

Beschreibung der EFA XML-Schnittstelle für die Open Government Data Initiative der WIENER LINIEN



30. September 2013

Inhalt

1. Grundlagen
2. Exkurs: Analysetools
3. Grundfunktionalität
4. Stop-Finder-Request - Haltestellensuche
5. Trip-Request – Fahrtanfrage
6. DM-Request – Abfahrtsmonitor

1. Grundlagen

1. Eingabe und Ausgabe
2. HTTP-Requests
3. Session-Handling
4. Umwandlung der Ausgabe
5. Spracheinstellung
6. Ausführungsanweisung
7. Generische Parameter für das User-Interface („Layout-Parameter“)

1.1 Eingabe und Ausgabe

Eingabe

Alle Anfragen werden über folgende URL durchgeführt:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?language=de

Ausgabe

XML-Struktur:

```
<itdRequest xsi:noNamespaceSchemaLocation="http://www.wienerlinien.at/ogd_routing/schema/9_16/itd.xsd" language="de" sessionID="0"
client="Firefox" serverID="VOR-OGDR01" clientIP="192.168.75.9" version="9.16.29.30" virtDir="ogd_routing" now="2013-09-10T10:38:11"
nowWD="2" lengthUnit="METER">
  <itdTripRequest requestID="0" execlnst="verifyOnly">
    <itdPrintConfiguration active="-1" layout="" shading="1" serverInfo="" serverAddress="" outputFormat="pdf"
printerDirect="0" fontSizeNormal="1" commuterFaresOutput="0" fareUnitOrientation="left"/>
    <itdAddress name="" place="" street="" addressExt1="" addressExt2=""/>
    <itdOdv type="stop" usage="origin" anyObjFilter="0"/>
    <itdOdv type="stop" usage="destination" anyObjFilter="0"/>
    <itdTripDateTime deparr="dep" ttpFrom="20121112" ttpTo="20131214"/>
    <itdTripOptions userDefined="0"/>
    <itdItinerary/>
  </itdTripRequest>
</itdRequest>
```

1.2. HTTP-Requests

Verschiedene Anfragearten (HTTP-Requests) stellen verschiedene Funktionalitäten zur Verfügung.

- Stop-Finder-Request - Haltestellensuche
- Trip-Request – Fahrplanfrage
- DM-Request – Abfahrtsmonitor

1.2 HTTP-Requests (2)

Alle Anfragearten leiten sich von einem gemeinsamen Anfrage-Objekt „*itdRequest*“ ab. Dieses stellt Grundfunktionalität wie

- das Session-Handling (1.3)
- die Umwandlung der Ausgabe (1.4)
- die Spracheinstellung (1.5)
- die Ausführungsanweisung (1.6)
- generische Parameter für das User-Interface (1.7)

zur Verfügung.

1.3 Session-Handling

Die Anfrage kann zustandslos oder zustandsbehaftet erfolgen. Für die zustandsbehaftete Anfrage werden folgende Parameter verwendet:

sessionID

Eindeutige ID einer internen Session. Für weitere Folgeanfragen muss die Session ID übergeben werden, um auf die in der Session gespeicherten Daten zugreifen zu können. Durch den Parameter *sessionID=0* wird eine neue Session erzeugt.

Beispiel 1:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?language=de

Beispiel 2:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?sessionID=0

1.3 Session-Handling (2)

requestID

ID eines Requests innerhalb einer Session. Eine Session kann mehrere Requests umfassen. Das System generiert bei einigen Funktionalitäten (z.B. Weiterfahrt, Rückfahrt) automatisch neue Request IDs.

```
<itdRequest xsi:noNamespaceSchemaLocation="http://ogd.vor.at/ogd/schema/9_16/itd.xsd"
language="de" sessionID="0" client="Firefox" serverID="VOR-OGDR01"
clientIP="192.168.75.9" version="9.16.29.30" virtDir="ogd_routing" now="2013-09-
10T10:38:11" nowWD="2" lengthUnit="METER">
<itdTripRequest requestID="0" execlnst="verifyOnly">
```

...

1.5 Spracheinstellung

language

Kontrolle sprachspezifischer Ausgabeinformation.

Wert: Die durch den Standard ISO 638:1988 (E/F) definierten Sprachkürzel.

Beispiele:

http://www.wienerlinien.at/ogd_routing/XSLT_TRIP_REQUEST2?language=de

http://www.wienerlinien.at/ogd_routing/XSLT_TRIP_REQUEST2?language=en

1.6 Ausführungsanweisung

execInst

Beeinflusst die Ausführung der Anweisung.

Werte:

- **normal** – Standardmäßige Ausführung.
- **verifyOnly** – Punkte werden verifiziert und Parameter in die XML-Ausgabe übernommen, es findet jedoch keine Berechnung von Fahrten, Abfahrten, etc. statt.
- **readOnly** – Die Eingaben durch Parameter werden in die XML-Ausgabe übernommen, es findet jedoch weder eine Punktverifikation noch eine Berechnung von Fahrten, Abfahrten, etc. statt.

