

EFA XML-Schnittstelle

Jänner 2013

Inhalt

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

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

http://www.linzag.at/static/XML_TRIP_REQUEST2?

Ausgabe

XML-Struktur:

```
- <itdRequest version="9.16.27.3" language="de" lengthUnit="METER" sessionID="0" client="Firefox" c
- <itdTripRequest requestID="0">
  <itdAddress name="" street="" place="" addressExt1="" addressExt2=""/>
  + <itdOdv type="unknown" usage="origin"></itdOdv>
  + <itdOdv type="unknown" usage="destination"></itdOdv>
  + <itdOdv type="unknown" usage="via"></itdOdv>
  + <itdTripDateTime deparr="dep" ttpFrom="20110110" ttpTo="20111031"></itdTripDateTime>
  + <itdTripOptions userDefined="0"></itdTripOptions>
  <itdItinerary/>
</itdTripRequest>
</itdRequest>
```

1.2. HTTP-Requests

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

- Stop-Finder-Request - Haltestellensuche
- Trip-Request – Fahrplananfrage
- DM-Request – Abfahrtsmonitor
- AddInfo -Request

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.linzag.at/static/XML_TRIP_REQUEST2?

Beispiel 2:

http://www.linzag.at/static/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 version="9.16.27.3" language="de" lengthUnit="METER" sessionID="0"
  - <itdTripRequest requestID="0" >
    <itdAddress name="" street="" place="" addressExt1="" addressExt2=""/>
```


1.5 Spracheinstellung

language

Kontrolle sprachspezifischer Ausgabeinformation.

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

Beispiele:

http://www.linzag.at/static/XSLT_TRIP_REQUEST2?language=de

http://www.linzag.at/static/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:

http://www.linzag.at/static/XML_TRIP_REQUEST2?type_origin=stopID&name_origin=60501200&type_destination=stopID&name_destination=60501070&execInst=verifyOnly

```
- <itdRequest version="9.16.27.3" language="de" lengthUnit="METER" sessionID="0" client="Firefox" clientIP="127.0.0.1" serverID="localhost" virtDir="linz_neu" now="2011-09-13T07:49:21" nowWD="3">
- <itdTripRequest requestID="0" execInst="verifyOnly">
  <itdAddress name="" street="" place="" addressExt1="" addressExt2=""/>
- <itdOdv type="stop" usage="origin">
  - <itdOdvPlace state="identified" method="itp">
    <odvPlaceElem omc="44001001" placeID="4" value="44001001:4" span="0" type="remote" mainPlace="1">Linz/Donau</odvPlaceElem>
    <odvPlaceInput/>
  </itdOdvPlace>
  - <itdOdvName state="identified" method="itp">
    <odvNameElem x="5448851" y="810583" mapName="NAV5" stopID="60501200" value="60501200:1" isTransferStop="0" matchQuality="100000">WIFI/LINZ AG</odvNameElem>
    <odvNameInput>60501200</odvNameInput>
  </itdOdvName>
</itdOdv>
- <itdOdv type="stop" usage="destination">
  - <itdOdvPlace state="identified" method="itp">
    <odvPlaceElem omc="44001001" placeID="4" value="44001001:4" span="0" type="remote" mainPlace="1">Linz/Donau</odvPlaceElem>
    <odvPlaceInput/>
  </itdOdvPlace>
  - <itdOdvName state="identified" method="itp">
    <odvNameElem x="5449235" y="811418" mapName="NAV5" stopID="60501070" value="60501070:1" isTransferStop="0" matchQuality="100000">Neue Welt</odvNameElem>
    <odvNameInput>60501070</odvNameInput>
  </itdOdvName>
</itdOdv>
```

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.linzag.at/static/XML_TRIP_REQUEST2?itdLPxx_paramName=paramValue

2. Exkurs: Analysetools

Folgende Tools sind für die Analyse der Anfrageparameter hilfreich:

- Fiddler
- Firebug

Fiddler (Freeware)

Web Debugging Proxi, 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.linzag.at/static/XML_TRIP_REQUEST2?itdDate=20110225

```
- <itdTripDateTime deparr="dep" ttpFrom="20110110" ttpTo="20111031">
  - <itdDateTime ttpFrom="20110110" ttpTo="20111031">
    <itdDate weekday="6" year="2011" month="2" day="25"/>
    <itdTime hour="7" minute="52"/>
  </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.linzag.at/static/XML_TRIP_REQUEST2?itdDate=201102

