



EFA XML-Schnittstelle des VRR

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
1.0		05.11.2014	SM	Initiale Version
1.1		11.02.2015	SM	Korrekturen
1.2		20.04.2017	SM	Parameter SpEncId entfernt, Parameter UTFMacro ergänzt

Freigabe			
	Datum	Name	Unterschrift
geprüft:			
freigegeben:			

Inhaltsverzeichnis

1	Allgemeines	13
2	Grundfunktionalität	14
2.1	Session-Handling	14
	requestID	14
	sessionID	14
2.2	Sprache	14
	language	14
2.3	Ausgabeformat	15
	outputFormat	15
2.4	Generische Parameter	15
	itdLPxx_<Name>	15
2.5	Koordinaten	15
	coordListOutputFormat	16
	coordOutputFormat	16
	coordOutputFormatTail	16
2.6	Sonstige	17
	execInst	17
	SpEnclId = 5	17
3	Datum und Uhrzeit	18
3.1	Datum	18
	itdDate	18
	itdDateDay	18
	itdDateMonth	18
	itdDateYear	18
	itdDateYearMonth	18
	itdDateDayMonthYear	19

3.2	Uhrzeit.....	19
	itdTime.....	19
	itdTimeHour.....	19
	itdTimeMinute.....	19
	itdTimeAMPM.....	19
	timeOffset.....	19
4	Punktverifikation	21
4.1	Obligatorische Parameter.....	21
	locationServerActive = 1.....	21
	name_<usage>	21
	stateless = 1	22
	type_<usage>.....	22
4.2	Hilfetext für Eingabefelder	23
	nameDefaultText_<usage>	23
4.3	Verschiedene Eingabemöglichkeiten ermöglichen	23
	nameInfo_<usage>	23
	typeInfo_<usage>.....	23
4.4	Optionale Parameter zur Optimierung der Punktsuche	23
	anyMaxSizeHitList.....	23
	anyResSort_<usage>.....	24
	anySigWhenPerfectNoOtherMatches = 1.....	24
	locationInfoActive = 1	24
	useHouseNumberList_<usage> = 1	24
	useLocalityMainStop = 1	24
	deleteAssignedStops_<usage>=1	24
	doNotSearchForStops_<usage>=1	25
	prMinQu.....	25
4.5	Suchkriterien	25

anyObjFilter_<usage>	25
anyOmc_<usage>	25
regionID_<usage>	26
tryFurtherAnyWithoutLoc = 1	26
4.6 Beeinflussung der Trefferqualität und Reduktion der Trefferliste	26
coord	27
objPref	27
prefStModes	27
regPref.....	28
rellnf	28
rMethod	28
4.7 Listenauswahl	31
4.8 Wahl einer zugeordneten Haltestelle.....	31
4.9 Eingabe von Haltestellen POIs und Koordinaten	31
Haltestellen.....	32
POI	32
Koordinaten	32
4.10 Ausgabeformat des EFALocationServers	33
convertAddressesITKernel2LocationServer	33
convertCoord2LocationServer	33
convertCrossingsITKernel2LocationServer	33
convertPOIsITKernel2LocationServer	33
convertStopsPTKernel2LocationServer.....	33
5 Linienverifikation.....	34
5.1 Eingabe bekannter Linien.....	34
line.....	34
IsShowTrainsExplicit = 1	35
lineVer	35

dateDay	35
6 StopFinder-Request	36
6.1 Ansteuerung des StopFinder-Requests	36
6.2 Hinweis zur Punktverifikation	37
6.3 Allgemeine Funktionalität	37
6.4 Verifizierten Punkt in einen anderen Request übernehmen	37
7 Trip-Request	38
7.1 Ansteuerung der Fahrtanfrage	38
7.2 Hinweis zur Punktverifikation	38
7.3 Optionaler Zwischenhalt.....	38
dwellTime_<usage>	38
7.4 Allgemeine Funktionalität	39
itdTripDateTimeDepArr.....	39
7.5 Allgemeine Optionen	39
calcNumberOfTrips.....	39
calcOneDirection = 1	39
useRealtime = 1	39
7.6 Verkehrsmittelausschluss und Verkehrsmiteinschluss	40
excludedMeans	40
exclMOT_<ID> = 1	41
includedMeans	41
inclMOT_<ID> = 1	41
7.7 Mobilitätseinschränkungen.....	41
impairedOptionsActive=1	42
lowPlatformVhcl.....	42
noElevators	42
noEscalators.....	42
noSolidStairs	42

wheelchair	42
7.8 Weitere Optionen für den öffentlichen Verkehr	43
ptOptionsActive = 1	43
changeSpeed	43
lineRestriction	44
maxChanges	44
routeType	44
useProxFootSearch = 1	44
7.9 Optionen für den Individualtransport	45
itOptionsActive = 1	45
changeSpeed	45
trITMOT	45
trITMOTvalue	45
7.10 Folgeanfragen	46
command.....	46
8 PS-Request	47
8.1 Ansteuerung der Pendlerfahrplans.....	47
8.2 Hinweis zur Punktverifikation	47
8.3 Allgemeine Funktionalität	48
itdDateTimeDepArr.....	48
psParamOneWay = 1	48
useAltOdv = 1	48
8.4 Datum und Uhrzeit für den Pendlerfahrplan	48
itdTime_<usagePS>	49
itdTimeAMPM_<usagePS>	49
itdTimeHour_<usagePS>	49
itdTimeMinute_<usagePS>	49
psParamGroupID	49

psParamMaxTimeHours	50
psParamSampleDate	50
psParamWeekday	50
useAllSampleDates = 1	51
8.5 Zusatzinformation in der PDF-Datei	51
psParamAddressExt1	51
psParamAddressExt2	51
psParamAddressName	51
psParamAddressPlace	51
psParamAddressStreet	51
9 DM-Request	52
9.1 Ansteuerung der Anfrage eines Abfahrtsmonitors	54
9.2 Hinweis zur Punktverifikation	54
9.3 Allgemeine Funktionalität	54
itdDateTimeDepArr	54
limit	54
useRealtime = 1	55
9.4 Haltestellenauswahl	55
9.5 Linienauswahl	56
mode = direct	56
9.6 Folgeanfragen	56
10 DMTTP-Request	58
10.1 Ansteuerung der Anfrage eines Abfahrtsmonitors	58
10.2 Hinweis zur Punktverifikation	58
10.3 Allgemeine Funktionalität	58
11 StopTimetable-Request	59
11.1 Ansteuerung des Aushangfahrplans	59
11.2 Hinweis zur Punktverifikation	59

11.3	Allgemeine Funktionalität	59
	mrgSt = 1	60
11.4	Haltestellenauswahl	60
11.5	Linienauswahl	60
	mode = direct.....	60
12	StopList-Request	61
12.1	Ansteuerung des StopList-Requests	61
12.2	Parameter zur Filterung.....	61
	stopListOMC.....	61
	stopListPlaceId	61
	stopListSubnetwork	61
	fromstop	62
	tostop.....	62
12.3	Parameter zur Ausgabe zusätzlicher Information	62
	servingLines = 1	62
	servingLinesMOTType = 1	62
	servingLinesMOTTypes = 1	62
	tariffZones = 1	62
13	AddInfo-Request	64
13.1	Ansteuerung der Anfrage von aktuellen Meldungen.....	64
13.2	Datums-Filter.....	64
	filterDateValid	64
	filterDateValidComponentsActive = 1	64
	filterDateValidDay.....	65
	filterDateValidMonth	65
	filterDateValidYear	65
	filterPublicationStatus	65
	filterValidIntervalStart und filterValidIntervalEnd.....	65

13.3 Orts-Filter	65
filterOMC	65
filterOMC_PlacelD.....	66
13.4 Linien-, Haltestellen-, Betreiber- und Verkehrsmittel-Filter	66
filterLineNumberIntervalStart und filterLineNumberIntervalEnd.....	66
filterMOTType.....	66
filterPNLineDir	66
filterPNLineSub	66
itdLPxx_selLine	66
itdLPxx_selOperator	67
itdLPxx_selStop.....	67
line.....	67
13.5 Filtern nach Meldungstypen und IDs	67
filterInfoID	67
filterInfoType.....	67
13.6 Filter nach Betreiber und Quelle.....	68
filterProviderCode.....	68
filterSourceSystemName.....	68
13.7 Reduktion	68
AIXMLReduction	68
filterXMLOutput = false	69
14 SystemInfo-Request	70
14.1 Ansteuerung der Anfrage von Systeminformation	70
14.2 Allgemeine Funktionalität	70
15 Map-Request	71
15.1 Ansteuerung der Anfrage von Systeminformation	71
15.2 Allgemeine Funktionalität	72
command.....	72

createImage = 0	72
mapName	72
mapNameOutput	72
moveDirection	73
omc.....	73
placeID	73
placeSelectionWithCoordinate	73
realHeight	74
realWidth	74
screenSize.....	74
screenHeight	75
screenWidth	75
xCenterReal	75
xScreen	76
xScreen2	76
yCenterReal	76
yScreen	76
yScreen2	76
zoomLevel	76
15.3 Kachel-Optionen.....	77
desiredRect	77
heightPixelsMax	77
heightPixelsMin	77
imageFormat	77
imageNoTiles = false.....	78
imageOnly	78
itDestLegPoint/itStartLegPoint.....	78
itDestLegStop/itStartLegStop	78

itDestMOT/itStartMOT	78
itDestPath = 1/itStartPath = 1	79
levelOfDetail	79
ptTripLeg_<ID> = 1	79
ptTripLeg_<ID>_MOT	80
ptTripLeg_<ID>_Stop	80
ptTripPart = 1	80
purposeSuffix	80
serverCopyright2Hint = 1	81
tripPartFocusIndex	81
widthPixelsMax	81
widthPixelsMin	81
16 Anhang	82
16.1 Verkehrsmitteltypen	82
16.2 Objekttypen	83
17 Glossar	84
17.1 HTTP-Parameter Makros	84

1 Allgemeines

Dieses Dokument beschreibt die HTTP-Parameter zur Nutzung der EFA XML-Schnittstelle für folgende Requests:

- [StopFinder-Request](#): Punkt- bzw. Haltestellen-Suche
- [Trip-Request](#): Anfrage von Fahrtauskünften
- [PS-Request](#): Anfrage eines Pendlerfahrplans
- [DM-Request](#): Anfrage eines Abfahrtsmonitors
- [DMTTP-Request](#): Anfrage des Soll-Abfahrtsmonitors
- [StopTimetable-Request](#): Anfrage eines Aushangfahrplans für eine Linie
- [StopList-Request](#): Anfrage aller Haltestellen eines Verbundsgebiets
- [SystemInfo-Request](#): Anfrage von Systeminformation
- [AddInfo-Request](#): Anfrage aktueller Meldungen
- [Map-Request](#): Anfrage von Karten

Die oben genannten Request nutzen die allgemeine Funktionalität der EFA, die im Kapitel [Grundfunktionalität](#) beschrieben ist. Einige der Requests nutzen gemeinsame Funktionalität zur [Punktverifikation](#) und der Eingabe von [Datum und Uhrzeit](#).

2 Grundfunktionalität

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

2.1 Session-Handling

Die EFA kann zustandslos oder zustandsbehaftet angefragt werden. Die zustandsbehaftete Anfrage ist nur notwendig, wenn Folgeanfragen verwendet werden. Diese sind in den Beschreibungen der jeweiligen Requests gesondert aufgelistet. Zum Beispiel für den Trip-Request im Abschnitt [Folgeanfragen](#).

Für die zustandsbehaftete Anfrage werden folgende Parameter verwendet:

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. Für Folgeanfragen muss die Request ID übergeben werden, sofern sie nicht 1 ist.

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.

2.2 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`, Italienisch `it` und Französisch `fr`.

Hinweis: Wird dieser Parameter nicht übergeben, wird die Standardsprache verwendet, sofern eine solche konfiguriert ist.

2.3 Ausgabeformat

outputFormat

Bestimmt das Format der Ausgabe. Mögliche Werte sind XSLT, XML und JSON. Standardmäßig erfolgt die Ausgabe, in Abhängigkeit von der Anfrage, in XML oder XSLT.