1.6 Ausführungsanweisung (2)

Beispiel – Verifikation einer Haltestelle mittels ID:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?type_origin=stopID&name_origin=60200334&type_destination=stopID&name_destination=60201015&execInst=verifyOnly

```
<itdOdvName state="identified" method="itp">
  <odvNameElem value="60200334:1" stopID="60200334" x="755059"
    y="653505" mapName="GIP1" isTransferStop="0"
    matchQuality="100000">Floridsdorf
  </odvNameElem>
  <odvNameInput>60200334</odvNameInput>
</itdOdvName>
```

1.7 Generische Parameter für das User-Interface

itdLPxx_paramName

Die generischen Parameter werden durchgereicht, ohne das Systemverhalten zu beeinflussen. Sie werden auch „Layout-Parameter“ genannt, da sie dazu dienen sich einen bestimmten Zustand der Programmoberfläche zu merken.

Sie beginnen mit „itdLPxx_“, das nur zur Kennzeichnung dient und weggeschnitten wird. Der Name *paramName* kann ebenso wie der Wert frei gewählt werden.

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdLPxx_paramName=paramValue

2. Exkurs: Analysetools

Folgende Tools sind für die Analyse der Anfrageparameter hilfreich:

- Fiddler
- Firebug

Fiddler (Freeware)

Web Debugging Proxy, das den HTTP(S)-Verkehr loggt.

www.fiddler2.com

Firebug (Addon für Firefox)

- HTML/CSS analysieren und in Echtzeit ändern
- JavaScript Debugger
- Analyse von Performance, Headern, Requests,...

3. Grundfunktionalität

Einige Funktionalitäten werden von mehreren HTTP-Requests (Trip-Request, DM-Request,...) benötigt, z.B.:

1. Datum und Uhrzeit
2. Verifikation von Punkten

3.1 Datum und Uhrzeit

XML-Schema: *itdDateTime* beinhaltet *itdDate* (Datum) und *itdTime* (Uhrzeit).

Parameter zur Wahl des Datums (*itdDate*)

Default: Anfragedatum

Parametername	Bedeutung	Format
itdDate	Jahr, Monat, Tag	JJJJMMTT JJMMTT

3.1 Datum und Uhrzeit (2)

Anfragebeispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdDate=20130910

```
<itdTripDateTime deparr="dep" ttpFrom="20121112" ttpTo="20131214">  
  <itdDateTime ttpFrom="20121112" ttpTo="20131214">  
    <itdDate day="10" month="9" year="2013" weekday="3"/>  
    <itdTime hour="14" minute="35"/>  
  </itdDateTime>  
</itdTripDateTime>
```

Wird die Zeit nicht angegeben (siehe später) so wird die aktuelle Systemzeit übernommen.

3.1 Datum und Uhrzeit (3)

Fehlercodes für das Datum (*itdMessage*)

Code	Bedeutung
-1	Ungültiger Datumswert
-10	Jahr außerhalb des Wertebereichs
-20	Monat außerhalb des Wertebereichs
-30	Tag außerhalb des Wertebereichs
-4001	Datum außerhalb der Fahrplanperiode

Beispiel: Fehler -1 und -30

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdDate=20130999

Datum und Uhrzeit (3a)

Beispiel – fehlerhafte Eingabe

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdDate=20130999

```
<itdDateTime ttpFrom="20121112" ttpTo="20131214">
  <itdDate day="99" month="09" year="2013" weekday="2">
    <itdMessage type="error" module="BROKER" code="-1">invalid date</itdMessage>
    <itdMessage type="error" module="BROKER" code="-30">day value out of
      range</itdMessage>
  </itdDate>
  <itdTime hour="14" minute="51"/>
</itdDateTime>
```

3.1 Datum und Uhrzeit (4)

Parameter zur Wahl der Uhrzeit (*itdTime*)

Default: Anfragezeit

Parametername	Bedeutung	Format
itdTime	Stunde, Minute	HHMM
timeOffset	Offset von der aktuellen Zeit (in Minuten)	MM
itdTimeAMPM	Zeit ist AM oder PM	am pm

3.1 Datum und Uhrzeit (5)

Anfragebeispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdTime=1654

```
<itdDateTime ttpFrom="20121112" ttpTo="20131214">  
  <itdDate day="30" month="9" year="2013" weekday="2"/>  
  <itdTime hour="16" minute="54"/>  
</itdDateTime>
```

Wird das Datum nicht angegeben, so wird das aktuelle Systemdatum übernommen.

3.1 Datum und Uhrzeit (6)

Fehlercodes für die Uhrzeit (*itdMessage*)

Code	Bedeutung
-1	Ungültige Uhrzeit
-10	Stunde außerhalb des Wertebereichs
-20	Minute außerhalb des Wertebereichs

Beispiel: Fehler -1 und -20

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdTime=1699

```
<itdTime hour="16" minute="99">  
  <itdMessage type="error" module="BROKER" code="-1">invalid time</itdMessage>  
  <itdMessage type="error" module="BROKER" code="-20">minute value out of range</itdMessage>  
</itdTime>
```

3.2 Verifikation von Punkten

Im Workshop wird die Punktverifikation mittels des EFA Location Servers (Einfeld-Eingabe) besprochen.

XML-Schema: *itdOdv*

Odv ist eine Abkürzung für „origin destination via“.

Das Element *itdOdv* enthält den Ort *itdOdvPlace* und den Namen *itdOdvName* eines Punktes.

