



Schnittstellenbeschreibung

XML

Stand: Dienstag, 19. Januar 2021



EINLEITUNG

Die KOMMSVZ-Schnittstelle organisiert den plattformunabhängigen Austausch der Grunddaten des KOMMSVZ-Modells und kommt überall dort zum Einsatz, wo KOMMSVZ praktiziert wird. Dabei wird die Schnittstelle unabhängig von der Datenstruktur und Datenhaltung dort eingerichtet, wo die jeweilige Verantwortung über die Strassen- und Wegedaten obliegt.

Die Schnittstelle für das kommunale Strassen- und Wegeverzeichnis in M-V wird aktuell über diese URL erreicht:

<https://web.btfietz.de/interfaces/kommsvz/xml.php>

Um die grundsätzlichen KOMMSVZ-Daten abzufragen bzw. zu speichern, gibt es eine Reihe von Argumenten, die an die URL als so genannten Query String angehängt werden müssen.

Der Aufbau der Schnittstelle ist damit eine Art Baukastensystem. Je nachdem was benötigt wird, kann angehängt werden.

Dabei gibt es je nach Abfragetyp Pflichtargumente und optionale Argumente.

Die Pflichtargumente müssen in jedem Fall angehängt und mit einem Wert versehen werden.

Die optionalen Argumente können angehängt werden und müssen nicht zwingend einen Wert erhalten.

Alle Argumente bzw. Bezeichner sind case-sensitiv, d.h. sie müssen genauso verwendet werden, wie sie hier beschreiben sind

Daraus ergeben sich die einzelnen Bestandteile des Baukastens:

- U = URL (<https://web.btfietz.de/interfaces/kommsvz/xml.php>)
- S = Nutzer- und Verbindungsdaten, bei allen Abfragen immer notwendig und gleich pro Quelle, die genutzt werden soll – `para[source_id]= testArea`

BEACHTE:

Diese Zugangsdaten werden einem Anwender der Schnittstelle vom jeweiligen Betreiber unter Berücksichtigung des Datenschutzes vertraulich auf Anfrage bereitgestellt.

- A = Abfragetyp (boolescher Wert) (z.B. Daten für einen Straßenabschnitt – `get[roadlinkdata]=1`)
- P = Pflichtargumente + Wert (z.B. eine RoadLinkId – `arg[rdl_id][]=31391`)
 - Bemerge:
Alle Argumente mit leeren eckigen Klammern am Ende können mehrfach in der Abfrage verwendet werden. Diese Daten werden als Array übergeben und werden einzeln ausgewertet.
- O = Optionale Argumente + Wert (z.B. für die Orientierung von Bildbefahrungsdaten – `arg[orientation]=P`)

Alle Ergebnisse der Abfragen werden in XML-Datenstrukturen zurückgegeben, für diese jeweils Dokumenttypdefinitionen (dtd) vorliegen. Diese sind im Anhang des Dokumentes zu finden.

Bei den Ergebnissen der Abfragen werden dort, wo es KOMMSVZ vorschreibt, Codes bzw. Schlüssel zurückgegeben. D.h. eine Entschlüsselung muss extern in der Applikation durchgeführt werden. In optimaler Weise sollte dabei auf die Codelisten der jeweiligen Verwaltungs- und Fachinstitutionen zugegriffen werden.

Als Beispiel sei hier die Entschlüsselung der KOMMSVZ-Widmungen erwähnt, die über die Codeliste [get_functionalroadclasscodelist](#) (Kapitel 6.3) vom anfragenden System decodiert werden kann. Alle KOMMSVZ-spezifischen Codelisten werden im Kapitel 6 beschrieben.

Für die Liegenschaftskataster-Codes bzw. Schlüssel, wie z.B. dem Gemeindegeschlüssel (`communecode`), gibt es noch keine offiziellen Codelisten.

Eine Fehlerüberprüfung ist in die Schnittstelle integriert. Wird eine fehlerhafte Abfrage gestellt oder eine Abfrage liefert ein fehlerhaftes Ergebnis, werden diese Fehler ebenfalls in XML zurückgegeben und können ausgewertet werden.

Schwerwiegende Fehler bei der Abfrage der Schnittstelle, wie z.B. die falsche Angabe der Nutzer- und

Verbindungsdaten werden mit dem Fehler-Tag [errorxmlcall](#) in einer unabhängigen Fehlerbeschreibung ausgegeben (siehe `kommsvz_errorxmlcall.dtd` im Anhang – Kapitel 7.1.7).
Alle anderen Fehler werden in Zusammenhang mit dem Abfragetyp ausgegeben und sind in der jeweiligen dtd zur Abfrage definiert.

Im Folgenden werden die einzelnen Abfragetypen erläutert und mit Ihren Parametern gelistet.

INHALTSVERZEICHNIS

1. Abfragen von Bestandsdaten zu einem Straßenabschnitt	6
1.1. Bestandsdaten zum Straßenabschnitt (get_roadlinkdata)	6
1.1.1. Alle Bestandsdaten zum Straßenabschnitt (arg_rdl_id)	6
1.1.1.1. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Straßenabschnitt	6
1.1.1.2. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Hausnummern	6
1.1.1.3. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Verwaltung	6
1.1.1.4. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Verknüpfungspunkte	6
1.1.1.5. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Querschnitte	7
1.1.1.6. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Ortsteile	7
1.1.1.7. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Merkmale	7
1.1.2. Aktuelle Gültigkeit-Bestandsdaten zu einem Straßenabschnitt (arg_roadlinklifespan)	7
1.1.2.1. Gültigkeit für das KOMMSVZ-Element Straßenabschnitt (arg_roadlinkid)	7
1.1.2.2. Gültigkeit für das KOMMSVZ-Element Hausnummern (arg_roadlinkaddressid)	7
1.1.2.3. Gültigkeit für das KOMMSVZ-Element Verwaltung (arg_roadlinkadministrationid)	7
1.1.2.4. Gültigkeit für das KOMMSVZ-Element Verknüpfungspunkte (arg_roadlinkconnectionnodesid)	8
1.1.2.5. Gültigkeit für das KOMMSVZ-Element Querschnitte (arg_roadlinkcrosssectionid)	8
1.1.2.6. Gültigkeit für das KOMMSVZ-Element Ortsteile (arg_roadlinkdistrictid)	8
1.1.2.7. Gültigkeit für das KOMMSVZ-Element Merkmale (arg_roadlinkpropertiesid)	8
1.1.3. Nächste freie Abschnittsnummer (arg_nextroadlinknr)	8
2. Abfragen von speziellen Bestandsdaten zu einem Straßenabschnitt	9
2.1. Verknüpfungs-Bestandsdaten zum Straßennetzknoten (get_roadnodeconnectdata)	9
3. Abfragen von Listen in Bezug auf vorhandene Straßenabschnitte	10
3.1. Liste Gemeinden (arg_getcommune)	10
3.2. Liste Ortsteile (arg_getdistrict)	10
3.3. Liste Straßen (arg_getroad)	10
3.3.1. Alle Straßen zu bestimmten Gemeinden (arg_communeid)	10
3.3.2. Alle Straßen zu einer Gemeinde und bestimmten Ortsteilen (arg_districtid)	11
3.4. Liste Straßenabschnitte (arg_getroadlink)	11
3.4.1. Alle Straßenabschnitte zu einer Gemeinde und bestimmten Straßen	11
3.4.2. Alle Straßenabschnitte zu einem Ortsteil einer Gemeinde und bestimmten Straßen	11
4. Speichern der KOMMSVZ-Bestandsdaten (set_roadlinksave)	12
4.1. Speichertyp (arg_type)	12
4.2. KOMMSVZ-Element (arg_tab)	12
4.3. Identifizierung (arg_id)	12
4.4. Bestandsdaten der KOMMSVZ-Elemente	13
4.4.1. KOMMSVZ-Element Straßenabschnitt	13
4.4.2. KOMMSVZ-Element Hausnummern	13
4.4.3. KOMMSVZ-Element Verwaltung	14
4.4.4. KOMMSVZ-Element Verknüpfungspunkte	14
4.4.5. KOMMSVZ-Element Querschnitte	14
4.4.6. KOMMSVZ-Element Ortsteile	14
4.4.7. KOMMSVZ-Element Merkmale	14
4.4.8. KOMMSVZ-Element Bildbefahrung	14
4.5. Beispiele für Speicherabfragen für das KOMMSVZ-Element Verwaltung	15
4.5.1. Einfügen (insert)	15
4.5.2. Aktualisieren (update)	15
4.5.3. Löschen (delete)	15

