



Statische EFA XML-Schnittstelle APB Bozen

HTTP-Parameter Dokumentation

Mentz Datenverarbeitung GmbH

Grillparzerstraße 18
81675 München
Tel: +49 89 41868-0
Fax: +49 89 41868-160
mdv@mentzdv.de
www.mentzdv.de

Versionsgeschichte				
Dokument Version	Software Version	Datum	Name	Änderungsgrund
0.1		31.07.2012	SM	Initiale Version
0.2		14.11.2012	SM	AddInfo-Request, StopTimetable-Request, TTB-Request, ROP-Request, Punktverifikation ergänzt
0.3		28.08.2014	SM	CoordInfo-Request

Freigabe			
	Datum	Name	Unterschrift
geprüft:			
freigegeben:			

Inhaltsverzeichnis

1	Allgemeines	10
2	Grundfunktionalität	11
2.1	Sprache	11
	language.....	11
2.2	Generische Parameter	11
	itdLPxx_<Name>	11
2.3	Koordinaten	12
	coordListOutputFormat.....	12
	coordOutputFormat	12
	coordOutputFormatTail.....	12
2.4	Sonstige	13
	execlnst	13
3	Datum und Uhrzeit.....	14
3.1	Datum.....	14
	itdDate	14
	itdDateDay.....	14
	itdDateMonth	14
	itdDateYear	14
	itdDateYearMonth	14
	itdDateDayMonthYear	14
3.2	Uhrzeit.....	15
	itdTime.....	15
	itdTimeHour.....	15
	itdTimeMinute	15
	itdTimeAMPM.....	15
	timeOffset	15

4	Punktverifikation	16
4.1	Obligatorische Parameter.....	16
	locationServerActive = 1	16
	name_<usage>	16
	odvMacro = true	17
	stateless = 1	17
	type_<usage>.....	17
4.2	Listenauswahl	18
4.3	Optionale Parameter zur Optimierung der Punktsuche	18
	anyMaxSizeHitList	18
	anyObjFilter_<usage>	18
	anySigWhenPerfectNoOtherMatches = 1	18
	useHouseNumberList_<usage> = 1	19
	useLocalityMainStop = 1	19
4.4	Hilfetext für Eingabefelder	19
	nameDefaultText_<usage>	19
4.5	Verschiedene Eingabemöglichkeiten ermöglichen	19
	nameInfo_<usage>	19
4.6	Eingabe von Haltestellen POIs und Koordinaten	19
4.7	Wahl einer zugeordneten Haltestelle.....	20
5	Linienverifikation.....	22
5.1	Obligatorische Parameter.....	22
	lineName	22
5.2	Optionale Parameter	22
	exactMatch = 1	22
5.3	Filter	22
	net	22
	opBranch	23

5.4	Eingabe bekannter Linien.....	23
	line.....	24
	IsShowTrainsExplicit = 1	24
	lineVer	24
	dateDay	24
6	StopFinder-Request	25
6.1	Ansteuerung des StopFinder-Requests	25
6.2	Verifizierten Punkt in einen anderen Request übernehmen	26
6.3	Allgemeine Funktionalität	26
7	Trip-Request	27
7.1	Ansteuerung der Fahrtanfrage	27
7.2	Allgemeine Funktionalität	27
	itdTripDateTimeDepArr.....	27
	SpEnclId = 0.....	27
7.3	Allgemeine Optionen	28
	calcNumberOfTrips.....	28
	calcOneDirection = 1	28
7.4	Verkehrsmittelausschluss und Verkehrsmiteinschluss	28
	exclMOT	28
	exclMOT_<ID> = 1	28
	includedMeans	28
	inclMOT_<ID> = 1	29
7.5	Weitere Optionen für den öffentlichen Verkehr	29
	ptOptionsActive = 1	29
	changeSpeed	29
	maxChanges	30
7.6	Weitere Optionen für den Individualtransport	31
	itOptionsActive = 1	31

8	DM-Request	32
8.1	Ansteuerung Anfrage zu einem Abfahrtsmonitor.....	33
8.2	Allgemeine Funktionalität	34
	itdDateTimeDepArr.....	34
	limit.....	34
8.3	Linienauswahl	34
	mode = direct.....	34
9	StopTimetable-Request	35
9.1	Ansteuerung des Aushangfahrplans	35
9.2	Allgemeine Funktionalität	35
	mrgSt = 1.....	35
9.3	Haltestellenauswahl	36
9.4	Linienauswahl	36
	mode = direct.....	36
10	TTB-Request	37
10.1	Ansteuerung der Fahrplanbuchseiten	37
10.2	Allgemeine Funktionalität	37
	contentFilter.....	37
10.3	Linienverifikation.....	38
	mode = line.....	38
10.4	Haltestellenverifikation	38
	mode = odv.....	38
10.5	Linienauswahl	38
	command = direct.....	38
11	ROP-Request.....	39
11.1	Ansteuerung der Anfrage von Stadtplanausschnitten/Linienverlaufsplänen	39
11.2	Allgemeine Funktionalität	39
11.3	Obligatorische Parameter für den Stadtplanausschnitt	40

reqType = spa	40
11.4 Generierung eines Stadtplanausschnitts mittels Punktverifikation	40
mode = odv.....	40
11.5 Direkte Generierung eines Stadtplanausschnitts	40
command = direct.....	40
Haltestelle.....	40
Wichtiger Punkt	40
Koordinate	41
11.6 Obligatorische Parameter für den Linienverlaufsplan.....	41
reqType = lvp.....	41
11.7 Generierung des Linienverlaufsplans mittels Linienverifikation	41
mode = line.....	41
11.8 Generierung des Linienverlaufsplans mittels Haltestellenverifikation und Linienauswahl	41
mode = odv.....	41
11.9 Direkte Generierung eines Linienverlaufsplans	41
command = direct.....	42
11.10 Obligatorische Parameter für den Linienverlaufsplan	42
mergeSup = 1	42
subReqType = overview	42
11.11 Linienverlaufsplan mit Stadtplanausschnitten für die durchfahrenen Haltestellen	42
printOutsPerStop	42
spaOut.....	42
12 AddInfo-Request.....	44
12.1 Ansteuerung der Anfrage von aktuellen Meldungen.....	44
12.2 Datums-Filter.....	44
filterDateValid	44
filterDateValidComponentsActive = 1	44

filterDateValidDay.....	45
filterDateValidMonth	45
filterDateValidYear	45
filterPublicationStatus	45
filterValidIntervalStart und filterValidIntervalEnd.....	45
12.3 Orts-Filter	45
filterOMC	45
filterOMC_PlacelD.....	46
12.4 Linien-, Haltestellen-, Betreiber- und Verkehrsmittel-Filter	46
filterLineNumberIntervalStart und filterLineNumberIntervalEnd.....	46
filterMOTType.....	46
filterPNLineDir	46
filterPNLineSub	46
itdLPxx_selLine	46
itdLPxx_selOperator.....	47
itdLPxx_selStop.....	47
line.....	47
12.5 Filtern nach Meldungstypen und IDs	47
filterInfOLD	47
filterInfoType.....	47
12.6 Filter nach Betreiber und Quelle.....	48
filterProviderCode.....	48
filterSourceSystemName.....	48
12.7 Reduktion	48
AIXMLReduction	48
filterXMLOutput = false	49
13 CoordInfo-Request	50
13.1 Ansteuerung der Koordinatenanfrage	50

13.2	Anfrage-Parameter.....	50
	coord	50
	deadline.....	51
	mapNameOutput	51
	max.....	51
13.3	Filter zur Auswahl der Punkt-Typen und zur Einschränkung des Suchraums	
	51	
	inclFilter = 1	52
	cluster_<Filterindex> = 1	52
	exclLayers_<Filterindex>.....	54
	inclDrawClasses_<Filterindex>	54
	inclPOIH_<Filterindex>.....	54
	radius_<Filterindex>	55
	type_<Filterindex>	55
14	Anhang	56
14.1	Verkehrsmittel	56

1 Allgemeines

Dieses Dokument beschreibt die HTTP-Parameter zur Nutzung der statischen EFA XML-Schnittstelle von APB Bozen. Das Interface umfasst folgende Requests:

- [StopFinder-Request](#): ermöglicht eine einfache Punktverifikation
- [Trip-Request](#): berechnet Fahrtauskünfte von einem Start- zu einem Zielpunkt, zu einer bestimmten Uhrzeit, an einem bestimmten Datum
- [DM-Request](#): erstellt den Abfahrtsmonitor einer oder mehrerer Haltestellen für alle oder bestimmte bedienenden Linien
- [StopTimetable-Request](#): erstellt den Aushangfahrplan einer Haltestelle für eine oder mehrere Linien
- [TTB-Request](#): generiert die Fahrplanbuchseite einer Linie
- [ROP-Request](#): generiert Stadtplanausschnitte und Linienverlaufspläne
- [AddInfo-Request](#): gibt die aktuellen Meldungen aus

Einige HTTP-Parameter stehen in allen Requests zur Verfügung. Diese sind im Kapitel [Grundfunktionalität](#) beschrieben. Von mehreren Requests gemeinsam genutzt werden die Eingabe von [Datum und Uhrzeit](#), die [Punktverifikation](#) und die [Linienverifikation](#).

Alle Anfragen erfolgen zustandslos. Bei Folgeanfragen, wie der Auswahl eines Punktes aus einer Trefferliste oder der bedienenden Linien, die auf einem Abfahrtsmonitor angezeigt werden sollen, müssen alle HTTP-Parameter, deren Wert bei der ersten Anfrage von der Standardeinstellung abgewichen ist, erneut übergeben werden.

2 Grundfunktionalität

Es gibt gewisse Grundfunktionalitäten, die von nahezu allen Requests verwendet werden können.

2.1 Sprache

language

Mit diesem Parameter wird die Sprache festgelegt. Als Wert können die durch den Standard ISO 639:1988 (E/F) definierten Sprachkürzel verwendet werden. Über den Sprach-Parameter wird die Sprache der Interfaces gesteuert. In der XML-Ausgabe wird die Sprache als Attribut `language` des Request-Elements `itdRequest` gespeichert.

Beispiel: Für Deutsch ist der Wert `de`, für Englisch `en` und Italienisch `it`.

Hinweis: Wird dieser Parameter nicht übergeben, wird die Standardsprache verwendet.