In der allerersten Beispielausgabe wurde noch keine Anfrage nach einem Punkt per XML Request gestellt. Ein Punkt kann eine Haltestelle, eine Adresse oder ein besonderer Punkt (POI) sein. Unter Punktverifikation wird hier die Suche nach Punkten im Auskunftssystem über Text / Strings verstanden.

```
<itdOdv type="unknown" usage="origin" anyObjFilter="0">
  <itdOdvPlace state="empty">
    <odvPlaceElem/>
  </itdOdvPlace>
  <itdOdvName state="empty" method="itp">
    <odvNameElem/>
  </itdOdvName>
</itdOdv>
```

3.2 Verifikation von Punkten (2)

Die Punkte werden durch eine Erweiterung des Parameternamens „usage“ unterschieden:

origin – Startpunkt (Trip-Request, PS-Request,...)

destination – Zielpunkt (Trip-Request, PS-Request,...)

via – Zwischenhalt (Trip-Request)

dm – Abfahrtshaltestelle (DM-Request)

sf – Haltestellensuche (Stop-Finder-Request)

...

3.2 Verifikation von Punkten (3)

Obligatorische Parameter:

locationServerActive = 1

Aktiviert den EFA Location Server.

name_<usage>

Start-, Ziel- oder Via-Punkt. Entspricht dem Eingabefeld.

type_<usage> = any

Spezifiziert den Punkt näher. Soll der EFA Location Server verwendet werden, ist der Wert *any*.

Aufgabe:

Verifizieren sie den Westbahnhof (1468) als Startpunkt.

3.2 Verifikation von Punkten (4)

Lösungsbeispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Meidling

Das Ergebnis ist kein einzelner Punkt sondern eine Liste. Eine Meldung `itdMessage` weist auch darauf hin. Die Listenelemente können über einen Index zugegriffen werden.

```
<itdOdvName state="list" method="itp">
<itdMessage type="error" module="BROKER" code="-8011"/>
  <odvNameElem value="0:1" selected="0" listIndex="0" streetName="" omc="49000000"
  placelD="50" x="750900" y="661996" mapName="GIP1" id="60200844" anyType="stop" anyTypeSort="2" nameKey=""
  locality="Wien" postCode="" objectName="Marx-Meidlinger-Straße" buildingName="" buildingNumber=""
  matchQuality="216" posttown="Wien">Wien, Marx-Meidlinger-Straße</odvNameElem>
  <odvNameElem value="1:2" selected="1" listIndex="1" streetName="" omc="49000000"
  placelD="50" x="749972" y="662665" mapName="GIP1" id="60201015" anyType="stop" anyTypeSort="2" nameKey=""
  locality="Wien" postCode="" objectName="Meidling" buildingName="" buildingNumber="" matchQuality="733"
  posttown="Wien">Wien, Meidling</odvNameElem>
  <odvNameElem value="2:3" selected="0" listIndex="2" streetName="" omc="49000000"
  placelD="50" x="750937" y="662741" mapName="GIP1" id="60201403" anyType="stop" anyTypeSort="2" nameKey=""
  locality="Wien" postCode="" objectName="UKH Meidling" buildingName="" buildingNumber="" matchQuality="251"
  posttown="Wien">Wien, UKH Meidling</odvNameElem>
  <odvNameElem value="3:4" selected="0" listIndex="3" streetName="" omc="49000000"
  placelD="50" x="749732" y="661644" mapName="GIP1" id="60200863" anyType="stop" anyTypeSort="2" nameKey=""
  locality="Wien" postCode="" objectName="Meidling Hauptstraße" buildingName="" buildingNumber=""
  matchQuality="272" posttown="Wien">Wien, Meidling Hauptstraße</odvNameElem>
  <odvNameElem value="4:5" selected="0" listIndex="4" streetName="" omc="49000000"
  placelD="50" x="749824" y="662252" mapName="GIP1" id="60200864" anyType="stop" anyTypeSort="2" nameKey=""
  locality="Wien" postCode="" objectName="Meidlinger Hauptstr/Pohlgasse" buildingName="" buildingNumber=""
  matchQuality="221" posttown="Wien">Wien, Meidlinger
  Hauptstr/Pohlgasse</odvNameElem><odvNameInput>Meidling</odvNameInput>
</itdOdvName>
```

3.2 Verifikation von Punkten (5)

Bei der Punktverifikation kann es neben Volltreffern zu Listen und Fehlern kommen. Das Attribut *state* des Elements *itdOdvName* beschreibt den Status der Punktverifikation und kann folgende Werte haben:

- **empty** – es hat keine Verifikation stattgefunden
- **identified** – Volltreffer
- **list** – Trefferliste
- **notidentified** – Verifikation fehlgeschlagen

3.2 Verifikation von Punkten (6)

Für die Auswahl aus einer Liste sind weitere Parameter notwendig:

Session angeben durch **sessionID** und **requestID**. (Hinweis: Diese sind nur vorhanden, wenn zustandsbehaftet angefragt wurde.)

nameState_<usage>

Beschreibt den Status Punktverifikation.

Wert: Wert des Attributs *state* des Elements *itdOdvName*.

3.2 Verifikation von Punkten (7)

Aufgabe:

Wählen Sie das Unfallkrankenhaus Meidling aus einer Suchanfrage „Meidling“ aus.

Hinweis:

Der Index des zu wählenden Listenelements *odvNameElem* befindet sich im Attribut *value* und wird durch den Parameter *name_<usage>* übergeben.