Hinweis: Die JSON-Ausgabe ist nicht für alle Requests möglich und bildet auch nicht immer den gleichen Umfang ab, wie die XML-Ausgabe.

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

2.5 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 Ausgabekoordinatensystem 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 Koordinatenformat)
- `MRCV` (MDV-eigenes Koordinatenformat, das auf EPSG 3857 (Spherical Mercator) basiert.)
- `PROJ[+init=epsg:<Code>]` (EPSG-Code)
- `MR_EPSG3857` (Alias für den EPSG-Code 3857, Spherical Mercator)
- `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.

2.6 Sonstige

execInst

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)

UTFMacro = true

Dieses Makro gibt die korrekte Zeichenencodierung an.

3 Datum und Uhrzeit

Die Eingabe von Datum und Uhrzeit wird von vielen verschiedenen Requests verwendet. 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.

Hinweis: Der [PS-Request](#) verwendet eine abweichende Datums- und Uhrzeiteingabe, da hier Stichtage und ein Zeitintervall benötigt werden. Siehe [Datum und Uhrzeit für den Pendlerfahrplan](#).

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 Wert 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-amerikanische 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 zählt zur Grundfunktionalität der EFA und wird von mehreren Requests benötigt. Zum Beispiel vom [StopFinder-Request](#), [Trip-Request](#) und [DM-Request](#). 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 eben genannte sowie 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 innehat. Die Funktion eines Punktes wird durch die Ergänzung `_<usage>` am Ende eines Parameters bestimmt. Parameter, die eine solche Ergänzung benötigen, werden für jeden Punkt separat definiert. Beim [Trip-Request](#) kann die Ergänzung `<usage>` folgende Werte annehmen:

- `origin` – Startpunkt
- `destination` – Zielpunkt
- `via` – Zwischenhalt

Welche Endungen für einen Request gültig sind, kann der Dokumentation des jeweiligen Requests entnommen werden.

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 `value` 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 `stateless` (siehe Parameter `stateless=1`) aus dem XML-Element `odvNameElem`.

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.

Hinweis: Ab EFA10 wird dieses Attribut automatisch ausgegeben und der Parameter somit nicht mehr benötigt.

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 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.

4.3 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.

typeInfo_<usage>

Dieser Parameter überschreibt den Parameter `type_<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.

4.4 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.

Falls die Trefferliste vor der Anwendung der definierten Bewertungskriterien reduziert wird, sollte der Wert nicht zu niedrig gewählt werden.

Hinweis: Ob die Liste vor oder nach der Anwendung der Bewertungskriterien reduziert wird, hängt von der Konfiguration des EFA Location Servers ab. Eine weitere Möglichkeit zur Reduktion der Trefferliste wird unter [Beeinflussung der Trefferqualität und Reduktion der Trefferliste](#) beschrieben.

anyResSort_<usage>

Normalerweise gibt der EFALocationServer die Trefferliste unsortiert zurück. Die Sortierung wird von der Benutzeroberfläche durchgeführt. Das Standardverhalten kann mit diesem Parameter geändert werden. Sein Wert entspricht dem Namen eines Sortierkriterien-Sets `ResultSorter<ID>`, das in der Konfigurationsdatei des EFA Location Servers festgelegt wird.

Hinweis: Die Verlagerung der Sortierung in den EFALocationServer hat den Vorteil, dass sie eine konstante Sortierreihenfolge in allen Benutzeroberflächen garantiert ist.

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.

locationInfoActive = 1

Aktiviert das Ermitteln aller Hausnummern zu einer verifizierten Straße. Die Ermittlung der Hausnummern erfolgt schließlich durch den Parameter `useHouseNumberList_<usage>`.

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.

deleteAssignedStops_<usage>=1

Dieser Parameter unterdrückt die Suche nach Zugeordneten Haltestellen. Die Haltestelle wird direkt verifiziert.

Hinweis: Dieser Parameter ist für den StopFinder-Request sinnvoll, falls dieser beispielsweise zur Anzeige einer Vorschlagsliste im Rahmen der Punktverifikation angeboten wird.

doNotSearchForStops_<usage>=1

Bewirkt, dass nach der Verifizierung einer Koordinate nicht zu umliegenden Haltestellen geroutet wird. Der Parameter sollte eingesetzt werden, wenn ausschließlich Fahrten ohne Öffentliche Verkehrsmittel berechnet werden.

Hinweis: Dieser Parameter ist für den StopFinder-Request sinnvoll, falls dieser beispielsweise zur Anzeige einer Vorschlagsliste im Rahmen der Punktverifikation angeboten wird.

prMinQu

Mit diesem Parameter kann eine Trefferqualität angegeben werden, die ein Treffer mindestens haben muss, um in die Trefferliste aufgenommen zu werden.

4.5 Suchkriterien

Mit den Suchkriterien kann der Raum für die Punktsuche eingeschränkt werden. Um die Anzahl der Parameter zu reduzieren und um Konsistenz über verschiedene Anwenderinterfaces hinweg zu erreichen, wird empfohlen die Suchkriterien in einem HTTP-Parameter Makro zusammen zu fassen. Das Konzept der Makros ist im Kapitel [HTTP-Parameter Makros](#) erläutert.

Die Kriterien für die Punktsuche, das heißt die passende Parameterkombination, sollten vor der Implementation mit dem EFALocationServer Monitor getestet werden. In der Regel dauert es eine Weile bis die ideale Kombination ermittelt ist, da durch die Konzentration auf Sonderfälle schnell die Gesamtübersicht verloren geht. In der Regel ist das Testen mit dem Monitor weniger aufwändig als in der Benutzeroberfläche.

anyObjFilter_<usage>

Die Suche kann auf bestimmte Objekttypen eingegrenzt werden. Der Wert des Parameters ist eine Bitmaske. Die einzelnen Objekttypen lassen sich beliebig kombinieren. Die Objekttypen sind dem Anhang [Objekttypen](#) zu entnehmen.

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

Hinweis: Möglicherweise sind nicht alle Objekttypen in den Daten verfügbar. Postleitzahlen und Kreuzungen sind bei deutschen Verbünden selten vorhanden.

anyOmc_<usage>

Bei der Verifikation werden nur Punkte innerhalb des Gemeindegebietes mit der durch den Parameter übergebenen Kennziffer berücksichtigt. Als Rückfallebene kann der Parameter `tryFurtherAnyWithoutLoc` aktiviert werden.

regionID_<usage>

Einschränkung des Suchraums auf die Region mit der durch den Parameter angegebenen ID. Die Regionen und ihre IDs müssen in einer separaten Datei definiert und die Funktionalität in der Konfigurationsdatei des EFALocationServers aktiviert werden.

tryFurtherAnyWithoutLoc = 1

Bei der Punktverifikation kann eine Gemeindekennziffer durch den Parameter `anyOmc_<usage>` angegeben werden. Wird dabei kein Treffer gefunden, wird eine erneute Suche ohne die vorgegebene Gemeindekennziffer gestartet, wenn dieser Parameter aktiv ist.

4.6 Beeinflussung der Trefferqualität und Reduktion der Trefferliste

Analog zu den Suchkriterien (siehe [Suchkriterien](#)) gilt auch für die folgenden Parameter, dass diese vor der Implementation mit dem EFALocationServer Monitor getestet werden sollten. Sie sollten auch dem Makro für die Suchkriterien hinzugefügt werden.

Kann ein Punkt nicht eindeutig verifiziert werden, wird in der XML-Ausgabe eine Liste von Punkten zurückgegeben (siehe auch [Listenauswahl](#)). Der beste Treffer ist durch das Attribut `selected=1` markiert. Standardmäßig ist der beste Treffer derjenige, dessen Name die größte Ähnlichkeit zum Anfragestring `name_<usage>` hat. Das Standardverhalten kann durch die Parameter `w_<Kriterium><Modus>` geändert werden, indem Einfluss auf die Bewertung der Treffer genommen wird. Die Bewertung des Treffers, d.h. die Trefferqualität, lässt sich in der XML-Ausgabe am Attribut `matchQuality` der Treffer `odvNameElem` ablesen.

Außer der Gewichtung, die sich durch den Parameter Präfix `w_` für „weighting“ auszeichnet, ist eine Reduktion der Liste durch die Parameter `r_<Kriterium><Modus>` möglich. In diesem Fall wird das Präfix `r_` für „relevance“ verwendet. Die Reduktion entspricht einer Gewichtung und der anschließenden Reduktion durch den Parameter `prMinQu`.

Sowohl für Gewichtung als auch für Reduktion stehen zwei unterschiedliche Modi `<Modus>` zur Verfügung:

- **Al („always“)** - Das Bewertungskriterium `<Kriterium>` wird auf alle Treffer angewendet. Dabei wird die Trefferqualität aller Treffer, die nicht dem durch den Parameter angegebenen Kriterium entsprechen, auf 0 gesetzt. Allerdings nur, wenn es mindestens einen Treffer gibt, der dem Kriterium entspricht.
- **Am („among best results“)** - Es kommt häufig vor, dass mehrere Treffer mit der gleichen Trefferqualität bewertet werden. So kann es auch vorkommen, dass theoretisch mehrere Elemente beste Treffer im Sinne der Trefferqualität sind.

Beim Standardbewertungsverhalten wird der als bester Treffer markierte Punkt zufällig unter diesen ausgewählt. Durch die Parameter `w_<Kriterium>` ist die Möglichkeit gegeben, die Auswahl durch Modifikation der Trefferqualität zu beeinflussen. Dazu wird die Trefferqualität für alle Treffer, die dem durch den Parameter angegebenen Kriterium entsprechen, um eins erhöht.

Der Modus wird dem Parameter als Suffix angehängt.

Folgende Kriterien `<Kriterium>` stehen zur Verfügung:

coord

Mit diesem Parameter wird die Koordinate des Zentrums `<x>:<y>:<Koordinatenformat>` für die Methode zur Reduzierung bzw. Gewichtung `rMethod DISTANCEFROMCENTER` angegeben. Der Wert des Parameters setzt sich zusammen aus dem x, dem y und dem Koordinatenformat der Koordinate.

objPref

Der Wert des Parameters stellt eine Bitmaske dar. Werden Treffer mit dem durch die Bitmaske angegebenen Objekttypen gefunden, wird der beste Treffer unter diesen gesucht. Die Objekttypen sind dem Anhang [Objekttypen](#) zu entnehmen.

Hinweis: Beim Modus `<Modus> A1` werden zunächst Objekte mit den durch den Parameter angegebenen Typen gesucht. Sind keine Objekte in der Trefferliste vorhanden, wird auch unter den anderen Objekttypen nach Treffern gesucht

Beispiel: Sollen die besten Treffer auf Haltestellen und POIs beschränkt werden, wird der Parameter folgendermaßen angegeben: `w_objPref<Modus>=34 (2 + 32)`.

prefStModes

Befinden sich unter den besten Treffern Haltestellen, wird der zu selektierende Treffer unter den Haltestellen gesucht, die von dem durch den Parameter angegebenen Verkehrsmitteln bedient werden. Gibt es keine Haltestellen, die von den angegebenen Verkehrsmitteln bedient werden, werden auch andere Haltestellen ausgegeben.

Der Wert des Parameters ist eine Bitmaske, die sich aus den Identifikationsnummern der Verkehrsmittel herleitet. Um eine Eindeutigkeit der Verkehrsmittel zu erzielen, wird die Verkehrsmittel-ID als Exponent zur Zweierbasis verwendet. Eine Übersicht über die Verkehrsmittel und ihrer Identifikationsnummern ist dem Anhang [Verkehrsmitteltypen](#) zu entnehmen.

Beispiel: Haltestellen, die von Zügen und S-Bahnen bedient werden. ID der Züge: 0 ($2^0 = 1$). ID der S-Bahnen: 1 ($2^1 = 2$). Daraus folgt: `w_prefStModesAm=3` (entspricht $1 + 2$).

Hinweis: Diesen Parameter gibt es nur im Modus `<Modus> Am`. Er kommt nur dann zum Einsatz, wenn sich unter den besten Treffern Haltestellen befinden.

regPref

Der Wert des Parameters gibt die ID einer Region an. Werden Treffer in der angegebenen Region gefunden, wird der beste Treffer unter diesen gesucht. Die Angabe der Region erfolgt analog zum Parameter `regionID_<usage>` (siehe [Suchkriterien](#)).