2.2 Generische Parameter

itdLPxx_<Name>

Die sogenannten Layout-Parameter sind generische Parameter, die in die XML-Antwort durchgereicht werden, ohne das Systemverhalten zu beeinflussen. Sie beginnen mit dem Präfix `itdLPxx_`, gefolgt von dem eigentlichen, frei wählbaren Parameternamen. Der Parameterwert ist ebenfalls frei wählbar.

In der XML-Ausgabe werden die Layout-Parameter `itdLayoutParam` in dem Container `itdLayoutParams` innerhalb des Anfrageobjektes `itdRequest` gespeichert. Der Parametername entspricht dem Wert des Attributs `name` und der Wert des Parameters dem Wert des Attributs `value`.

Beispiel: Im Browser werden zwei Eingabemasken für die Fahratanfrage angeboten. Eine Variante enthält nur die notwendigsten Eingabefelder. Die zweite Variante bietet zusätzlich verschiedene Fahrtoptionen an. Wird der Startpunkt allein verifiziert, kann anhand der XML-Ausgabe nicht entschieden werden, über welche der beiden Eingabemasken die Eingabe erfolgte. In diesem Fall kann beispielsweise ein Parameter `itdLPxx_advancedInterface=true` für die Eingabemaske mit den Fahrtoptionen übergeben werden. Ist der Wert `false`, wird die minimale Eingabemaske ange-

zeigt. Auf diese Weise kann festgehalten werden, welche der beiden Masken angezeigt wird.

2.3 Koordinaten

Das voreingestellte MDV-interne Koordinatensystem kann mitunter von externen Anwendungen, die über die XML-Schnittstelle auf die EFA-Programme zugreifen, nicht verwendet werden. Alternativ zur externen Umrechnung der Koordinaten kann das Ein- und/oder Ausgabekordinatensystem der EFA geändert werden. Die Koordinatenumrechnung findet dann intern statt. Zur Wahl steht neben dem MDV-Koordinatensystem das WGS84-Format sowie EPSG-Codes.

In diesem Kapitel wird die Ausgabe von Koordinaten behandelt. Die Eingabe wird im Kapitel [Eingabe von Haltestellen, POIs und Koordinaten](#) im Rahmen der [Punktverifikation](#) beschrieben.

coordListOutputFormat

Standardmäßig werden die Koordinaten als Liste ausgegeben. Um die Dateigröße zu reduzieren, empfiehlt es sich die Koordinatenfolge stattdessen als String auszugeben. Dazu wird dem Parameter der Wert `string` statt `list` (Default) übergeben.

coordOutputFormat

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 Ausgabe eines EPSG-Codes. Der Wert des Parameters kann auf folgende Weisen angegeben werden:

- `MDV` (Default, MDV-eigenes Koordinatensystem)
- `PROJ[+init=epsg:<Code>]` (EPSG-Code)
- `WGS84 [DD.DDDDD]` (WGS84-Koordinaten)
- `WGS84 [GGZHTXX]` (WGS84-Koordinaten, Faktor 100.000)
- `WGS84 [GGZHT]` (WGS84-Koordinaten, Faktor 1.000)

coordOutputFormatTail

Standardmäßig werden die Koordinaten mit fünf Nachkommastellen ausgegeben. Die Anzahl der Nachkommastellen kann durch den Wert des Parameters beschränkt werden. Ist der Wert 0, werden keine Nachkommastellen ausgegeben.

Hinweis: Dieser Parameter dient lediglich dazu nicht benötigte Nachkommastellen abzuschneiden. Er erhöht nicht die Genauigkeit der Koordinaten.

2.4 Sonstige

execlnst

Dieser Parameter beschreibt die Ausführung der Anfrage. Mögliche Werte sind:

- `normal` (Default, Berechnung findet statt)
- `verifyOnly` (Nur Punktverifikation, keine Berechnung)
- `readOnly` (Eingaben werden in die XML-Ausgabe übernommen, es finden weder eine Punktverifikation noch eine Berechnung statt)

3 Datum und Uhrzeit

Die Eingabe von Datum und Uhrzeit wird von vielen Requests verwendet. So zum Beispiel beim [Trip-Request](#) und [DM-Request](#). Sie kann auf viele verschiedene Arten erfolgen, um unterschiedliche Datums- und Zeitformate abbilden zu können. Die Angabe der Parameter ist immer optional. Fehlende HTTP-Parameter werden so ergänzt, dass sie den aktuellen Datums- bzw. Uhrzeiteinstellungen des Fahrplanauskunft-Servers entsprechen.

3.1 Datum

itdDate

Eingabe eines vollständigen Datums. Der Wert setzt sich dabei aus der vier- oder zweistelligen Jahreszahl, gefolgt von der zweistelligen Monatsangabe und der zweistelligen Angabe des Tages, zusammen.

itdDateDay

Eingabe des Tages. Der Tag wird als ein- oder zweistelliger Wert angegeben.

itdDateMonth

Eingabe des Monats. Der Monat wird als ein- oder zweistelliger angegeben.

itdDateYear

Eingabe des Jahres. Das Jahr wird als zwei- oder vierstelliger Wert angegeben.

itdDateYearMonth

Eingabe von Jahr und Monat. Der Wert setzt sich dabei aus der vierstelligen Jahreszahl, gefolgt von der zweistelligen Monatszahl, zusammen.

itdDateDayMonthYear

Eingabe von Tag, Monat und Jahr. Der Wert setzt sich aus der zweistelligen Tagsangabe, gefolgt von der zweistelligen Monatsangabe, gefolgt von der zwei- oder vierstelligen Jahreszahl, zusammen. Auf Wunsch kann ein beliebiges Trennzeichen verwendet werden. In diesem Fall muss die Angabe der Jahreszahl jedoch immer vierstellig sein.

3.2 Uhrzeit

itdTime

Eingabe der Uhrzeit. Der Wert kann in einem der folgenden Formate angegeben werden:

- HHMM
- HH : MM
- HH . MM
- MM
- HHMMa (anglo-amerikanisches Format „am“)
- HHMMh (24-Stunden Format)
- HHMMp (anglo-amerikanisches Format „pm“)

itdTimeHour

Eingabe der Stunden. Der Wert gibt in ein- oder zweistelliger Form die Stunden an.

itdTimeMinute

Eingabe der Minuten. Der Wert gibt in ein- oder zweistelliger Form die Minuten an.

itdTimeAMPM

Bei der anglo-amerikanischen Zeitangabe im 12-Stunden-Format unterscheidet der Wert mit `am` (Default) oder `pm` den Vor- und Nachmittag.

timeOffset

Der Wert gibt einen Zeit-Offset an, der auf die aktuelle Zeit des Fahrplanauskunft-Servers bzw. die Anfragezeit aufgerechnet wird. Die Angabe erfolgt in Minuten.

4 Punktverifikation

Die Punktverifikation wird von nahezu allen Requests benötigt. Sie kann auf viele verschiedene Arten erfolgen. Punkte können beispielsweise über Koordinaten, über Haltestellen-Identifikationsnummern oder durch freien Suchtext ermittelt werden. Im Folgenden werden die HTTP-Parameter vorgestellt, durch die die genannten und weitere Eingabearten realisiert werden können. Zusätzliche Parameter dienen dazu das Suchverhalten zu beeinflussen und zu optimieren. Andere wirken als Filter und schränken den Suchraum ein oder erwirken eine Reduzierung der Trefferlisten.

Die im Zusammenhang der Punktverifikation häufig gebrauchte Abkürzung ODV steht für Origin-Destination-Via. Sie bezieht sich auf die Funktion, die der Punkt innerhalb eines Requests inne hat. Die Funktion eines Punktes wird durch die Ergänzung `<usage>` am Ende eines Parameters bestimmt. Parameter, die eine solche Ergänzung benötigen, werden für jeden Punkt gemäß seiner Funktionalität separat definiert. Folgende Werte kann `<usage>` annehmen:

- `origin, destination, via` (für Trip-Request)
- `dm` (für DM-Request)
- `sf` (für StopFinder-Request)
- `stt` (für StopTimetable-Request)
- `ttb` (für TTB-Request)
- `rop` (für ROP-Request)

Es wird die Ermittlung von Punkten (Haltestellen, Adressen, wichtige Punkte,...) mittels des EFA Location Servers beschrieben. Er ermöglicht eine Einfeld-Suche.

4.1 Obligatorische Parameter

locationServerActive = 1

Aktiviert den EFA Location Server. Der EFA Location Server stellt eine Einfeld-Suche zur Verfügung, die es ermöglicht alle Angaben zum Namen des Punktes über die im Folgenden beschriebenen Parameter anzugeben.

name_<usage>

Dieser Parameter enthält den Namen des Punktes. Ein Punkt entspricht dem Element `itdOdv` in der XML-Ausgabe. Seine Funktion ist im Attribut `usage` angegeben, das Element `itdOdvName` bezieht sich auf den Namen des Punktes.

Ein Punkt kann eine Haltestelle, eine Adresse, ein POI, usw. sein. Der Wert des Parameters ist abhängig vom Status der Punktverifikation. Der Status des Punktes ist dem Attribut `state` des Elements `itdOdvName` zu entnehmen:

- `state= empty` oder `state=notidentified: name_<usage>` enthält die Eingabe des Anwenders.
- `state= list: name_<usage>` enthält den Wert (Attribut `stateless` aus XML-Element `odvNameElem`) des ausgewählten Elements. Dieser Fall tritt ein, wenn kein Volltreffer ermittelt werden konnte und in der vorherigen Anfrage eine Liste zurückgegeben wurde.
- `state=identified: name_<usage>` enthält den Wert des Attributs `statless` (siehe Parameter `stateless=1`) aus dem XML-Element `odvNameElem`.

odvMacro = true

Spricht ein für den EFA Controller definiertes Parameter-Macro an. In diesem Macro sind die Bewertungskriterien bezüglich der Trefferqualität definiert.

stateless = 1

Fügt identifizierten Punkten im Element in der XML-Ausgabe das Attribut `stateless` hinzu. Der Wert des Attributs ergibt bei der Punktsuche immer einen Volltreffer, wenn er als Wert des Parameters `name_<usage>` übergeben wird.

type_<usage>

Dieser Parameter beschreibt den Typ des Punktes. Mögliche Werte sind:

- `poiID`
- `stopID`
- `coord`
- `any`

Für die Einfeld-Suche muss der Wert `any` sein. Die übrigen Werte dienen zum Eingeben von Volltreffern (siehe Kapitel [Eingabe von Haltestellen, POIs und Koordinaten](#)). Zur Einschränkung der Suchmenge bei der Einfeld-Suche steht der Parameter `anyObjFilter_<usage>` zur Filterung bereit.