3.2 Verifikation von Punkten (8)

Lösung:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=<sessionID>&requestID=<requestID>&type_origin=any&name_origin=2:3&nameState_origin=list

```
<itdRequest xsi:noNamespaceSchemaLocation="http://ogd.vor.at/ogd/schema/9_16/itd.xsd" language="de" sessionID="VOR-OGDR011715536156" client="Firefox" serverID="VOR-OGDR01" clientIP="192.168.75.9" version="9.16.29.30" virtDir="ogd_routing" now="2013-09-30T15:05:52" nowWD="2" lengthUnit="METER">
<itdTripRequest requestID="1">
...
<itdOdv type="any" usage="origin" anyObjFilter="0">
<itdOdvPlace state="identified" method="itp">
    <odvPlaceElem ...>Wien</odvPlaceElem>
    <odvPlaceInput/>
</itdOdvPlace>
<itdOdvName state="identified" method="itp">
    <odvNameElem streetName="" omc="49000000" placelD="50" x="750937" y="662741" mapName="GIP1" id="60201403" anyType="stop" nameKey="" locality="Wien" postCode="" objectName="UKH Meidling" buildingName="" buildingNumber="" posttown="Wien">Wien, UKH Meidling
    </odvNameElem>
    <odvNameInput>Meidling</odvNameInput>
</itdOdvName>
```

RequestID und SessionID sind entsprechend der vorigen Beispielanfrage (3.2.(4)) zu setzen.

3.2 Verifikation von Punkten (9)

Die Parameter *type_<usage>* und *name_<usage>* können durch die Parameter

typInfo_<usage>

nameInfo_<usage>

übersteuert werden. Wird der Parameter übergeben, soll aber deaktiviert sein, ist sein Wert *invalid*.

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Westbahnhof&nameInfo_origin=Stephansplatz

3.2 Verifikation von Punkten (10)

Das Eingabefeld mit einem Hinweistext vorbelegen:

nameDefaultText_<usage>

Der Wert des Parameters ist ein Text, der bei der Verifikation des durch name_<usage> angegebenen Punktes nicht berücksichtigt werden soll. Auf diese Weise kann ein Hinweistext (z.B. „Bitte Halteestelle, Adresse oder POI eingeben!“) im Eingabefeld angezeigt werden.

3.2 Verifikation von Punkten (11)

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Haltestelle, Adresse oder wichtiger Punkt

Im Vergleich zu:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Haltestelle, Adresse oder wichtiger Punkt&nameDefault_origin=Haltestelle, Adresse oder wichtiger Punkt

3.2 Verifikation von Punkten (12)

Einschränkung des Suchraums:

anyObjFilter_<usage>

Der Suchraum kann auf bestimmte Objekttypen eingegrenzt werden. Der Wert ist eine Bitmaske mittels der die Objekttypen miteinander kombiniert werden können.

Beispiel:

anyObjFilter_<usage>=34 → (2 + 32)
Haltestellen und wichtige Punkte.

Wert	Bedeutung
0	Kompletter Suchraum
1	Orte
2	Haltestellen
4	Straßen
8	Adressen
16	Kreuzungen
32	Wichtige Punkte
64	Postleitzahlen

3.2 Verifikation von Punkten (13)

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Erdberg&anyObjFilter_origin=2

```
<itdOdvName state="list" method="itp">
  <itdMessage type="error" module="BROKER" code="-8011"/>
  <odvNameElem value="0:1" selected="1" listIndex="0" streetName="" omc="49000000"
placeID="50" x="756109" y="660805" mapName="GIP1" id="60200289" anyType="stop" anyTypeSort="2"
nameKey="" locality="Wien" postCode="" objectName="Erdberg" buildingName="" buildingNumber=""
matchQuality="360" posttown="Wien">Wien, Erdberg</odvNameElem>
  <odvNameElem value="1:2" selected="0" listIndex="1" streetName="" omc="49000000"
placeID="50" x="757139" y="661470" mapName="GIP1" id="60201579" anyType="stop" anyTypeSort="2"
nameKey="" locality="Wien" postCode="" objectName="Erdbergstraße/Gaswerk" buildingName=""
buildingNumber="" matchQuality="237" posttown="Wien">Wien, Erdbergstraße/Gaswerk</odvNameElem>
  <odvNameElem value="2:3" selected="0" listIndex="2" streetName="" omc="49000000"
placeID="50" x="755638" y="660344" mapName="GIP1" id="60200283" anyType="stop" anyTypeSort="2"
nameKey="" locality="Wien" postCode="" objectName="Erdbergstr./Schlachthausg." buildingName=""
buildingNumber="" matchQuality="219" posttown="Wien">Wien,
Erdbergstr./Schlachthausg.</odvNameElem>
  <odvNameInput>Erdberg</odvNameInput>
</itdOdvName>
```

3.2 Verifikation von Punkten (14)

Soll ein Punkt nicht verifiziert werden, sondern ein bekannter Punkt verwendet werden, kann dieser über seine ID oder Koordinate identifiziert werden.

Beispiel Haltestelle:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?type_origin=stopID&name_origin=60501170

Beispiel POI:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?type_origin=poiID&name_origin=45:44010021:-1

Hinweis: Zusätzlich zur ID des POIs müssen die GKZ und die ID des Ortes angegeben werden (Bsp. Touristeninfo am HBF).

3.2 Verifikation von Punkten (15)

Beispiel Koordinate:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?type_origin=coord&name_origin=16.338705:48.197833:WGS84 →
Europaplatz, Westbahnhof, schnappt auf die nächste Kante

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?type_origin=coord&name_origin=16.338705:48.197833:WGS84:Ein schöner Ort

→ Beliebige Koordinate mit Freitext (z.B. „Ein schöner Ort“)

Hinweis: Um eine Koordinate identifizieren zu können, müssen (mindestens) x, y und das Koordinatensystem angegeben werden.