Hinweis: Ist der Modus `<Modus> Al`, werden andere Regionen nur berücksichtigt, wenn in der angegebenen Region keine Treffer vorhanden sind.

relInf

Durch diesen Parameter fließt die Relevanz des Treffers in die Berechnung seiner Trefferqualität mit ein. Sein Wert ist ganzzahlig und liegt zwischen 0 und 100. Er beschreibt, mit wie viel Prozent die Relevanz in die Trefferqualität einfließt. Standardmäßig wird die Relevanz nicht berücksichtigt, was einem Wert 0 entspricht. Soll ausschließlich die Relevanz für die Trefferqualität berücksichtigt werden, muss der Wert 100 sein.

Die Relevanz der Treffer wird aus einer externen Datei, die automatisch oder händisch erstellt sein kann, eingelesen. Der Pfad zu dieser Datei muss in der Konfigurationsdatei des EFALocationServers angegeben werden.

Hinweis: Der Standardwert dieses Parameters muss nicht unbedingt 0 sein. Er kann in der Sektion in der Konfigurationsdatei des EFALocationServers abweichend definiert sein.

rMethod

Mit diesem Parameter wird die für die Reduzierung bzw. Gewichtung verwendete Methode angegeben. Folgende Methoden stehen zur Verfügung:

- `DISTANCEFROMCENTER` - Distanz zum Zentrum. Die Distanz wird mit `coord` angegeben.

- **BESTCOVER** - Größtmögliche Übereinstimmung von Wörterbuchtext und Benutzereingabe.
- **PREFIXCOUNT** - Anzahl gleicher Präfixbuchstaben von Wörterbuchtext und Benutzereingabe.
- **PREFIXRATIO** - Anteil gleicher Präfixbuchstaben von Wörterbuchtext und Benutzereingabe.

Das Präfix **w_** steht für „weighting“, da durch Kombination verschiedener Parameter eine Gewichtung nach verschiedenen Kriterien vorgenommen werden kann. Das heißt, dass eine beliebige Anzahl dieser Parameter verwendet werden kann; inklusive doppelter Verwendung gleicher Parameter mit unterschiedlichem Wert. Gleiches gilt für die Reduzierung der Trefferlisten mittels der Parameter mit Präfix **r_**. Die Parameter werden gemäß des Kriteriums **<Kriterium>** in folgender Reihenfolge ausgeführt:

1. Objekttypen (**objPref**)
2. Regionen (**regPref**)
3. Verkehrsmittel (**prefStModes**)

Innerhalb eines Kriteriums **<Kriterium>** werden die Parameter gemäß ihrem Modus in folgender Reihenfolge ausgeführt:

1. always (**A1**)
2. among best results (**Am**)

Werden Parameter mehrfach verwendet, erfolgt die Abarbeitung hierarchisch. Der erste Parameter hat die höchste Priorität, der letzte die niedrigste.

Beispiel zur Hierarchie: Werden die Parameter **w_regPrefAl=1** und **w_regPrefAl=2** (in dieser Reihenfolge) verwendet, so wird zunächst geprüft, ob sich unter den Treffern welche befinden, die in der Region 1 liegen. Ist dies der Fall kommt der Parameter **w_regPrefAl2** nicht mehr zum Einsatz. Die Treffer aus der Region 1 werden hoch bewertet und für alle anderen Treffer wird die Trefferqualität auf 0 gesetzt. Ist jedoch kein Treffer aus der Region 1 vorhanden, wird geprüft, ob Treffer aus der Region 2 vorhanden sind. Ist dies der Fall, werden die Treffer aus der Region 2 hoch bewertet und für alle anderen Treffer die Trefferqualität mit 0 angegeben. Siehe hierzu auch das Beispiel zur Kombination von Treffern weiter unten.

Anmerkung: Die Anzahl der Treffer wird durch die Gewichtung nicht beeinflusst. Es

wird lediglich die Trefferqualität modifiziert, wodurch die Selektion des besten Treffers beeinflusst werden kann. Eine Reduktion der Liste kann durch die Parameter `anyMaxSizeHitList` oder `prMinQu=1` erreicht werden (siehe [Optionale Parameter zur Optimierung der Punktsuche](#)).

Beispiel zur Kombination von Kriterien: Es werden folgende Parameter verwendet:

```
w_regPrefAm=1    // 3. Kriterium
w_objPrefAl=2    // 1. Kriterium
w_objPrefAl=12   // 2. Kriterium (12=8+4)
```

Die Haltestellenliste bestehe aus:

- Haltestelle "A" mit Trefferqualität 101 in der Region mit der ID 1
- Haltestelle "B" mit Trefferqualität 102 in der Region mit der ID 1
- POI "C" mit Trefferqualität 103 in der Region mit der ID 1
- Haltestelle "D" mit Trefferqualität 102 in der Region mit der ID 2

Als bester Treffer wird die Haltestelle "B" durch das Attribut `selected=1` gekennzeichnet, da:

1. Kriterium: Die Trefferqualität von POI "C" wird auf 0 gesetzt, da es sich nicht um eine Haltestelle (Objekttyp 2) handelt. Die modifizierten Trefferqualitäten sind wie folgt:
 - Haltestelle "A" mit Trefferqualität 101 in der Region mit der ID 1
 - Haltestelle "B" mit Trefferqualität 102 in der Region mit der ID 1
 - POI "C" mit Trefferqualität **0** in der Region mit der ID 1
 - Haltestelle "D" mit Trefferqualität 102 in der Region mit der ID 2
2. Kriterium: Kommt nicht zum Einsatz, da bereits ein anderes Kriterium für Objekttypen (1. Kriterium) zuvor angewendet wurde. Die Trefferqualitäten ändern sich somit nicht.
3. Kriterium: Bezieht aufgrund des Modus `<Modus> Am` ausschließlich auf die besten Treffer. Dies sind Haltestelle "B" und "D" mit der Trefferqualität 102. Da sich Haltestelle B in der Region mit der ID 1 befindet, wird die Trefferqualität auf 103 erhöht. Die Trefferqualitäten sind nun wie folgt:
 - Haltestelle "A" mit Trefferqualität 101 in der Region mit der ID 1
 - Haltestelle "B" mit Trefferqualität **103** in der Region mit der ID 1
 - POI "C" mit Trefferqualität 0 in der Region mit der ID 1
 - Haltestelle "D" mit Trefferqualität 102 in der Region mit der ID 2

Damit kann Haltestelle "B" eindeutig als bester Treffer gekennzeichnet werden.

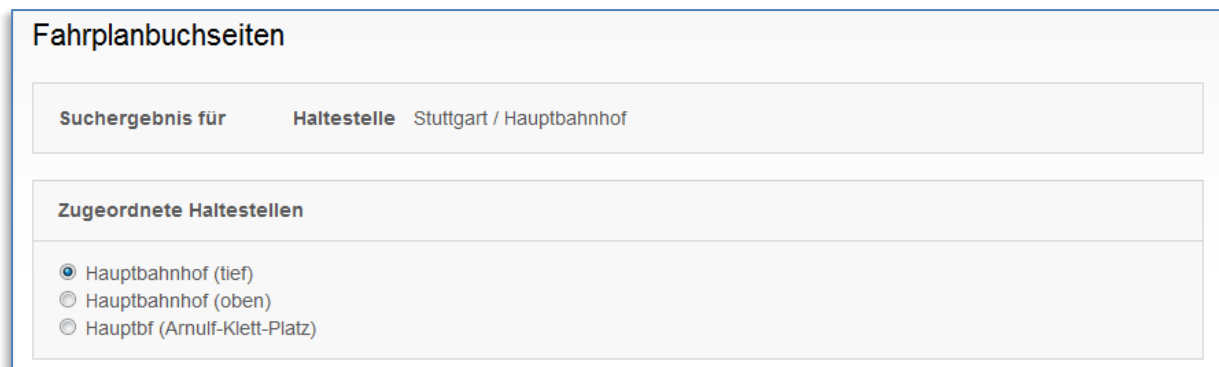
4.7 Listenauswahl

Ergibt ein Punkt kein Volltreffer, enthält das Element `itdOdvName` mehrere Elemente `odvNameElem`. Der beste Treffer ist durch das Attribut `selected=1` markiert (siehe hierzu auch [Beeinflussung der Trefferqualität und Reduktion der Trefferliste](#)). Zur Auswahl eines Punktes wird der Wert des Attributes `stateless` durch den Parameter `name_<usage>` übergeben.

4.8 Wahl einer zugeordneten Haltestelle

Einige Anfragearten, wie zum Beispiel der [StopTimetable-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 (siehe Abbildung 1). 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 erfolgt über den Parameter `name_<usage>`. Als Wert wird der Wert des Attributes `stateless` übergeben.



The screenshot shows a web interface titled 'Fahrplanbuchseiten'. It contains a search result section with the text 'Suchergebnis für Haltestelle Stuttgart / Hauptbahnhof'. Below this, there is a section titled 'Zugeordnete Haltestellen' which lists three options with radio buttons: 'Hauptbahnhof (tief)', 'Hauptbahnhof (oben)', and 'Hauptbf (Arnulf-Klett-Platz)'. The first option, 'Hauptbahnhof (tief)', is selected.

Abbildung 1 - Wahl einer zugeordneten Haltestelle am Beispiel Fahrplanbuchseiten

4.9 Eingabe von Haltestellen POIs und Koordinaten

Soll kein Punkt gesucht werden, sondern ein bereits bekannter Punkt verwendet werden, kann dieser über seine ID oder Koordinate identifiziert werden. Da keine Punktverifikation stattfindet, ist es nicht notwendig den EFALocationServer zu verwenden. Den EFA Location Server betreffende Parameter können daher weggelassen werden.

den. Es sind lediglich die Parameter `type_<usage>`, `name_<usage>` und gegebenenfalls `stateless=1` notwendig. Eventuell ist es notwendig das Ausgabeformat zu konvertieren. Die hierfür notwendigen Parameter sind im Kapitel [Ausgabeformat des EFALocationServers](#) beschrieben.

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=20018235`

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=3377:5111000:-1`

Koordinaten

Eine Koordinate 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=6787835:51231000:WGS84[dd.ddddd]` oder `name_origin=3345565:480331:NAV3:ein schöner Ort`

4.10 Ausgabeformat des EFALocationServers

Wird ein Punkt nicht über den EFALocationServer gesucht, sondern direkt vom EFAITKernel ausgegeben, hat er ein anderes Ausgabeformat. Das Format des EFALocationServers ist gegenüber dem des EFAITKernels um einige Attribute erweitert. Werden sowohl vom EFALocationServer als auch vom EFAITKernel ausgegebene Punkte verwendet, empfiehlt es sich das EFAITKernel Format in das des EFALocationServers zu konvertieren, um eine einheitliche Schnittstelle zu erhalten. Im Zweifelsfall empfiehlt es sich die unten beschriebenen Parameter standardmäßig zu verwenden, sofern der EFAController nicht bereits entsprechend konfiguriert ist.

Hinweis: Wird ein Volltreffer übergeben (siehe [Eingabe von Haltestellen POIs und Koordinaten](#)), ist das Ausgabeformat das des EFAITKernels. Auch wenn der Parameter `type_<usage>=any` verwendet wird.

convertAddressesITKernel2LocationServer

Konvertiert das intern verwendete Adress-Format des EFAITKernels in das vom EFALocationServer verwendete Format.

convertCoord2LocationServer

Konvertiert das intern verwendete Koordinaten-Format des EFAITKernels in das vom EFALocationServer verwendete Format.

convertCrossingsITKernel2LocationServer

Konvertiert das intern verwendete Kreuzungs-Format des EFAITKernels in das vom EFALocationServer verwendete Format.

convertPOIsITKernel2LocationServer

Konvertiert das intern verwendete POI-Format des EFAITKernels in das vom EFALocationServer verwendete Format.

convertStopsPTKernel2LocationServer

Konvertiert das intern verwendete Haltestellen-Format des EFAITKernels in das vom EFALocationServer verwendete Format.

5 Linienverifikation

Da die Dokumentation keine Requests mit Liniensuche umfasst, soll hier lediglich die Eingabe bekannter Linien erläutert werden. Diese wird zum Beispiel für die Linienauswahl im [DM-Request](#) benötigt.

