p>Beispiel 2: Zwei VIA mit 10min und 20min Wartezeit: <code>via=801234 10;801235 20</code></p>
viaId	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</p> <p>Wird <code>via</code> genutzt, sind <code>viaId</code> und <code>viaWaitTime</code> wirkungslos</p>
viaWaitTime	Optional		0	<p>Legt die Verweilzeit in der Station (in Minuten) fest.</p> <p>Wird <code>via</code> genutzt, sind <code>viaId</code> und <code>viaWaitTime</code> wirkungslos.</p>
minChangeTime	Optional		-	<p>Minimale Umstiegszeit in Minuten.</p>
maxChangeTime	Optional		-	<p>Maximale Umstiegszeit in Minuten.</p>
addChangeTime	Optional		-	<p>Zeitwert, welcher der Umstiegszeit bei jedem Halt zugerechnet wird.</p>
maxChange	Optional	0 – 11	-	<p>Maximale Anzahl der Umstiege</p>



date	Optional		Aktuelles Server-Datum	Setzt das Datum für welches Abfahrten gesucht werden im Format YYYY-MM-DD (Abfahrtsdatum).
time	Optional		Aktuelle Server-Zeit	Setzt die Startzeit für welche Abfahrten gesucht werden (Abfahrtszeit)
searchForArrival	Optional	0 / 1	0	Wenn gesetzt, definieren <code>date</code> und <code>time</code> die Ankunfts- nicht die Abfahrtszeit.
numF	Optional	1 - 6	5	Mindestanzahl von Fahrten nach der Suchzeit. Summe von <code>numF</code> und <code>numB</code> muss ≤ 6 betragen (also maximal 6).
numB	Optional	0 - 3	0	Mindestanzahl von Fahrten vor der Suchzeit. Summe von <code>numF</code> und <code>numB</code> muss ≤ 6 betragen (also maximal 6).
products	Optional	-	-	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse 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: $2^2 + 2^4 = 20$
categories	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>
context	Optional	Werte von <code>scrB</code> , <code>scrF</code> aus Trip Response	-	Definiert den Startpunkt für Vor- oder Rückwärtsblättern (Scrolling). Mit dem <code>scrB</code> - Wert eines früheren Ergebnisses kann in der Zeit zurückgeblättert werden und mit <code>scrF</code> kann vorwärts geblättert werden.



poly	Optional	0 / 1	0	Aktiviert/Deaktiviert die Polylinie für jede Etappe (leg) der Route.
polyEnc	Optional	GPA	-	Aktiviert die Ausgabe der Polylinie im Format „Google Encoded Polyline Format“.
passlist	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.
showPassingPoints	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>
originWalk	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 <code>originwalk=1,0,1000</code> lautet</p>
originBike	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 originbike=1,0,1000
originCar	Optional	0 / 1	-	<p>Aktiviert (originCar=1) /Deaktiviert (originCar=0) 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 originCar=1,2000,100000</p>
originPark	Optional	0 / 1	-	<p>Zur Kalibrierung des Park&Ride-Vorlaufes bei Auto-Routen (groupFilter API_PARK_AND_RIDE).</p> <p>Aktiviert (originPark=1) /Deaktiviert (originPark=0) Park & 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 & Ride- Vorlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet originPark=1,0,1000</p>
originMeta	Optional	siehe Abschnitt 11.5	-	<p>Aktiviert (originMeta=1) /Deaktiviert (originMeta=0) die Verwendung von vordefinierten IV-Profilen <metaprofil> im Vorlauf der Strecke.</p> <p>originMeta=<metaprofil> Mehrere Meta-Profile können mit , (Beistrich) getrennt mitgegeben werden.</p>
destWalk	Optional	0 / 1	1	<p>Aktiviert (destWalk=1) /Deaktiviert (destWalk=0) 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, können diese Werte (in Metern) mit</p>



				<p>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>
destBike	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>
destCar	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>
destPark	Optional	0 / 1	-	<p>Aktiviert (<code>destPark=1</code>) /Deaktiviert (<code>destPark=0</code>) den Park & 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 1 oder 0 (für aktiviert/deaktiviert) gesetzt werden.</p> <p>Beispiel:</p>



				Park & Ride- Nachlauf aktiv, minimale Distanz 0m, maximale Distanz 1000m → Parameter lautet destPark=1,0,1000
destMeta	Optional		-	Aktiviert (destMeta=1) /Deaktiviert (destMeta=0) die Verwendung von vordefinierten IV-Profilen <metaprofil> im Nachlauf der Strecke. destMeta=<metaprofil> Mehrere Meta-Profile können mit , (Beistrich) getrennt mitgegeben werden.
totalMeta	Optional		-	Aktiviert (totalMeta=1) /Deaktiviert (totalMeta=0) die Verwendung von vordefinierten IV-Profilen <metaprofil> im Verlauf der gesamten Strecke. totalMeta=<metaprofil> Mehrere Meta-Profile können mit , (Beistrich) getrennt mitgegeben werden.
totalCar	Optional			Aktiviert (totalCar=1)/Deaktiviert (totalCar=0) Autorouten für die gesamte Route und lässt die Konfiguration des IV Router zu
bikeCarriage	Optional	0 / 1	0	Aktiviert (bikeCarriage=1) /Deaktiviert (bikeCarriage=0) die Suche nach Routen, welche ausdrücklich Fahrradtransport (Fahrradmitnahme) erlauben.
rtMode	Optional	OFF, SERVER_DEFAULT	SERVER_DEFAULT	Aktiviert (1)/Deaktiviert (0) den Realtime-Modus bei der ÖV Routenberechnung. rtMode=OFF Verbindungen werden auf Basis der Plandaten berechnet. Echtzeitinformation werden ignoriert (RT- Info). rtMode=SERVER_DEFAULT Echtzeitinformationen werden in die Routenberechnung miteinbezogen.
unsharp	Optional	0 / 1	0	Veraltet/aufgelassen da nun standardmäßig eine unscharfe Suche bei der