5. Weitere Abfragen	16
5.1. Abfrage der aktuellen Serverzeit (get_servertime)	16
5.2. Bildbefahrungs-Bestandsdaten zum Straßenabschnitt (get_roadlinkpictures)	16
5.2.1. Alle Bildbefahrungs-Bestandsdaten (arg_all)	16
5.2.2. Bildbefahrungs-Bestandsdaten zum Straßenabschnitt (arg_rdl_id)	16
5.2.3. Bildbefahrungs-Bestandsdaten zum Straßenabschnitt mit einer Orientierung (arg_orientation)	17
5.3. Bestandsdaten zum Straßennetzknoten aus dem ZKM (get_roadnodedata)	17
5.3.1. Alle Bestandsdaten zum Straßennetzknoten mit einer NetzknotenId (arg_rdn_id)	17
5.3.2. Alle Bestandsdaten zum Straßennetzknoten mit einer Netzknottennummer (arg_rdn_code)	17
6. Datendienste für Codelisten	18
6.1. Codeliste für KOMMSVZ-Fahrtrichtungen (get_directioncodelist)	18
6.2. Codeliste für KOMMSVZ-Straßentypen (alt: -arten) (get_formofwaycodelist)	18
6.3. Codeliste für KOMMSVZ-Straßengruppen (get_functionalroadgroupcodelist)	18
6.4. Codeliste für KOMMSVZ-Straßenarten (alt: -Widmungen) (get_functionalroadclasscodelist)	19
6.5. Codeliste für KOMMSVZ-Straßenmeistereien (get_maintenanceauthoritycodelist)	19
6.6. Codeliste für KOMMSVZ-Verknüpfungspunktarten (get_nodecrossreferencecodelist)	19
6.7. Codeliste für KOMMSVZ-Straßenservicebereiche (get_noderoadserviceareacodelist)	19
6.8. Codeliste für KOMMSVZ-Baulasten (get_ownerauthoritycodelist)	20
6.9. Codeliste für KOMMSVZ-Baulastenträger Dritter (get_ownerauthorityagencycodelist)	20
6.10. Codeliste für KOMMSVZ-Ortsdurchfahrten (get_roadmainthroughcodelist)	20
6.11. Codeliste für KOMMSVZ-Belagsarten (get_roadsurfacecategorycodelist)	20
6.12. Codeliste für KOMMSVZ-Querschnittsstreifenart (get_stripetypecodelist)	21
7. Anhang	22
7.1. Dokumententypdefinitionen (dtd)	22
7.1.1. kommsvz_roadlink.dtd	22
7.1.2. kommsvz_roadlinkbrowser.dtd	23
7.1.3. kommsvz_roadlinksave.dtd	23
7.1.4. kommsvz_nextroadlinknumber.dtd	23
7.1.5. kommsvz_roadnodeconnect_roadlink.dtd	24
7.1.6. kommsvz_timestamp.dtd	24
7.1.7. kommsvz_errorxmlcall.dtd	24

1. Abfragen von Bestandsdaten zu einem Straßenabschnitt

1.1. Bestandsdaten zum Straßenabschnitt (get_roadlinkdata)

Alle oder aktuelle Bestandsdaten zu einem oder mehreren Straßenabschnitten bzw. deren sekundären Bestandsdaten abfragen.
In weiteren Pflichtargumenten wird festgelegt, welche Bestandsdaten abgefragt werden sollen.

A get[roadlinkdata]=1

1.1.1. Alle Bestandsdaten zum Straßenabschnitt (arg_rdl_id)

Alle Bestandsdaten zu einem oder mehreren Straßenabschnitten abfragen.
Abfrage über eine oder mehrere Straßenabschnitts-Id.
Die Straßenabschnitts-Id ist eine interne, eindeutige Id innerhalb des jeweiligen Datenquellenverzeichnisses (Datenbank). Die Id ist nicht das Straßenabschnittskennzeichen.
Die Straßenabschnitts-Id erhält man derzeit über die Liste Straßenabschnitte (Kapitel 3.4) oder als Ergebnis einer Selektion im KOMMSVZ-Webdienst (Kapitel 4.4.1).

dtd: kommsvz_roadlink.dtd (Kapitel 7.1.1)

P arg[rdl_id][]=DEMVKS59c4e98ceca7b	gml_id
O arg[datatab]=main	wenn separate Datentabelle angegeben wird dann werden nur Daten für diese zurückgegeben, ansonsten alle
O arg[mb_date]=...	(timestamp(o) with time zone)

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionales Argument + Wert) für alle Bestandsdaten:
[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVKS59c4e98ceca7b&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVKS59c4e98ceca7b&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.1. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Straßenabschnitt

O arg[datatab]=main

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):
[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVKS59c4e98ceca7b&arg\[datatab\]=main&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVKS59c4e98ceca7b&arg[datatab]=main&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.2. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Hausnummern

O arg[datatab]= adress

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):
[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVKS59c4e98ceca7b&arg\[datatab\]=adress&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVKS59c4e98ceca7b&arg[datatab]=adress&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.3. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Verwaltung

O arg[datatab]= admin

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):
[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVKS59c4e98ceca7b&arg\[datatab\]=admin&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVKS59c4e98ceca7b&arg[datatab]=admin&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.4. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Verknüpfungspunkte

O arg[datatab]= connect

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):
[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVKS59c4e98ceca7b&arg\[datatab\]=connect&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVKS59c4e98ceca7b&arg[datatab]=connect&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.5. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Querschnitte

O arg[datatab]= cross

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVK559c4e98ceca7b&arg\[datatab\]=cross&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVK559c4e98ceca7b&arg[datatab]=cross&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.6. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Ortsteile

O arg[datatab]= district

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVK559c4e98ceca7b&arg\[datatab\]=district&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVK559c4e98ceca7b&arg[datatab]=district&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.1.7. Alle Bestandsdaten zum Straßenabschnitt für das KOMMSVZ-Element Merkmale

O arg[datatab]=prop

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + optionale Argumente + Werte):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[rdl_id\]\[\]=DEMVK559c4e98ceca7b&arg\[datatab\]=prop&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[rdl_id][]=DEMVK559c4e98ceca7b&arg[datatab]=prop&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

1.1.2. Aktuelle Gültigkeit-Bestandsdaten zu einem Straßenabschnitt (arg_roadlinklifespan)

Aktuelle Gültigkeit-Bestandsdaten zu einem Straßenabschnitt bzw. deren sekundären Bestandsdaten abfragen.

Wird die Abfrage nur mit arg[roadlinklifespan][]=1 gestellt, dann wird die aktuellste Gültigkeit aller Straßenabschnitte zurückgegeben.