5.1 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](#)) verlinkt werden sollen. Dort befinden sich die Angaben zur Linie im Element `motDivaParams` im Container `itdMean-sOfTransport` 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 `itdServing-Line` in der XML Ausgabe. Es bietet sich die Implementation einer Linienauswahl an (siehe Abbildung 2).



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> : <Ergän-`

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

Eine typische Anwendung für den StopFinder-Request ist das Erstellen einer Vorschlagsliste (siehe Abbildung 3), 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 Hinweis zur Punktverifikation

Bei der [Punktverifikation](#) wird als Erweiterung `<usage>` der Wert `sf` angegeben.

6.3 Allgemeine Funktionalität

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

6.4 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.

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 Verbindungsoptionen. Die als optionalen Verbindungsoptionen bieten umfassende Möglichkeiten die Fahrt-ergebnisse nach eigenen Bedürfnissen zu optimieren.

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?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

```
http://server:port/virtuellesVerzeichnis/XSLT_TRIP_REQUEST2?HTTP-Parameter
```

7.2 Hinweis zur Punktverifikation

Bei der [Punktverifikation](#) werden als Erweiterung `<usage>` folgende Werte angegeben:

- `origin` – Startpunkt
- `destination` – Zielpunkt
- `via` – Zwischenhalt

7.3 Optionaler Zwischenhalt

Der Zwischenhalt hat die Parameterergänzung `via` und kann ausschließlich beim [Trip-Request](#) oder [PS-Request](#) verwendet werden. Ausschließlich auf den Zwischenhalt beziehen sich folgende optionale Parameter:

dwellTime_<usage>

Aufenthaltsdauer an einem Zwischenhalt (Via-Punkt). Der Präfix `<usage>` bezieht sich in diesem Fall ausschließlich auf Zwischenhalte; nicht auf den Start- oder Zielpunkt. Die Aufenthaltsdauer wird in Minuten angegeben.

Hinweis: Wird eine Aufenthaltsdauer von 0 Minuten gewählt, werden Adressen und wichtige Punkte als Zwischenhalt nicht berücksichtigt. Dieses Verhalten kann durch den Parameter `AlwaysDwellTimeJourneyCalculation=1` in der Konfigurationsdatei des EFAControllers geändert werden.

7.4 Allgemeine Funktionalität

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`.

7.5 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.

useRealtime = 1

Aktiviert die Berücksichtigung der Echtzeitüberwachung (siehe Abbildung 4).





			Stadtbahn U7 Richtung Nellingen Ostfildern	
		heute		
	ab 08:26	08:36		Hauptbf (A.-Klett-Pl.) Gleis 4
	an 08:36	08:46		Ruhbank (Fernsehturm) #

Abbildung 4 - Fahrtauskunft mit Echtzeitüberwachung

Dies bewirkt, dass in der XML-Ausgabe (siehe Abbildung 5) für jeden echtzeitüberwachten Teilweg ein zusätzliches Element `itdRBLControlled` ausgegeben wird. Die Verspätung wird in Minuten erfasst. Für die Ankunft wird sie als Attribut `delayMinutes` und für die Abfahrt als Attribut `delayMinutesArr` angegeben.

```
<itdPartialRoute type="PT" active="1" timeMinute="10" bookingCode="Sefa01.dc.vvs.de_507847682R1T0PR2"
  partialRouteType="normalPartialRoute" bufBef="-1" bufAft="-1">
+ <itdPoint stopID="5006112" area="1" platform="4" name="Hauptbf (A.-Klett-Pl.)" nameWO="Hauptbf (A.-
  Klett-Pl.)" platformName="Gleis 4" usage="departure" x="3513396.00000" y="754922.00000"
  mapName="NBWT" omc="8111000" placeID="51" locality="Stuttgart">
+ <itdPoint stopID="5006128" area="2" platform="1" name="Ruhbank (Fernsehturm)" nameWO="Ruhbank
  (Fernsehturm)" platformName="" usage="arrival" x="3514399.00000" y="758213.00000"
  mapName="NBWT" omc="8111000" placeID="51" locality="Stuttgart">
+ <itdMeansOfTransport name="Stadtbahn U7" shortname="U7" symbol="U7" type="1" motType="3"
  productName="Stadtbahn" destination="Nellingen Ostfildern" network="vvs" TTB="1" STT="1" ROP="1"
  destID="5002974" spTr="" tC="63">
  <itdRBLControlled delayMinutes="10" delayMinutesArr="10" />
</itdPartialRoute>
```

Abbildung 5 - Echtzeitüberwachung in der XML Antwort des Trip-Requests

Änderungen des Steigs können als generische Attribute erfasst werden.

Das Element `itdDateTime` enthält die Echtzeit, die Sollzeit wird in einem zusätzlichen Element `itdDateTimeTarget` ausgegeben (siehe Abbildung 6).

```
<itdPoint stopID="5006112" area="1" platform="4" name="Hauptbf (A.-Klett-Pl.)" nameWO="Hauptbf (A.-
  Klett-Pl.)" platformName="Gleis 4" usage="departure" x="3513396.00000" y="754922.00000"
  mapName="NBWT" omc="8111000" placeID="51" locality="Stuttgart">
+ <itdMapItemList>
- <itdDateTime>
  <itdDate year="2012" month="6" day="27" weekday="4" />
  <itdTime hour="8" minute="36" />
</itdDateTime>
- <itdDateTimeTarget>
  <itdDate year="2012" month="6" day="27" weekday="4" />
  <itdTime hour="8" minute="26" />
</itdDateTimeTarget>
</itdPoint>
```

Abbildung 6 - Echt- und Sollzeit

7.6 Verkehrsmittelausschluss und Verkehrsmiteinschluss

Eine Übersicht über die Standard-Verkehrsmittelbelegung ist dem Anhang [Verkehrsmitteltypen](#) zu entnehmen.

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.

7.7 Mobilitätseinschränkungen

Die unter den Mobilitätseinschränkungen zusammengefassten Parameter ermöglichen es auf die Bedürfnisse von Menschen mit eingeschränkter Mobilität einzugehen. Diese Parameter können nur verwendet werden, wenn Treppen, Rolltreppen, etc. in den Daten erfasst sind.

impairedOptionsActive=1

Aktiviert die Parameter für eingeschränkte Mobilität.

lowPlatformVhcl

Ist dieser Parameter aktiviert, werden ausschließlich Niederflurfahrzeuge berücksichtigt. Es ist nicht notwendig dem Parameter einen bestimmten Wert zuzuweisen. Sobald der Parameter übergeben wird, ist diese Option aktiv.

noElevators

Wird dieser Parameter übergeben, werden ausschließlich Fahrten mit Umstiegen ohne Verwendung von Aufzügen beauskunftet. Es ist nicht notwendig dem Parameter einen bestimmten Wert zuzuweisen. Sobald der Parameter übergeben wird, ist diese Option aktiv.

noEscalators

Wird dieser Parameter übergeben, werden ausschließlich Fahrten mit Umstiegen, die nicht über Rolltreppen erfolgen, beauskunftet. Es ist nicht notwendig dem Parameter einen bestimmten Wert zuzuweisen. Sobald der Parameter übergeben wird, ist diese Option aktiv.

noSolidStairs

Wird dieser Parameter übergeben, werden ausschließlich Fahrten mit Umstiegen, die nicht über normale Treppen erfolgen, beauskunftet. Es ist nicht notwendig dem Parameter einen bestimmten Wert zuzuweisen. Sobald der Parameter übergeben wird, ist diese Option aktiv.

wheelchair

Wird dieser Parameter übergeben, werden ausschließlich rollstuhlgerechte Fahrzeuge bei der Auskunft berücksichtigt. Rollstuhlgerechte Fahrzeuge haben einen Hublift bzw. eine Rampe oder ermöglichen den niveaugleichen Einstieg. Es ist nicht notwendig dem Parameter einen bestimmten Wert zuzuweisen. Sobald der Parameter übergeben wird, ist diese Option aktiv.

7.8 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:

`Umstiegszeit [min] = (Zeit aus der Umstiegsmatrix [min] * 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 [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:

`Geschwindigkeit [km/h] = (Standardgeschwindigkeit [km/h] * Wert des Parameters) / 100`

Die Standardgeschwindigkeit beträgt, falls nicht anders konfiguriert, 4 km/h.

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.

lineRestriction

Mit diesem Parameter können bestimmte Linien ausgeschlossen werden. Er kann folgende Werte annehmen:

- 400 (alle Linien, Default)
- 401 (alle Linien außer ICE)
- 402 (Verbund ohne Aufschlag)
- 403 (Verbund und Nahverkehr)

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 Konfigurationsdatei des EFA Controllers definierte Maximum, Default)

routeType

Gibt an, nach welchem Kriterium die Fahrtauskünfte optimiert werden sollen:

- `leastinterchange` (Verbindungen mit möglichst wenigen Umstiegen)
- `lasttime` (Möglichst schnelle Verbindungen, Default)
- `leastwalking` (Verbindungen mit möglichst wenigen Fußwegen)

useProxFootSearch = 1

Berücksichtigung von nahegelegenen Haltestellen.

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

7.9 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.

trITMOT

Mit diesem Parameter wird der Transportmitteltyp für den Weg vom Startpunkt zur Starthaltestelle und von der Zielhaltestelle zum Zielpunkt gewählt. Folgende Werte sind möglich:

- 100 (Fußweg)
- 101 (Bike & Ride)
- 102 (Fahrradmitnahme)
- 103 (Kiss & Ride)
- 104 (Park & Ride)
- 105 (Taxi)

Alternativ können die Parameter `trITArrMOT` und `trITDepMOT` verwendet werden. Mit ihnen kann man den Transportmitteltyp für Start- und Zielpunkt separat setzen.

trITMOTvalue

Der Wert des Parameters gibt die maximale Zeit an, die der Weg vom Startpunkt zur Starthaltestelle sowie der Weg von der Zielhaltestelle zum Zielpunkt dauern soll. Die Zeit wird in Minuten angegeben. Voreingestellt sind 10 Minuten, falls nichts anderes in der Konfigurationsdatei des EFA Controllers bzw. des EFA Servers definiert ist.

Alternativ können die Parameter `trITArrMOTvalue` und `trITDepMOTvalue` verwendet werden. Mit ihnen kann man die maximale Zeit für Start- und Zielpunkt separat setzen. Sollen unterschiedliche maximale Zeiten für die einzelnen Transportmitteltypen angegeben werden, kommen die Parameter `trITMOTvalue<Transportmitteltyp>` bzw. `trITArrMOTvalue<Transportmitteltyp>` und `trITDepMOTvalue<Transportmitteltyp>` zum Einsatz. Die verfügbaren Transportmitteltypen sind für den Parameter `trITMOT` aufgelistet.

7.10 Folgeanfragen

Bei Folgeanfragen müssen die Parameter `sessionID` und `requestID` übergeben werden (siehe [Session-Handling](#)).

command

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
- `printOverview` Übersicht drucken (PDF)
- `printSingleview:n` Einzelansicht drucken (n - Fahrtindex)

8 PS-Request

Im Folgenden wird die Ansteuerung zum Erstellen eines Pendlerfahrplans (PS-Request), auch „Persönlicher Fahrplan“ genannt, erläutert. Die Abkürzung „PS“ steht für „Personal Schedule“. Der Pendlerfahrplan hat große Ähnlichkeit zur [Fahrtanfrage](#). Im Gegensatz zu dieser werden jedoch Fahrten innerhalb eines Zeitintervalls ermittelt. Daher unterscheidet sich die Eingabe von Datum und Uhrzeit. Auf Wunsch können Zeitintervalle für eine einfache Fahrt oder für eine Hin- und Rückfahrt angegeben werden.

8.1 Ansteuerung der Pendlerfahrplans

Der Pendlerfahrplan wird über HTTP-Parameter angesteuert. Als Antwort wird eine PDF-Datei generiert. Zusätzlich liegt die Antwort in einer XML-Struktur vor. Dort ist auch der Link zur PDF-Datei im Attribut `src` des Elements `itdDownload` gespeichert.