4.2 Listenauswahl

Ergibt ein Punkt kein Volltreffer, enthält das Element `itdOdvName` mehrere Elemente `odvNameElem`. Zur Auswahl eines Punktes wird der Wert des Attributes `stateless` durch den Parameter `name_<usage>` übergeben.

4.3 Optionale Parameter zur Optimierung der Punktsuche

anyMaxSizeHitList

Gibt die maximale Länge der Trefferliste an. Übersteigt die die Anzahl der gefundenen Treffer den angegebenen Wert, wird die Liste reduziert. Die Reduktion erfolgt anhand der Trefferqualität. Da die Trefferliste vor der Anwendung der durch das Parameter-Macro `odvMacro` definierten Bewertungskriterien reduziert wird, sollte der Wert jedoch nicht zu niedrig gewählt werden.

anyObjFilter_<usage>

Die Suche kann auf bestimmte Objekttypen eingegrenzt werden. Der Wert des Parameters ist eine Bitmaske. Die einzelnen Objekttypen lassen sich beliebig kombinieren:

- 0 (Kein Filter aktiv)
- 1 (Orte)
- 2 (Haltestellen-IDs und -Aliasnamen)
- 4 (Straßennamen)
- 8 (Adressen)
- 16 (Kreuzungen)
- 32 (POI-IDs und -Aliasnamen)
- 64 (Postleitzahlen)

Beispiel: `anyObjFilter_origin=34` (2 + 32 → Haltestellen und POIs)

anySigWhenPerfectNoOtherMatches = 1

Wenn der gesuchte Punkt als perfekter Treffer erkannt wird, wird nur dieser berücksichtigt. Ansonsten werden auch Treffer, die der Eingabe ähnlich sind gefunden. Auf diese Weise kann entweder eine kurze Trefferlisten oder eine tolerante Suche ermöglicht werden.

useHouseNumberList_<usage> = 1

Wird dieser Parameter eingeschaltet, wird im Rahmen der Adress-Verifikation eine Liste möglicher Hausnummern ausgegeben.

useLocalityMainStop = 1

Ein verifizierter Ort entspricht einer Koordinate, es müssen umliegende Haltestellen gesucht werden. Dieser Parameter ändert dieses Standardverhalten. Statt der Suche nach umliegenden Haltestellen werden die dem Ort zugeordneten Haltestellen verwendet. Auf diese Weise kann ein Ort mit einer Haupthaltestelle assoziiert werden.

4.4 Hilfetext für Eingabefelder

nameDefaultText_<usage>

Der durch den Parameter übergebene Text wird bei der Punktsuche nicht berücksichtigt. Dadurch ist es möglich im Eingabefeld einen Hilfetext anzuzeigen.

4.5 Verschiedene Eingabemöglichkeiten ermöglichen

nameInfo_<usage>

Dieser Parameter überschreibt den Parameter `name_<usage>`. Dies kann notwendig sein, wenn mehrere Eingabemöglichkeiten zur Verfügung gestellt werden sollen. Muss der Parameter übergeben werden, soll jedoch keine Wirkung zeigen, kann er deaktiviert werden, indem der Wert auf `invalid` gesetzt wird.

4.6 Eingabe von Haltestellen POIs und Koordinaten

Soll ein Punkt nicht verifiziert werden, sondern ein bekannter Punkt verwendet werden, kann dieser über seine ID oder Koordinate identifiziert werden. Da keine Punktverifikation stattfindet, ist es nicht notwendig den EFA Location Server zu verwenden. Den EFA Location Server betreffende Parameter können daher weggelassen werden. Es sind lediglich die Parameter `type_<usage>`, `name_<usage>` und `stateless=1` notwendig.

Haltestellen

Eine Haltestelle kann eindeutig über ihre Haltestellen ID identifiziert werden. In der XML-Antwort kann diese nach abgeschlossener Haltestellenverifikation dem Attribut `stateless` des Punktes `odvNameElem` entnommen werden. Sie wird als Wert des Parameters `name_<usage>` übergeben. Der Parameter `type_<usage>` hat den Wert `stopID`.

Beispiel: `type_origin=stopID&name_origin=66000468`

POI

Ein POI kann eindeutig über seine ID in Kombination mit der Gemeindekennziffer und der ID des Ortes identifiziert werden. In der XML-Antwort können diese Angaben nach abgeschlossener POI-Verifikation dem Attribut `stateless` des Elements `odvNameElem` entnommen werden. Der Suffix `poiID` wird weggelassen. Parameters `name_<usage>` übergeben. Der Parameter `type_<usage>` hat den Wert `poiID`.

Beispiel: `type_origin=poiID&name_origin= 895:8111000:-1`

Koordinaten

Eine Koordinaten wird über die beiden Parameter `type_<usage>=coord` und `name_<usage>=<x>:<y>:<Koordinatenformat>:<Freitext>` eingegeben. Der Wert des Parameters `name_<usage>` setzt sich aus drei obligatorischen Werten und einem optionalen Wert, getrennt durch Doppelpunkte, zusammen. `<x>` und `<y>` stellen die x- und die y-Koordinate da. Der für die Wahl des Koordinatenformat entscheidende Parameter ist `<Koordinatenformat>`. Dieser gibt das Koordinatenformat (siehe Kapitel [Koordinaten](#)) an. Der letzte Wert, `<Freitext>` ist optional. Wird kein Freitext gewählt, versucht das System auf die nächstgelegene Straße zu schnappen.

Beispiel: `name_origin=6.775652:51.224942:WGS84[DD.DDDDD]` oder `name_origin=1678773:352096:APBV:ein schöner Ort`

4.7 Wahl einer zugeordneten Haltestelle

Einige Anfragearten, wie zum Beispiel der [StopTimetable](#)- oder der [TTB-Request](#), benötigen eine eindeutig identifizierte Haltestelle. Diese kann über die Punktverifikation ermittelt werden. Wird jedoch nach einer Adresse, einem wichtigen Punkt oder

einer Koordinate gesucht, muss anschließend eine zugeordnete Haltestelle ausgewählt werden. Auch komplexere Haltestellen wie ein Hauptbahnhof können durch mehrere Haltestellen modelliert sein. Es werden maximal zehn zugeordnete Haltestellen ermittelt. Dies sind die am nächsten gelegenen Haltestellen.

Sind zugeordnete Haltestellen vorhanden, werden diese als `itdAssignedStop` im Punkt `itdOdv` in der XML Antwort ausgegeben. Die Wahl einer zugeordneten Haltestelle (siehe Abbildung 1 - Wahl einer zugeordneten Haltestelle am Beispiel Fahrplanbuchseiten) erfolgt über den Parameter `name_<usage>`. Als Wert wird der Wert des Attributs `stateless` übergeben.

Fahrplanbuchseiten

Suchergebnis für Haltestelle Stuttgart / Hauptbahnhof

Zugeordnete Haltestellen

- Hauptbahnhof (tief)
- Hauptbahnhof (oben)
- Hauptbf (Arnulf-Klett-Platz)

Abbildung 1 - Wahl einer zugeordneten Haltestelle am Beispiel Fahrplanbuchseiten

5 Linienvifikation

Verschiedene Requests, wie zum Beispiel der [TTB-Request](#) oder der Linienvverlaufplan (siehe [ROP-Request](#)) ermöglichen die Vifikation einer Linie bzw. benötigen eine Linie als Eingabe. Die Linie wird anhand ihres Namens gesucht. Filter können den Suchraum eingrenzen. Alternativ können bereits bekannte Linien als eindeutige Treffer übergeben werden.

5.1 Obligatorische Parameter

lineName

Gibt den Liniennamen an. Als Wert kann eine Liniennummer oder ein Teil einer Liniennummer eingegeben werden. Dieser wird mit dem Liniennamen (auch: Linienbezeichner für BildschirmAusgabe) verglichen. In der XML-Ausgabe entspricht der Liniennamen dem Attribut `number` einer Linie `itdServingLine`.

Hinweis: Bei der Eingabe von Zahlen werden diese nur mit den kompletten Zahlen der Liniennamen verglichen. So wird bei der Eingabe von `linienName=10` die Linie U10 und U9/U10 gefunden, nicht aber die Linien 100 und 101.

5.2 Optionale Parameter

exactMatch = 1

Bewirkt, dass bei der Liniensuche nach dem Liniennamen durch den Parameter `lineName` nur Volltreffer berücksichtigt werden.

5.3 Filter

net

Filter für Linien des als Parameterwert übergebenen Teilentzes. Der Wert des Parameters entspricht dem Attribut `network` des Elements `motDivaParams` der XML-Ausgabe.

opBranch

Mit dem Parameter kann ein Betriebszweig übergeben werden. Auf diese Weise kann nach Betreibern gefiltert werden, denn es werden ausschließlich Linie der Betreiber des gewählten Betriebszweigs ausgegeben. Durch mehrfache Verwendung des Parameters können mehrere Betriebszweige gewählt werden.

5.4 Eingabe bekannter Linien

Ist die Linie bereits bekannt, so kann diese direkt als Volltreffer übergeben werden. Dies ist beispielsweise der Fall, wenn auf der Ergebnisseite des [Trip-Requests](#) Aushangfahrpläne (siehe [StopTimetable-Request](#)) oder Fahrplanbuchseiten (siehe [TTB-Request](#)) verlinkt werden sollen. Dort befinden sich die Angaben zur Linie im Element `motDivaParams` im Container `itdMeansOfTransport` jedes Teilwegs `itdPartialRoute`.

Ein weiterer Anwendungsfall ist die Wahl einer zugeordneten Linie einer Haltestelle, eines wichtigen Punktes oder einer Adresse. Nach der Punktverifikation befinden sich die bedienenden Linien im Container `itdServingLines` als `itdServingLine` in der XML Ausgabe. Es bietet sich die Implementation einer Linienauswahl an (siehe Abbildung 2 - Wahl zweier bedienender Linien am Beispiel Aushangfahrplan).