Anhand einer zusätzlich übergebenen ID wird festgelegt, für welche Daten dessen Gültigkeit abgefragt wird.

In der Antwort wird das Datum der letzten Änderung zurückgegeben.

dtd: kommsvz_timestamp.dtd (Kapitel 7.1.6) – gültig für alle Unterkapitel

P arg[roadlinklifespan][]=1

1.1.2.1. Gültigkeit für das KOMMSVZ-Element Straßenabschnitt (arg_roadlinkid)

P arg[roadlinkid][]=DEMVK559c4e98ceca7b

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinklifespan\]\[\]=1&arg\[roadlinkid\]\[\]=DEMVK559c4e98ceca7b](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinklifespan][]=1&arg[roadlinkid][]=DEMVK559c4e98ceca7b)

1.1.2.2. Gültigkeit für das KOMMSVZ-Element Hausnummern (arg_roadlinkaddressid)

Der zu übergebene Wert für roadlinkaddressid ist die gml_id des Roadlinks.

P arg[roadlinkaddressid][]=DEMVK559c4e98ceca7b

P arg[location]=10

Station des Straßenabschnitts

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinklifespan\]\[\]=1&arg\[roadlinkaddressid\]\[\]=DEMVK559c4e98ceca7b&arg\[location\]=10](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinklifespan][]=1&arg[roadlinkaddressid][]=DEMVK559c4e98ceca7b&arg[location]=10)

1.1.2.3. Gültigkeit für das KOMMSVZ-Element Verwaltung (arg_roadlinkadministrationid)

Der zu übergebene Wert für roadlinkadministrationid ist die gml_id des Roadlinks.

P arg[roadlinkadministrationid][]=DEMVK559c4e98ceca7b

P arg[location]=10

Station des Straßenabschnitts

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinklifespan\]\[\]=1&arg\[roadlinkadministrationid\]\[\]=DEMVK559c4e98ceca7b&arg\[location\]=10](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinklifespan][]=1&arg[roadlinkadministrationid][]=DEMVK559c4e98ceca7b&arg[location]=10)

1.1.2.4. Gültigkeit für das KOMMSVZ-Element Verknüpfungspunkte (arg_roadlinkconnectionnodesid)

Aktuelle Gültigkeit-Bestandsdaten für die KOMMSVZ-Datentabelle roadlinkConnectionNodes, in der alle Verknüpfungs- bzw. Übergangsknoten (in andere Transportnetze, klassifizierte und kommunale Straßennetze) stationär enthalten sind. Der zu übergebene Wert für roadlinkconnectionnodesid ist die gml_id des Roadlinks.

P arg[roadlinkconnectionnodesid][]=DEMVK59c4e98ceca7b
P arg[location]=10 Station des Straßenabschnitts

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfiget.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinkifespanspan\]=1&arg\[roadlinkconnectionnodesid\]\[\]=DEMVK59c4e98ceca7b&arg\[location\]=10](https://web.btfiget.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinkifespanspan]=1&arg[roadlinkconnectionnodesid][]=DEMVK59c4e98ceca7b&arg[location]=10)

1.1.2.5. Gültigkeit für das KOMMSVZ-Element Querschnitte (arg_roadlinkcrosssectionid)

Der zu übergebene Wert für roadlinkcrosssectionid ist die gml_id des Roadlinks.

P arg[roadlinkcrosssectionid][]=DEMVK59c4e98ceca7b
P arg[location]=10 Station des Straßenabschnitts

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfiget.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinkifespanspan\]=1&arg\[roadlinkcrosssectionid\]\[\]=DEMVK59c4e98ceca7b&arg\[location\]=10](https://web.btfiget.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinkifespanspan]=1&arg[roadlinkcrosssectionid][]=DEMVK59c4e98ceca7b&arg[location]=10)

1.1.2.6. Gültigkeit für das KOMMSVZ-Element Ortsteile (arg_roadlinkdistrictid)

Der zu übergebene Wert für roadlinkdistrictid ist die gml_id des Roadlinks.

P arg[roadlinkdistrictid][]=DEMVK59c4e98ceca7b

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfiget.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinkifespanspan\]=1&arg\[roadlinkdistrictid\]\[\]=DEMVK59c4e98ceca7b](https://web.btfiget.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinkifespanspan]=1&arg[roadlinkdistrictid][]=DEMVK59c4e98ceca7b)

1.1.2.7. Gültigkeit für das KOMMSVZ-Element Merkmale (arg_roadlinkpropertiesid)

Der zu übergebene Wert für roadlinkpropertiesid ist die gml_id des Roadlinks.

P arg[roadlinkpropertiesid][]= DEMVK59c4e98ceca7b
P arg[location]=10 Station des Straßenabschnitts

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfiget.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[roadlinkifespanspan\]=1&arg\[roadlinkpropertiesid\]\[\]=DEMVK59c4e98ceca7b&arg\[location\]=10](https://web.btfiget.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[roadlinkifespanspan]=1&arg[roadlinkpropertiesid][]=DEMVK59c4e98ceca7b&arg[location]=10)

1.1.3. Nächste freie Abschnittsnummer (arg_nextroadlinknr)

Nächste freie Abschnittsnummer für eine Straße innerhalb einer Gemeinde abfragen.

dtd: kommsvz_nextroadlinknumber.dtd (Kapitel 7.1.4)

P arg[arg_nextroadlinknr]=1
P arg[communecode]= 13074050
P arg[roadcode]=00104

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + ...):

[https://web.btfiget.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkdata\]=1&arg\[arg_nextroadlinknr\]=1&arg\[communecode\]=13074050&arg\[roadcode\]=00104](https://web.btfiget.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkdata]=1&arg[arg_nextroadlinknr]=1&arg[communecode]=13074050&arg[roadcode]=00104)

2. Abfragen von speziellen Bestandsdaten zu einem Straßenabschnitt

2.1. Verknüpfungs-Bestandsdaten zum Straßennetzknoten (get_roadnodeconnectdata)

Ermittelt zu einem oder mehreren Straßennetzknoten alle Verknüpfungs-Bestandsdaten, die sich innerhalb des KOMMSVZ-Modells aus den Bestandsdaten der Straßenabschnitte ableiten lassen.

D. h. zu einem wird geprüft, ob ein Netzknoten innerhalb von Straßenabschnitten Stationen (locations) bildet, so dass der Netzknoten ein Verknüpfungs- bzw. Übergangsknoten in andere Netze ist (siehe die KOMMSVZ-Datentabelle roadlinkConnectionNodes). Diese Prüfung erfolgt nur, wenn kommunale Straßennetze abgefragt werden.

Zum anderen werden alle Straßenabschnitte zurückgegeben, die an den Netzknoten innerhalb des abgefragten Straßennetzes anschließen. Hierbei wird in der Rückgabe die Station (location) leer gelassen.

Dabei gibt es 2 Pflichtfelder. Zum einen die Straßengruppe des Straßennetzes (roadgroup), das abgefragt wird, und zum anderen die Nummer des Netzknosens (rdn_code).

dtd: kommsvz_roadnodeconnect_roadlink.dtd (Kapitel 7.1.5)

A get[roadnodeconnectdata]=1

P arg[roadgroup]=G (siehe Codeliste für KOMMSVZ-Straßengruppen (Kapitel 6.3))

P arg[rdn_code][]= 439833940058

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadnodeconnectdata\]=1&arg\[roadgroup\]=G&arg\[rdn_code\]\[\]=439833940058](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadnodeconnectdata]=1&arg[roadgroup]=G&arg[rdn_code][]=439833940058)

3. Abfragen von Listen in Bezug auf vorhandene Straßenabschnitte

Diese Abfragen dienen dazu, um Listen von Bestandsdaten unter einer bestimmten Gruppierung/Thematik zu erstellen.