3.2 Exkurs: Koordinaten

coordOutputFormat

Ausgabeformat der Koordinaten.

Werte: **MDV** | WGS84[DD.ddddd] | <EPSG-Code> (ab Version 9.17)

coordListOutputFormat

Ausgabe der Koordinaten als Liste von Elementen (*itdCoordinateBaseElemList*) oder String (*itdCoordinateList*).

Werte: **list**, string

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=any&name_origin=Westbahnhof&type_destination=stopID&name_destination=60200001&computeMonomodalTripCar=1&itOptionsActive=1&coordListOutputFormat=list

3.2 Exkurs: Koordinaten (2)

coordOutputFormatTail

Beschränkung der Nachkommastellen der Koordinaten.

Wert: Anzahl der Nachkommastellen

Hinweis:

Werden die Koordinaten aus der XML-Antwort herausreduziert, können diese mit folgenden Parametern bei Bedarf aktiviert werden:

showItdInterchangePathCoordinates = 1

Aktiviert die Koordinaten der Umstiegswege.

showItdPathCoordinates = 1

Aktiviert die Koordinaten für die Teilwege.

3.2 Exkurs: Koordinaten (3)

- **coordOutputFormat** kann die Werte **MDV | WGS84[DD.ddddd]** | **<EPSG-Code>** annehmen. Dieser Parameter bestimmt das Ausgabeformat der Koordinaten. Standardmässig wird das MDV eigene Koordinatensystem MDV verwendet. Ebenfalls möglich ist die Wahl von WGS 84 oder die Angabe eines EPSG Codes. Die Angabe des EPSG Codes erfolgt in folgendem Format: PROJ[+init=epsg:<Code>].
- **coordOutputFormatTeil** kann den Wert **<Anzahl>** annehmen. Standardmässig werden die Koordinaten mit Nachkommastellen ausgegeben. Die Anzahl der Nachkommastellen kann durch den Wert **<Anzahl>** beschränkt werden. Ist der Wert 0, werden keine Nachkommastellen ausgegeben.

3.2 Verifikation von Punkten (16)

Fehlerbehandlung:

Analog zum Datums- und Uhrzeitelement werden im Element *itdOdv* Fehlermeldungen *itdMessage* ausgegeben. Die Fehlerbeschreibungen sind dem **Dokument EFA9_Errorcodes_V1.0** zu entnehmen.

4. Stop-Finder-Request

Die Verifikation eines Punktes kann auch zur Suche von Haltestellen verwendet werden. Dies wird mit folgender Anfrage erreicht:

http://www.wienerlinien.at/ogd_routing/XML_STOPFINDER_REQUEST?

Teil der Haltestellensuche:

- Verifikation von Punkten (3.2)

Punktverifikation: Der Parameter-Suffix *<usage>* nimmt den Wert *sf* an.

4. Stop-Finder-Request (2)

Der Stop-Finder-Request nutzt die Parameter zur Punktverifikation, hat aber keine eigenen Parameter.

Anwendungsbeispiel Vorschlagsliste bei der Punkt-Eingabe

http://www.wienerlinien.at/ogd_routing/XML_STOPFINDER_REQUEST?locationServerActive=1&outputFormat=XML&type_sf=any&name_sf=Erdberg

4.1 Anwendungsbeispiel Vorschlagsliste



Fahrplanauskunft



Fahrplan

▸ **Auskunft**

▸ Pers. Fahrplan

▸ Abfahrtsmonitor

▸ Echtzeitanzeige

▸ Meine Route - Anmeldung

▸ Meine Route -
Einstellungen

▸ Meine Route - Abmeldung

Auskunft

Start

Stadt/Gemeinde

Wien

Haltestelle Straße/Nr. Wichtiger Punkt

Erd



Ziel

Stadt/Gemeinde

Wien

Haltestelle Straße/Nr. Wichtiger Punkt

Erdberg



Zeit

Abfahrt Ankunft

Datum



Uhrzeit



4.1 Anwendungsbeispiel Vorschlagsliste (2)

outputFormat

Bestimmt das Format der Ausgabe.

Werte: HTML, XML, JSON

Hinweis: Der Parameter gilt auch bei verschiedenen anderen Requests.

Für eine mittels AJAX angefragte Vorschlagsliste empfiehlt sich die Ausgabe im JSON-Format.

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_STOPFINDER_REQUEST?locationServerActive=1&outputFormat=JSON&type_sf=any&name_sf=Erdberg