99

Datum und Uhrzeit (3a)

Beispiel – fehlerhafte Eingabe

http://www.linzag.at/static/XML_TRIP_REQUEST2?itdDate=20110299

```
- <itdTripDateTime deparr="dep" ttpFrom="20110110" ttpTo="20111031">
  - <itdDateTime ttpFrom="20110110" ttpTo="20111031">
    - <itdDate weekday="3" year="2011" month="02" day="99">
      <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="7" minute="54"/>
  </itdDateTime>
</itdTripDateTime>
```

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.linzag.at/static/XML_TRIP_REQUEST2?itdTime=1654

```
- <itdTripDateTime deparr="dep" ttpFrom="20110110" ttpTo="20111031">  
  - <itdDateTime ttpFrom="20110110" ttpTo="20111031">  
    <itdDate weekday="3" year="2011" month="9" day="13"/>  
    <itdTime hour="16" minute="54"/>  
  </itdDateTime>  
</itdTripDateTime>
```

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.linzag.at/static/XML_TRIP_REQUEST2?itdTime=1699

```
- <itdTripDateTime deparr="dep" ttpFrom="20110110" ttpTo="20111031">
  - <itdDateTime ttpFrom="20110110" ttpTo="20111031">
    <itdDate weekday="3" year="2011" month="9" day="13"/>
    - <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>
  </itdDateTime>
</itdTripDateTime>
```

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 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">  
  - <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 Hauptbahnhof Linz (1170) als Startpunkt.

3.2 Verifikation von Punkten (4)

Lösungsbeispiel:

http://www.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Hauptbahnhof Linz

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

```
- <itdOdv type="any" usage="origin" anyObjFilter="0">
  - <itdOdvPlace state="empty">
    <odvPlaceElem/>
  </itdOdvPlace>
  - <itdOdvName state="list" method="itp">
    <itdMessage type="error" module="BROKER" code="-8011"/>
    <odvNameElem listIndex="0" selected="1" matchQuality="1000" x="5447580" y="809422"
      mapName="NAV5" id="60501720" omc="44001001" placeID="4" anyType="stop" anyTypeSort="2"
      locality="Linz/Donau" objectName="Hauptbahnhof" buildingName="" buildingNumber="" postCode=""
      streetName="" nameKey="" posttown="Linz/Donau" value="0:1">Linz/Donau,
      Hauptbahnhof</odvNameElem>
    <odvNameElem listIndex="1" selected="0" matchQuality="980" x="5447564" y="809391"
      mapName="NAV5" id="1" omc="44001001" placeID="-1" anyType="poi" anyTypeSort="4"
      locality="Linz/Donau" objectName="Hauptbahnhof Linz" buildingName="" buildingNumber=""
      postCode="" streetName="" nameKey="" posttown="" value="1:2">Linz/Donau, Hauptbahnhof
      Linz</odvNameElem>
    <odvNameElem listIndex="2" selected="0" matchQuality="950" x="5447498" y="809550"
      mapName="NAV5" id="45" omc="44001001" placeID="-1" anyType="poi" anyTypeSort="4"
      locality="Linz/Donau" objectName="Touristeninformation am HBF" buildingName="" buildingNumber=""
      postCode="" streetName="" nameKey="" posttown="" value="2:3">Linz/Donau, Touristeninformation am
      HBF</odvNameElem>
    <odvNameInput>Hauptbahnhof Linz</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 die Touristeninformation am Hauptbahnhof 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.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=<sessionID>&requestID=<requestID>&type_origin=any&name_origin=2:3&nameState_origin=list