Bei jeder Abfrage besteht eine direkte Abhängigkeit zum Straßenabschnitt, d. h. im Ergebnis werden nur die Bestandsdaten gelistet, zu denen Straßenabschnitte vorhanden sind.

Die in den Abfragen verwendeten amtlichen Codes beziehen sich alle auf die amtlichen Daten von MV, die bisher nicht in Codelisten organisiert sind.

dtd: kommsvz_roadlinkbrowser.dtd (Kapitel 7.1.2) – gültig für alle Unterkapitel

A get[roadlinkbrowser]=1

3.1. Liste Gemeinden (arg_getcommune)

Alle Gemeinden zu einer oder mehreren Verwaltungen (departmentcode) abfragen.

Es werden nur Gemeinden gelistet, die auch aktuelle Straßenabschnitte enthalten.

P arg[getcommune]=1
P arg[departmentcode][]=5455 (Code aus der amtlichen Verwaltungs-Liste von MV)
O arg[mb_date]=... (timestamp(o) with time zone)
O arg[picvisit]=... (o oder 1, bei 1 nur Gem. mit Befahrungsbildern)

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkbrowser\]=1&arg\[getcommune\]=1&arg\[departmentcode\]\[\]=5455&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkbrowser]=1&arg[getcommune]=1&arg[departmentcode][]=5455&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

3.2. Liste Ortsteile (arg_getdistrict)

Alle Ortsteile zu einer oder mehreren Gemeinden (communecode) abfragen.

Es werden nur Ortsteile gelistet, die auch aktuelle Straßenabschnitte enthalten.

Straßenabschnitte, die keinem Ortsteil zugeordnet sind, werden unter „Ortsteil unbekannt“ gruppiert, für den es einen internen definierten Code gibt.

P arg[getdistrict]=1
P arg[communecode][]= 13074050 (Code aus der amtlichen Gemeinde-Liste von MV)
O arg[mb_date]=... (timestamp(o) with time zone)
O arg[picvisit]=... (o oder 1, bei 1 nur Ortst. mit Befahrungsbildern)

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkbrowser\]=1&arg\[getdistrict\]=1&arg\[communecode\]\[\]=13074050&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkbrowser]=1&arg[getdistrict]=1&arg[communecode][]=13074050&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

3.3. Liste Straßen (arg_getroad)

Alle Straßen zu einer oder mehreren Gemeinden (communecode) oder zu einer Gemeinde und einem oder mehreren in der Gemeinde befindlichen Ortsteil/e (districtcode) abfragen.

Es werden nur Straßen gelistet, die auch aktuelle Straßenabschnitte enthalten.

P arg[getroad]=1

3.3.1. Alle Straßen zu bestimmten Gemeinden (arg_communecode)

P arg[communecode][]= 13074050 (Code aus der amtlichen Gemeinde-Liste von MV)
O arg[mb_date]=... (timestamp(o) with time zone)
O arg[picvisit]=... (o oder 1, bei 1 nur Str. mit Befahrungsbildern)

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkbrowser\]=1&arg\[getroad\]=1&arg\[communecode\]=13074050&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkbrowser]=1&arg[getroad]=1&arg[communecode]=13074050&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

3.3.2. Alle Straßen zu einer Gemeinde und bestimmten Ortsteilen (arg_districtcode)

P	arg[communecode]= 13074050	(Code aus der amtlichen Gemeinde-Liste von MV)
P	arg[districtcode][]=001	(Code aus der amtlichen Ortsteile-Liste von MV)
O	arg[mb_date]=...	(timestamp(o) with time zone)
O	arg[picvisit]=...	(o oder 1, bei 1 nur Str. mit Befahrungsbildern)

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkbrowser\]=1&arg\[getroadlink\]=1&arg\[communecode\]=13074050&arg\[districtcode\]\[\]=001&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkbrowser]=1&arg[getroadlink]=1&arg[communecode]=13074050&arg[districtcode][]=001&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

3.4. Liste Straßenabschnitte (arg_getroadlink)

Alle Straßenabschnitte zu einer oder mehreren Straßen (roadcode) abfragen, die sich in einer oder mehreren Gemeinden (communecode) befinden oder in einem oder mehreren Ortsteil/en (districtcode) einer Gemeinde liegen.

Es werden nur Straßen gelistet, die auch aktuelle Straßenabschnitte enthalten.

P arg[getroadlink]=1

3.4.1. Alle Straßenabschnitte zu einer Gemeinde und bestimmten Straßen

P	arg[communecode]= 13074050	(Code aus der amtlichen Gemeinde-Liste von MV)
P	arg[roadcode][]=00104	(Code aus der amtlichen Straßen-Liste von MV)
O	arg[mb_date]=...	(timestamp(o) with time zone)
O	arg[picvisit]=...	(o oder 1, bei 1 nur Abs. mit Befahrungsbildern)

Beispiel (URL + Abfragetyp + Pflichtargumente + Werte):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkbrowser\]=1&arg\[getroadlink\]=1&arg\[communecode\]=13074050&arg\[roadcode\]\[\]=00104&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkbrowser]=1&arg[getroadlink]=1&arg[communecode]=13074050&arg[roadcode][]=00104&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

3.4.2. Alle Straßenabschnitte zu einem Ortsteil einer Gemeinde und bestimmten Straßen

P	arg[communecode]= 13074050	(Code aus der amtlichen Gemeinde-Liste von MV)
P	arg[districtcode]=001	(Code aus der amtlichen Ortsteile-Liste von MV)
P	arg[roadcode][]=00104	(Code aus der amtlichen Straßen-Liste von MV)
O	arg[mb_date]=...	(timestamp(o) with time zone)
O	arg[picvisit]=...	(o oder 1, bei 1 nur Abs. mit Befahrungsbildern)

Beispiel (URL + Abfragetyp + Pflichtargumente + Werte):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkbrowser\]=1&arg\[getroadlink\]=1&arg\[communecode\]=13074050&arg\[districtcode\]=001&arg\[roadcode\]\[\]=00104&arg\[mb_date\]=2017-09-22%2014:41:57.345221+02](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkbrowser]=1&arg[getroadlink]=1&arg[communecode]=13074050&arg[districtcode]=001&arg[roadcode][]=00104&arg[mb_date]=2017-09-22%2014:41:57.345221+02)

4. Speichern der KOMMSVZ-Bestandsdaten (set_roadlinksave)

Mit der KOMMSVZ-Schnittstelle können KOMMSVZ-Bestandsdaten zu einem Straßenabschnitt gespeichert werden.

Abfragetyp:

A set[roadlinksave]=1

Für jede Speicherabfrage gibt es Pflichtargumente. Diese werden in den Punkten 4.1 – 4.3 behandelt. Die eigentlichen Bestandsdaten sind dann im Punkt 4.4 zu jedem KOMMSVZ-Element aufgelistet.

Um eine Speicherungsabfrage zu generieren, muss man die Punkte 4.1 – 4.4 hintereinander abarbeiten, d.h. die einzelnen Argumente an die URL anhängen. Im Punkt 4.5 wird dies am Beispiel des KOMMSVZ-Elements Verwaltung gezeigt (Hinweis: jede Ausführung der Abfrage speichert die Daten physisch in der DB).