Je nach Anfrageart wird die Antwort direkt in XML (`XML_PS_REQUEST2`) oder in HTML (`XSLT_PS_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_PS_REQUEST2?HTTP-Parameter`

Für eine HTML-Antwort ist folgende URL einzugeben:

`http://server:port/virtuellesVerzeichnis/XSLT_PS_REQUEST2?HTTP-Parameter`

Hinweis: Die generierten PDF-Dateien werden in einem Verzeichnis `PersonalSchedule` abgelegt. Dies wird nicht automatisch generiert, sondern muss manuell für das EFA Web-Interface angelegt werden.

8.2 Hinweis zur Punktverifikation

Bei der [Punktverifikation](#) werden als Erweiterung `<usage>` folgende Werte angegeben:

- `origin` – Startpunkt
- `destination` – Zielpunkt
- `via` – Zwischenhalt (standardmäßig deaktiviert)

8.3 Allgemeine Funktionalität

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

itdDateTimeDepArr

Schaltet zwischen einem Abfahrts- (`dep`) und Ankunftsmonitor (`arr`) um.

Mit den Werten `firstService` bzw. `lastService` werden die ersten bzw. letzten Fahrten ausgegeben. Der Uhrzeit-Parameter `itdTime` wird in diesem Fall nicht berücksichtigt. Standardmäßig findet die erste Fahrt um 4 Uhr morgens und die letzte Fahrt um 2 Uhr des Folgetages statt. Diese Zeiten können in der Konfigurationsdatei des EFA Controllers bzw. des EFA Servers geändert werden.

psParamOneWay = 1

Standardmäßig werden sowohl Hin- als auch Rückfahrten berechnet. Wird dieser Parameter aktiviert wird der Fahrplan nur für eine Richtung berechnet.

useAltOdv = 1

Aktiviert die Verwendung eines Zwischenhalts, d.h. einer Haltestelle, an der bei den ausgegebenen Fahrten gehalten wird. Zwischenhalte mit `name_via` und `type_via` angegeben. Außerdem können die in Kapitel [Optionaler Zwischenhalt](#) für den [Trip-Request](#) beschriebenen optionalen Parameter für den Zwischenhalt verwendet werden.

8.4 Datum und Uhrzeit für den Pendlerfahrplan

Die Berechnung des Pendlerfahrplans erfolgt aufgrund von Stichtagen. Diese sind vordefiniert, so dass statt einem Datum ein Wochentagstyp gewählt werden muss. Zusätzlich wird ein Uhrzeitintervall benötigt. Start und Ende des Intervalls für die Hinfahrt und des optionalen Intervalls für die Rückfahrt werden anhand des Suffixes `_<usagePS>` unterschieden. Folgende Werte sind möglich:

- `arrFrom` - Der Wert des Parameters bezieht sich auf den Beginn des Intervalls für die Rückfahrt, d.h. die frühestmögliche Hinfahrt.
- `arrTo` - Der Wert des Parameters bezieht sich auf das Ende des Intervalls für die Rückfahrt, d.h. die spätestmögliche Hinfahrt.
- `depFrom` - Der Wert des Parameters bezieht sich auf den Beginn des Intervalls für die Rückfahrt, d.h. die frühestmögliche Hinfahrt.
- `depTo` - Der Wert des Parameters bezieht sich auf das Ende des Intervalls für die Rückfahrt, d.h. die spätestmögliche Hinfahrt.

Hinweis: Die maximale Größe des Intervalls und die Vorbelegung der Zeiten sind in der Konfigurationsdatei des EFAControllers festgelegt. Die Intervallgröße kann durch den HTTP-Parameter `psParamMaxTimeHours` überschrieben werden.

itdTime_<usagePS>

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“)

itdTimeAMPM_<usagePS>

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

itdTimeHour_<usagePS>

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

itdTimeMinute_<usagePS>

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

psParamGroupID

Gelegentlich sind verschiedene Fahrplandaten mit unterschiedlichen Gültigkeitsperioden verfügbar. Dies ist insbesondere in der Zeit um den Fahrplanwechsel herum der Fall. Da der Persönliche Fahrplan sich nicht auf ein konkretes Datum bezieht, kann im Fall von mehreren Fahrplandaten der Satz mit der gewünschten Fahrplanperiode durch diesen Parameter ausgewählt werden.

Als Wert wird das Attribut `groupID` einer der Datengruppen `itdValidityPeriod`, die im Element `itdValidityInfo` der XML-Ausgabe (siehe Abbildung 7) zur Auswahl stehen, angegeben. Als textuelle Beschreibung für eine Auswahlbox eignet sich das Attribut `description`. Für jeden Wochentagstypen gibt es eigene Fahrplanperioden. Der Wochentagstyp kann dem Attribut `weekdayType` entnommen werden. Die

aktuell ausgewählte Fahrplanperiode ist im Attribut `groupID` des Elements `itdPS-Date` vermerkt, der aktive Wochentagstyp im Attribut `weekday`.

```
<itdPSDateTime>
  <itdValidityInfo messageCode="0" validFrom="20101001" validTo="20111210">
    <itdValidityPeriod validFrom="20101001" validTo="20101211" weekdayType="8" groupID="1" description="Jahresfahrplan 2010">
      <refDate>20101209</refDate>
      <refDate>20101210</refDate>
    </itdValidityPeriod>
    <itdValidityPeriod validFrom="20101001" validTo="20101211" weekdayType="6" groupID="1" description="Jahresfahrplan 2010">
      <refDate>20101204</refDate>
    </itdValidityPeriod>
    <itdValidityPeriod validFrom="20101001" validTo="20101211" weekdayType="10" groupID="1" description="Jahresfahrplan 2010">
      <refDate>20101205</refDate>
    </itdValidityPeriod>
    <itdValidityPeriod validFrom="20101212" validTo="20111210" weekdayType="8" groupID="2" description="Jahresfahrplan 2011">
      <refDate>20101217</refDate>
      <refDate>20101223</refDate>
      <refDate>20101224</refDate>
      <refDate>20101227</refDate>
    </itdValidityPeriod>
    <itdValidityPeriod validFrom="20101212" validTo="20111210" weekdayType="6" groupID="2" description="Jahresfahrplan 2011">
      <refDate>20101218</refDate>
    </itdValidityPeriod>
    <itdValidityPeriod validFrom="20101212" validTo="20111210" weekdayType="10" groupID="2" description="Jahresfahrplan 2011">
      <refDate>20101225</refDate>
      <refDate>20101226</refDate>
      <refDate>20110101</refDate>
    </itdValidityPeriod>
  </itdValidityInfo>
  <itdPSDate weekday="6" groupID="1" validityFrom="20101001" validityTo="20101211">
    <itdDate year="2010" month="12" day="4" weekday="7"/>
  </itdPSDate>
</itdPSDateTime>
```

Abbildung 7 - Unterschiedliche Gültigkeitsperioden

psParamMaxTimeHours

Der Parameter gibt die maximal zulässige Größe für das Intervall an, in dem Fahrten für den Persönlichen Fahrplan berechnet werden sollen. Wertebereich: [0..24]. Liegen die Uhrzeiten für den Start und das Ende des Intervalls weiter auseinander als durch den Parameter angegeben, wird eine Fehlermeldung ausgegeben.

psParamSampleDate

Der Wert gibt einen Stichtag an, der die in den Daten vorhandenen übersteuert. Er setzt sich aus der vierstelligen Jahreszahl, gefolgt von der zweistelligen Angabe des Monats und der zweistelligen Angabe des Tages, zusammen.

psParamWeekday

Der Parameter bestimmt für welchen Wochentagstypen Fahrten berechnet werden sollen. Mögliche Werte sind:

- 6 : SAMSTAG - Es werden Fahrten berechnet, die samstags gültig sind.

- 8 : MO-FR - Es werden Fahrten berechnet, die montags bis freitags gültig sind.
- 10 : SONNTAG - Es werden Fahrten berechnet, die sonn- und feiertags gültig sind.

useAllSampleDates = 1

Standardmäßig werden für die Berechnung eines Persönlichen Fahrplans nur die gewählten Stichtage verwendet. Dieses Verhalten wird durch diesen Parameter geändert. Er bewirkt, dass alle Stichtage berücksichtigt werden, unabhängig davon, ob sie zum gewählten Wochentagstypen passen.

8.5 Zusatzinformation in der PDF-Datei

psParamAddressExt1

Ermöglicht das Einblenden eines Zusatztextes unterhalb des Namens im Kopf der PDF-Version des Persönlichen Fahrplans.

psParamAddressExt2

Ermöglicht das Einblenden eines zweiten Zusatztextes unterhalb des ersten Zusatztexts im Kopf der PDF-Version des Persönlichen Fahrplans.

psParamAddressName

Ermöglicht das Einblenden des Namens im Kopf der PDF-Version des Persönlichen Fahrplans.

psParamAddressPlace

Ermöglicht das Einblenden von PLZ und Ort im Kopf der PDF-Version des Persönlichen Fahrplans.

psParamAddressStreet

Ermöglicht das Einblenden der Straße und Hausnummer <Straße> im Kopf der PDF-Version des Persönlichen Fahrplans.

9 DM-Request

Im Folgenden wird die Anfrage eines Abfahrtsmonitors (DM-Request) erläutert. Grundsätzlich gilt dabei zu beachten, dass es zwei unterschiedliche Typen von Abfahrtsmonitoren gibt: den Abfahrtsmonitor einer konkreten Haltestellen (siehe Abbildung 8) und einen kombinierten Abfahrtsmonitor für mehrere Haltestellen (siehe Abbildung 9 bzw. Abbildung 10).

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 [Haltestellenauswahl](#)).

Abfahrtsmonitor			
Datum: 13.03.2013 von: München, Hauptbahnhof Abfahrt: 09:00 Uhr			
Datum/Zeit	Verkehrsmittel	Linie	Richtung
13.03. - 09:00 Uhr	U	U2	Feldmoching
13.03. - 09:00 Uhr	U	U4	Westendstraße
13.03. - 09:00 Uhr	Tram	20	Moosach Bahnhof
13.03. - 09:00 Uhr	Tram	19	Willibaldplatz
13.03. - 09:00 Uhr	U	U2	Messestadt Ost
13.03. - 09:01 Uhr	S	S4	Geltendorf
13.03. - 09:01 Uhr	Tram	16	Romanplatz
13.03. - 09:01 Uhr	S	S2	Ostbahnhof
13.03. - 09:01 Uhr	R	RB 57016 Regionalbahn	Donauwörth
13.03. - 09:01 Uhr	R	ALX 84124 alex - Länderbahn und Vogtlandbahn GmbH	Hof Hbf
13.03. - 09:01 Uhr	R	RB 57212 Regionalbahn	Dinkelscherben

Abbildung 8 - 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 Haltestelle gruppiert werden (siehe Abbildung 9) oder für jede Abfahrt der Haltestellenname angegeben werden (siehe Abbildung 10). 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 9 - Kombiniertes Abfahrtsmonitor I

Abfahrtsmonitor				
Datum: 13.03.2013 von: München, Thierschstraße 2				
Abfahrt: 09:00 Uhr				
Datum/Zeit	Haltestelle	Verkehrsmittel	Linie	Richtung
13.03. - 09:00 Uhr	Isartor		S8	Flughafen München
13.03. - 09:00 Uhr	Ludwigsbrücke		132	Isartor
13.03. - 09:00 Uhr	Isartor		16	Romanplatz
13.03. - 09:01 Uhr	Isartor		S8	Herrsching

Abbildung 10 - Kombiniertes Abfahrtsmonitor II

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.

9.1 Ansteuerung der Anfrage eines Abfahrtsmonitors

Die Anfrage des Abfahrtsmonitors wird über HTTP-Parameter gesteuert. 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`

9.2 Hinweis zur Punktverifikation

Bei der [Punktverifikation](#) wird als Erweiterung `<usage>` der Wert `dm` angegeben.

9.3 Allgemeine Funktionalität

Es können die in den Kapiteln [Grundfunktionalität](#), [Datum und Uhrzeit](#) sowie [Punktverifikation](#) beschriebenen HTTP-Parameter verwendet werden. Ebenso gilt der für den [Trip-Request](#) beschriebene [Verkehrsmittelausschluss und Verkehrsmiteinschluss](#). Außerdem folgende Parameter:

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.

useRealtime = 1

Aktiviert die Berücksichtigung der Echtzeitüberwachung (siehe Abbildung 11).