```
- <itdRequest version="9.16.27.3" language="de" lengthUnit="METER" sessionID="68150581" client="Firefox"
  clientIP="127.0.0.1" serverID="localhost" virtDir="linz_neu" now="2011-09-13T08:14:07" nowWD="3">
  - <itdTripRequest requestID="1">
    <itdAddress name="" street="" place="" addressExt1="" addressExt2=""/>
    - <itdOdv type="any" usage="origin" anyObjFilter="0">
      - <itdOdvPlace state="identified">
        <odvPlaceElem omc="44001001" placeID="-1" value="44001001:-1" span="0" type="remote"
          mainPlace="0">Linz/Donau</odvPlaceElem>
        <odvPlaceInput/>
      </itdOdvPlace>
      - <itdOdvName state="identified" method="itp">
        <odvNameElem x="5447498" y="809550" mapName="NAV5" id="45" omc="44001001" placeID="4"
          anyType="poi" locality="Linz/Donau" objectName="Touristeninformation am HBF" buildingName=""
          buildingNumber="" postCode="" streetName="" nameKey="" posttown="">Linz/Donau,
          Touristeninformation am HBF</odvNameElem>
        <odvNameInput>Hauptbahnhof Linz</odvNameInput>
      </itdOdvName>
    </itdOdv>
  </itdTripRequest>
</itdRequest>
```

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

typelInfo_<usage>

nameInfo_<usage>

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

Beispiel:

http://www.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Hauptbahnhof Linz&nameInfo_origin=Neues Rathaus

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.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Haltestelle, Adresse oder wichtiger Punkt

Im Vergleich zu:

http://www.linzag.at/static/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.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=LinzWifi&anyObjFilter_origin=2

```

- <itdOdv type="any" usage="origin" anyObjFilter="2">
  - <itdOdvPlace state="identified" method="itp">
    <odvPlaceElem omc="44001001" placeID="4" value="44001001:4" span="0" type="remote"
      mainPlace="1">Linz/Donau</odvPlaceElem>
    <odvPlaceInput/>
  </itdOdvPlace>
  - <itdOdvName state="identified" method="itp">
    <odvNameElem x="5448851" y="810583" mapName="NAV5" id="60501200" omc="44001001"
      placeID="4" anyType="stop" locality="Linz/Donau" objectName="WIFI/LINZ AG" buildingName=""
      buildingNumber="" postCode="" streetName="" nameKey="" posttown="Linz/Donau">Linz/Donau,
      WIFI/LINZ AG</odvNameElem>
    <odvNameInput>LinzWifi</odvNameInput>
  </itdOdvName>
  - <itdOdvAssignedStops select="0">
    <itdOdvAssignedStop stopID="60501200" x="5448851" y="810583" mapName="NAV5"
      value="60501200:WIFI/LINZ AG" place="Linz/Donau" nameWithPlace="Linz WIFI/LINZ AG"
      distanceTime="0" isTransferStop="0">WIFI/LINZ AG</itdOdvAssignedStop>
  </itdOdvAssignedStops>
  - <genAttrList>
    - <genAttrElem>
      <name>anyObjFilter</name>
      <value>STOP</value>
    </genAttrElem>
  </genAttrList>
</itdOdv>

```

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.linzag.at/static/XML_TRIP_REQUEST2?type_origin=stopID&name_origin=60501170

Beispiel POI:

http://www.linzag.at/static/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.linzag.at/static/XML_TRIP_REQUEST2?type_origin=coord&name_origin=5448851:810583:NAV5 → WIFI, schnappt auf die nächste Kante

[http://www.linzag.at/static/XML_TRIP_REQUEST2?type_origin=coord&name_origin=5443851:812583:NAV5:Ein schöner Ort](http://www.linzag.at/static/XML_TRIP_REQUEST2?type_origin=coord&name_origin=5443851:812583:NAV5:Ein%20sch%C3%B6ner%20Ort)
→ 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.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=any&name_origin=Landeskulturzentrum&type_destination=stopID&name_destination=60501200&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.linzag.at/static/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.linzag.at/static/XML_STOPFINDER_REQUEST?locationServerActive=1&outputFormat=XML&type_sf=any&name_sf=Hauptbahnhof