Als Ergebnis einer Speicherungsabfrage wird mittels XML eine Erfolgsmeldung und im Falle des Einfügens und Aktualisierens die gespeicherte Id (Identifizierung eines KOMMSVZ-Datensatzelementes) zurückgegeben. Mit dieser Id können dann die gespeicherten Bestandsdaten wieder abgefragt werden (s.o.).

dtd: kommsvz_roadlinksave.dtd (Kapitel 7.1.3) – für Fehler, Erfolgsmeldung und Rückgabe-Id

4.1. Speicherungstyp (arg_type)

Um KOMMSVZ-Bestandsdaten zu speichern, bietet die Schnittstelle die Speicherungstypen einfügen (insert), aktualisieren (update) und löschen (delete).

Je nach Typ muss das entsprechende Argument (arg_type) an die URL angehängt werden. Pflichtargumente (eines pro Abfrage):

P	arg[type]=insert	Einfügen
P	arg[type]=update	Aktualisieren
P	arg[type]=delete	Löschen

4.2. KOMMSVZ-Element (arg_tab)

Als weiteres Pflichtargument muss die Bestandsdatentabelle für das jeweilige KOMMSVZ-Element angegeben werden, in der die Bestandsdaten gespeichert werden sollen.

Je nach KOMMSVZ-Element muss das entsprechende Argument (arg_tab) an die URL angehängt werden.

Pflichtargumente (eines pro Abfrage):

P	arg[tab]=roadlink	KOMMSVZ-Element Straßenabschnitt
P	arg[tab]=roadlinkaddress	KOMMSVZ-Element Hausnummern
P	arg[tab]=roadlinkadministration	KOMMSVZ-Element Verwaltung
P	arg[tab]=roadlinkconnectionnodes	KOMMSVZ-Element Verknüpfungspunkte
P	arg[tab]=roadlinkcrosssection	KOMMSVZ-Element Querschnitte
P	arg[tab]=roadlinkdistrict	KOMMSVZ-Element Ortsteile
P	arg[tab]=roadlinkproperties	KOMMSVZ-Element Merkmale
P	arg[tab]=roadlinkpictures	KOMMSVZ-Element Bildbefahrung

4.3. Identifizierung (arg_id)

Als weiteres Pflichtargument muss für die Speicherungstypen aktualisieren und löschen eine eindeutige Identifizierung übergeben werden, die das zu speichernde KOMMSVZ-Datensatzelement genau identifiziert. Dabei können ein oder mehrere Datensätze gleichzeitig übergeben werden.

Das entsprechende Argument (arg_id) muss an die URL angehängt werden.

Pflichtargument (eine oder mehrere pro Abfrage):

P	arg[id][]=DEMVK59c4e98ceca7b	GML-ID eines KOMMSVZ-Datensatzelementes
---	------------------------------	---

4.4. Bestandsdaten der KOMMSVZ-Elemente

In diesem Abschnitt werden alle Bestandsdaten der einzelnen KOMMSVZ-Elemente beschrieben, die gespeichert werden können.

Durch die Abkürzungen i = insert (einfügen), u = update (aktualisieren), d = delete (löschen) wird gekennzeichnet, welche Bestandsdaten bei welchen Speichertypen verwendet werden müssen (p = Pflichtfelder) bzw. können.

werden Pflichtfelder bei einer Speicherungsanfrage nicht übergeben, schlägt diese fehl.

Die Bezeichnungen der Bestandsdaten-Argumente für die Abfrage sind an die Bezeichnung der Datenfelder des KOMMSVZ-Modells angelehnt und sind dadurch eindeutig identifizierbar.

Die Werte der Datenfelder sollten, wenn möglich, aus den bekannten Codelisten und Verzeichnissen entnommen werden. In den unteren Auflistungen der Argumente werden für die entsprechenden Quellen Kürzel verwendet (code = Codeliste, verz = amtliche Verzeichnisse von MV, zkm = zentrales Knotenmanagement)

4.4.1. KOMMSVZ-Element Straßenabschnitt

Der Straßenabschnitt, als Hauptelement im KOMMSVZ-Modell, enthält neben den definierenden Bestandsdaten auch die Geodaten.

Die Geodaten können nicht über diese Schnittstelle gespeichert werden. Dies erfolgt direkt über den KOMMSVZ-Webdienst in einem webfähigen Geodatenbrowser.

Exemplarisch der DescribeFeatureType eines Beispiel KOMMSVZ – WFS – Dienstes:

http://195.98.208.14:8080/kommsvz/wfs?request=DescribeFeatureType&version=1.1.0&service=WFS&typeName=kommsvz_wfs:roadlink

Wird ein Straßenabschnitt neu gezeichnet, werden im Geodatenbrowser auch die Bestandsdaten dazu gespeichert.

Mit dieser Schnittstelle können somit die Bestandsdaten eines Straßenabschnittes aktualisiert werden oder ein Straßenabschnitt kann komplett (mit seinen Bestandsdaten) gelöscht werden.

Als Ergebnis der Aktualisierungsabfrage wird die neue interne Id des Straßenabschnitts (siehe Kapitel 1.1.1) zurückgegeben. Aufgrund der Archivierungs- bzw. Gültigkeitsfunktionalität innerhalb des KOMMSVZ-Modells werden die alten Bestandsdaten mit der übergebenen Id archiviert. Dieses gilt nicht nur für den Straßenabschnitt, sondern auch für alle Untertabellen (Kapitel 4.4.2 – 4.4.8).

Bestandsdaten-Argumente:

arg[data][rdlcode]=DE1307405000104050	u, p	
arg[data][councode]=DE	u, p	verz Staat
arg[data][commcode]=13074050	u, p	verz Gemeinde
arg[data][roadcode]=00104	u, p	verz Strasse
arg[data][rdlnr]=050	u, p	
arg[data][begnode]=4547210	u, p	zkm
arg[data][endnode]=4547180	u, p	zkm
arg[data][eunr]=	u	
arg[data][len]=184	u, p	
arg[data][frcc]=Z	u, p	code Straßenart (Kap. 6.4)
arg[data][grco]=G	u, p	code Straßengru. (Kap. 6.3)
arg[data][proc]=Tom UptoDate	u, d	

4.4.2. KOMMSVZ-Element Hausnummern

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][loc]=40	i, u, p	
arg[data][hnr]=19	i, u	
arg[data][hnraddon]=	i, u	
arg[data][hnrpos]=links	i, u	links, rechts
arg[data][hnrdesc]=	i, u	
arg[data][hnrcode]=	i, u	
arg[data][proc]=Max Mustermann	i, u, d	

4.4.3. KOMMSVZ-Element Verwaltung

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][loc]=40	i, u, p	
arg[data][mainauthcode]=130202	i, u	code Straßenmei. (Kap. 6.4)
arg[data][ownauthagencycode]=00004	i, u	code Baulastentr. (Kap. 6.8)
arg[data][ownauthcode]=D	i, u	code Baulasten (Kap. 6.7)
arg[data][proc]=Max Mustermann	i, u, d	
arg[data][lowerauthgmlrdlid]= DEMVKS59c4e98ceca7b	i	

4.4.4. KOMMSVZ-Element Verknüpfungspunkte

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][loc]=40	i, u, p	
arg[data][nodecode]=4547181	i, u	zkm
arg[data][proc]=Max Mustermann	i, u, d	

4.4.5. KOMMSVZ-Element Querschnitte

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][loc]=75	i, u, p	
arg[data][typecode]=100	i, u	code Querschn. (Kap. 6.11)
arg[data][leftx]=-354	i, u	
arg[data][rightx]=265	i, u	
arg[data][diffwidth]=	i, u	
arg[data][proc]=Max Mustermann	i, u, d	