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

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 *monomoden* 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 *monomoden* 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
Öffentlicher Verkehr monomodal	API_OEV	monomodale Modus sucht eine direkte ÖV Route mit allen Produktklassen
Auto monomodal	API_CAR	monomodaler Modus sucht eine direkte Auto-Route (IV)



e-Auto monomodal	API_ECAR	monomodaler Modus sucht eine direkte E-Auto Route (IV)
Fußweg monomodal	API_WALK	monomodaler Modus sucht eine direkte Fußwegroute (IV)
Fahrrad monomodal	API_BIKE	monomodaler Modus sucht eine direkte Fahrrad-Route (IV) mit Alltagsrad-Profil
Bike & Ride 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.
Bike & Ride mit VMobil Radboxen 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.
Fahrrad-mitnahme intermodal	API_BIKECARRIAGE	intermodaler Modus sucht eine Route bestehend aus einem Fahrradvorlauf sowie einer ÖV-Verbindung mit Fahrradbeförderung im Hauptlauf und einem Fahrradnachlauf.
Park & Ride 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.
Autoreisezug intermodal	API_CARTRAIN	intermodaler Modus sucht eine Route bestehend aus einem Autovorlauf, einem Autoreisezug im Hauptlauf und einem Autonachlauf.
groupFilter BIKE Sharing Provider (oft lokal begrenzt Angebote, nicht immer flächendeckend)		
NEXTBIKE monomodal	API_BIKESHR_NEXTBIKE_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem Leihfahrrad des Betreibers NEXTBIKE. Enthält komplette Nextbike-Abdeckung exklusive WienMobilRad und CityBike Linz (siehe groupFilter CityBike)
NEXTBIKE intermodal	API_BIKESHR_NEXTBIKE	intermodaler Modus sucht eine Route mit einem stationsbasiertem Leihfahrrad des Betreibers NEXTBIKE und ÖV/anderer Modalität.
CITYBIKE monomodal	API_BIKESHR_CITYBIKE_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem Leihfahrrad des Betreibers WienMobil Rad (ehemals CityBikes) und CityBike Linz.
CITYBIKE intermodal	API_BIKESHR_CITYBIKE	intermodaler Modus sucht eine kombinierte Route mit einem stationsbasiertem Leihfahrrad des Betreibers CITYBIKE und ÖV/anderer Modalität. WienMobil Rad (ehemals CityBikes) und CityBike Linz.
*LIME Bikes monomodal	API_BIKESH_LIME_ONLY	monomodaler Modus sucht eine Direktroute mit einem Leihfahrrad des Betreibers LIME.
*LIME Bikes intermodal	API_BIKESH_LIME	intermodaler Modus sucht eine kombinierte Route mit einem Leihfahrrad des Betreibers LIME und ÖV/anderer Modalität.
groupFilter SCOOTER Sharing Provider (oft lokal begrenzt Angebote, nicht immer flächendeckend)		
EASYWAY Roller monomodal	API_SCOOTERSHR_EASYWAY_ONLY	monomodaler Modus sucht eine Direktroute mit einem stationsbasiertem E-Roller des Betreibers EASYWAY.
EASYWAY Roller intermodal	API_SCOOTERSHR_EASYWAY	intermodaler Modus sucht eine kombinierte Route mit einem stationsbasiertem E-Roller des Betreibers EASYWAY und ÖV/anderer Modalität.



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



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

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

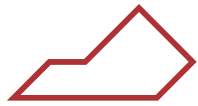
Beispiel für eine Abfrage an der Produktvariante Radrouter

https://radrouting.verkehrsauskunft.at/vao/restproxy/{version}/trip?accessId={accessId}&format=json&originExtId=490118400&destCoordLat=48.199126&destCoordLong=16.368503&groupFilter=API_BR_TOURINGBIKE

11.3 Meta-Profile 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.

Meta-Profile 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:					
Berücksichtigung von Verkehrsmeldungen in der Routenberechnung	totalCar	single	evnt	0/1	aktiviert (1)
Berücksichtigung von Sperren (Umfahren) in der Routenberechnung	totalCar	single	aevnt	0/1	aktiviert (1)
Berücksichtigung von Verkehrslage (Echtzeit & Statistik) in der Routenberechnung (Dynamik 3)	totalCar	single	htsta	0/1	aktiviert (1)
		single	cdyn	0/1	aktiviert (1)
Berücksichtigung aller Dynamiken (1-3) in der Routenberechnung	totalCar	multiple	1 evnt=0,aevnt=0,tsta=0,htsta=0,cdyn=0	0/1	aktiviert (1)
Parksuchzeiten (in Wien)	totalCar	single	1 getInitEndTimes=0	0/1	aktiviert (1)
Anbindung auf Autobahnen in der Routenberechnung	totalCar	single	1 allowFirstLastSarcHighwayConnection=1	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)					
	totalMeta	single	bike_speed_slow (10km/h)	--	normal