```

- <itdStopFinderRequest requestID="0">
  - <itdOdv type="any" usage="sf" anyObjFilter="0">
    - <itdOdvPlace state="empty">
      <odvPlaceElem/>
    </itdOdvPlace>
    - <itdOdvName state="list">
      <itdMessage type="error" module="BROKER" code="-8011"/>
      <odvNameElem listIndex="0" selected="1" matchQuality="1000" x="5447580" y="809422"
        mapName="NAV5" id="60501720" omc="44001001" placeID="4" anyType="stop" anyTypeSort="2"
        locality="Linz/Donau" objectName="Hauptbahnhof" buildingName="" buildingNumber="" postCode=""
        streetName="" nameKey="" posttown="Linz/Donau" value="0:1">Linz/Donau,
        Hauptbahnhof</odvNameElem>
      <odvNameElem listIndex="1" selected="0" matchQuality="965" x="5447564" y="809391"
        mapName="NAV5" id="1" omc="44001001" placeID="-1" anyType="poi" anyTypeSort="4"
        locality="Linz/Donau" objectName="Hauptbahnhof Linz" buildingName="" buildingNumber=""
        postCode="" streetName="" nameKey="" posttown="" value="1:2">Linz/Donau, Hauptbahnhof
        Linz</odvNameElem>
    </itdOdvName>
  </itdOdv>
</itdStopFinderRequest>

```

4.1 Anwendungsbeispiel Vorschlagsliste

Von Haupt

Nach

- Linz/Donau, Brucknerhaus Haupteingang
- Linz/Donau, Hauptbahnhof
- Linz/Donau, Hauptplatz
- Linz/Donau, Anton-Bruckner-Privatuniversität/Hauptgebäude
- Linz/Donau, Hauptbahnhof Linz
- Linz/Donau, Hauptbücherei der Stadt Linz/Mediathek
- Linz/Donau, Hauptplatz
- Linz/Donau, Touristeninformation am HBF
- Ansfelden (OÖ) Gerhart-Hauptmann-Straße
- Ansfelden (OÖ) Hauptplatz
- Linz/Donau Hauptplatz
- Linz/Donau Hauptstraße
- Traun (OÖ) Hauptplatz

08:52


08:58 an Linz WIFI/LINZ AG

	3. Fahrt	08:57	09:03	00:06	0x
	4. Fahrt	09:02	09:08	00:06	0x

Später Letzte Fahrt

[Details für alle Fahrten anzeigen](#) [Rückfahrt](#) [Weiterfahrt](#) [Neue Fahrt](#)

Karte



Linieninformation

Drucken

- 1. 08:47 - 08:53 Uhr
 2
- 2. 08:52 - 08:58 Uhr
 1
- 3. 08:57 - 09:03 Uhr
 2
- 4. 09:02 - 09:08 Uhr
 1

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.linzag.at/static/XML_STOPFINDER_REQUEST?locationServerActive=1&outputFormat=JSON&type_sf=any&name_sf=Hauptbahnhof

Ausgabebeispiel – JSON

- http://www.linzag.at/static/XML_STOPFINDER_REQUEST?locationServerActive=1&outputFormat=JSON&type_sf=any&name_sf=Hauptbahnhof

```
{ "parameters": [ { "name": "requestID", "value": "0" }, { "name": "sessionID", "value": "0" } ], "message": [ { "name": "code", "value": "-8011" }, { "name": "error", "value": "" } ], "input": [ { "name": "input", "value": "Hauptbahnhof" } ], "stopFinder": [ { "usage": "sf", "type": "any", "name": "Linz/Donau, Hauptbahnhof", "stateless": "60501720", "anyType": "stop", "sort": "2", "quality": "1000", "object": "Hauptbahnhof", "posttown": "Linz/Donau", "ref": { "id": "60501720", "omc": "44001001", "placeID": "4", "place": "Linz/Donau", "coords": "5447580,809422" } }, { "usage": "sf", "type": "any", "name": "Linz/Donau, Hauptbahnhof Linz", "stateless": "poiID:1:44001001:-1", "anyType": "poi", "sort": "4", "quality": "965", "object": "Hauptbahnhof Linz", "ref": { "id": "1", "omc": "44001001", "placeID": "-1", "place": "Linz/Donau", "coords": "5447564,809391" } }, { "usage": "sf", "type": "any", "name": "Linz/Donau, Touristeninformation am HBF", "stateless": "poiID:45:44001001:-1", "anyType": "poi", "sort": "4", "quality": "932", "object": "Touristeninformation am HBF", "ref": { "id": "45", "omc": "44001001", "placeID": "-1", "place": "Linz/Donau", "coords": "5447498,809550" } } ] ] }
```

5. Trip-Request – Fahrthanfrage

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.linzag.at/static/XML_TRIP_REQUEST2?

Infolge werden benötigt:

Teil der Fahrthanfrage:

- 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
6. Echtzeit

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.linzag.at/static/XML_TRIP_REQUEST2?itdDate=20110225&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
 - **imparedOptionsActive=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

*Die Pöstlingbergbahn wird unter der Verkehrsmittelklasse Seil-/Zahnradbahnen synonym mit Bergbahnen geführt.

5.2 Fahrtoptionen (7)

Hinweis:

Die Verkehrsmittel sind verbundspezifisch und können von den Standardverkehrsmitteln abweichen.

Aufgabe:

Fahren Sie von WIFI Linz AG zur Johannes Kepler Universität ohne die Tram zu benutzen.

Tipp:

http://www.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=any&name_origin=WIFI Linz AG&type_destination=any&name_destination=Johannes Kepler Universität

5.2 Fahrtoptionen (8)

Lösung:

http://www.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=any&name_origin=WIFI Linz AG&type_destination=any&name_destination=Linz Johannes Kepler Universität&ptOptionsActive=1&excludedMeans=4