Ausgabebeispiel – JSON

- http://www.wienerlinien.at/ogd_routing/XML_STOPFINDER_RE_QUEST?locationServerActive=1&outputFormat=JSON&type_sf=any&name_sf=Erdberg

```
{ "parameters": [ { "name": "requestID", "value": "0" }, { "name": "sessionID", "value": "0" } ], "message": [ { "name": "code", "value": "-8011" }, { "name": "error", "value": "" } ], "input": [ { "name": "input", "value": "Erdberg" } ], "stopFinder": [ { "usage": "sf", "type": "any", "name": "Wien, Erdberg", "stateless": "60200289", "anyType": "stop", "sort": "2", "quality": "360", "object": "Erdberg", "posttown": "Wien", "modes": "2,11", "ref": { "id": "60200289", "omc": "49000000", "placeID": "50", "place": "Wien", "coords": "756109.00000,660805.00000" } }, { "usage": "sf", "type": "any", "name": "Wien, Erdbergstraße/Gaswerk", "stateless": "60201579", "anyType": "stop", "sort": "2", "quality": "237", "object": "Erdbergstraße/Gaswerk", "posttown": "Wien", "modes": "5", "ref": { "id": "60201579", "omc": "49000000", "placeID": "50", "place": "Wien", "coords": "757139.00000,661470.00000" } }, { "usage": "sf", "type": "any", "name": "Wien, Erdbergstr./Schlachthausg.", "stateless": "60200283", "anyType": "stop", "sort": "2", "quality": "219", "object": "Erdbergstr./Schlachthausg.", "posttown": "Wien", "modes": "4,5,11", "ref": { "id": "60200283", "omc": "49000000", "placeID": "50", "place": "Wien", "coords": "755638.00000,660344.00000" } } ] }
```

5. Trip-Request – Fahrtanfrage

Berechnung von Fahrten (ÖV) von A nach B (über C) zu einem bestimmten Datum und einer bestimmten Uhrzeit. Die Anpassung an individuelle Bedürfnisse erfolgt über die Fahrtoptionen (4.2).

Anfrage:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?

Infolge werden benötigt:

Teil der Fahrtanfrage:

- Datum und Uhrzeit (3.1)
- Verifikation von Punkten (3.2)

Punktverifikation: Der Parameter-Suffix *<usage>* kann den Wert *origin*, *destination* oder *via* annehmen.

5. Trip-Request – Fahrtanfrage (2)

1. Ergänzung zu Datum und Uhrzeit
2. Fahrtoptionen
3. Trip-Request Folgeanfragen
4. Fußwegbeschreibung
5. Exkurs: Koordinaten

5.1 Ergänzung zu Datum und Uhrzeit

itdTripDateTimeDepArr

Bestimmt, ob sich Datum und Uhrzeit einer Fahrt auf Abfahrt oder Ankunft beziehen.

Werte: **dep** | arr

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?itdDate=20130919&itdTime=16:45&itdTripDateTimeDepArr=arr

5.2 Fahrtoptionen

Beeinflussung der Fahrtergebnisse, so dass sie den persönlichen Bedürfnissen entsprechen.

XML-Schema: *itdTripOptions*

5.2 Fahrtoptionen (2)

Die Optionen sind in verschiedenen Untergruppen organisiert:

XMLSchema:

- *itdUsedOptions* – allgemeine Optionen
- *itdPtOptions* – Optionen für den ÖV
- *itdItOptions* – Optionen für den Individualtransport

Parameter zur Aktivierung verschiedener Gruppen von Optionen:

- **ptOptionsActive=1** – Optionen für den ÖV
- **impairedOptionsActive=1** – Mobilitätseinschränkungen
- **itOptionsActive=1** – Optionen für den Individualtransport

Standardmäßig sind die Parameter deaktiviert (0).

4.2 Fahrtoptionen (3)

Allgemeine Parameter (Auswahl): (*itdUsedOptions*)

calcNumberOfTrips

Anzahl der ÖV Fahrten, die berechnet werden (Alternativen nicht eingerechnet)

Hinweis: Mit *calcNumberOfTrips=0* ist Berechnung von ÖV-Fahrten deaktiviert.

5.2 Fahrtoptionen (4)

Parameter für ÖV Optionen (Auswahl): `ptOptionsActive = 1`

useProxFootSearch = 1

Berücksichtigung von nahegelegenen Haltestellen.

Zur Unterscheidung der Start- und Zielhaltestelle kann der Parametername durch die Erweiterung *Orig* bzw. *Dest* ergänzt werden.

5.2 Fahrtoptionen (5)

maxChanges

Anzahl der Umstiege.

Werte:

- 0 – Direktfahrt
- 1 – 1 x umsteigen
- 2 – 2 x umsteigen
- 9 – 9 x umsteigen

routeType

Optimierung der Fahrtauskunft:

Werte:

- | | |
|--------------------|--------------------------------|
| • LEASTTIME | Zeitkürzeste Verbindung |
| • LEASTINTERCHANGE | Verbindung mit wenig Umsteigen |
| • LEASTWALKING | Verbindung mit wenig Fußwegen |

5.2 Fahrtoptionen (6)

Parameter zum Ausschluss von Verkehrsmitteln:

excludedMeans = checkbox

Aktiviert den Verkehrsmittelausschluss mittels des Parameters `exclMOT_<ID> = 1`.

exclMOT_<ID> = 1

Schließt das Verkehrsmittel mit der ID `<ID>` aus. Kann mehrfach verwendet werden.

Alternative:

excludedMeans = <ID>

Mehrfachverwendung möglich.

<ID>	Verkehrsmitteltyp
0	Zug
1	S-Bahn
2	U-Bahn
3	Stadtbahn
4	Straßen-/Trambahn
5	Stadtbus
6	Regionalbus
7	Schnellbus
8	Seil-/Zahnradbahn*
9	Schiff
10	AST/Rufbus
11	Sonstige

5.2 Fahrtoptionen (7)

Hinweis:

Die Verkehrsmittel sind betriebsspezifisch und können von den Standardverkehrsmitteln abweichen.

Aufgabe:

Fahren Sie von Westbahnhof zum Stephansplatz ohne die Tram zu benutzen.