Radgeschwindigkeit			bike_speed_normal (15km/h)	--	normal
			bike_speed_fast (20km/h)	--	normal
			bike_speed_ebike (25km/h)	--	normal
Steigungen vermeiden (Fahrrad)	totalMeta	single	bike_noascents	--	deaktiviert
Radinfrastruktur bevorzugen	totalMeta	single	bike_preferbikeroute	--	deaktiviert
Radschieben vermeiden	totalMeta	single	bike_nopush	--	deaktiviert

11.4 Meta-Profile für E-Auto Routing

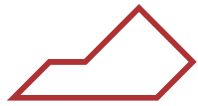
Die zu übergebenden Parameter für den IV Router werden als totalMeta Aufzählung übergeben. Die Werte sind alle nach dem Schema *ecar_Meta-Profil_Zahlenwert* vergeben.

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.

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
Meta-Profil für den Modus E-Auto (groupFilter=API_ECAR)							
Fahrzeugtechnische Parameter							
Verbrauch in kWh/100km (EC)	totalMeta	single	ecar_ec_X	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 30, 35	20		



Nutzbare Batterie-Kapazität in kWh (MC)	totalMeta	single	ecar_mc_x	20, 25, 30, 35, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110	35
Batterie-Ladestand bei Start in % (AC)	totalMeta	single	ecar_ac_x	100, 90, 80, 70, 60, 50, 40, 30, 20	80
Maximale Ladeleistung in kW (PC)	totalMeta	single	ecar_pc_x	25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350	50
Ladestellen-bezogene Parameter (nur eingeschränkt nutzbar *siehe wichtigen Hinweis)					
Steckertyp (PLT) (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
Zahlungsmittel (PAM)	totalMeta	multiple	ecar_pam_x	cash, rfid, credit_card, debit_card, app, sms, nfc, website	keiner
Ökostrom nutzen (GEN)	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	

11.5 Besonderheiten für ÖV Routensuche

Dieses Kapitel beschreibt spezifische Eigenschaften, Konfigurationsmöglichkeiten und Besonderheiten der ÖV-Routenberechnung in der VAO REST API. Es ergänzt die Beschreibung des Trip Service und verweist auf die zugrunde liegenden Algorithmen und Datenmodelle.

11.5.1 Suchmodi und Konfiguration der Routensuche im ÖV

Der Standardsuchmodus der ÖV-Routenberechnung liefert für einen gegebenen Suchzeitpunkt bevorzugt schnelle und komfortable Verbindungen mit möglichst wenigen Umstiegen. Ist die schnellste Verbindung keine Direktverbindung, werden zusätzlich sogenannte „bequeme“ Alternativen mit geringerer Umstiegsanzahl ermittelt, die nur geringfügig langsamer sind.

Optional kann der ökonomische Suchmodus (Parameter `economic`) aktiviert werden. In diesem Modus werden zusätzliche Suchläufe mit unterschiedlichen Bewertungsmethoden durchgeführt. Dadurch können auch gute **Direktverbindungen** ausgegeben werden, die **langsamer** als die Ergebnisse der Standardsuche sind.

Die ÖV-Routensuche arbeitet zudem mit einer **unscharfen Suche**, bei der umliegende Haltestellen rund um Start- und Zielpunkt in die Berechnung einbezogen werden. Entsprechende Fußwegvorläufe werden automatisch berücksichtigt.

11.5.2 Berücksichtigung von Echtzeitdaten

Bei der ÖV-Routenberechnung werden standardmäßig **Echtzeitinformationen** wie Verspätungen oder Gleisänderungen berücksichtigt. Die in der Response ausgegebene Reisedauer enthält diese Abweichungen bereits.

Soll die Routenauskunft ausschließlich auf Basis von **Plandaten (SOLL)** erfolgen, kann der Echtzeitmodus über den Parameter `rtMode=OFF` deaktiviert werden.

Weiterführende Information zu Fahrplandaten und Echtzeitdaten siehe Abschnitt 21.1 Fahrplandaten (SOLL) 21.2 Verarbeitung von Realtime-Daten (ÖV).

11.5.3 Vorwärts- und Rückwärts-Suche (Parameter `numB` und `numF`)

Standardmäßig werden **bis zu fünf (maximal) Routingvorschläge** ab dem angegebenen Suchzeitpunkt geliefert. Über die Parameter `numF` (Vorwärtssuche) und `numB` (Rückwärtssuche) kann das **zeitliche Suchfenster erweitert** werden:

- `numF`: Anzahl zusätzlicher Verbindungen nach dem Suchzeitpunkt (max. 6)
- `numB`: Anzahl zusätzlicher Verbindungen vor dem Suchzeitpunkt (max. 3)

Die **Summe** aus `numF` und `numB` darf **nicht größer als 6** sein.

Die Wirkung dieser Parameter hängt von der Suchrichtung ab, die über `searchForArrival` gesteuert wird:

- Bei Vorwärtssuche (`searchForArrival=0`) bezieht sich `numF` auf zukünftige Verbindungen
- Bei Rückwärtssuche (`searchForArrival=1`) bezieht sich `numF` auf vergangene Verbindungen

So kann man beispielsweise mit `numB=3` und `numF=3` jeweils 3 Ergebnisse vor und nach dem Suchzeitpunkt berechnen lassen.

Rückwärtssuchen sind algorithmisch komplexer und rechenintensiver und sollten daher nur gezielt eingesetzt werden.