```
- <itdTripOptions userDefined="0">
- <itdPtOptions active="1" maxChanges="9" maxTime="360" maxWait="120" routeType="LEASTTIME"
  changeSpeed="normal" lineRestriction="400" useProxFootSearch="0" useProxFootSearchOrigin="0"
  useProxFootSearchDestination="0" bike="0" plane="0" noCrowded="0" noSolidStairs="0"
  noEscalators="0" noElevators="0" lowPlatformVhcl="0" wheelchair="0" SOSAvail="0"
  noLonelyTransfer="0" illumTransfer="0" overgroundTransfer="0" noInsecurePlaces="0"
  privateTransport="0">
- <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 Linz unterwegs. Sie wollen vom Hauptbahnhof zum Landeskulturzentrum 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.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&type_origin=stopID&name_origin=LinzHauptbahnhof&type_destination=any&name_destination=LinzLandeskulturzentrum

5.2 Fahrtoptionen (11)

Lösung:

z.B:

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

```
- <itdItinerary>  
  - <itdRouteList>  
    - <itdRoute active="1" selected="1" changes="0" distance="167" routeIndex="1" routeTripIndex="1" alternative="0" print="0" delete="0" searchMode="-1" cTime="20110913094632060" method="drs" individualDuration="00:06" vehicleTime="6" publicDuration="00:12">
```

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 Landeskulturzentrum. Lassen Sie sich anschließend auch die Rückfahrt ausgeben.

5.4 Rückfahrt / Weiterfahrt (3)

Hilfe (Hinfahrt):

1. Schritt:

http://www.linzag.at/static/XML_TRIP_REQUEST2?locationServerActive=1&sessionID=0&type_origin=any&name_origin=Hauptbahnhof&type_destination=any&name_destination=Landeskulturzentrum

2. Schritt (Vorschlag):

http://www.linzag.at/static/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.linzag.at/static/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.

5.5 Echtzeit

useRealtime = 1

Aktiviert die Berücksichtigung der Echtzeitinformation.

XML Schema:

Das Element *itdDateTime* enthält die aktuellen Abfahrts- und Ankunftspunkte *itdPoint* enthält die Echtzeit. In der Ausgabe wird allerdings der Unterschied zwischen Echtzeit und Sollzeit nicht sichtbar..

5.5 Echtzeit (2)

Beispiel:

[http://www.linzag.at/static/XML_TRIP_REQUEST2?useRealtime=1
&locationServerActive=1&type_origin=any&name_origin=Wifi%20Linz%20AGi&type_destination=any&name_destination=Goethekreuzung](http://www.linzag.at/static/XML_TRIP_REQUEST2?useRealtime=1&locationServerActive=1&type_origin=any&name_origin=Wifi%20Linz%20AGi&type_destination=any&name_destination=Goethekreuzung)

5.5 Echtzeit (3)

6. DM-Request - Abfahrtsmonitor

Abfahrtsmonitor einer Haltestelle.

Anfrage:

http://www.linzag.at/static/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
5. Echtzeit

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

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.linzag.at/static/XML_DM_REQUEST?sessionID=0&locationServerActive=1&type_dm=any&name_dm=WIFI Linz AG

Schritt 2:

http://www.linzag.at/static/XML_DM_REQUEST?sessionID=14211722&requestID=1&locationServerActive=1&type_dm=any&name_dm=WIFI Linz AG&nameState_dm=identified&selectAssignedStops_dm=1&assignedStop_dm=4:Rosenheimer Platz

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.linzag.at/static/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 die Linz AG an. Eine Einschränkung auf bestimmte Linien soll nicht vorgenommen werden.