Tipp:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=any&name_origin=Westbahnhof&type_destination=any&name_destination=Stephansplatz

5.2 Fahrtoptionen (8)

Lösung:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=any&name_origin=Westbahnhof&type_destination=any&name_destination=Stephansplatz&ptOptionsActive=1&excludedMeans=4

<excludedMeans>

```
<meansElem value="0" selected="0">Zug</meansElem>
<meansElem value="1" selected="0">S-Bahn</meansElem>
<meansElem value="2" selected="0">U-Bahn</meansElem>
<meansElem value="3" selected="0">Stadtbahn</meansElem>
<meansElem value="4" selected="1">Straßen-/Trambahn</meansElem>
<meansElem value="5" selected="0">Stadtbus</meansElem>
<meansElem value="6" selected="0">Regionalbus</meansElem>
<meansElem value="7" selected="0">Schnellbus</meansElem>
<meansElem value="8" selected="0">Seil-/Zahnradbahn</meansElem>
<meansElem value="9" selected="0">Schiff</meansElem>
<meansElem value="10" selected="0">AST/Rufbus</meansElem>
<meansElem value="11" selected="0">Sonstige</meansElem>
```

</excludedMeans>

5.2 Fahrtoptionen (9)

Außerdem:

changeSpeed

Gehgeschwindigkeit beim Umsteigen bei *ptOptionsActive=1* und Geschwindigkeit für Weg zur Starthaltestelle bzw. von der Zielhaltestelle bei *itOptionsActive=1*.

Geschwindigkeit [km/h] = (100 x Standardgeschwindigkeit) / Wert
Werte: **normal** (->100), **slow** (->50), **fast** (->200), [25..400]

5.2 Fahrtoptionen (10)

Aufgabe 1: Menschen mit speziellen Bedürfnissen

Stellen Sie sich vor, Sie sind mit einer älteren Dame in Wien unterwegs. Sie wollen vom Hauptbahnhof zum Stephansplatz fahren und haben noch ein wenig Zeit. Die ältere Dame möchte auf keinen Fall die Strassenbahn benutzen. Außerdem kann Sie nicht gut Treppen steigen und braucht etwas länger zum umsteigen. Welche Optionen würden Sie wählen?

Tipp:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=stopID&name_origin=Hauptbahnhof&type_destination=any&name_destination=Stephansplatz

5.2 Fahrtoptionen (11)

Lösung:

z.B:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=stopID&name_origin=Wien%20Hauptbahnhof&type_destination=any&name_destination=Stephansplatz&ptOptionsActive=1&itOptionsActive=1&excludedMeans=4&changeSpeed=slow&routeType=LEASTINTERCHANGE&impairedOptionsActive=1&noSolidStairs=1&lowPlatformVhcl=1

5.3 Trip-Request Folgeanfragen

Parameter:

command

Ermöglicht Folgeanfragen.

Mögliche Werte (Auswahl):

- `nop` no Operation
- `changeRequest` Anfrage modifizieren
- `tripRetoure` neuer Request mit Vorbelegung
- `tripGoOn` neuer Request mit Vorbelegung des Starts
- `tripPrev` frühere Fahrt
- `tripNext` spätere Fahrt
- `tripLast` letzte Fahrt
- `tripFirst` erste Fahrt
- `printSingleview:n` Einzelansicht drucken (n - Fahrtindex)

5.3 Trip-Request Folgeanfragen (2)

Zur Erinnerung:

Bei Folgeanfragen müssen die Parameter **sessionID** und **requestID** übergeben werden.

5.4 Rückfahrt / Weiterfahrt

command = tripRetoure

Rückfahrt. Start und Zielpunkt werden vertauscht und vorbelegt.

command = tripGoOn

Weiterfahrt. Der Startpunkt wird mit dem Zielpunkt vorbelegt.

Es muss sichergestellt werden, dass die vorbelegten Punkte Volltreffer sind!

nameValue_<usage>

Verhindert die erneute Verifikation des Punktes, wenn der Parameter name_<value> übergeben werden muss, sein Wert dem Wert des Parameters name_<value> entspricht und sich der Parameter type_<usage> gegenüber der vorherigen Anfrage nicht geändert hat.

5.4 Rückfahrt / Weiterfahrt (2)

Aufgabe:

Berechnen Sie eine Fahrtauskunft vom Hauptbahnhof zum Stephansplatz. Lassen Sie sich anschließend auch die Rückfahrt ausgeben.

5.4 Rückfahrt / Weiterfahrt (3)

Hilfe (Hinfahrt):

1. Schritt:

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Wien%20Hauptbahnhof&type_destination=any&name_destination=Stephansplatz

2. Schritt (Vorschlag):

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=10690982&requestID=1&nameState_origin=list&name_origin=0:1&type_origin=any

5.4 Rückfahrt / Weiterfahrt (4)

Lösung (Rückfahrt):

http://www.wienerlinien.at/ogd_routing/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=10782945&requestID=1&command=tripRetoure

Hinweis:

Der Parameter nameValue_<usage> ist nicht notwendig, da der Parameter name_<usage> nicht verwendet wird.

6. DM-Request - Abfahrtsmonitor

Abfahrtsmonitor einer Haltestelle.

Anfrage:

http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?

Teil des Abfahrtsmonitors:

- Datum und Uhrzeit (3.1)
- Verifikation von Punkten (3.2)

Punktverifikation: Der Parameter-Suffix *<usage>* nimmt den Wert *dm* an.