11.5.4 Scrolling bei der ÖV-Routenberechnung (Parameter context)

Auf Basis eines bereits durchgeführten Trip Requests können weitere frühere oder spätere Verbindungen für dieselbe Route abgefragt werden („**Blättern in der Zeit**“). Das wird erreicht, durch Beibehaltung der ursprünglichen Request- Parameter und Festlegung eines Startpunktes für das Blättern mit Hilfe eines zusätzlichen Kontextparameter.

Jede Trip-Response enthält im Element `TripList` die Attribute:

- `scrB` (Scroll Backward)
- `scrF` (Scroll Forward)

Diese Werte können in einem Folge-Request als Parameter `context` verwendet werden. Bei unveränderten Request-Parametern liefert der Server daraufhin frühere bzw. spätere Verbindungen relativ zur ursprünglichen Anfrage.

11.5.5 Filterung von Verkehrsmittel (Produktklassen)

Für Trip-Requests können die zu berücksichtigenden **ÖV-Produktklassen** über den Parameter `products` eingeschränkt werden.

Die Filterung erfolgt über eine **Bit-Maske**, wobei jede Produktklasse durch eine Zweierpotenz repräsentiert ist. Die ÖV-Produktklassen und typische Bit-Masken → siehe Abschnitt 21.5.

- Beispiel: `products=64` entspricht 2^6 und damit der Produktklasse Bit 6 (Regionalbus).

Bei der Kombination mehrerer Produktklassen wird die Summe der jeweiligen Bitwerte gebildet. Beispielsweise ergibt sich für die Kombination von U-Bahn (Bit 2) und Straßenbahn (Bit 4):

- Beispiel: `products=20` entspricht $2^2 + 2^4 = 4 + 16 = 20$

Die Rückrechnung eines `products`-Werts auf einzelne Produktklassen kann auch arithmetisch erfolgen, ohne Bitoperatoren zu verwenden. Der `products`-Wert wird durch Zerlegung in Zweierpotenzen auf die enthaltenen Produktklassen zurückgeführt.

Einschränkung:

Die Filterung der Produktklasse Anrufsammeltaxi / Rufbus (AST, Produktklasse 10) ist bei Departure- und Arrivalboards nicht wirksam, da bedarfsorientierte Angebote keinen festen Fahrplan besitzen. Wird ausschließlich mit diesem Produktfilter angefragt, wird der Filter ignoriert und es werden alle verfügbaren Fahrten ausgegeben.

11.5.6 Zwischenziele (Via-Orte) im ÖV-Routing

Die ÖV-Routenberechnung kann um **Via-Orte** (Zwischenhalte) erweitert werden. Via-Orte können Haltestellen oder Adressen sein und sind **ausschließlich für monomodale ÖV-Routen** zulässig. Für kombinierte Routen stehen sie nicht zur Verfügung.

- Ein einzelner Via-Ort wird über `viaId` definiert.
- Eine optionale Verweilzeit kann über `viaWaitTime` (in Minuten) angegeben werden.
- Mehrere Via-Orte muss mit einer komplexen Struktur im Parameter `via` erfolgen:



- Aufbau: `viaId|waittime|viastatus|products`.
- Allerdings ist `viastatus` bei VAO nicht relevant, daher ist die Struktur folgendermaßen (ohne `viastatus`) mitzugeben: `viaId|waittime||products`
- Die IDs für den Parameter `VIA` und `viaID` werden über Locations Service identifiziert
- Alle Ortspunkte haben in der Response ein Attribut `ID` oder `extID`, das entsprechend hier weiterverwendet werden muss.

11.5.7 Vor- und Nachläufe bei adressgenauen ÖV-Routen

Bei adressgenauen ÖV-Routen sowie bei intermodalen Routen mit ÖV-Anteil existiert immer ein entsprechender **Vor- und Nachlauf** zwischen Adresse und Haltestelle. Dieser erfolgt je nach Konfiguration zu Fuß, mit dem Fahrrad oder mit dem PKW.

Standardmäßig ist für ÖV-Routen ein Vor- und Nachlauf von **2000 m** konfiguriert. Die Basisinformationen zu Länge und Dauer werden in den einzelnen **Legs** des Trip-Responses ausgegeben.

Für Vor- und Nachläufe können spezifische Modalitäten definiert werden, z. B.:

- Vorlauf: `originWalk`, `originBike`, `originCar`, `originPark`, `originMeta`
- Nachlauf: `destWalk`, `destBike`, `destCar`, `destPark`, `destMeta`

Für eine detaillierte kartografische Darstellung (z. B. Polylinien für Vor- und Nachläufe) muss anhand der im Trip-Response enthaltenen `gisRef` ein zusätzlicher `GisRoute-Request` (→ siehe Abschnitt 12 GIS Routen Service) durchgeführt werden. Dies gilt auch für kombinierte ÖV-Routen wie Park-&-Ride oder Bike-&-Ride.

11.5.8 Alternative Fußwegverbindungen

Bei ÖV-Abfragen wird geprüft, ob bei kurzen Luftliniendistanzen zwischen Start und Ziel (unter ca. **1 500 m**) eine reine Fußwegverbindung effizienter ist. In diesem Fall kann eine alternative Fußwegroute ausgegeben werden.

Zusätzlich werden bei nahe beieinanderliegenden Start- und Zielpunkten Fußwegverbindungen im Rahmen einer unscharfen Suche mitberechnet.