6.3 DM-Optionen (4)

Lösung:

1. Schritt:

http://www.linzag.at/static/XML_DM_REQUEST?sessionID=0&locationServerActive=1&type_dm=any&name_dm=WIFI Linz AG&limit=20

2. Schritt:

http://www.linzag.at/static/XML_DM_REQUEST?sessionID=2426560047&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.linzag.at/static/XML_DM_REQUEST?sessionID=14948624&requestID=1&dmLineSelectionAll=1&command=dmLast
- http://www.linzag.at/static/XML_DM_REQUEST?sessionID=14948624&requestID=1&dmLineSelectionAll=1&command=dmFirst

6.5 Echtzeit

useRealtime = 1

Aktiviert die Berücksichtigung der Echtzeitinformation.

XML Schema:

Für jede echtzeitüberwachte Linie *itdServingLine* ist das Attribut *realtime=1*.

6.5 Echtzeit (2)

Zur Erinnerung: Beim Trip-Request enthält das Element *itdDateTime* auch die Echtzeit. Es wird immer die aktuelle Abfahrtszeit ausgegeben. In der Ausgabe wird nicht zwischen Echtzeit und Sollzeit unterschieden.

6.5 Echtzeit (3)

Beispiel:

1. Schritt:

[http://www.linzag.at/static/XML_DM_REQUEST?useRealtime=1&sessionID=0&locationServerActive=1&type_dm=any&name_dm=Goet
hekreuzung](http://www.linzag.at/static/XML_DM_REQUEST?useRealtime=1&sessionID=0&locationServerActive=1&type_dm=any&name_dm=Goet
hekreuzung)

2. Schritt:

http://www.linzag.at/static/XML_DM_REQUEST?useRealtime=1&sessionID=1394024437&requestID=1&dmLineSelectionAll=1

6.5 Echtzeit – Abfahrtsmonitor (4)

- Die resultierende Ausgabe weist für die betreffende Linie die aktuelle Abfahrtszeit an der angefragten Haltestelle der Abfahrtsmonitors aus. In der Ausgabe wird nicht zwischen Sollzeit und Echtzeit unterschieden. Damit wird auch der Unterschied zwischen Soll- und Echtzeit nicht sichtbar.

```

- <itdDeparture stopID="60500990" x="5447613" y="808864" mapName="NAV5" stopName="Linz Goethekreuzung" area="1" platform="1" platformName="" countdown="1">
- <itdDateTime>
  <itdDate day="17" month="12" year="2012" weekday="2"/>
  <itdTime hour="18" minute="6" ap=""/>
</itdDateTime>
- <itdServingLine key="642" code="4" number="3" symbol="3" direction="Linz Landgutstraße" motType="4">
  <itdNoTrain name="Straßenbahn"/>
  <motDivaParams project="w12" network="esg" line="01003" supplement="E" direction="R"/>
</itdServingLine>
</itdDeparture>
- <itdDeparture stopID="60500990" x="5447613" y="808866" mapName="NAV5" stopName="Linz Goethekreuzung" area="1" platform="0" platformName="" countdown="2">
- <itdDateTime>
  <itdDate day="17" month="12" year="2012" weekday="2"/>
  <itdTime hour="18" minute="8" ap=""/>
</itdDateTime>
- <itdServingLine key="543" code="4" number="3" symbol="3" direction="Doblerholz" motType="4">
  <itdNoTrain name="Straßenbahn"/>
  <motDivaParams project="w12" network="esg" line="01003" supplement="E" direction="H"/>
</itdServingLine>
</itdDeparture>
- <itdDeparture stopID="60500990" x="5447613" y="808864" mapName="NAV5" stopName="Linz Goethekreuzung" area="1" platform="1" platformName="" countdown="1">
- <itdDateTime>
  <itdDate day="17" month="12" year="2012" weekday="2"/>
  <itdTime hour="18" minute="7" ap=""/>
</itdDateTime>
- <itdServingLine key="457" code="4" number="2" symbol="2" direction="Linz Universität" motType="4">
  <itdNoTrain name="Straßenbahn"/>
  <motDivaParams project="w12" network="esg" line="01002" supplement="E" direction="R"/>
</itdServingLine>
</itdDeparture>
- <itdDeparture stopID="60500990" x="5447613" y="808866" mapName="NAV5" stopName="Linz Goethekreuzung" area="1" platform="0" platformName="" countdown="4">
- <itdDateTime>
  <itdDate day="17" month="12" year="2012" weekday="2"/>
  <itdTime hour="18" minute="10" ap=""/>
</itdDateTime>