Abbildung 2 - Wahl zweier bedienender Linien am Beispiel Aushangfahrplan

Folgende Parameter werden für die Übergabe von Linien benötigt:

line

Eine konkrete Linie wird durch eine durch Doppelpunkt getrennte Liste von Teilnetz, DIVA-Liniennummer, Ergänzung und Richtung `<Netz>:<DIVA-Linie>:<Ergänzung>:<Richtung>` angegeben. Soll dabei einer der Parameter nicht berücksichtigt werden, wird der Wert leer gelassen. Zur Wahl mehrerer Linien kann der Parameter mehrfach verwendet werden.

Beispiel: `apb:05136:12b:R` oder `apb:05136::`

IsShowTrainsExplicit = 1

Standardmäßig werden keine Züge bei den bedienenden Linien ausgegeben. Die Anzeige der Züge in der Linienauswahl muss explizit durch diesen Parameter aktiviert werden.

Der Aushangfahrplan (siehe [StopTimetable-Request](#)), die Fahrplanbuchseiten (siehe [TTB-Request](#)) und der Linienverlaufsplan (siehe [ROP-Request](#)) benötigen zusätzlich weitere Parameter für die Linienauswahl.

lineVer

Wählt die Linienversion aus. Die Linienversion ist dem Attribut `spTr` des Linien-Elements `itdServingLine` der XML-Ausgabe zu entnehmen. Ist die Linienversion nicht bekannt, kann alternativ das Datum mit dem Parameter `dateDay` übergeben werden. Die Linienversion wird dann anhand dessen intern ermittelt.

dateDay

Mit diesem Parameter kann der das Datum, für den der Aushangfahrplan, die Fahrplanbuchseite oder der Linienverlaufsplan erstellt werden soll, angegeben werden. Der Wert ist die vierstellige Jahreszahl, gefolgt von dem zweistelligen Monat und dem zweistelligen Tag. Die Angabe des Datums ist nur notwendig, wenn die Linienversion nicht bekannt ist.

6 StopFinder-Request

Im Folgenden wird die Punktverifikation mittels des StopFinder-Requests erläutert. Über HTTP-Parameter wird ein Punkt, zum Beispiel eine Adresse, ein wichtiger Punkt oder eine Haltestelle, angefragt. Die Punktverifikation ist ebenfalls Teil anderer Requests, zum Beispiel des [Trip-Requests](#) und des [DM-Requests](#). Soll jedoch außer der Ermittlung eines Punkts keine weitere Berechnung vorgenommen werden, empfiehlt sich die Verwendung des StopFinder-Requests aus Performancegründen.

Eine typische Anwendung für den StopFinder-Request ist das Erstellen einer Vorschlagsliste (siehe Abbildung 3 - Vorschlagsliste), die den Anwender bei der Eingabe unterstützt.



Abbildung 3 - Vorschlagsliste

6.1 Ansteuerung des StopFinder-Requests

Der StopFinder-Request wird mit HTTP-Parametern gesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (XML_STOPFINDER_REQUEST) oder in HTML (XSLT_STOPFINDER_REQUEST) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_STOPFINDER_REQUEST?HTTP-Parameter
```

Für eine HTML Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_STOPFINDER_REQUEST?HTTP-Parameter
```

6.2 Verifizierten Punkt in einen anderen Request übernehmen

Durch Verwendung des Werts des Attributs `stateless`, kann ein mit dem StopFinder-Request ermittelter Punkt an andere Requests als Volltreffer übergeben werden. Dies ist im Kapitel [Punktverifikation](#) näher beschrieben.

6.3 Allgemeine Funktionalität

Der StopFinder-Request verwendet die im Kapitel [Punktverifikation](#) beschriebenen Parameter zur Punktverifikation. Die Parameter verwenden das Suffix `sf`.

7 Trip-Request

Im Folgenden wird die Ansteuerung der Fahrthanfrage (Trip-Request) erläutert. Die Fahrthanfrage berechnet Fahrten zu einem angegebenen Datum und einer angegebenen Uhrzeit. Mit An-/Abkennung, Start- und Zielpunkt sowie Fahrtoptionen.

7.1 Ansteuerung der Fahrthanfrage

Die Fahrthanfrage wird über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (`XML_TRIP_REQUEST2`) oder in HTML (`XSLT_TRIP_REQUEST2`) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_TRIP_REQUEST2?HT  
TP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_TRIP_REQUEST2?HT  
TP-Parameter
```

7.2 Allgemeine Funktionalität

Für die Fahrtberechnung müssen mindestens ein Start- und ein Zielpunkt verifiziert werden. Für die Parameter des Startpunktes wird das Suffix `origin`, für den Zielpunkt der Suffix `destination` verwendet. Die Angabe von einer Abfahrts- oder Ankunftszeit ist optional.

Es können die in den Kapiteln [Grundfunktionalität](#), [Datum und Uhrzeit](#) sowie [Punktverifikation](#) beschriebenen HTTP-Parameter verwendet werden. Außerdem:

itdTripDateTimeDepArr

Der Wert des Parameters bestimmt, ob sich Datum und Uhrzeit auf die Abfahrt (`dep`) oder die Ankunft (`arr`) beziehen. Default ist `dep`.

SpEnclId = 0

Gibt die Zeichenencodierung an. Dieser Parameter ist notwendig, wenn das Anfrage-System UTF-8-codiert anfragt.

7.3 Allgemeine Optionen

calcNumberOfTrips

Gibt die Anzahl der Fahrten mit Öffentlichen Verkehrsmitteln an. Standardmäßig werden vier Fahrten berechnet. Gibt es alternative Fahrten, kann die angegebene Anzahl überschritten werden. Alternative Fahrten sind daran zu erkennen, dass in der XML-Antwort das Attribut `alternative` einer Fahrt `itdRoute` den Wert 1 hat.

calcOneDirection = 1

Standardmäßig wird eine der Fahrten vor der gewählten Abfahrtszeit bzw. nach der gewählten Ankunftszeit ausgegeben. Dieser Parameter unterdrückt das Standardverhalten.

7.4 Verkehrsmittelausschluss und Verkehrsmiteinschluss

Die Belegung der Verkehrsmittel-IDs von APB Bozen kann dem Kapitel [Verkehrsmittel](#) entnommen werden.

excludedMeans

Dieser Parameter kann auf zwei Arten verwendet werden:

Der Wert `checkbox` aktiviert das Ausschließen von Verkehrsmitteltypen. Die auszuschließenden Verkehrsmittel werden dann mit dem Parameter `exclMOT_<ID>` angegeben. Standardmäßig sind alle Verkehrsmittel aktiviert.

Alternativ kann die ID des auszuschließenden Verkehrsmittels als Wert angegeben werden. Der Parameter kann mehrfach verwendet werden.

exclMOT_<ID> = 1

Dieser Parameter bewirkt dass der Verkehrsmitteltyp mit der als Wert übergebenen ID ausgeschlossen wird. Zum Ausschließen mehrerer Verkehrsmittel kann der Parameter mehrfach verwendet werden.

Achtung: Das gewählte Verkehrsmittel wird ausgeschlossen sobald der entsprechende Parameter übergeben wird. Auch wenn dieser keinen Wert oder die Werte 0 bzw. `false` hat.

includedMeans

Dieser Parameter kann auf zwei Arten verwendet werden:

Der Wert `checkbox` aktiviert das Einschließen von Verkehrsmitteltypen. Die einzuschließenden Verkehrsmittel werden dann mit dem Parameter `inclMOT_<ID>` angegeben. Standardmäßig sind alle Verkehrsmittel deaktiviert.

Alternativ kann die ID des einzuschließenden Verkehrsmittels als Wert angegeben werden. Der Parameter kann mehrfach verwendet werden.

inclMOT_<ID> = 1

Dieser Parameter bewirkt dass der Verkehrsmitteltyp mit der als Wert übergebenen ID eingeschlossen wird. Zum Einschließen mehrerer Verkehrsmittel kann der Parameter mehrfach verwendet werden.

Achtung: Das gewählte Verkehrsmittel wird eingeschlossen sobald der entsprechende Parameter übergeben wird. Auch wenn dieser keinen Wert oder die Werte 0 bzw. `false` hat.

7.5 Weitere Optionen für den öffentlichen Verkehr

ptOptionsActive = 1

Sollen die Optionen für den öffentlichen Verkehr verwendet werden, müssen diese zunächst mit diesem Parameter aktiviert werden. Standardmäßig sind die Optionen für den öffentlichen Verkehr deaktiviert.

changeSpeed

Modifiziert die für Umstiegsfußwege benötigte Zeit. Folgende Werte sind möglich:

- `fast` (entspricht 50)
- `normal` (default, entspricht 100)
- `slow` (entspricht 200)

Der Wert stellt einen Faktor dar, mit dem die in der Umstiegsmatrix definierte Zeit nach folgender Formel modifiziert wird:

$$\text{Umstiegszeit [min]} = (\text{Zeit aus der Umstiegsmatrix [min]} * \text{Wert des Parameters}) / 100$$

Falls sich dieser Parameter auch auf den Weg vom Startpunkt zur ersten Haltestelle und von der letzten Haltestelle zum Zielpunkt beziehen soll, muss sichergestellt sein, dass die Optionen für den Individualtransport durch den Parameter `itOptionsActive=1` aktiviert sind (siehe Kapitel [Weitere Optionen für den Individualtransport](#)).

Für den Individualtransport sind die Alias-Namen wie folgt definiert:

- `fast` (entspricht 120)
- `normal` (default, entspricht 100)
- `slow` (entspricht 80)

Der Wert stellt einen Faktor dar, mit dem die für den EFA IT Kernel konfigurierte Standardgeschwindigkeit nach folgender Formel modifiziert wird:

$$\text{Geschwindigkeit [km/h]} = (\text{Standardgeschwindigkeit [km/h]} * \text{Wert des Parameters}) / 100$$

Die Standardgeschwindigkeit beträgt 4 km/h. Bei starken Steigungen (>10%) wird diese jedoch automatisch herabgesetzt.

In der Regel wird für einen Umstieg mehr Zeit benötigt als für den Individualtransport, da der Passagier Treppen überwinden und sich seinen Weg durch andere Passagiere bahnen muss. Daher sind die Werte für die Aliasnamen *fast* und *slow* für den Individualtransport nicht nur gegenüber dem Umstiegsfußweg vertauscht, sondern auch erhöht.

Achtung: Alternativ zu den Alias-Namen können auch Werte zwischen 25 und 400 angegeben werden. Wirkt sich der Parameter *changeSpeed* sowohl auf den Umstiegsfußweg als auch auf den Individualtransport aus, sollte immer ein Alias-Name verwendet werden. Da der Wert des Parameters eine unterschiedliche Bedeutung für die Berechnung dieser beiden Fälle hat, verhält er sich gegenläufig.

maxChanges

Maximale Anzahl von Umstiegen einer Fahrt. Fahrten mit mehr als den angegebenen Umstiegen werden bei der Fahrtberechnung verworfen. Standardmäßig sind 9 Umstiege als maximale Anzahl definiert, dieser Wert kann jedoch in der Konfigurationsdatei des EFA Controllers angepasst werden. Mögliche Werte sind:

- 0 (Direktverbindung)
- 1
- 2
- 9 (bzw. das in der Konfigurationdatei des EFA Controllers definierte Maximum, Default)

7.6 Weitere Optionen für den Individualtransport

itOptionsActive = 1

Sollen die Optionen für den Individualtransport verwendet werden, müssen diese zunächst mit diesem Parameter aktiviert werden. Standardmäßig sind die Optionen für den Individualtransport ausgeschaltet.

changeSpeed

Dieser Parameter nicht nur für den Individualtransport, sondern auch für den öffentlichen Verkehr. Er ist im Kapitel [Weitere Optionen für den öffentlichen Verkehr](#) beschrieben.

8 DM-Request

Im Folgenden wird die Ansteuerung der Anfrage zu einem Abfahrtsmonitor (DM-Request) erläutert. Grundsätzlich gilt dabei zu beachten, dass es zwei unterschiedliche Typen von Abfahrtsmonitoren gibt: den Abfahrtsmonitor einer konkreten Haltestellen (siehe Abbildung 4 - Abfahrtsmonitor einer Haltestelle) und einen kombinierten Abfahrtsmonitor für mehrere Haltestellen (siehe Abbildung 6 - Kombiniertes Abfahrtsmonitor).

Für den Abfahrtsmonitor einer Haltestelle muss zunächst die Haltestelle verifiziert werden. Die Haltestelle kann dabei auch über die Anfrage einer Adresse, eines POIs oder einer Koordinate ermittelt werden. In diesem Fall muss in einem Zwischenschritt eine der vom System ermittelten zugeordneten Haltestellen ausgewählt werden (siehe Kapitel [Wahl einer zugeordneten Haltestelle](#)).

Abfahrten		
Zeit	Linie / Richtung	
<input type="checkbox"/> Abfahrtstafel aktualisieren (minütlich)		
Haltestelle: Bozen, Bahnhof 1		<input checked="" type="checkbox"/> Später
08:57	 poonte nova-collepietra-bolzano/birch... / Kreuzwirt	
09:12	 bolzano-siusi-ortisei/bozen-seis-st.u... / St.Ulrich Antoniusplatz	
09:18	 castelrotto-bolzano/kastelruth-bozen / Perathonerstr.	
09:25	 bolzano-soprabolzano/bozen-oberbozen / Oberbozen	
09:31	 bolzano-soprabolzano/bozen-oberbozen / Perathonerstr.	
09:33	 Collepietra-prato isarco-bolzano/Stei... / Perathonerstr.	
09:42	 castelrotto-bolzano/kastelruth-bozen / Kastelruth	
09:46	 bolzano-siusi-ortisei/bozen-seis-st.u... / Perathonerstr.	
09:58	 plan-novale/ried-bz / Perathonerstr.	
10:12	 bolzano-siusi-ortisei/bozen-seis-st.u... / St.Ulrich Antoniusplatz	
10:12	 castelrotto-bolzano/kastelruth-bozen / Kastelruth	
10:13	 bressanone-bolzano/brixen-bozen / Perathonerstr.	
10:17	 bressanone-bolzano/brixen-bozen / Busbahnhof	
10:18	 castelrotto-bolzano/kastelruth-bozen / Perathonerstr.	
10:27	 Collepietra-prato isarco-bolzano/Stei... / Steinegg	

Abbildung 4 - Abfahrtsmonitor einer Haltestelle

Wird ein kombinierter Abfahrtsmonitor für mehrere Haltestellen angezeigt, fällt der Schritt der Wahl einer zugeordneten Haltestelle weg. Allerdings sollte im Abfahrtsmonitor darauf hingewiesen werden an welcher Haltestelle abgefahren wird bzw. sollen die Abfahrten pro Halteselle gruppiert werden. Dies ist dem Attribut `stopName` in der Abfahrt `itdDeparture` in der XML-Antwort zu entnehmen.



Abfahrten	
<input type="checkbox"/> Abfahrtstafel aktualisieren (minütlich)	
Zeit	Linie / Richtung
Haltestelle: Gelsenkirchen Freiheit ▼ Später	
12:14	 255 / GLADBECK OBERHOF * ÜBER GE-SCHOLVEN
12:15	 210 / Recklinghausen Marderweg
Haltestelle: Gelsenkirchen St.-Marien-Hospi (2 Min. Fußweg) ▼ Später	
12:16	 247 / Gelsenkirchen Buer Rathaus
Haltestelle: Gelsenkirchen Königswiese (6 Min. Fußweg) ▼ Später	
12:15	 211 / Gelsenkirchen Buer Rathaus
12:16	 211 / Gelsenkirchen Buer Rathaus
12:16	 91 / Gelsenkirchen Buer Rathaus

Abbildung 5 - Kombiniertes Abfahrtsmonitor

In beiden Fällen können nach der Punktverifikation optional in einem Zwischenschritt eine oder mehrere bedienenden Linien ausgewählt werden (siehe Kapitel [Linienauswahl](#)). Der Abfahrtsmonitor zeigt nur die Abfahrten der ausgewählten Linien an.

8.1 Ansteuerung Anfrage zu einem Abfahrtsmonitor

Die Anfrage zur Ansteuerung eines Abfahrtsmonitors wird über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (`XML_DM_REQUEST`) oder in HTML (`XSLT_DM_REQUEST`). Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_DM_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_DM_REQUEST?HTTP-Parameter
```