11.6 Besonderheiten für IV-Routensuche

11.6.1 Suchmodi und Konfiguration der Routensuche im IV

Standardmäßig wird bei IV-Routen die **schnellste Route** berechnet. VAO stellt **Meta-Profile** bereit, die die Routensuche anpassen können.

Meta-Profile

- Optimierung nach **Zeit oder Distanz**
- Anpassung von **Geschwindigkeiten** (Gehen, Auto)
- **Spezielle Konfigurationen für Radverkehr**
- Aktivierung/Deaktivierung von Verkehrsmeldungen oder Kurzparkzoneninformationen

Weiterführende Information → Abschnitt 11.3 und 11.4



Anbindung auf Autobahnen

- Start- oder Zielpunkte können standardmäßig nicht direkt auf der Autobahn gesetzt werden (nur Anschlussstellen).
 - Aktivierung über Parameter-Pipe:
`totalCar=1|allowFirstLastSarcHighwayConnection=1`

Parksuchzeiten

- Parksuchzeiten (derzeit z. B. im Raum Wien) werden standardmäßig in die IV-Berechnung einbezogen.
 - Deaktivierung über die Parameter-Pipe
`totalCar=1|getInitEndTimes=0`

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

Für das E-Auto Routing (`groupFilter=API_ECAR`) stehen weitere Konfigurationsmöglichkeiten zur Verfügung. Das E-Auto Routing befindet sich in einer österreichweiten Testphase und wird evaluiert.

11.6.2 IV-Verkehrsmeldungen aktivieren

Standardmäßig sind IV-Meldungen bei allen IV Modalitäten (Auto, E-Auto, Rad) **inaktiv**, da die Ausgabe die Response stark vergrößert.

- Aktivierung: `totalMeta=event`
- Für IV-Anteile in kombinierten Routen (z. B. Park&Ride) müssen Verkehrsmeldungen im Vorlauf über `originMeta=event` aktiviert werden.

Weiterführende Information → Abschnitt 11.3 und 11.4 und 22.2

11.6.3 Berücksichtigung von Echtzeitdaten

Bei der Routenberechnung werden standardmäßig Echtzeitinformationen einbezogen, sodass die Dauer der Reise/Fahrt die Verzögerungen beinhaltet.

Standardmäßig aktivierte Dynamiken:

- Verkehrsmeldungen (`evnt`)
- Umfahren von Sperren (`aevnt`)
- Verkehrslage (Echtzeitinformationen & Statistiken, wie Tagesganglinien) (`tasta`, `htsta`)
- Kuzzeit-prognosedaten (1-4h) (`cdyn`)

Deaktivieren aller Dynamiken mit der Parameter-Pipe:

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

11.6.4 Alternative Route

Standardmäßig Im PKW Modus (`groupFilter=API_CAR`) aktiviert

- in passenden Fällen wird eine Alternativroute berechnet.
- Ausgabe von 1 oder maximal 2 Routen



Berechnung der Alternativroute deaktivieren

- Da die Berechnung der Alternativroute auch Rechenzeit benötigt, kann diese deaktiviert werden.
- `totalMeta= disableAlternativeRoutes`

Weiterführende Information → Abschnitt 22.1

11.6.5 Zwischenziele (Via-Orte) im IV-Routing

Das VIA-Routing im IV unterstützt:

- **Adressen und POIs** (IDs über Location Service oder `location.name` / `location.nearbystops`)
- **Koordinaten**: z. B. `Via=geo:14.667915,47.169457`
- Mehrere Koordinaten: durch **Semikolon** getrennt (;)

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
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
ctx	Erforderlich	-	-	Legt den GIS Routen Kontext fest. D.h. die entsprechende Definition den Vor- oder Nachlauf einer Route.
poly	Optional	0 / 1	0	Aktiviert/Deaktiviert die Berechnung der Polylinie für jede Routenetappe.
polyEnc	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@\\$C%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@\\$C%A7tf@\\$f@\\$f@\\$f@\\$f@\\$f@\\$f@\\$C%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@$C%A7bt@0@2000@120@-1@50@1@1000@0@@@@false@0@-1@$f@$f@$f@$f@$t@2000@50000@120@-1@100@1@15000@0@@@@false@0@-1@$f@$C%A7tf@$f@$f@$f@$f@$f@$f@$C%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 `JourenltailRef` 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.

Abfragezeitpunkt

kann durch den Parameter `date` und `time` angepasst werden. bei der Abfrage in die Zukunft und in die Vergangenheit gibt es bestimmte Faktoren zu berücksichtigen:

Weiterführende Information → Abschnitt 21.1.3

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).

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 Abfahrtshaltestellen (`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 (`displayName`) und Betreiber (`operator`) vorausgesetzt diese Daten sind vom Verkehrsverbund in erfasst und geführt.