6. DM-Request – Abfahrtsmonitor (2)

1. Ergänzung zu Datum und Uhrzeit
2. Ergänzung zur Verifikation von Punkten
3. DM-Optionen
4. DM-Request Folgeanfragen

6.1 Ergänzung zu Datum und Uhrzeit

Parameter zur Unterscheidung von Abfahrt und Ankunft:

itdDateTimeDepArr

Unterscheidung zwischen Abfahrts- und Ankunftsmonitor

Werte: **dep** | arr

XML-Schema:

Die Ankünfte *itdArrival* werden in einer Liste *itdArrivalList* gespeichert. Die Abfahrten *itdDeparture* in einer List *itdDepartureList*.

6.2 Ergänzung zur Verifikation von Punkten

Zugeordnete Haltestellen auswählen:

Notwendig, wenn der Punkt eine Haltestelle sein soll, aber eine beliebige Eingabe möglich ist. (Beim Abfahrtsmonitor nicht zwingend.)

XML-Schema: *itdOdvAssignedStops*

selectAssignedStops_<usage> = 1

Aktiviert die Auswahl zugeordneter Haltestellen.

assignedStop_<usage>

Auswahl einer zugeordneten Haltestelle.

Wert: Attribut *value* einer zugeordneten Haltestelle

itdOdvAssignedStop

6.2 Ergänzung zur Verifikation von Punkten (2)

assignedStopList_<usage>

Auswahl mehrerer zugeordneter Haltestellen.

Wert: Eine oder mehrere Haltestellen IDs, getrennt durch Doppelpunkt. (Die ID befindet sich im Attribut *stopID* der Elemente *itdOdvAssignedStop*.)

Hinweis:

Bei der Wahl einer zugeordneten Haltestelle müssen nicht nur **sessionID** und **requestID** übergeben werden. Auch der Verifikationsstatus des Punktes muss mit **nameState_<usage>=identified** übermittelt werden.

6.2 Ergänzung zur Verifikation von Punkten (3)

Beispiel:

Schritt 1:

http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?sessionID=0&locationServerActive=1&type_dm=any&name_dm=Westbahnhof

Bevor die Abfahrten angezeigt werden, müssen die Linien ausgewählt werden!

6.3 DM-Optionen

Parameter zur Linienauswahl:

Die Parameter **sessionID** und **requestID** müssen angegeben werden (Folgeanfrage).

dmLineSelectionAll = 1

Wählt alle Linien aus.

dmLineSelection

Auswahl einer Linie. Der Parameter kann mehrfach verwendet werden.

Wert:

- Index der Linie - Attribut *index* der Linie *itdServingLine*
- all – alle Linien
- 0:x – Züge in alle Richtungen – Attribut *traininfo=show* muss im Element *itdServingLines* vorhanden sein

6.3 DM-Optionen (2)

Beispiel:

http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?sessionID=2425800329&requestID=1&dmLineSelectionAll=1

6.3 DM-Optionen (3)

Einige der Fahrtoptionen des Trip-Requests gelten auch für den Abfahrtsmonitor. Zum Beispiel der Ausschluss von Verkehrsmitteln (4.2). Weitere Parameter:

limit

Maximale Anzahl der Abfahrten.

Wert: Integer (Default: 40)

IsShowTrainsExplicit = 1

Anzeige der Züge in der Linienauswahl.

XML-Schema: Sind die Züge nicht aktiviert, enthält *itdServingLines* das Attribut *@trainInfo=show*. Durch *dmLineSelection=0:x* kann die Auswahl aller Züge ermöglicht werden.

6.3 DM-Optionen (3)

Aufgabe:

Zeigen Sie einen Abfahrtsmonitor mit den nächsten 20 Abfahrten für den Wiener Westbahnhof an. Eine Einschränkung auf bestimmte Linien soll nicht vorgenommen werden.

6.3 DM-Optionen (4)

Lösung:

1. Schritt:

http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?sessionID=0&locationServerActive=1&type_dm=any&name_dm=Westbahnhof&limit=20

2. Schritt:

http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?sessionID=<HIER DIE SESSION ID ERGÄNZEN>&requestID=1&dmLineSelectionAll=1

6.4 DM-Request Folgeanfragen

command

Ermöglicht Folgeanfragen.

Mögliche Werte (Ausschnitt):

- nop no Operation
- changeRequest Anfrage modifizieren
- dmFirst erste Abfahrt
- dmLast letzte Abfahrt
- dmNext nächste Abfahrten
- dmPrev vorherige Abfahrten

6.4 DM-Request Folgeanfragen - Beispiele

Beispiele:

- http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?sessionID=<HIER DIE SESSION ID ERGÄNZEN>&requestID=1&dmLineSelectionAll=1&command=dmlast
- http://www.wienerlinien.at/ogd_routing/XML_DM_REQUEST?sessionID=<HIER DIE SESSION ID ERGÄNZEN>&requestID=1&dmLineSelectionAll=1&command=dmlfirst

Fehler & Fragen

Es wird ersucht etwaige Fragen bzw. Unklarheiten, die aus dieser Dokumentation nicht eindeutig hervorgehen vorrangig durch Erfahrungen **innerhalb der Developercommunity** zu klären. Wir bitten in diesem Zusammenhang um Verständnis, dass es seitens Wiener Linien bzw. VOR personell nicht möglich ist jeder technischen Anfrage nachzugehen.