8.2 Allgemeine Funktionalität

Für die Anfrage eines Abfahrtsmonitors muss mindestens ein Abfahrtspunkt verifiziert werden. Im Falle des Abfahrtsmonitors für eine Haltestelle ist dies eine Haltestelle. Dabei wird das Parameter-Suffix `dm` verwendet.

Es können die in den Kapiteln [Grundfunktionalität](#), [Datum und Uhrzeit](#) sowie [Punktverifikation](#) beschriebenen HTTP-Parameter verwendet werden. Außerdem:

itdDateTimeDepArr

Bestimmt, ob sich die Zeit auf die Abfahrt oder Ankunft bezieht. Auf diese Weise kann durch die Werte `dep` (Default) und `arr` zwischen einem Abfahrts- und Ankunftsmonitor unterschieden werden.

limit

Dieser Parameter legt die maximale Anzahl an Abfahrten fest, die ausgegeben werden. Standardmäßig werden bis zu 40 Fahrten innerhalb eines Zeitraums von maximal zwei Tagen ausgegeben.

8.3 Linienauswahl

Der letzte Schritt vor der Anzeige des Abfahrtsmonitors ist die Auswahl der Linien. Diese wird im Kapitel [Eingabe bekannter Linien](#) beschrieben. Zusätzlich wird folgender Parameter benötigt:

mode = direct

Soll keine Auswahl der Linien stattfinden und der Abfahrtsmonitor für alle Linien ausgegeben werden, muss der Direkt-Modus des Abfahrtsmonitors verwendet werden. Der Parameter `mode=direct` wird übergeben, sobald die Haltestelle identifiziert ist. Ansonsten wird er erst mit der Linienwahl übergeben.

9 StopTimetable-Request

Im Folgenden wird die Ansteuerung des Aushangfahrplans (StopTimetable-Request) erläutert. Aushangfahrpläne werden für bestimmte Haltestellen und Linien erstellt.

Der Ablauf der Anfrage sieht vor zunächst eine Haltestelle auszuwählen. Die Vorgehensweise ist im Kapitel [Haltestellenauswahl](#) beschrieben. Anschließend werden eine oder mehrere der bedienenden Linien ausgewählt, für die jeweils ein Aushangfahrplan erstellt wird. Die Auswahl der Linie wird im Kapitel [Linienauswahl](#) erläutert.

9.1 Ansteuerung des Aushangfahrplans

Der Aushangfahrplan wird über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (XML_STT_REQUEST) oder in HTML (XSLT_STT_REQUEST) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_STT_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_STT_REQUEST?HTTP-Parameter
```

Der Aushangfahrplan wird als PDF-Datei generiert. Der Link zu dieser Datei befindet sich im Attribut `src` des Elements `itdDownloadFile` im Container `itdDownload` in der XML Antwort.

9.2 Allgemeine Funktionalität

Um einen Aushangfahrplan generieren zu können, müssen mindestens eine Haltestelle verifiziert und eine ihrer bedienenden Linie ausgewählt werden. Bei der Verifikation der Haltestelle wird das Parameter-Suffix `stt` verwendet.

Es können die in den Kapiteln [Grundfunktionalität](#) sowie [Punktverifikation](#) beschriebenen HTTP-Parameter verwendet werden. Außerdem:

mrgSt = 1

Die Aushangfahrpläne werden standardmäßig steiggenau ausgegeben. Das kann zu doppelten Aushangfahrplänen führen, wenn Aushangfahrpläne für mehrere Steige gelten. Ist dieser Parameter aktiv, werden doppelt vorhandene Aushangfahrpläne nur einfach ausgegeben.

Chelsea, Royal Hospital
Sloane Square
Brompton (London), Blacklands Terrace
Chelsea, Kensington & Chelsea College
South Kensington, Fulham Road / Sloane Avenue
South Kensington
South Kensington
Brompton (London), Victoria & Albert Museum
Knightsbridge (London), Imperial College
Knightsbridge (London), Kensington / Royal Albert Hall
Knightsbridge (London), Kensington / Queens Gate
Knightsbridge (London), Kensington / Prince Consort Rd

Abbildung 6 - Aushangfahrplan der Linie 360 von Traveline South East mit mrgSt=0

9.3 Haltestellenauswahl

Die Haltestellenauswahl erfolgt mit den im Kapitel [Punktverifikation](#) beschriebenen Parametern. Wird die Haltestelle nicht als Volltreffer übergeben, muss sichergestellt werden, dass am Ende des Punktverifikationsprozesses eine Haltestelle gewählt wurde. Bei der Suche nach wichtigen Punkten oder Adressen, erfolgt dies beispielsweise wie im Abschnitt [Wahl einer zugeordneten Haltestelle](#) beschriebenen.

9.4 Linienauswahl

Ist die Haltestelle verifiziert, werden eine oder mehrere bedienende Linien gewählt, für die jeweils ein Aushangfahrplan als PDF generiert wird. Die Wahl der Linie erfolgt wie im Kapitel [Eingabe bekannter Linien](#) beschrieben. Zusätzlich ist folgender Parameter notwendig:

mode = direct

Dieser Parameter ist für die Wahl einer oder mehrerer Linien notwendig.

10 TTB-Request

Im Folgenden wird die Ansteuerung der Fahrplanbuchseiten (TTB-Request) erläutert. Eine Fahrplanbuchseite wird für eine Linie generiert. Diese kann auf zwei Arten ausgewählt werden:

1. [Linienverifikation](#)
2. [Haltestellenverifikation](#) und [Linienauswahl](#)

10.1 Ansteuerung der Fahrplanbuchseiten

Die Fahrplanbuchseiten werden über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (`XML_TTB_REQUEST`) oder in HTML (`XSLT_TTB_REQUEST`) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_TTB_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_TTB_REQUEST?HTTP-Parameter
```

Die Fahrplanbuchseiten werden als PDF-Datei generiert. Der Link zu dieser Datei befindet sich im Attribut `src` des Elements `itdDownloadFile` im Container `itdDownload` in der XML Antwort.