Jede Abfahrt enthält auch einen Verweis (`JourenyDetailRef`) 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
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
id	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
extId	Erforderlich, wenn <code>id</code> nicht angegeben ist	-	-	Veraltet/aufgelassen da der Parameter <code>id</code> nun auch externe IDs unterstützt.
direction	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.
date	Optional		Aktuelles Serverdatum	Setzt das Abfahrtsdatum für die Suche im Format YYYY-MM-DD
time	Optional		Aktuelle Serverzeit	Setzt die Abfahrtszeit für die Suche im Format hh:mm in 24h Nomenklatur.
duration	Optional	0 / 1439	60	Setzt die Intervallzeit in Minuten
dur	Optional	0 / 1439	60	Veraltet/aufgelassen wird mit <code>duration</code> abgedeckt
products	Optional	-	-	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse 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: $2^2 + 2^4 = 20$
passlist	Optional	0 / 1	0	Aktiviert die durchfahrenen Wegpunkte (Haltestellen), an denen nicht gehalten wird.
lines	Optional	-	-	<p>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.</p> <p>Für ÖV Produktklasse 1 bis 11 anwendbar.</p> <p>z.B. Filter für Linien 120 und 140 lines=120,140</p>
maxJourneys	Optional	-	-	<p>Maximale Anzahl von Fahrten im Ergebnis. Kein Wert liefert alle An- und Abfahrten innerhalb der gesetzten Zeitspanne.</p> <p>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.</p>
rtMode	Optional	OFF / SERVER_DEFAULT	SERVER_DEFAULT (aktiviert)	<p>SERVER_DEFAULT= Echtzeit ist berücksichtigt (siehe Abschnitt 5.7)</p> <p>OFF = Suche nur auf Basis der Plandaten (SOLL-Daten)</p>
type	Optional	DEP_MAST, DEP_STATION, DEP_EQUIVS	DEP_EQUIVS	<p>ab V1.3.0</p> <p>DEP_MAST: DepartureBoard am Masten</p> <p>DEP_STATION: DepartureBoard mit allen Fahrten an beliebigen Masten der angeforderten Station</p> <p>DEP_EQUIVS: DepartureBoard mit allen Fahrten an beliebigen Masten und äquivalenten Haltestellen</p>



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="A1@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" txN="Niederflurfahrzeug">Niederflurfahrzeug</Note>

    <Note key="n0" type="A" priority="50" routeIdxFrom="18" routeIdxTo="29" txN="barrierefrei">barrierefrei</Note>