Zeit	heute	Linie	in Richtung	Gleis / Bussteig	Hinweise
 08:48	08:50	 44	Killesberg		Niederflurbus
 08:49	08:50	 U9	Hedelfingen	2	
 08:50		 U12	Killesberg	3	

Abbildung 11 - Abfahrtsmonitor mit Echtzeit

Dies bewirkt, dass in der XML-Ausgabe (siehe Abbildung 12) jede echtzeitüberwachte Linie `itdServingLine` mit dem Attribut `realtime=1` gekennzeichnet ist. Die Verspätung wird in Minuten als Attribut `delay` des Knotens `itdNoTrain` erfasst. Ist der Wert `-9999`, fällt die Fahrt aus. Änderungen des Steigs können als generische Attribute erfasst werden. Die Echtzeit wird in einem zusätzlichen Element `itdRTDateTime` ausgegeben.

```
<itdDeparture stopID="5006112" x="3513385" y="754905" mapName="NBWT" area="6" platform="5" platformName=""
stopName="Hauptbf (A.-Klett-Pl.)" nameWO="Hauptbf (A.-Klett-Pl.)" countdown="1">
- <itdDateTime>
  <itdDate year="2012" month="6" day="27" weekday="4" />
  <itdTime hour="8" minute="48" ap="" />
</itdDateTime>
- <itdRTDateTime>
  <itdDate year="2012" month="6" day="27" weekday="4" />
  <itdTime hour="8" minute="50" ap="" />
</itdRTDateTime>
- <itdServingLine key="365" code="3" number="44" symbol="44" motType="5" realtime="1" direction="Killesberg"
destID="-1" stateless="vvs.30044:R:j12">
  <itdNoTrain name="Bus" delay="2">Niederflurbus</itdNoTrain>
  <motDivaParams line="30044" project="j12" direction="R" supplement="" network="vvs" />
+ <itdOperator>
</itdServingLine>
<genAttrList />
<itdServingTrip tripCode="365" AVMSTripID="113016000021" />
</itdDeparture>
```

Abbildung 12 - Echtzeitüberwachung in der XML Antwort des DM-Requests

9.4 Haltestellenauswahl

Soll der Abfahrtsmonitor nur für eine Haltestelle angezeigt werden, muss gegebenenfalls die gewünschte nahegelegene bzw. zugeordnete Haltestelle in einem Zwischenschritt ermittelt werden. Dies erfolgt wie in Kapitel [Wahl einer zugeordneten Haltestelle](#) beschrieben.

9.5 Linienauswahl

Der Abfahrtsmonitor kann für eine oder mehrere bedienende Haltestellen angezeigt werden. Die Linienauswahl kann sowohl in einem Zwischenschritt (siehe Abbildung 13) als auch automatisch erfolgen (siehe Parameter `mode=direct`).

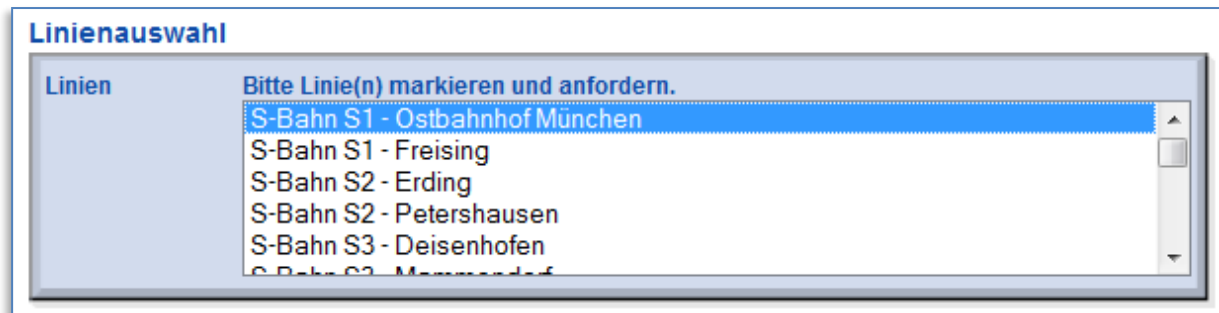


Abbildung 13 - Wahl der Linien

Sollen in einem Zwischenschritt eine oder mehrere bedienende Linien gewählt werden, erfolgt die Wahl der Linie wie im Kapitel [Eingabe bekannter Linien](#) beschrieben. Zusätzlich ist folgender Parameter notwendig:

mode = direct

Soll keine Auswahl der Linien stattfinden und der Abfahrtsmonitor für alle Linien ausgegeben werden, muss der Direkt-Modus des Abfahrtsmonitors bereits im ersten Schritt verwendet werden. Ansonsten wird dieser Parameter erst bei der Auswahl der Linien mit übergeben.

Hinweis: Beim Direkt-Modus werden alternative Haltestellen gesucht, sofern diese per Konfiguration aktiviert wurde. Um zu gewährleisten, dass der Abfahrtsmonitor nur für die konkret angefragte Haltestelle gilt, muss die Suche von alternativen Haltestellen in diesem Fall mittels des Parameters `useProxFootSearch=0` (siehe [Weitere Optionen für den öffentlichen Verkehr](#)) ausgeschaltet werden.

9.6 Folgeanfragen

Bei Folgeanfragen müssen die Parameter `sessionID` und `requestID` übergeben werden (siehe [Session-Handling](#)).

command

Mögliche Werte (Ausschnitt):

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

10 DMTTP-Request

Im Folgenden wird die Ansteuerung der Soll-Abfahrtstafel (DMTTP-Request) erläutert. Sie beinhaltet die Abfahrten ausgewählter Linien einer Haltestelle für die komplette aktuelle Fahrplanperiode. Die Abkürzung „TTP“ steht in diesem Sinne für „Time Table Period“.

10.1 Ansteuerung der Anfrage eines Abfahrtsmonitors

Die Anfrage der Soll-Abfahrtstafel wird über HTTP-Parameter gesteuert. Die Antwort der Anfrage liegt in einer XML-Struktur vor. Je nach Anfrageart wird die Antwort direkt in XML (XML_DMTTP_REQUEST) oder in HTML (XSLT_DMTTP_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_DMTTP_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

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

10.2 Hinweis zur Punktverifikation

Bei der [Punktverifikation](#) wird als Erweiterung `<usage>` der Wert `ss` angegeben.

10.3 Allgemeine Funktionalität

Die Soll-Abfahrtstafel funktioniert analog zum Abfahrtsmonitor und verwendet die gleichen HTTP-Parameter. Lediglich die Parametererweiterung `_<usage>` für die [Punktverifikation](#) unterscheidet sich.

11 StopTimestable-Request

Im Folgenden wird die Ansteuerung des Aushangfahrplans (StopTimestable-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.

11.1 Ansteuerung des Aushangfahrplans

Der Aushangfahrplan wird über HTTP-Parameter angesteuert. Als Antwort wird eine PDF-Datei generiert. Zusätzlich liegt die Antwort in einer XML-Struktur vor. Dort ist auch der Link zur PDF-Datei im Attribut `src` des Elements `itdDownload` gespeichert.

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`

11.2 Hinweis zur Punktverifikation

Bei der [Punktverifikation](#) wird als Erweiterung `<usage>` der Wert `stt` angegeben.

11.3 Allgemeine Funktionalität

Um einen Aushangfahrplan generieren zu können, müssen mindestens eine Haltestelle verifiziert und eine ihrer bedienenden Linie ausgewählt werden.

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 (siehe Abbildung 14). Ist dieser Parameter aktiv, werden doppelt vorhandene Aushangfahrpläne nur einfach ausgegeben.

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

Abbildung 14 - Aushangfahrplan der Linie 360 von Traveline South East mit mrgSt=0

11.4 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.

11.5 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.

12 StopList-Request

Mit dem StopList-Request können alle Haltestellen des Verbund-Gebietes ausgegeben werden. Bei großen Netzen empfiehlt sich eine Filterung durch die im Abschnitt [Parameter zur Filterung](#) beschriebenen Parameter. Ansonsten kann es zu langen Antwortzeiten oder zum Absturz des Browsers kommen. Zusätzlich gibt es Parameter, die die Ausgabe von ergänzender Information bewirken. Diese sind im Abschnitt [Parameter zur Ausgabe zusätzlicher Information](#) beschrieben.

12.1 Ansteuerung des StopList-Requests

Der StopList-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_STOPLIST_REQUEST) oder in HTML (XSLT_STOPLIST_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_STOPLIST_REQUEST?HTTP-Parameter`

Für eine HTML Antwort ist folgende URL einzugeben: `http://server:port/virtuellesVerzeichnis/XSLT_STOPLIST_REQUEST?HTTP-Parameter`

12.2 Parameter zur Filterung

stopListOMC

Mit diesem Parameter kann eine Gemeindekennziffer (GKZ bzw. OMC) angegeben werden. Es werden ausschließlich Haltestellen gefunden, die innerhalb der angegebenen Gemeinde liegen.

stopListPlaceld

Mit diesem Parameter kann die ID eines Ortes angegeben werden. Es werden ausschließlich Haltestellen gefunden, die in Orten mit der entsprechenden ID liegen. Sinnvoll ist die Kombination von GKZ und ID des Ortes.

stopListSubnetwork

Mit diesem Parameter kann ein Netz angegeben werden. Es werden ausschließlich Haltestellen gefunden, die von Verkehrsmitteln dieses Netzes bedient werden.

fromstop

Mit diesem Parameter kann eine Haltestellen ID angegeben werden, die die Untergrenze eines Intervalls darstellt. Es werden ausschließlich Haltestellen ausgegeben, die in dem Intervall liegen. Die Obergrenze des Intervalls wird durch den Parameter `tostop` angegeben.

tostop

Mit diesem Parameter kann eine Haltestellen ID angegeben werden, die die Obergrenze eines Intervalls darstellt. Es werden ausschließlich Haltestellen ausgegeben, die in dem Intervall liegen. Die Untergrenze des Intervalls wird durch den Parameter `fromstop` angegeben.

12.3 Parameter zur Ausgabe zusätzlicher Information

Bei der Ausgabe von Zusatzinformation muss beachtet werden, dass diese die Performance der Anfrage verschlechtert.

servingLines = 1

Dieser Parameter bewirkt, dass die bedienenden Linien für jede Haltestelle ausgegeben werden.

servingLinesMOTType = 1

Dieser Parameter bewirkt die Ausgabe des Hauptverkehrsmittels jeder Haltestelle. Er kann nicht in Kombination mit dem Parameter `servingLinesMOTTypes = 1` verwendet werden.

servingLinesMOTTypes = 1

Dieser Parameter bewirkt die Ausgabe aller Verkehrsmittel der Haltestellen. Die Verkehrsmittel werden pro Haltestelle als kommasetrennte Liste ausgegeben. Der Parameter kann nicht in Kombination mit dem Parameter `servingLinesMOTType = 1` verwendet werden.

tariffZones = 1

Dieser Parameter bewirkt, dass die Tarifzone für jede Haltestelle ausgegeben wird. Alternativzonen werden durch einen Schrägstrich getrennt mit ausgegeben.

13 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 lange sein können. Daher empfiehlt es sich nicht, die Meldungen vollständig auszugeben, sondern beispielsweise nur die Meldungen für das heutige Datum aktiven.

13.1 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
```

13.2 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.

13.3 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 Gemeindekennziffer und eine ID des Ortes bestimmt. Die beiden Werte werden durch einen Doppelpunkt getrennt angegeben. Der Parameter kann mehrfach verwendet werden.

13.4 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.

13.5 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.

13.6 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.

13.7 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.

14 SystemInfo-Request

Im Folgenden wird das Anfragen von Systeminformation (SystemInfo-Request) erläutert. Es wird die verwendete Programmversion ausgegeben, sowie Information darüber welches Format die Daten haben und wann diese zuletzt aktualisiert wurden.

14.1 Ansteuerung der Anfrage von Systeminformation

Die Anfrage von Systeminformation 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_SYSTEMINFO_REQUEST) oder in HTML (XSLT_SYSTEMINFO_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_SYSTEMINFO_REQUEST?HTTP-Parameter
```

Für eine HTML-Antwort ist folgende URL einzugeben:

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

14.2 Allgemeine Funktionalität

Der SystemInfo-Request verwendet die [Grundfunktionalität](#) und hat keine eigenen HTTP-Parameter.