10.2 Allgemeine Funktionalität

Es können die in den Kapiteln [Grundfunktionalität](#) sowie [Punktverifikation](#) und [Linienverifikation](#) beschriebenen HTTP-Parameter verwendet werden. Bei der Punktverifikation wird das Parameter-Suffix `ttb` verwendet. Außerdem können folgende Parameter übergeben werden:

contentFilter

Auswahl welche Haltestellen in den Fahrplanbuchseiten berücksichtigt werden. Ist der Wert des Parameters `allstops`, was der Standardeinstellung entspricht, werden alle Haltestellen der Linie aufgelistet. Bei `timingpoints` hingegen werden nur die Haltestellen, für die explizite Abfahrtszeiten vorliegen, angezeigt. Nur an diesen Haltestellen ist garantiert, dass die Transportmittel die Abfahrtszeiten abwarten.

Liegen nicht für alle Haltestellen Abfahrtszeiten vor, aber trotzdem alle Haltestellen

angezeigt, so werden die Abfahrtszeiten für alle Haltestellen ohne explizite Abfahrtszeiten interpoliert.

10.3 Linienvorifikation

Die Linienvorifikation erfolgt wie im Kapitel [Linienvorifikation](#) beschrieben. Zusätzlich ist folgender Parameter notwendig:

mode = line

Dieser Parameter leitet die Linienvorifikation ein.

10.4 Haltestellenverifikation

Die Haltestellenverifikation erfolgt wie im Kapitel [Punktverifikation](#) beschrieben. Wird die Haltestelle nicht als Volltreffer übergeben, muss sichergestellt werden, dass am Ende des Punktverifikationsprozesses eine Haltestelle gewählt wurde. Bei der Suche nach wichtigen Punkten oder Adressen, erfolgt dies beispielsweise wie im Abschnitt [Wahl einer zugeordneten Haltestelle](#) beschrieben. Findet eine Haltestellensuche statt, ist folgender zusätzlicher Parameter notwendig:

mode = odv

Dieser Parameter leitet die Haltestellenverifikation ein.

10.5 Linienauswahl

Ist die Haltestelle verifiziert, werden eine bedienende Linien gewählt, für die die Fahrplanbuchseite als PDF generiert wird. Die Wahl der Linie erfolgt wie im Kapitel [Eingabe bekannter Linien](#) beschrieben. Zusätzlich ist folgender Parameter notwendig:

command = direct

Dieser Parameter ist für die Wahl einer oder mehrerer Linien notwendig.

11 ROP-Request

Im Folgenden wird die Ansteuerung der Anfrage eines Stadtplanausschnittes oder eines Linienverlaufsplans (ROP-Request) erläutert. Ein Stadtplanausschnitt wird für die Umgebung von Haltestellen, Adressen, wichtigen Punkten oder Koordinaten erstellt. Diese können mittels einer Punktverifikation ermittelt (siehe Kapitel [Generierung eines Stadtplanausschnitts mittels Punktverifikation](#)) oder direkt eingegeben werden (siehe Kapitel [Direkte Generierung eines Stadtplanausschnitts](#)).

Der Linienverlaufsplan zeigt den kompletten Verlauf einer Linie auf einer Karte. Die Linie kann auf zwei Arten ausgewählt werden:

1. Linienverifikation (siehe [Generierung des Linienverlaufsplans mittels Linienverifikation](#))
2. Haltestellenverifikation und Linienauswahl (siehe [Generierung des Linienverlaufsplans mittels Haltestellenverifikation und Linienauswahl](#))

11.1 Ansteuerung der Anfrage von Stadtplanausschnitten/Linienverlaufsplänen

Die Anfrage des Stadtplanausschnitts bzw. Linienverlaufsplänen wird über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (`XML_ROP_REQUEST`) oder in HTML (`XSLT_ROP_REQUEST`) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_ROP_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_ROP_REQUEST?HTTP-Parameter
```

Die Stadtplanausschnitte und Linienverlaufspläne werden als PDF-Datei generiert. Der Link zu dieser Datei befindet sich im Attribut `src` des Elements `itdDownloadFile` im Container `itdDownload` in der XML Antwort.

11.2 Allgemeine Funktionalität

Es können die in den Kapiteln [Grundfunktionalität](#) sowie [Punktverifikation](#) und [Linienverifikation](#) beschriebenen HTTP-Parameter verwendet werden. Bei der Punktverifikation wird das Parameter-Suffix `rop` verwendet.

11.3 Obligatorische Parameter für den Stadtplanausschnitt

reqType = spa

Aktiviert die Generierung eines Stadtplanausschnitts.

11.4 Generierung eines Stadtplanausschnitts mittels Punktverifikation

Die Punktverifikation erfolgt mit den im Kapitel [Punktverifikation](#) beschriebenen Parametern. Zusätzlich ist folgender Parameter notwendig:

mode = odv

Aktiviert die Punktverifikation für den Stadtplanausschnitt.

11.5 Direkte Generierung eines Stadtplanausschnitts

Wenn der Punktverifikationsprozess übersprungen wird, muss sichergestellt werden, dass es sich bei dem mit den folgenden Parametern übergebenen Punkt um einen Volltreffer handelt. Für die direkte Generierung ist folgender Parameter obligatorisch:

command = direct

Dieser Parameter ermöglicht die direkte Anfrage eines Stadtplanausschnitts ohne vorher den Punktverifikationsprozess zu durchlaufen.

Der Volltreffer kann eine Haltestelle, ein wichtiger Punkt oder eine Koordinate sein:

Haltestelle

Eine Haltestelle kann über ihre ID eindeutig identifiziert werden. Die ID wird als Wert des Parameters `stopID` übergeben.

Wichtiger Punkt

Um einen wichtigen Punkt als eindeutig identifizierbaren Treffer zu übergeben sind zwei Parameter notwendig. Der Parameter `omc` hat die Gemeindekennziffer (GKZ bzw. OMC) als Wert. Der Parameter `poiID` die ID des wichtigen Punktes.

Koordinate

Eine Koordinate wird über x, y und das Koordinatensystem eindeutig identifiziert. Die Parameter zum Übergeben der Werte lauten `xcoord`, `ycoord` und `namecoord`.

11.6 Obligatorische Parameter für den Linienverlaufsplan

reqType = lvp

Aktiviert die Generierung eines Linienverlaufsplans.

11.7 Generierung des Linienverlaufsplans mittels Linienverifikation

Die Linienverifikation erfolgt mit den im Kapitel [Linienverifikation](#) beschriebenen Parametern. Zusätzlich muss diese wie folgt aktiviert werden.

mode = line

Aktiviert die Linienverifikation

11.8 Generierung des Linienverlaufsplans mittels Haltestellenverifikation und Linienauswahl

Die Haltestellensuche erfolgt mit den im Kapitel [Punktverifikation](#) beschriebenen Parametern. Es muss sichergestellt sein, dass es sich bei dem verifizierten Punkt um eine Haltestelle handelt. Gegebenenfalls muss eine zugeordnete Haltestelle gewählt werden (siehe Kapitel [Wahl einer zugeordneten Haltestelle](#)).

Im Anschluss an die Punktverifikation kann eine bedienende Linie gewählt werden. Dazu werden die im Kapitel [Eingabe bekannter Linien](#) beschriebenen Parameter verwendet.

mode = odv

Aktiviert die Punktverifikation.

11.9 Direkte Generierung eines Linienverlaufsplans

Der Linienverlaufsplan kann für eine Linie direkt angefragt werden. Dazu sind folgende Parameter notwendig:

command = direct

Dieser Parameter ermöglicht die direkte Anfrage des Linienvverlaufsplans über die Angabe einer Linie.

11.10 Obligatorische Parameter für den Linienvverlaufsplan

mergeSup = 1

Es werden alle Präsentationen außer der Basislinie verworfen. Es ist nicht möglich verschiedene Präsentationen in einem Linienvverlaufsplan darzustellen.

subReqType = overview

Mit diesem Parameter kann die Ausgabe dahingehend übersteuert werden, dass die Übersichtskarte ohne einzelne Detailkarten des Linienvverlaufs ausgegeben wird. Standardmäßig werden die Detailkarten ausgegeben.

11.11 Linienvverlaufsplan mit Stadtplanausschnitten für die durchfahrenden Haltestellen

printOutsPerStop

Dieser Parameter kann die Werte 1 oder 2 haben. Damit werden für jede durchfahrende Haltestelle zusätzlich zum Linienvverlaufsplan ein bzw. zwei Stadtplanausschnitte generiert.

spaOut

Beeinflusst die Reihenfolge der Ausgabe der Stadtplanausschnitte in der PDF-Datei. Durch den Wert `bothdirections` werden die Stadtplanausschnitte für beide Richtungen ausgegeben, sofern der Linienvverlauf unterschiedlich ist. Der Wert `mainroutefirst` bewirkt, dass die Stadtplanausschnitte der Hauptroute zuerst ausgegeben werden, anschließend die der Nebenroute.



12 AddInfo-Request

Mit dem AddInfo-Request werden aktuelle Meldungen, die über das ICS (Incident Capturing System) eingepflegt wurden, angefragt. Mittels HTTP-Parametern können die Meldungen nach verschiedenen Kriterien gefiltert werden. Werden keine Filterparameter angegeben, erfolgt die Ausgabe aller Meldungen. Dabei ist jedoch zu beachten, dass die Antwortzeiten je nach Menge der eingepflegten Meldungen extrem lang sein können. Daher empfiehlt es sich nicht die Meldungen vollständig auszugeben, sondern beispielsweise nur die für das heutige Datum aktiven.

12.1 Ansteuerung der Anfrage von aktuellen Meldungen

Der AddInfo-Request wird über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (`XML_ADDINFO_REQUEST`) oder in HTML (`XSLT_ADDINFO_REQUEST`) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_ADDINFO_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_ADDINFO_REQUEST?HTTP-Parameter
```