4.4.6. KOMMSVZ-Element Ortsteile

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][distcode]=001	i, u, p	verz Ortsteile
arg[data][proc]=Max Mustermann	i, u, d	

4.4.7. KOMMSVZ-Element Merkmale

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][loc]=25	i, u, p	
arg[data][rdwidth]=150	i, u	
arg[data][rdheight]=150	i, u	
arg[data][dircode]=B	i, u	code Fahrtrichtg. (Kap. 6.1)
arg[data][lanes]=1	i, u	
arg[data][minmaxlanes]=2	i, u	
arg[data][rdsignid]=0	i, u	verz Verkehrszeichen
arg[data][rdrestcode]=A	i, u	A=Anfang, E=Ende
arg[data][rdmaincode]=O	i, u	code Ortsdurchf. (Kap. 6.9)
arg[data][formcode]=007	i, u	code Straßenart (Kap. 6.2)
arg[data][rdsurfcatcode]=P	i, u	code Belagsarten (Kap. 6.10)
arg[data][proc]=Max Mustermann	i, u, d	

4.4.8. KOMMSVZ-Element Bildbefahrung

Bestandsdaten-Argumente:

arg[data][rdlid]=DEMVKS59c4e98ceca7b	i, u, p	
arg[data][loc]=30	i, u, p	
arg[data][lon]=11.345	i, u	
arg[data][lat]=10.345	i, u	
arg[data][ori]=P	i, u	P, V, G, L (Kap. 5.2.3)
arg[data][file]=bild.jpg	i, u	URL zum Bild

4.5. Beispiele für Speicherabfragen für das KOMMSVZ-Element Verwaltung (roadlinkadministration)

4.5.1. Einfügen (insert)

Beispiel (URL + Abfragetyp + Speicherungstyp + KOMMSVZ-Element + Bestandsdaten):

```
https://web.btftetz.de/interfaces/kommsvz/xml.php
?para[source_id]=testArea
&set[roadlinksave]=1
&arg[type]=insert
&arg[tab]=roadlinkadministration
&arg[data][rdlid]=DEMVK559c4e98ceca7b
&arg[data][loc]=40
&arg[data][mainauthcode]=130202
&arg[data][ownauthagencycode]=00004
&arg[data][ownauthcode]=D
&arg[data][proc]=Max Mustermann
```

4.5.2. Aktualisieren (update)

Beispiel (URL + Abfragetyp + Speicherungstyp + KOMMSVZ-Element + Identifizierung + Bestandsdaten):

```
https://web.btftetz.de/interfaces/kommsvz/xml.php
?para[source_id]=testArea
&set[roadlinksave]=1
&arg[type]=update
&arg[tab]=roadlinkadministration
&arg[id][]=14
&arg[data][rdlid]=DEMVK559c4e98ceca7b
&arg[data][loc]=40
&arg[data][mainauthcode]=130202
&arg[data][ownauthagencycode]=00004
&arg[data][ownauthcode]=D
&arg[data][proc]=Max Mustermann
```

4.5.3. Löschen (delete)

Beispiel (URL + Abfragetyp + Speicherungstyp + KOMMSVZ-Element + Identifizierung + Bearbeiter):

```
https://web.btftetz.de/interfaces/kommsvz/xml.php
?para[source_id]=testArea
&set[roadlinksave]=1
&arg[type]=delete
&arg[tab]=roadlinkadministration
&arg[id][]=14
&arg[data][proc]=Max Mustermann
```

5. Weitere Abfragen

Dieses Kapitel behandelt Abfragen an die aktuell realisierte Schnittstelle, die zum einen kein Bestandteil im Sinne des grundsätzlichen KOMMSVZ-Modells sind (5.1 Serverzeit, 5.2 Bildbefahrungsdaten) und zum anderen von einer anderen grundlegenden Schnittstelle zur Verfügung gestellt werden müssen (5.3 ZKM).

Diese Abfragen müssen kein Bestandteil der grundsätzlichen KOMMSVZ-Schnittstelle sein, sind aber bei der funktionellen Umsetzung des KOMMSVZ-Modells notwendig.

Die Abfragen zeigen, welche weiteren bzw. weiterführenden Schnittstellen geschaffen werden müssen (5.3 ZKM) und wie weitere Fachdaten (5.2 Bildbefahrung) angeschlossen werden können.

5.1. Abfrage der aktuellen Serverzeit (get_servertime)

Die Abfrage gibt die aktuelle Zeit des Servers, auf dem die POSTGIS-Datenbank installiert ist, im ISO-konformen Format an.

dtd: https://web.btfietz.de/interfaces/kommsvz/dtd/kommsvz_servertime.dtd

A get[servertime]=1

Abfrage:

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[servertime\]=1](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[servertime]=1)

5.2. Bildbefahrungs-Bestandsdaten zum Straßenabschnitt (get_roadlinkpictures)

Abfragen aller Bildbefahrungs-Bestandsdaten zu einem oder mehreren Straßenabschnitten mit oder ohne einer bestimmten Orientierung, d.h. Aufnahme-richtung des Bildes in Bezug zur Fahrtrichtung.

dtd: https://web.btfietz.de/interfaces/kommsvz/dtd/kommsvz_roadlinkpicture.dtd

A get[roadlinkpictures]=1

5.2.1. Alle Bildbefahrungs-Bestandsdaten (arg_all)

Alle Bildbefahrungs-Bestandsdaten zu allen Straßenabschnitten des abgefragten Straßennetzes.

P arg[all][]=1

Beispiel (URL + Abfragetyp + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkpictures\]=1&arg\[all\]\[\]=1](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkpictures]=1&arg[all][]=1)

5.2.2. Bildbefahrungs-Bestandsdaten zum Straßenabschnitt (arg_rdl_id)

Alle Bildbefahrungs-Bestandsdaten zu einem oder mehreren Straßenabschnitten.

P arg[all][]=1

P arg[rdl_id][]=DEMVK59c4e98ceca7b

Beispiel (URL + Abfragetyp + Pflichtargument + Wert):

[https://web.btfietz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkpictures\]=1&arg\[all\]\[\]=1&arg\[rdl_id\]\[\]=DEMVK59c4e98ceca7b](https://web.btfietz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkpictures]=1&arg[all][]=1&arg[rdl_id][]=DEMVK59c4e98ceca7b)

5.2.3. Bildbefahrungs-Bestandsdaten zum Straßenabschnitt mit einer Orientierung (arg_orientation)

Alle Bildbefahrungs-Bestandsdaten zu einem oder mehreren Straßenabschnitten mit einer bestimmten Orientierung abfragen. Für die Orientierung wird ein Code angegeben, der momentan noch nicht in Codelisten organisiert ist:

- P – Primär (nach vorne)
- V – Vegetation (nach rechts, Fahrbahnrand)
- G – Retrospektiv (nach hinten)
- L – Links (nach links, Gegenverkehr)

P arg[rdl_id][]=DEMVKS59c4e98ceca7b
P arg[orientation]=P

Beispiel (URL + Abfragetyp + Pflichtargument + Wert + Pflichtargument + Wert):

[https://web.btfigetz.de/interfaces/kommsvz/xml.php?para\[source_id\]=testArea&get\[roadlinkpictures\]=1&arg\[rdl_id\]\[\]=DEMVKS59c4e98ceca7b&arg\[orientation\]=P](https://web.btfigetz.de/interfaces/kommsvz/xml.php?para[source_id]=testArea&get[roadlinkpictures]=1&arg[rdl_id][]=DEMVKS59c4e98ceca7b&arg[orientation]=P)

5.3. Bestandsdaten zum Straßennetzknoten aus dem ZKM (get_roadnodedata)

Alle oder spezielle Bestandsdaten zu einem oder mehreren Straßennetzknoten abfragen. In weiteren Pflichtargumenten wird festgelegt, welche Bestandsdaten abgefragt werden sollen.