15 Map-Request

Im Folgenden wird die Karten-Anfrage (Map-Request) erläutert. Durch die Kartenanfrage wird eine Karte generiert. Über HTTP-Parameter kann der gewünschte Kartenausschnitt gewählt, das Bildformat geändert, die Kartengröße definiert und Interaktion, wie Zoomen, Navigieren und das Auswählen von POIs oder Haltestellen, realisiert werden (siehe Abbildung 15). Außerdem können alle Objekte eines bestimmten Typs, wie zum Beispiel Haltestellen oder wichtige Punkte, innerhalb des von der Karte abgedeckten Gebiets ermittelt werden.



Abbildung 15 - Interaktive Karte mit dem Map-Request

Hinweis: Der Map-Request ist veraltet und wird nicht mehr gewartet oder weiterentwickelt. Mittlerweile stehen komfortable Plugins als Freeware zur Verfügung, die dank JavaScript und AJAX Karten mit besseren Navigationsmöglichkeiten bereitstellen. Um Objekte, wie beispielsweise POIs oder Haltestellen auf der Karte darzustellen, kann zwar weiterhin der Map-Request verwendet werden, jedoch empfiehlt sich die für diese Zwecke optimierte Koordinaten-Anfrage (Coord-Request).

15.1 Ansteuerung der Anfrage von Systeminformation

Die Karten-Anfrage 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_MAPREQUEST) oder in HTML (XSLT_MAP_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_MAP_REQUEST?HTTP-Parameter`

Für eine HTML-Antwort ist folgende URL einzugeben:

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

Die Karte selbst wird als Bild generiert. Standardmäßig im Format GIF, was aber über den Parameter `imageFormat` geändert werden kann. Der Link zur Karte ist in der XML-Ausgabe im Attribut `fileName` des Elements `itdMapRequest` hinterlegt.

15.2 Allgemeine Funktionalität

Der Karten-Request verwendet die Grundfunktionalität und außerdem folgende Parameter:

command

Mit diesem Parameter kann die Verarbeitung der Anfrage beeinflusst werden. Folgende Werte sind möglich:

- `directMap` - direkte Generierung der Karte
- `mapCenter` bzw. `mapSetODV` - Sprung ins Kartenzentrum
- `mapMove` - ermöglicht die Karte zu verschieben
- `mapRect` - fragt den durch die Parameter `xScreen`, `yScreen`, `xScreen2` und `yScreen2` aufgezogenen rechteckigen Kartenausschnitt an

createImage = 0

Standardmäßig wird bei der Karten-Anfrage eine Karte generiert. Mit diesem Parameter wird die Generierung unterdrückt.

mapName

Mit diesem Parameter wird das Koordinatenformat der Daten übergeben.

Beispiel: GIP, NAV3

mapNameOutput

Mit diesem Parameter wird das Koordinatenformat der Karte angegeben. Der Parameter ist unbedingt notwendig, wenn das Koordinatenformat, das für die Karten verwendet wird, nicht dem Koordinatenformat, in dem die Daten vorliegen, übereinstimmt.

Beispiel: GIP, NAV3

moveDirection

Mit diesem Parameter kann der Kartenausschnitt bewegt werden. Der Wert gibt die Bewegungsrichtung an. Die möglichen Werte werden in Abbildung 16 grafisch veranschaulicht.

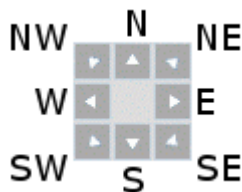


Abbildung 16 - Bewegungsrichtungen

omc

Mit diesem Parameter wird eine Gemeindekennziffer angegeben. Auf diese Weise ist es möglich eine Gemeinde direkt auf der Karte anzuspringen. Sie wird zentriert in der Kartenmitte dargestellt.

placeID

Mit diesem Parameter wird die Identifikationsnummer eines Ortes angegeben. Auf diese Weise ist es möglich einen Ort direkt auf der Karte anzuspringen. Er wird zentriert in der Kartenmitte dargestellt.

placeSelectionWithCoordinate

Dieser Parameter stellt eine verkürzende Schreibweise zur Anforderung einer Karte durch Kombination verschiedener Parameter dar. Sein Wert `<x>:<y>:<Zoomstufe>:<Breite>:<Höhe>:<Haltestellen>:<POIs>:<Koordinatenformat>` setzt sich aus den durch Doppelpunkt getrennten Parametern x-Koordinate des Kartenzentrums `<x>`, y-Koordinate des Kartenzentrums `<y>`, Zoomstufe `<Zoomstufe>`, Breite der Karte `<Breite>`, Höhe der Karte `<Höhe>`, Anzeige der Haltestellen `<Haltestellen>`, Anzeige der wichtigen Punkte `<POIs>` und dem Koordinatenformat `<Koordinatenformat>` zusammen. Die Werte für die Anzeige der Haltestellen `<Haltestellen>` bzw. wichtiger Punkte `<POIs>` können 0 oder 1 sein. 0 bedeutet dabei, dass die Haltestellen bzw. wichtigen Punkte nicht angezeigt werden; 1 bedeutet, dass sie angezeigt werden.

Beispiel: Der Parameter kann beispielsweise verwendet zu werden, wenn die Einstiegskarte eine statische Karte ohne anklickbare Bereiche darstellt. Die Wahl der Einstiegspunkte muss dann über eine Liste erfolgen, deren Einträge mittels des Parameters `placeSelectionWithCoordinate` gewählt werden können. Der Wert des Parameters kann zum Beispiel wie folgt sein:

4417528:801425:4:640:480:1:1:NAV4. Das Zentrum des Kartenausschnitts stellt die Koordinate (4417528, 801425) dar. Es wird die 4. Zoomstufe angezeigt und die Karte hat eine Größe von 640 px x 480 px. Sowohl die Haltestellen als auch die wichtigen Punkte werden auf der Karte eingezeichnet.

Hinweis: Werden einzelne der bereits im Parameter `placeSeletionWithCoordinate` enthaltenen Parameter zusätzlich separat übergeben, so werden diese nicht berücksichtigt, sondern durch den Parameter `placeSeletionWithCoordinate` überschrieben.

realHeight

Mit diesem Parameter wird die reale Höhe des dargestellten Kartenausschnitts in Metern angegeben. Dies ist nicht mit der Höhe der Karte zu verwechseln, die mit dem Parameter `screenSize` bzw. `screenHeight` angegeben wird.

Die reale Höhe spiegelt sich in der XML-Ausgabe im Attribut `realHeight` des Elements `itdMapRequest` wieder.

Hinweis: Um diesen Parameter verwenden zu können, ist die Angabe einer gültigen Zoomstufe, durch den Parameter `zoomLevel`, notwendig. Außerdem muss ebenfalls die reale Breite angegeben werden. Hierfür wird der Parameter `realWidth` verwendet.

realWidth

Mit diesem Parameter wird die reale Breite des dargestellten Kartenausschnitts in Metern angegeben. Dies ist nicht mit der Höhe der Karte zu verwechseln, die mit dem Parameter `screenSize` bzw. `screenWidth` angegeben wird.

Die reale Breite spiegelt sich in der XML-Ausgabe im Attribut `realWidth` des Elements `itdMapRequest` wieder.

Hinweis: Um diesen Parameter verwenden zu können, ist die Angabe einer gültigen Zoomstufe, durch den Parameter `zoomLevel`, notwendig. Außerdem muss ebenfalls die reale Höhe angegeben werden. Hierfür wird der Parameter `realHeight` verwendet.

screenSize

Mit diesem Parameter kann eine von drei vordefinierten Kartengrößen gewählt werden:

- `large` - 700 px x 525 px

- `medium` - 480 px x 360 px
- `small` - 300 px x 225 px

Standardmäßig ist die Kartengröße `medium` gewählt. In der XML-Ausgabe spiegelt sich der Wert des Parameters im Attribut `screenSize` des Elements `itdMapRequest` wieder.

Hinweis: Alternativ zur Wahl einer vordefinierten Größe kann die Breite und Höhe der Karte durch die Parameter `screenWidth` und `screenHeight` frei gewählt werden. Zu beachten ist, dass sie jedoch durch diesen Parameter überschrieben werden.

screenHeight

Mit diesem Parameter wird die Höhe der Karte angegeben. Dies ist nicht mit der Höhe des Kartenausschnitts zu verwechseln, die durch den Parameter `realHeight` angegeben wird.

Der Wert des Parameters spiegelt sich in der XML-Ausgabe im Attribut `screenHeight` des Elements `itdMapRequest` wieder.

Hinweis: Dieser Parameter kann durch den Parameter `screenSize` überschrieben werden. In diesem Fall wird die durch den Parameterwert gewählte feste Kartengröße verwendet.

screenWidth

Mit diesem Parameter wird die Breite der Karte angegeben. Dies ist nicht mit der Breite des Kartenausschnitts zu verwechseln, der durch den Parameter `realWidth` angegeben wird.

Der Wert des Parameters spiegelt sich in der XML-Ausgabe im Attribut `screenWidth` des Elements `itdMapRequest` wieder.

Hinweis: Dieser Parameter kann durch den Parameter `screenSize` überschrieben werden. In diesem Fall wird die durch den Parameterwert gewählte feste Kartengröße verwendet.

xCenterReal

Dieser Parameter gibt die reale x-Koordinate des Kartenzentrums an. Der Wert des Parameters spiegelt sich in der XML-Ausgabe im Attribut `xCenterReal` des Elements `itdMapRequest` wieder.

xScreen

Soll der angezeigte Kartenausschnitt über eine Boundingbox definiert werden, wird mit diesem Parameter die x-Koordinate der linken oberen Ecke angegeben.

Hinweis: Dieser Parameter kann nur in Kombination mit dem Parameter `command=mapRect` verwendet werden.

xScreen2

Soll der angezeigte Kartenausschnitt über eine Boundingbox definiert werden, wird mit diesem Parameter die x-Koordinate der rechten unteren Ecke angegeben.

Hinweis: Dieser Parameter kann nur in Kombination mit dem Parameter `command=mapRect` verwendet werden.

yCenterReal

Dieser Parameter gibt die reale y-Koordinate des Kartenzentrums an. Der Wert des Parameters spiegelt sich in der XML-Ausgabe im Attribut `yCenterReal` des Elements `itdMapRequest` wieder.

yScreen

Soll der angezeigte Kartenausschnitt über eine Boundingbox definiert werden, wird mit diesem Parameter die y-Koordinate der linken oberen Ecke angegeben.

Hinweis: Dieser Parameter kann nur in Kombination mit dem Parameter `command=mapRect` verwendet werden.

yScreen2

Soll der angezeigte Kartenausschnitt über eine Boundingbox definiert werden, wird mit diesem Parameter die y-Koordinate der rechten unteren Ecke angegeben.

Hinweis: Dieser Parameter kann nur in Kombination mit dem Parameter `command=mapRect` verwendet werden.

zoomLevel

Dieser Parameter gibt an, welche Zoomstufe der Karte angezeigt werden soll. Die angefragte Zoomstufe kann dem Attribut `zoomLevel` des Elements `itdMapRequest` der XML-Ausgabe entnommen werden.

Hinweis: Dieser Parameter kann nur in Kombination mit dem Parameter `command=mapRect` verwendet werden.