12.2 Datums-Filter

filterDateValid

Filtert nach Meldungen, die für einen bestimmten Tag gültig sind. Das Datum wird über die zweistellige Angabe des Tages, gefolgt von der zweistelligen Angabe des Monats und der vierstelligen Angabe des Jahres, angegeben. Als Separator dient das Minuszeichen. Der Parameter kann mehrfach verwendet werden.

filterDateValidComponentsActive = 1

Dieser Parameter ermöglicht eine Alternative zu `filterDateValid`. Tag, Monat und Jahr können getrennt über die Parameter `filterDateValidDay`, `filterDateValidMonth` und `filterDateValidYear` eingegeben werden.

filterDateValidDay

Dieser Parameter ist Bestandteil des komponentenweisen Datumsfilters, der durch den Parameter `filterDateValidComponentsActive=1` aktiviert wird. Mit ihm gibt man den Tag (zweistellig) an, für den die Meldungen angezeigt werden sollen.

filterDateValidMonth

Dieser Parameter ist Bestandteil des komponentenweisen Datumsfilters, der durch den Parameter `filterDateValidComponentsActive=1` aktiviert wird. Mit ihm gibt man den Monat (zweistellig) an, für den die Meldungen angezeigt werden sollen.

filterDateValidYear

Dieser Parameter ist Bestandteil des komponentenweisen Datumsfilters, der durch den Parameter `filterDateValidComponentsActive=1` aktiviert wird. Mit ihm gibt man das Jahr (vierstellig) an, für das die Meldungen angezeigt werden sollen.

filterPublicationStatus

Ausschließlich aktuelle Meldungen (`current`) oder abgelaufene Meldungen (`history`) anzeigen.

filterValidIntervalStart und **filterValidIntervalEnd**

Diese Parameter geben ein Datumsintervall an. Es werden ausschließlich Meldungen ausgegeben, die innerhalb des angegebenen Intervalls liegen. Das Start- bzw. End-Datum wird über die zweistellige Angabe des Tages, gefolgt von der zweistelligen Angabe des Monats und der vierstelligen Angabe des Jahres, angegeben. Als Separator dient das Minuszeichen.

12.3 Orts-Filter

filterOMC

Filtert nach Meldungen, die bestimmte Orte betreffen. Der Wert des Filters ist eine Liste von Gemeindekennziffern (GKZ bzw. OMC), die durch einen Doppelpunkt voneinander getrennt werden. Auf diese Weise können beispielsweise Landkreise modelliert werden.

filterOMC_PlacelD

Filtert nach Meldungen, die einen bestimmten Ort betreffen. Der Ort wird durch die Gemeindegrenznummer und eine ID des Ortes bestimmt. Die beiden Werte werden durch einen Doppelpunkt getrennt angegeben. Der Parameter kann mehrfach verwendet werden.

12.4 Linien-, Haltestellen-, Betreiber- und Verkehrsmittel-Filter

filterLineNumberIntervalStart und filterLineNumberIntervalEnd

Angabe eines Liniennummer-Intervalls. Es wird nach Meldungen gefiltert, die Linien betreffen, deren Liniennummern (entspricht den letzten 3 Stellen der DIV-Linie) in dem Intervall liegen. Der Wert der Parameter muss größer als Null sein und kann maximal 999 sein, außerdem muss der Wert von `filterLineNumberIntervalEnd` größer sein als der von `filterLineNumberIntervalStart`.

filterMOTType

Filtert nach Meldungen, die ein bestimmtes Verkehrsmittel betreffen. Zur Wahl mehrerer Verkehrsmittel, wird der Parameter mehrfach verwendet. Als Wert des Parameters wird die Verkehrsmittel-ID angegeben. Die Belegung der Verkehrsmittel-IDs von APB Bozen kann dem Kapitel [Verkehrsmittel](#) entnommen werden.

filterPNLineDir

Filtert nach Meldungen, die Linien betreffen, die den Kriterien „Teilnetz“, „DIVA Linie“ und „Richtung“ entsprechen. Die Kriterien werden durch Doppelpunkt getrennt angegeben. Der Parameter kann mehrfach verwendet werden.

filterPNLineSub

Filtert nach Meldungen, die Linien betreffen, die den Kriterien „Teilnetz“, „DIVA Linie“ und „Ergänzung“ entsprechen. Die Kriterien werden durch Doppelpunkt getrennt angegeben. Der Parameter kann mehrfach verwendet werden.

itdLPxx_selLine

Filtert nach Meldungen, die eine bestimmte DIVA-Linie betreffen. Als Wert des Parameters wird die DIVA-Linien-Nummer übergeben. Zur Wahl mehrerer DIVA-Linien, wird der Parameter mehrfach verwendet.

itdLPxx_selOperator

Filtert nach Meldungen, die die Linien eines bestimmten Betreibers betreffen. Als Wert des Parameters wird die Betreiber-Kennung übergeben. Zur Wahl mehrerer Betreiber, wird der Parameter mehrfach verwendet.

itdLPxx_selStop

Filtert nach Meldungen, die eine bestimmte Haltestelle betreffen. Als Wert wird die Haltestellen-ID übergeben. Zur Wahl mehrerer Haltestellen, wird der Parameter mehrfach verwendet.

line

Dieser Parameter wählt eine Linie aus. Er kann mehrfach übergeben werden. Sein Wert setzt sich zusammen aus dem Teilnetz, der DIVA Liniennummer, der Ergänzung, der Richtung und dem Fahrplanprojekt, getrennt durch Doppelpunkte. Für die Ergänzung gilt zu beachten, dass der Unterstrich "_" durch ein Leerzeichen " " ersetzt werden muss.

Mit dem Parameter `passedStops=1` können dann Meldungen zurückgegeben werden, die die durchfahrenen Haltestellen der ausgewählten Linien betreffen.

12.5 Filtern nach Meldungstypen und IDs

filterInfoID

Mit dem Parameter kann eine bestimmte Meldung anhand ihrer ID gesucht werden. Die ID wird als Wert des Parameters übergeben. Der Parameter kann zur Suche mehrerer Meldungen mehrfach verwendet werden.

filterInfoType

Filtert nach Meldungen eines bestimmten Typs. Für den ÖV sind folgende Meldungstypen definiert:

- stopInfo
- stopBlocking
- lineInfo
- lineBlocking
- routeInfo
- routeBlocking

Des Weiteren gibt es die Typen:

- `generalInfo`
- `bannerInfo`

Der Parameter kann mehrfach verwendet werden.

12.6 Filter nach Betreiber und Quelle

filterProviderCode

Filtert nach Meldungen, die von einem bestimmten Autor oder Anbieter eingegeben oder zur Verfügung gestellt wurden. Zur Wahl mehrerer Betreiber, wird der Parameter mehrfach verwendet. Als Wert wird der Code des Autors bzw. Anbieters übergeben.

filterSourceSystemName

Filtern nach Meldungen, die von dem als Wert übergebenen Quellsystem eingegeben wurden. Der Parameter kann mehrfach verwendet werden.

12.7 Reduktion

AIXMLReduction

Dieser Parameter kann mehrfach verwendet werden und reduziert die XML-Ausgabe um verschiedene Elemente. Folgende Werte können übergeben werden und entfernen die durch → angegebenen Elemente:

- | | |
|---|-------------------------------------|
| • <code>removeCreationTime</code> | → <code>creationTime</code> |
| • <code>removeExpirationTime</code> | → <code>expirationDateTime</code> |
| • <code>removeGenericParams</code> | → <code>genericParams</code> |
| • <code>removeInfoLink</code> | → <code>infoLink</code> |
| • <code>removeITIncident</code> | → <code>itIncident</code> |
| • <code>removeLastModificationTime</code> | → <code>lastModificationTime</code> |
| • <code>removeLines</code> | → <code>concernedLines</code> |
| • <code>removePublication</code> | → <code>publicationDuration</code> |
| • <code>removeSourceSystem</code> | → <code>sourceSystem</code> |
| • <code>removeStops</code> | → <code>concernedStops</code> |
| • <code>removeValidity</code> | → <code>validityPeriod</code> |

filterXMLOutput = false

Entfernt des kompletten Inhalts des Elements `itdAdddInfoRequest` aus der XML-Ausgabe.

13 CoordInfo-Request

Im Folgenden wird die Ansteuerung der Koordinatenanfrage (CoordInfo-Request) erläutert. Diese dient dazu Koordinaten verschiedener Objekte, wie Haltestellen oder wichtiger Punkte, anzufragen. Diese Punkte können anschließend, beispielsweise auf einer interaktiven Karte, dargestellt werden. Die Auswahl der Punkt-Typen und die Einschränkung des Suchraums erfolgt über die weiter unten beschriebenen http-Parameter. Einschränkungen des Suchraums werden durch eine Filterung nach verschiedenen Kriterien vorgenommen. Das kann beispielsweise die Entfernung zum Zentrum oder eine bestimmte Zeichenklasse sein.

13.1 Ansteuerung der Koordinatenanfrage

Die Koordinatenanfrage wird über HTTP-Parameter angesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (XML_COORD_REQUEST) oder in HTML (XSLT_COORD_REQUEST) ausgegeben. Um eine Antwort in HTML generieren zu können, ist die Transformation (z.B. eine XSL-Transformation) der XML-Struktur in HTML notwendig.