Diese Abfragen sind hier nur beispielhaft aufgeführt, da sie eigentlich an das zentrale Netznotenmanagement ZKM gestellt werden müssen. Das ZKM befindet sich in der Entwicklung. Die Anfrage an das simulierte ZKM kann ohne Authentifizierung erfolgen.

dtd: https://web.btfigetz.de/interfaces/zkm_sim/dtd/kommsvz_roadnode.dtd

A get[roadnodedata]=1

5.3.1. Alle Bestandsdaten zum Straßennetzknoten mit einer Netznotenid (arg_rdn_id)

Alle Bestandsdaten zu einem oder mehreren Straßennetzknoten abfragen. Abfrage über eine oder mehrere Straßennetzknoten-Id.

P arg[rdn_id][]=25566

Beispiel (URL + Abfragetyp + Pflichtargument + Wert):

[https://web.btfigetz.de/interfaces/zkm_sim/xml.php?get\[roadnodedata\]=1&arg\[rdn_id\]\[\]=25566](https://web.btfigetz.de/interfaces/zkm_sim/xml.php?get[roadnodedata]=1&arg[rdn_id][]=25566)

5.3.2. Alle Bestandsdaten zum Straßennetzknoten mit einer Netznotennummer (arg_rdn_code)

Alle Bestandsdaten zu einem oder mehreren Straßennetzknoten abfragen. Abfrage über eine oder mehrere Straßennetzknoten-Nummer.

P arg[rdn_code][]=439833940010

Beispiel (URL + Abfragetyp + Pflichtargument + Wert):

[https://web.btfigetz.de/interfaces/zkm_sim/xml.php?get\[roadnodedata\]=1&arg\[rdn_code\]\[\]=439833940010](https://web.btfigetz.de/interfaces/zkm_sim/xml.php?get[roadnodedata]=1&arg[rdn_code][]=439833940010)

6. Datendienste für Codelisten

Die Codelisten dienen zur Entschlüsselung aller innerhalb des KOMMSVZ-Modells verwendeten Schlüssel.

Hierbei wird unterschieden zwischen Codelisten des amtlichen Liegenschaftskatasters, die auch für andere Fachanwendungen verwendet werden können und den KOMMSVZ-Codelisten, die nur im Zusammenhang mit dem Infrastrukturvermögen Straße ihre Funktionalität haben.

Im Folgenden wird in dieser Beschreibung nur auf die Codelisten eingegangen, die direkt mit dem Infrastrukturvermögen Straße verknüpft sind.
Die Codelisten des amtlichen Liegenschaftskatasters stehen in der Verantwortung des Bundeslandes MV. Im Dokument „BTFietz_Codelisten.doc“ ist beispielhaft beschrieben, wie diese aufgebaut sein sollten.

6.1. Codeliste für KOMMSVZ-Fahrtrichtungen (`get_directioncodelist`)

Die KOMMSVZ-Fahrtrichtungen entschlüsseln, ob ein Straßenabschnitt in beide Richtungen gleichzeitig oder nur in die Hin- oder Gegenrichtung befahrbar ist.
Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_directionCodeList.dtd

A	<code>get[directioncodelist]=1</code>	
P	<code>arg[all][]=1</code>	gesamte Liste
P	<code>arg[directioncode][]=B</code>	Liste für einen bzw. mehrere Werte

Beispiel (gesamte Liste):

[https://web.btfietz.de/interfaces/kommsvzcodelists/xml.php?get\[directioncodelist\]=1&arg\[all\]\[\]=1](https://web.btfietz.de/interfaces/kommsvzcodelists/xml.php?get[directioncodelist]=1&arg[all][]=1)

Beispiel (Liste für einen oder mehrere Werte):

[https://web.btfietz.de/interfaces/kommsvzcodelists/xml.php?get\[directioncodelist\]=1&arg\[directioncode\]\[\]=B&arg\[directioncode\]\[\]=R](https://web.btfietz.de/interfaces/kommsvzcodelists/xml.php?get[directioncodelist]=1&arg[directioncode][]=B&arg[directioncode][]=R)

6.2. Codeliste für KOMMSVZ-Straßentypen (alt: -arten) (`get_formofwaycodelist`)

Die KOMMSVZ-Straßentypen entschlüsseln, welche spezieller Typ bzgl. seiner Benutzung ein Straßenabschnitt ist.

Mit Typ ist hier gemeint, wer oder was sich auf der Straße als Hauptbenutzer bewegen darf, wie z. B. Auto-, Fahrradstraße oder Fußgängerzone.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_formOfWayCodeList.dtd

A	<code>get[formofwaycodelist]=1</code>	
P	<code>arg[all][]=1</code>	gesamte Liste
P	<code>arg[formofwaycode][]=007</code>	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.3. Codeliste für KOMMSVZ-Straßengruppen (`get_functionalroadgroupcodelist`)

Die KOMMSVZ-Straßengruppen entschlüsseln, in welches übergeordnete Element innerhalb der Widmungsstruktur in der Verwaltung von Straßennetzen ein Straßenabschnitt zugeordnet ist. Jeder Straßengruppe sind eine oder mehrere KOMMSVZ-Straßenarten untergeordnet (siehe Punkt 6.4).

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_functionalRoadGroupCodeList.dtd

A	<code>get[functionalroadgroupcodelist]=1</code>	
P	<code>arg[all][]=1</code>	gesamte Liste
P	<code>arg[funcgroupcode][]=K</code>	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.4. Codeliste für KOMMSVZ-Straßenarten (alt: -Widmungen)

(get_functionalroadclasscodelist)

Die KOMMSVZ-Straßenarten entschlüsseln, welcher Art ein Straßenabschnitt (bzw. ein Netzknoten) innerhalb der Widmungsstruktur in der Verwaltung von Straßennetzen zugeordnet ist, wie z. B. Bundes-, Orts- oder Privatstraße.

Neben der Straßenart als Code werden in der gesamten Liste Verknüpfungen zur KOMMSVZ-Straßengruppe und KOMMSVZ-Baulast jeweils gebildet. Dabei kann eine Straßenart nur einer Gruppe bzw. nur einer Baulast zugeordnet sein.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

Zudem können alle Arten zu einer Gruppe bzw. zu einer Baulast abgefragt werden.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_functionalRoadClassCodeList.dtd

A	get[functionalroadclasscodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[funcclasscode][]=Z	Liste für einen bzw. mehrere Straßenarten-Codes
P	arg[funcgroupcode][]=G	Liste für eine bzw. mehrere Straßengruppe(n)
P	arg[ownauthcode][]=G	Liste für eine bzw. mehrere Baulaste(n)

Beispiel s. o.