```

7. Add-Info-Request – Aktuelle Hinweise

Das Fahrplanauskunftssystem der Linz Linien verfügt auch über eine Störungsmeldungssystem, in Folge ICS genannt. Anfrage der über ein ICS erfassten Meldungen können an die XML Schnittstelle gerichtet werden.

Anfrage:

http://www.linzag.at/static/XML_ADDINFO_REQUEST?

Der Aufbau der Anfragen geschieht analog zu den in den früheren Beispielen. Wir beschreiben im folgenden:

- weitere Parameter zum Filtern
- weitere Parameter

7.1 Parameter zum Filtern

Die Anfrage von Aktuellen Meldungen liefert alle erfassten Meldungen zurück. Diese können mittels Parametern nach verschiedenen Kriterien gefiltert werden. Einige Beispiele:

filterDateValid=<TT>-<MM>-<JJJJ>

Meldungen, die an diesem Tag gültig sind. Der Parameter kann mehrfach übergeben werden.

filterProviderCode=<Betreibercode>

Meldungen, die den Betreiber mit der ID <Betreibercode> betreffen. Der Parameter kann mehrfach übergeben werden.

filterSourceSystemName=<Quelle>

Meldungen, die vom System <Quelle> eingegeben wurden. Der Parameter kann mehrfach übergeben werden.

7.1 Parameter zum Filtern (2)

filterInfoType

Meldungen eines bestimmten Typs. Der Parameter kann mehrfach übergeben werden.

Folgende Werte sind möglich:

- stopInfo
- stopBlocking
- lineInfo
- lineBlocking
- routeInfo
- routeBlocking
- trafficSituation
- generalInfo
- bannerInfo

7.1 Parameter zum Filtern (3)

filterOMC=<GKZ1>:<GKZ2>:...

Meldungen, die die Gemeinden mit der Kennziffer <GKZ1>, <GKZ2>, ... betreffen. Es können mehrere Gemeindekennziffern durch Doppelpunkt getrennt angegeben werden.

filterOMC_PlaceID=<GKZ>:<ID des Ortes>

Meldungen, den Ort mit der Gemeindekennziffer <GKZ> und der ID <ID des Ortes> betreffen. Der Parameter kann mehrfach übergeben werden.

7.1 Parameter zum Filtern (4)

filterMOTType=<ID>

Meldungen, die das Verkehrsmittel mit der ID <ID> betreffen. Der Parameter kann mehrfach übergeben werden.

Hinweis:

Die Verkehrsmittel sind verbundspezifisch und können von den Standardverkehrsmitteln abweichen.

<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

7.1 Parameter zum Filtern (5)

itdLPxx_selLine=<DIVA Liniennummer>

Meldungen mit der DIVA Liniennummer <DIVA Liniennummer>. Der Parameter kann mehrfach übergeben werden.

itdLPxx_selStop=<Haltestellen ID>

Meldungen, die die Haltestelle mit der ID <Haltestellen ID> betreffen. Der Parameter kann mehrfach übergeben werden.

7.2 Weitere Parameter

AIXMLReduction=<Wert>

Reduziert die XML-Ausgabe. Der Wert bestimmt, welche Elemente entfernt werden sollen. Der Parameter kann mehrfach übergeben werden.

<Wert>	Entferntes Element
removeCreationTime	creationTime
removeExpirationTime	expirationDateTime
removeGenericParams	genericParams
removeInfoLink	infoLink
removeITIncident	itIncident
removeLastModificationTime	lastModificationTime
removeLines	concernedLines
removePublication	publicationDuration
removeSourceSystem	sourceSystem
removeStops	concernedStops
removeValidity	validityPeriod