Für eine XML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XML_COORD_REQUEST?HTT  
P-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_COORD_REQUEST?HT  
TP-Parameter
```

13.2 Anfrage-Parameter

coord

Mit diesem Parameter wird die Zentrumskoordinate angegeben, die den Mittelpunkt der Suche nach den Punkten darstellt. Die Koordinate setzt sich zusammen aus der x-Koordinate, der y-Koordinate und dem Koordinatensystem (dies entspricht dem Attribut `mapName`), jeweils getrennt durch einen Doppelpunkt.

Hinweis: Der Suchradius `radius_<Filterindex>` wird für jeden Punkt-Typ `type_<Filterindex>` separat angegeben.

deadline

Mit diesem Parameter kann ein Datum angegeben werden, dass die zu diesem Zeitpunkt gültigen Haltestellen ermittelt. Das Datum wird als vierstellige Jahreszahl, gefolgt von der zweistelligen Monatszahl und der zweistelligen Jahreszahl angegeben. Wird dieser Parameter nicht verwendet, werden die Haltestellen, die zu der aktuellen Server-Zeit gültig sind, ermittelt.

mapNameOutput

Mit diesem Parameter wird das Koordinatensystem (dies entspricht dem Attribut `mapName`) des GIS Netzes angegeben. Der Parameter ist insbesondere dann sinnvoll, wenn das Koordinatensystem der Karte nicht mit dem Koordinatensystem, in dem die Daten vorliegen (GIS Netz), übereinstimmt. Entsprechen sich das Koordinatensystem der Karten und der Daten, kann der Parameter weggelassen werden. Aus Performancegründen ist es jedoch sinnvoll ihn immer zu übergeben, da so die Ermittlung des Koordinatensystems durch den EFA IT-Kernel wegfallen kann.

Beispiel: NAV3

max

Mit diesem Parameter wird die maximale Anzahl an Elementen angegeben, die mittels der Koordinatenanfrage ermittelt und ausgegeben werden sollen. Standardmäßig liegt keine Beschränkung der Anzahl vor.

13.3 Filter zur Auswahl der Punkt-Typen und zur Einschränkung des Suchraums

Für die Koordinatenanfrage stehen erweiterte Filter zur Verfügung. Um diese verwenden zu können, ist ihre Aktivierung durch den Parameter `inclFilter=1` notwendig. Die Filter zeichnen sich durch ein Suffix `_<Filterindex>` aus. Auf diese Weise können verschiedene Filter miteinander kombiniert werden und auf einen Punkt-Typ (siehe `type_<Filterindex>`) angewendet werden. Kombinierte Filter haben das gleiche Suffix.

Beispiel: Durch die Angabe von Filtern ist es möglich nur ausgewählte Punkt-Typen, wie Haltestellen, anzufragen und diese gegebenenfalls durch ihre Zeichenklasse einzuschränken. Der Typ "Haltestelle" wird durch den Parameter `type_<Filterindex>=STOP` gewählt; die Zeichenklasse durch den Parameter `inclDrawClasses_<Filterindex>`. Die beiden Suffixe müssen identisch sein.

inclFilter = 1

Aktivierung der Filter zur Einschränkung des Suchraums bei der Ermittlung von Objekten. Auf diese Weise können Objekt nach bestimmten Kriterien gefiltert werden.

cluster_<Filterindex> = 1

Dieser Parameter aktiviert das Zusammenfassen der Objekte. Auf diese Weise ist eine sinnvolle Darstellung von Haltestellen oder wichtigen Punkten auch auf Übersichtskarten möglich, ohne dass diese überfrachtet werden.

Die Größe Fläche, für die die Objekte zusammengefasst werden, wird über den Parameter `radius_<Filterindex>` gesteuert. Es werden die Objekte innerhalb von Quadraten mit der doppelten Länge und Breite des Radius zusammengefasst.

In der XML-Ausgabe wird ein Attribut `clusterNum` innerhalb der Elemente `coordInfoItem` in der `coordInfoItemList` ausgegeben. Dies bezeichnet wie viele Objekte des durch das Attribut `typ` bestimmten Typs (Haltestelle oder POI) zusammengefasst wurden. Die gemittelte Koordinate, zur Darstellung eines Symbol für den Cluster auf einer Karte genutzt werden kann, wird im Element `coordInfoItem` ausgegeben.

```
coordInfoItemList>
  coordInfoItem type="STOP" clusterNum="1359" id="" name="" addName="" omc="0" placeID="0" locality="" gisLayer="" gisID="0" distance="7071">
    itdPathCoordinates>
      coordEllipsoid=UNKNOWN ELLIPSOID
      coordType=PROJECTION_MDV
      itdCoordinateString decimal="." cs=" " ts="5mx20:">4884050.00000,3751572.00000
    </itdPathCoordinates>
  </coordInfoItem>
+ coordInfoItem type="STOP" clusterNum="891" id="" name="" addName="" omc="0" placeID="0" locality="" gisLayer="" gisID="0" distance="7071">
+ coordInfoItem type="STOP" clusterNum="881" id="" name="" addName="" omc="0" placeID="0" locality="" gisLayer="" gisID="0" distance="7071">
```



In der JSON-Ausgabe wird pro Cluster ein Objekt ausgegeben. Dies enthält ebenfalls ein Element `clusterNum`, sowie den Objekttyp `typ`. Die Koordinate zur Darstellung des Symbols wird als `coords` ausgegeben.

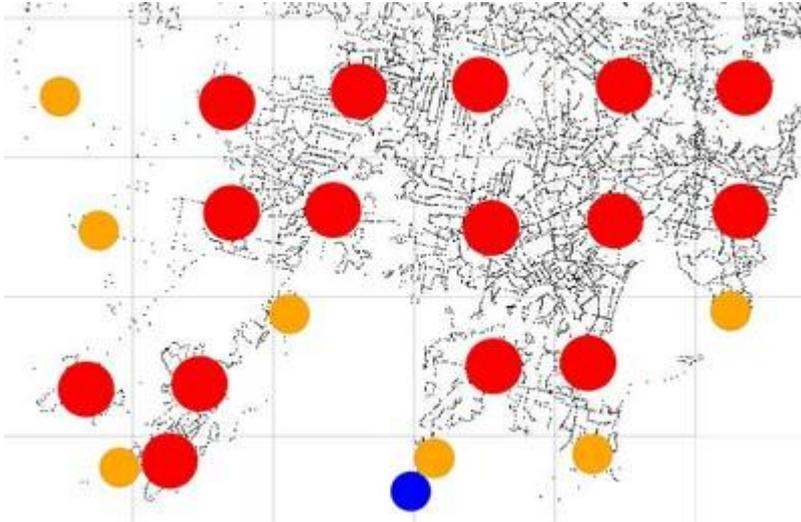
```
gisID: "0",
distance: "7071",
coords: "4884050,3751572",
attrs: [ ]
},
- {
  desc: "",
  addDesc: "",
  type: "STOP",
  clusterNum: "891",
  id: "",
  omc: "0",
  placeID: "0",
  locality: "",
  layer: "",
  gisID: "0",
  distance: "7071",
  coords: "4892324,3760606",
  attrs: [ ]
},
- {
  desc: "",
  addDesc: "",
  type: "STOP",
  clusterNum: "881",
```



Beispiel 1: radius_1=50000 bewirkt eine Clustergröße von 100 km x 100 km. Beispielanfrage:

```
http://<server>:<port>/<virtuelles
Verzeichnis>/XML_COORD_REQUEST?&coord=4889145:3756744:GDAV&inclFilter=1&coordListOutputFormat=STRING&type_1=STOP&radius_1=50000&cluster_1=1
```

Beispiel 2: Die Anzahl der Elemente pro Cluster kann auf der Karte beispielsweise durch die Größe der Symbole dargestellt werden.



exclLayers_<Filterindex>

Mit diesem Parameter kann man GIS-Layer angeben, der nicht berücksichtigt werden soll. Dadurch reduziert sich die Suchzeit. Mehrere GIS-Layer können durch einen Doppelpunkt getrennt angegeben werden.

inclDrawClasses_<Filterindex>

Mit diesem Parameter kann eine Zeichenklasse mittels ihrer ID angegeben werden, die die gesuchten Punkte haben müssen. Als Wert wird der numerische Code der Zeichenklasse (siehe Handbuch zur Karten-Konfiguration) übergeben. Mehrere Zeichenklassen können durch Doppelpunkt getrennt angegeben werden.

inclPOIH_<Filterindex>

Mit diesem Parameter können eine oder mehrere POI Hierarchien angegeben werden, die die gesuchten wichtigen Punkte (`type_<Filterindex>=POI_AREA` oder `type_<Filterindex>=POI_POINT`) haben müssen. Sollen mehrere Hierarchien angegeben, werden diese durch einen Doppelpunkt getrennt. Es werden auch die Punkte einer durch diesen Parameter angegebenen Hierarchie angefordert, wenn die Ausgabe der entsprechenden Hierarchie über die Konfigurationsdatei des EFA IT-Kernels unterdrückt wird.

Beispiel: `inclPOIH_1=A:B:D`

radius_<Filterindex>

Mit diesem Parameter wird der Radius für die Umkreissuche in Metern angegeben. Den Mittelpunkt der Suche stellt die Zentrumsordinate `coord` dar.

type_<Filterindex>

Mit diesem Parameter kann ein bestimmter Punkt-Typ ausgewählt werden. Mehrere Punkt-Typen können durch die mehrfache Verwendung dieses Parameters angefragt werden. Dabei wird ein anderer Index <Filterindex> vergeben. Folgende Typen sind möglich:

Punkt-Typ	Beschreibung
ANY	Alle Punkte
BUS_POINT	Steige
ENTRANCE	Eingänge (zum Beispiel von U-Bahn-Haltestellen)
GIS_AREA	GIS-Gebiet (kann je nach GIS-Daten variieren)
GIS_POINT	GIS-Punkt (kann je nach GIS-Daten variieren)
LINE	Linien, die über das Straßensegment der übergebenen Koordinate <code>coord</code> fahren
POI_AREA	Flächen-POIs (wichtige Flächen-Punkte)
POI_POINT	Punkt-POIs (wichtige Punkte)
STOP	Haltestellen
STREET	Straßen

Sollen weitere Filter auf einen Element-Typen angewendet werden, so haben diese den gleichen Suffix `_<Filterindex>`.

14 Anhang

14.1 Verkehrsmittel

Belegung der Verkehrsmittel-IDs von APB Bozen:

ID	Verkehrsmittel
0	Zug
5	Stadtbus
6	Regionalbus (zukünftig vorgesehen)
7	Nightliner
8	Seilbahn
10	AST/Rufbus (zukünftig vorgesehen)