  </Notes>

  <altId>at:49:1184:0:3</altId>
</Departure>
```

13.2 Scrolling in den Tafeln

Um die Abfahrtstafeln durchzublätern, 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 Ankunstafel. Die Response enthält eine Liste der Ankünfte im Root Element `Arrival`. Dieses enthält die Ankunftsstationen (`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.

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.

14.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
id	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
extId	Erforderlich, wenn <code>id</code> nicht angegeben ist		-	Veraltet/aufgelassen da der Parameter <code>id</code> nun auch externe IDs unterstützt.
direction	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.
date	Optional		Aktuelles Serverdatum	Setzt das Ankunftsdatum für die Suche im Format YYYY-MM-DD



time	Optional		Aktuelle Serverzeit	Setzt die Ankunftszeit für die Suche im Format hh:mm in 24h Nomenklatur.
duration	Optional	0 / 1439	60	Setzt die Intervallzeit in Minuten
dur	Optional	0 / 1439	60	Veraltet/aufgelassen wird mit <code>duration</code> abgedeckt
products	Optional	-	-	Dezimalwert für die zu berücksichtigenden ÖV Produktklasse in der Suche. Dieser Wert wird wie folgt berechnet: Beispiel: Bei Berücksichtigung von lediglich U-Bahnen und Straßenbahnen sind in der Bit-Maske die Bits 2 und 4 gesetzt. Dadurch ergibt sich: $2^2 + 2^4 = 20$
lines	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>
maxJourneys	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: Ankunststafel für Wien Schottentor am 12.02.2021 um 17:00h.

<https://routenplaner.verkehrsauskunft.at/vao/restproxy/{{version}}/arrivalBoard?accessId={{accessId}}&format=xml&iid=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 Ankunststafeln durchzublättern, 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 *Zwischenstopps* in der konkreten Start-Ziel-Relation ausgegeben (mit dem Parameter `passPoints`).

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

15.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
id	Erforderlich	-	-	<code>id</code> für die abzufragende Fahrt Ab Version 1.3.3 kann auch die <code>altId</code> für Haltestellen verwendet werden
date	Optional		Aktuelles Serverdatum	Setzt das Ankunftsdatum für die Suche im Format YYYY-MM-DD
poly	Optional	0/1	0	Aktiviert die Berechnung einer Polylinie für jeden leg-Abschnitt der Reise
polyEnc	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)
showPassingPoint	Optional	0/1	0	Aktiviert oder Deaktiviert die Anzeige von Halten, in welchen nicht zugestiegen oder ausgestiegen werden kann (PassingPoints 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

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" catOutI="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 im 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. Bei der Abfrage in die Zukunft und in die Vergangenheit gibt es bestimmte Faktoren zu berücksichtigen:

Weiterführende Information → Abschnitt 21.1.3

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.

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
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
ctx	Erforderlich	-	-	Legt den Rekonstructions-Text fest
poly	Optional	0/1	0	Aktiviert/ Deaktiviert die Berechnung der Polylinie für jede Routenetape.
polyEnc	Optional	GPA	-	Aktiviert die Ausgabe der Polyline im Format „Google Encoded Polyline Format“.
date	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 SVC_NO_RESULT 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. bei der Abfrage in die Zukunft und in die Vergangenheit gibt es bestimmte Faktoren zu berücksichtigen:

Weiterführende Information → Abschnitt 21.1.3

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
accessId	Erforderlich	-	-	Access ID zur Authentifizierung



18 DATA INFORMATION (DATAINFO)

Dieser Service bietet detaillierte Informationen über alle Betreiber (Element `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.

Weiterführende Information → Abschnitt 21

18.1 Request Parameter

Name	Benutzung	Werte	Default	Beschreibung
accessId	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
accessId	Erforderlich	-	-	Access ID zur Authentifizierung
lineId				Interne LineId der Linie
date				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">tcp=16585#16784 plancode0=agr46 planid=1753822184
planid0=1753822184 planid_adr=1753273272 plancode_adr=9401m planid_adr=1662723912
plancode_adr=r779m planid_poi=1753826766 plancode_poi=agvog srvv=5.45.KGOEVV.17.3.16
(customer/hcuva0/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>

      ...

    </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“
API_QUOTA	400	400	Quotenüberlauf für „Schlüssel“ zum „Service“
API_TOO_MANY_REQUESTS	429	403	too many requests
API_PARAM	400	400	Erforderlicher Parameter <<name>> fehlt
API_PARAM	400	400	numB falsch, nur Werte im Bereich [0,6] erlaubt
API_PARAM	400	400	numF falsch, nur Werte im Bereich [0,6] erlaubt
API_PARAM	400	400	numF + numB dürfen 6 nicht überschreiten
API_FORMAT	400	400	Ausgabeformat nicht unterstützt
SVC_PARAM	400	400	Request Parameter falsch oder fehlend
SVC_LOC	400	400	Ortsangabe falsch oder fehlend/ Es konnte keine Haltestelle in der Umgebung der Adresse gefunden werden
SVC_LOC_ARR	400	400	Ankunftsort falsch oder fehlend
SVC_LOC_DEP	400	400	Abfahrtstort falsch oder fehlend
SVC_LOC_VIA	400	400	Unbekannter Umstiegshalt
SVC_LOC_EQUAL	400	400	Start/Ziel oder Vias identisch
SVC_LOC_NEAR	400	400	Start und Ziel liegen zu nahe beieinander
SVC_DATETIME	400	400	Datum/Zeit ungültig oder fehlend
SVC_DATETIME_PERIOD	400	400	Datum/Zeit außerhalb der gültigen Periode oder nicht im Fahrplan



SVC_PROD	400	400	Produkt (Transportmittel) Feld fehlt oder ungültig
SVC_CTX	400	400	Kontext ungültig
SVC_MAIL_ADR	400	400	Absender/Empfänger- Mailadresse ungültig oder fehlend
SVC_NO_RESULT	400	200	Kein Ergebnis gefunden
SVC_SMS_NUM	400	400	Empfängertelefonnummer für SMS ungültig oder fehlend
Server Errors			
SVC_MAIL	500	500	Fehler beim Mailversand
BAIM_ERROR	500	500	Konfiguration oder Routing Fehler in BAIM
SVC_SMS	500	500	Fehler beim SMS- Versand
SVC_FAILED_SEARCH	500	500	Nicht erfolgreiche Suche
SVC_NO_MATCH	422	422	Keine Übereinstimmung gefunden/ kein Treffer
SVC_TOO_MANY	400	422	Zu viele Suchergebnisse
INT_ERR	500	500	Interner Fehler
INT_HAFAS_CONNECTION_ERROR	503	503	Verbindung zu [HOST]:[PORT] ist nicht möglich; Time-Out der Verbindung (ConnectionException)
INT_HAFAS_CONNECTION_ERROR	503	503	Verbindung zu [HOST]:[PORT] durch HAFAS Server geschlossen (ConnectionException)
INT_HAFAS_CONNECTION_ERROR	503	503	Verbindung zu [HOST]:[PORT] durch HAFAS Server geschlossen (ConnectionException) (HAFAS Server Timeout-Limit überschritten)
INT_HAFAS_CONNECTION_OPENING_ERROR	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
INT_TIMEOUT	503	503	Timeout während der Verarbeitung (API Timeout-Limit überschritten)
INT_TIMEOUT	503	503	Timeout während der Kommunikation mit dem Backend (Backend-Timeout-Limits überschritten)
Spezielle Errors			
SOT_AT_DEST	400	400	Trip bereits erreicht
SOT_BEFORE_START	400	400	Trip nicht gestartet



SOT_CANCELLED	400	400	Trip gelöscht
SOT_ALL_TRAINS_FILTERED	400	400	Alle Trips gefiltert
SOT_STAY_IN_CURRENT_CONNECTION	400	200	Keine Änderung. In dem Trip verbleibend.
PARTIALSEARCH_INCORRECT_PARAM	400	400	Mit einer invaliden Parameterkombination wurde angefragt, z.B. wurde versucht rückwärts/vorwärts vom Beginn/Ende einer Verbindung zu suchen.
SOT_TRANSIT_LOCATION_MISSING_FOR_JOURNEY	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
TOO_MANY	Das Ergebnis enthält zu viele Elemente. Einige Elemente wurden verworfen.
TOO_MANY_SCROLL_CTX	Das Ergebnis enthält zu viele Scroll Kontext Elemente. Einige Elemente wurden verworfen.
H887	Timeout überschritten. Das Ergebnis kann unvollständig sein.
H889	Zu viele Routen gefunden. Einige Elemente wurden verworfen.

21 WEITERFÜHRENDE INFORMATIONEN ZU ÖV

21.1 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.

21.1.1 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.

21.1.2 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.

21.1.3 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.

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.

21.2 Verarbeitung von Realtime-Daten (ÖV)

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).

21.2.1 Fahrplanabweichungen (Realtime)

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).

21.2.2 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

21.2.1 Auslastungsinformationen (Besetzungsgrad)

Aktuell werden über die Echtzeitdaten im Verbundgebiet Vorarlberg auch Auslastungsinformationen zu Fahrten geliefert.



Integration von Echtzeit-Auslastungsinformationen (Besetzungsgrad) in den Responses der folgenden Services:
Trip, Stationboard (Departures / Arrivals), Journey Details

- keine flächendeckende Verfügbarkeit von Auslastungsdaten, aktuell nur im Verbundgebiet des VVV (Vorarlberg)
- Für Züge erfolgt – sofern die entsprechenden Daten geliefert werden – eine getrennte Ausgabe für erste und zweite Klasse.
- Liegt keine Klassentrennung vor (z. B. bei Bussen), wird der identische Belegungsgrad für FIRST und SECOND ausgegeben.
- Der Auslastungsgrad wird als numerischer Rohwert im Feld rawRt ausgegeben und im Feld sRt (OccupancyStatusType) klassifiziert
- sRt kann folgende Werte annehmen:
 - U – Occupancy status not set
 - NA – Occupancy information not available
 - L – Low seat occupancy
 - M – Medium seat occupancy
 - H – High seat occupancy

Beispiel:

```
<Occupancy name="FIRST" rawRt="11" sRt="L"/>  
<Occupancy name="SECOND" rawRt="11" sRt="L"/>
```

21.3 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.

21.4 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`).



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 einer 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.

Beispiel:

```
<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">
```

21.4.1 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.

21.5 ÖV- Verkehrsmittel & Produktklassen

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

Belegung der Bit-Maske:

Bit (n) catCode-Wert	Bitmaske (2^n) cls-Wert	Produktklasse catoutL-Wert
0	1	Zug
1	2	S-Bahn
2	4	U-Bahn
3	8	Stadtbahn
4	16	Straßenbahn
5	32	Schnellbus



6	64	Regionalbus
7	128	Stadtbus
8	256	Seil-/Zahnradbahn
9	512	Schiff
10	1024	Anrufsammeltaxi/Rufbus (AST)
11	2084	Sonstige
12	4096	Autoreisezug
13-15	8192-32768	Nicht in Verwendung

Zur Erleichterung soll die folgende Tabelle mit Standardkombinationen helfen:

Kombination	Berechnung	cls
U-Bahn + Straßenbahn	$4 + 16$	20
S-Bahn + U-Bahn	$2 + 4$	6
S-Bahn + U-Bahn + Straßenbahn	$2 + 4 + 16$	22
S-Bahn + U-Bahn + Stadtbahn + Straßenbahn	$2 + 4 + 8 + 16$	30
Alle Busse (Schnellbus + Regionalbus + Stadtbus)	$32 + 64 + 128$	224
Zug + S-Bahn	$1 + 2$	3
Zug + S-Bahn + Stadtbahn	$1 + 2 + 8$	11
Schiene (urban + Zug)	$1 + 2 + 4 + 8 + 16$	31
Schiene inkl. Autoreisezug	$31 + 4096$	4127
Seil-/Zahnradbahn + Schiff	$256 + 512$	768
Alle Klassen 0–12	Summe $2^0 \dots 2^{12}$	8191
Alle außer AST	$8191 - 1024$	7167
Nur AST	1024	1024

Interpretation: Jede gesetzte 1 in der Binärdarstellung von cls entspricht einer enthaltenen Produktklasse.
Beispiel: $\text{cls}=224 = 11100000_2 \rightarrow$ Bits 5, 6, 7 gesetzt \rightarrow **Schnellbus, Regionalbus,**

21.5.1 Interpretation der Produktklasse in der Response

Diese Produktklassen werden dabei als Bitmaske ($2^{\text{Produktklasse}}$) 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="645142">

  <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="O01AA_" 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="O01AA_" matchId="xxx">
```


22 WEITERFÜHRENDE INFORMATIONEN ZU IV

22.1 Alternative Route in der Response

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.

22.2 IV-Verkehrsmeldungen in der Trip Response

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</v  
ao:routable><vao:link_url>http://www.graztourismus.at/de/anreise-und-  
verkehr/parken/kurzparkzonen</vao:link_url></vao:poi_original></wfs:member></wfs:FeatureC  
ollection>"><?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=Describ  
eFeatureType&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><va  
o: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</v  
ao:routable><vao:link_url>http://www.graztourismus.at/de/anreise-und-  
verkehr/parken/kurzparkzonen</vao:link_url></vao:poi_original></wfs:member></wfs:FeatureC  
ollection></Note>
```

22.3 Mautinformationen in der Trip-Response

Bei IV Routen werden den einzelnen Streckenabschnitten die Mautinformationen beigefügt, die in der GIP 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>` beigefügt mit den folgenden Keys. Der Key muss zusammen mit dem Feld `txtN` ausgewertet werden.

Erläuterungen zur Auswertung der Mautinformation im <code><GisRoute><Seg><Note></code>		
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

22.4 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>
```



22.5 VIA-Orte in der Trip Response einer IV -Route

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.

22.6 Praxis zum Decodieren der Routen-Polylinie

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/unescaped werden, um im String verwendet zu werden. Ein Tool zum unescape <https://www.freeformatter.com/json-escape.html>.

23 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 geliefert. 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 erreichbar. Ein Berggipfel ist beispielsweise oft nicht mit der Modalität PKW erreichbar, sondern nur zu Fuß.

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 geliefert 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.



24 EINSCHRÄNKUNGEN BEI ROUTEN IM AUSLAND

Die VAO umfasst 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.