15.3 Kachel-Optionen

Alle folgenden Parameter können nur in Kombination mit dem Parameter `command=directMap` verwendet werden.

desiredRect

Mit diesem Parameter können Real Estate Karten erzeugt werden. Der Wert `<x>:<y>:<Koordinatenformat>:<halbe Höhe>:<halbe Breite>` setzt sich zusammen aus der Zentrumskoordinate, die durch die Parameter `<x>` und `<y>` angegeben wird, das Koordinatenformat, sowie die halbe Höhe und die halbe Breite des Gebiets bestimmt.

heightPixelsMax

Gibt eine maximale Höhe der Karte an. Letztendlich handelt es sich dabei jedoch nur an eine Empfehlung an den EFA IT-Kernel. Die tatsächliche Höhe kann abweichen.

heightPixelsMin

Gibt eine minimale Höhe der Karte an. Letztendlich handelt es sich dabei jedoch nur an eine Empfehlung an den EFA IT-Kernel. Die tatsächliche Breite kann am Ende abweichen.

imageFormat

Legt fest, in welchem Format die Karte erzeugt wird. Standardmäßig werden PDF-Dateien erzeugt. Ebenfalls möglich ist die Erzeugung von GIF-, JPG- und PNG-Dateien. Standard ist GIF. Mögliche Werte sind:

- GIF
- JPEG
- JPG
- PDF
- PNG

Hinweis: Das Format der Karte entspricht zwar dem durch diesen Parameter gewählten, die Endung der Datei ist jedoch immer `.pdf`.

imageNoTiles = false

Für den Individualtransport existieren aufgrund der fehlenden Umstiege an Haltestellen keine Umgebungspläne. Um die Fahrtübersicht übersichtlicher zu gestalten, werden Detailkarten erzeugt, die Ausschnitte der Karte vergrößern. Diese Funktionalität kann mit diesem Parameter deaktiviert werden.

imageOnly

Standardmäßig werden Bilder mit Zusatzinformation, wie Legenden oder Überschriften, ausgegeben. Dies kann man durch diesen Parameter abschalten. Es werden ausschließlich die Bilder ausgegeben.

itDestLegPoint/itStartLegPoint

Mit diesem Parameter wird der Steigname für den Ziel-/Startpunkt in der Überschrift ausgegeben. In der Regel ist der Steigname sinnvoll vorbelegt und muss nicht geändert werden.

Hinweis: Um diesen Parameter zu verwenden, muss der Parameter `itDestPath/itStartPath` aktiviert sein. Zusätzlich muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, wird der Parameter nicht unterstützt.

itDestLegStop/itStartLegStop

Mit diesem Parameter wird der Haltestellenname für den Ziel-/Startpunkt in der Überschrift ausgegeben. In der Regel ist der Haltestellenname sinnvoll vorbelegt und muss nicht geändert werden.

Hinweis: Um diesen Parameter zu verwenden, muss der Parameter `itDestPath/itStartPath` aktiviert sein. Zusätzlich muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, wird der Parameter nicht unterstützt.

itDestMOT/itStartMOT

Mit diesem Parameter wird der Verkehrsmitteltext für den Ziel-/Startpunkt in der Überschrift ausgegeben. In der Regel ist der Verkehrsmitteltext sinnvoll vorbelegt und muss nicht geändert werden.

Hinweis: Um diesen Parameter zu verwenden, muss der Parameter `itDestPath/itStartPath` aktiviert sein. Zusätzlich muss bereits eine Fahrt berechnet

sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, wird der Parameter nicht unterstützt.

itDestPath = 1/itStartPath = 1

Mit diesem Parameter werden die Parameter zur Modifizierung des Ziel-/Startpunktes in der Überschrift aktiviert.

Hinweis: Es ist nicht notwendig, dass der Parameter einen Wert hat. Sobald dieser vorhanden ist, können die entsprechenden Parameter verwendet werden.

levelOfDetail

Mit diesem Parameter können verschiedene Kartentypen ausgewählt werden, die eine unterschiedliche Detailgenauigkeit bieten. Folgende Werte sind möglich:

- `destpath` - Zielkarte mit der Strecke von der Zielhaltestelle zum Zielpunkt
- `overall` - Übersichtskarte über die gesamte Fahrtstrecke
- `partialpath` - Umstiegskarte (im Kartenmittelpunkt ist die Umstiegshaltestelle zu sehen)
- `startpath` - Startkarte mit der Strecke vom Startpunkt zur Starthaltestelle

Hinweis: Es muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Dieser Parameter wird für die Berechnung von Wegeketten, d.h. sobald Zwischenpunkte verwendet werden, nicht unterstützt.

ptTripLeg_<ID> = 1

Mit diesem Parameter wird die Teilstrecke mit der Identifikationsnummer `<ID>` ausgewählt, für die Modifizierungen in der Überschrift vorgenommen werden sollen.

Hinweis: Es ist nicht notwendig, dass der Parameter einen Wert hat. Sobald dieser vorhanden ist, können die entsprechenden Parameter verwendet werden. Die Modifizierung der Teilstrecken in der Überschrift muss durch den Parameter `ptTripPart` aktiviert sein. Zusätzlich muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Dieser Parameter wird bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, nicht mehr unterstützt.

ptTripLeg_<ID>_MOT

Mit diesem Parameter wird der Verkehrsmitteltext für die Teilstrecke mit der Identifikationsnummer <ID> in der Überschrift ausgegeben. In der Regel ist der Verkehrsmitteltext sinnvoll vorbelegt und muss nicht geändert werden.

Hinweis: Die Modifizierung der Teilstrecken in der Überschrift muss durch den Parameter `ptTripPart` aktiviert sein und die Teilstrecke durch den Parameter `ptTripLeg_<ID>`, mit identischem Wert <ID>, ausgewählt sein. Zusätzlich muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Dieser Parameter wird bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, nicht mehr unterstützt.

ptTripLeg_<ID>_Stop

Mit diesem Parameter wird der Haltestellenname für die Teilstrecke mit der Identifikationsnummer <ID> in der Überschrift ausgegeben. In der Regel ist der Haltestellenname sinnvoll vorbelegt und muss nicht geändert werden.

Hinweis: Die Modifizierung der Teilstrecken in der Überschrift muss durch den Parameter `ptTripPart` aktiviert sein und die Teilstrecke durch den Parameter `ptTripLeg_<ID>` ausgewählt sein. Zusätzlich muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Dieser Parameter wird bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, nicht mehr unterstützt.

ptTripPart = 1

Mit diesem Parameter werden die Parameter zur Modifizierung der Teilstrecken in der Überschrift aktiviert. Eine bestimmte Teilstrecke wird über den Parameter `ptTripLeg_<ID>` gewählt.

Hinweis: Es ist nicht notwendig, dass der Parameter einen Wert hat. Sobald dieser vorhanden ist, können die entsprechenden Parameter verwendet werden. Es muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionID` angegeben werden. Dieser Parameter wird bei der Verwendung von Wegeketten, d.h. sobald Zwischenpunkte vorhanden sind, nicht mehr unterstützt.

purposeSuffix

Mit diesem Parameter gibt man den Verwendungszweck einer angefragten Karte an. Der Verwendungszweck ist wichtig, um unterschiedliche Arten von Karten unterschiedlich konfigurieren zu können. Die Konfiguration erfolgt in der Konfigurationsdatei des EFA IT-Kernels. Der Verwendungszweck findet sich im Namen der Sektion

wieder. Er wird durch einen Punkt getrennt an die Bildart angehängt.

Die Angabe von mehreren Verwendungszwecken ist möglich. Diese werden durch einen Punkt getrennt aneinandergehängt und bilden eine hierarchische Ordnung. Weiter hinten stehende Verwendungszwecke sind den weiter vorne stehenden untergeordnet. Auf die gleiche Weise können mehrere Verwendungszwecke als Wert des Parameters angegeben werden.

serverCopyright2Hint = 1

Durch diesen Parameter wird eine zweite Copyright-Zeile ausgegeben, die rechts neben der Karte steht. Der Text des Copyrights wird in der Konfigurationsdatei des EFA IT-Kernels definiert. Dazu dient der Parameter `Copyright2` in der Sektion `[Maps]`.

tripPartFocusIndex

Dieser Parameter gibt den Index der Teilstrecke an, für die eine Karte ausgegeben werden soll. Die erste Teilstrecke hat den Index 0.

Hinweis: Dieser Parameter wird für die Berechnung von Wegeketten, d.h. sobald Zwischenpunkte verwendet werden, nicht unterstützt. Es muss bereits eine Fahrt berechnet sein und die Sitzung durch den Parameter `sessionId` angegeben werden.

widthPixelsMax

Gibt eine maximale Breite der Karte an. Letztendlich handelt es sich dabei jedoch nur an eine Empfehlung an den EFA IT-Kernel. Die tatsächliche Breite kann am Ende abweichen.

widthPixelsMin

Gibt eine minimale Breite der Karte an. Letztendlich handelt es sich dabei jedoch nur an eine Empfehlung an den EFA IT-Kernel. Die tatsächliche Breite kann am Ende abweichen.

16 Anhang

16.1 Verkehrsmitteltypen

Belegung der Standard Verkehrsmittel-IDs:

ID	Verkehrsmittel
0	Zug
1	S-Bahn
2	U-Bahn
3	Stadtbahn
4	Straßenbahn
5	Stadtbus
6	Regionalbus
7	Schnellbus
8	Seil-/Zahnradbahn
10	AST/Rufbus
11	Schwebebahn
12	Flugzeug
13	Regionalzug (z.B. IRE, RE und RB)
14	Nationaler Zug (z.B. IR und D)
15	Internationale Zug (z.B. IC und EC)
16	Hochgeschwindigkeitszüge (z.B. ICE)
17	Schienenersatzverkehr
18	Schuttlezug
19	Bürgerbus

Kundenspezifische Verkehrsmittel können von der Standardbelegung abweichen.

16.2 Objekttypen

Bei der Suche nach Objekten ist manchmal die Angabe des Objekttyps notwendig. Bei der [Punktverifikation](#) ist diese optional, kann aber als zusätzliches Filterkriterium verwendet werden. Folgende Objekttypen sind möglich:

Objekttyp	ID (Dezimalwert)	Beschreibung
-	0	Kein Filter aktiv. Suche im kompletten Suchraum.
PLACE	1	Suche in allen Orten des abgedeckten GIS-Gebietes.
STOP	2	Suche innerhalb aller im abgedeckten GIS-Gebiet erfassten Haltestellen IDs und Alias-Namen für Haltestellen.
STREET	4	Suche innerhalb aller im abgedeckten GIS-Gebiet definierten Straßennamen.
ADDRESS	8	Suche innerhalb aller im abgedeckten GIS-Gebiet definierten Adressen.
CROSSING	16	Suche innerhalb aller im abgedeckten GIS-Gebiet erfassten Kreuzungen.
POI	32	Suche innerhalb aller im abgedeckten GIS-Gebiet definierten IDs und Alias-Namen wichtiger Punkte.
POSTCODE	64	Suche innerhalb aller im abgedeckten GIS-Gebiet erfassten Postleitzahlen.

Die IDs der Objekttypen sind so angelegt, dass sie als Bitmaske verwendet werden können. Auf diese Weise ist es möglich die Objekttypen beliebig zu kombinieren. Soll nach keinem speziellen Objekttyp gefiltert werden, muss 0 angegeben werden. Die Kombination von mehreren Objekttypen kann bei der Punktverifikation eingesetzt werden. Für die Suche nach Betreibern oder Linien macht sie keinen Sinn.

Beispiel: Die Objekttypen zur Punktverifikation werden beispielsweise als Wert des Parameters `anyObjFilter_usage` in Form einer Bitmaske zur Filterung des Suchraums übergeben. Soll eine Filterung nach Adressen, Straßen und Haltestellen erfolgen, muss der Filter den Wert 14 (8+4+2) haben.

17 Glossar

17.1 HTTP-Parameter Makros

HTTP-Parameter Makros werden in der Konfigurationsdatei des EFAControllers definiert. Sie fassen mehrere HTTP-Parameter in einem einzigen zusammen. Name und Wert des Makros können in der Konfigurationsdatei frei gewählt werden.

Die Verwendung von Makros bietet sich insbesondere dann an, wenn kundenspezifische Kriterien für eine Suchanfrage oder Berechnung vorliegen. Ein Beispiel sind die Kriterien für die Punktsuche. Diese werden in der Regel durch eine Kombination mehrerer Parameter abgebildet und als Makro abgelegt. Ändern sich die Kriterien, wird das Makro in der Konfigurationsdatei angepasst. Eine Anpassung in der Benutzeroberfläche ist nicht notwendig, was insbesondere dann von Vorteil ist, wenn die Kriterien für mehrere gelten. Ein weiterer Vorteil von Makros ist, dass durch die Zusammenfassung mehrerer HTTP-Parameter zu einem Parameter die Anzahl der mitgeschickten Parameter bzw. die URLs für die Requests kürzer werden.