6.5. Codeliste für KOMMSVZ-Straßenmeistereien (get_maintenanceauthoritycodelist)

Die KOMMSVZ-Straßenmeistereien entschlüsseln, von welchen Straßenmeistereien ein Straßenabschnitt unterhalten wird.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_maintenanceAuthorityCodeList.dtd

A	get[maintenanceauthoritycodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[authcode][]=130101	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.6. Codeliste für KOMMSVZ-Verknüpfungspunktarten

(get_nodecrossreferencecodelist)

Die KOMMSVZ-Verknüpfungspunktarten entschlüsseln, welche Netzknoten Übergangsknoten in andere Transportverkehrsnetze sind. Diese Codeliste ist Bestandteil des ZKM.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_nodeCrossReferenceCodeList.dtd

A	get[nodecrossreferencecodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[crosscode][]= roadBnet	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.7. Codeliste für KOMMSVZ-Straßenservicebereiche

(get_noderoadserviceareacodelist)

Die KOMMSVZ-Straßenservicebereiche entschlüsseln, welche Netzknoten Übergangsknoten in Straßenservicebereiche, wie z. B. Rastplätze, sind. Diese Codeliste ist Bestandteil des ZKM.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_nodeRoadServiceAreaCodeList.dtd

A	get[noderoadserviceareacodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[areacode][]= restarea	Liste für einen bzw. mehrere Werte

6.8. Codeliste für KOMMSVZ-Baulasten (get_ownerauthoritycodelist)

Die KOMMSVZ-Baulasten entschlüsseln, bei wem die Baulasten für einen Straßenabschnitt liegen.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_ownerAuthorityAgencyCodeList.dtd

A	get[ownerauthoritycodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[authcode][]=D	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.9. Codeliste für KOMMSVZ-Baulastenträger Dritter

(get_ownerauthorityagencycodelist)

Die KOMMSVZ-Baulastenträger entschlüsseln, wenn die Baulast bei Dritten liegt (s. Punkt 6.7), welcher Baulastenträger für einen Straßenabschnitt verantwortlich ist.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_ownerAuthorityAgencyCodeList.dtd

A	get[ownerauthorityagencycodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[authcode][]=B	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.10. Codeliste für KOMMSVZ-Ortsdurchfahrten (get_roadmainthroughcodelist)

Die KOMMSVZ-Ortsdurchfahrten entschlüsseln, ob ein Straßenabschnitt innerhalb eines Ortes oder in einem anderen bestimmten Bereich liegt.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_roadMainThroughCodeList.dtd

A	get[roadmainthroughcodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[mainthroughcode][]=E	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.11. Codeliste für KOMMSVZ-Belagsarten (get_roadsurfacecategorycodelist)

Die KOMMSVZ-Belagsarten entschlüsseln, welche Straßenbelagsarten innerhalb eines Straßenabschnittes auftreten.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_roadSurfaceCategoryCodeList.dtd

A	get[roadsurfacecategorycodelist]=1	
P	arg[all][]=1	gesamte Liste
P	arg[surfacecode][]=P	Liste für einen bzw. mehrere Werte

Beispiel s. o.

6.12. Codeliste für KOMMSVZ-Querschnittstreifenart (get_stripetypecodelist)

Die KOMMSVZ-Querschnittstreifenart entschlüsseln, welche Art bzw. Funktion ein Querschnittstreifen innerhalb eines Straßenabschnittes hat.

Die Codeliste kann komplett abgefragt werden oder nur für einen bzw. mehrere Werte.

dtd: https://web.btfietz.de/interfaces/kommsvzcodelists/dtd/kommsvz_stripeTypeCodeList.dtd

A get[stripetypecodelist]=1

P arg[all][]=1

gesamte Liste

P arg[stripecode][]=122

Liste für einen bzw. mehrere Werte

Beispiel s. o.

7. Anhang

7.1. Dokumententypdefinitionen (dtd)

7.1.1. kommsvz_roadlink.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DTD erstellt mit XMLSpy v2012 rel. 2 (X64) (http://www.altova.com)-->
<ELEMENT roadlink ((xmlErrors, xmlNumberOfResults, roadlinkId*))>
<ELEMENT xmlErrors ((roadlinkError*))>
<ELEMENT roadlinkError (#PCDATA)>
<ATTLIST roadlinkError roadlinkId CDATA #REQUIRED
        errorNumber CDATA #REQUIRED>
<ELEMENT xmlNumberOfResults (#PCDATA)>
<ELEMENT roadlinkId ((wkb_geometry, roadlinkCode, countryCode, communeCode, roadCode, roadlinkNumber, startNode,
endNode, europeanRouteNumber, length, functionalroadclassCode, beginLifespan, endLifespan, forwardroadlinkId, processor,
roadlinkAdministrationGroup, roadlinkDistrictGroup, roadlinkPropertiesGroup, roadlinkCrossSectionGroup,
roadlinkAddressGroup, roadlinkConnectionNodesGroup))>
<ATTLIST roadlinkId id CDATA #REQUIRED>
<ELEMENT roadlinkCode (#PCDATA)>
<ELEMENT countryCode (#PCDATA)>
<ELEMENT communeCode (#PCDATA)>
<ELEMENT roadCode (#PCDATA)>
<ELEMENT roadlinkNumber (#PCDATA)>
<ELEMENT startNode (#PCDATA)>
<ELEMENT endNode (#PCDATA)>
<ELEMENT europeanRouteNumber (#PCDATA)>
<ELEMENT length (#PCDATA)>
<ATTLIST length unit CDATA #FIXED "meter">
<ELEMENT forwardroadlinkId (#PCDATA)>
<ELEMENT roadlinkAdministrationGroup ((roadlinkAdministration*))>
<ELEMENT roadlinkDistrictGroup ((roadlinkDistrict*))>
<ELEMENT roadlinkPropertiesGroup ((roadlinkProperties*))>
<ELEMENT roadlinkCrossSectionGroup ((roadlinkCrossSection*))>
<ELEMENT roadlinkAddressGroup ((roadlinkAddress*))>
<ELEMENT roadlinkConnectionNodesGroup ((roadlinkConnectionNodes*))>
<ELEMENT roadlinkAdministration ((roadlinkAdministrationId, location, maintenanceAuthorityCode, ownerAuthorityCode,
ownerAuthorityAgencyCode, beginLifespan, endLifespan, processor))>
<ELEMENT roadlinkAdministrationId (#PCDATA)>
<ELEMENT maintenanceAuthorityCode (#PCDATA)>
<ELEMENT ownerAuthorityCode (#PCDATA)>
<ELEMENT ownerAuthorityAgencyCode (#PCDATA)>
<ELEMENT roadlinkDistrict ((roadlinkDistrictId, districtCode, beginLifespan, endLifespan, processor))>
<ELEMENT roadlinkDistrictId (#PCDATA)>
<ELEMENT districtCode (#PCDATA)>
<ELEMENT roadlinkProperties ((roadlinkPropertiesId, location, roadWidth, roadHeight, directionCode, numberOfLanes,
minMaxNumberOfLanes, roadsignId, roadRestriction, roadmainthroughCode, formofwayCode, roadsurfacecategoryCode,
beginLifespan, endLifespan, processor))>
<ELEMENT roadlinkPropertiesId (#PCDATA)>
<ELEMENT roadWidth (#PCDATA)>
<ATTLIST roadWidth unit CDATA #FIXED "centimeter">
<ELEMENT roadHeight (#PCDATA)>
<ATTLIST roadHeight unit CDATA #FIXED "centimeter">
<ELEMENT directionCode (#PCDATA)>
<ELEMENT numberOfLanes (#PCDATA)>
<ELEMENT minMaxNumberOfLanes (#PCDATA)>
<ELEMENT roadsignId (#PCDATA)>
<ELEMENT roadRestriction (#PCDATA)>
<ELEMENT roadmainthroughCode (#PCDATA)>
<ELEMENT formofwayCode (#PCDATA)>
<ELEMENT roadsurfacecategoryCode (#PCDATA)>
<ELEMENT roadlinkCrossSection ((roadlinkCrossSectionId, location, stripeTypeCode, stripeLeftXValue, stripeRightXValue,
stripeDiffuseWidth, beginLifespan, endLifespan, processor))>
<ELEMENT roadlinkCrossSectionId (#PCDATA)>
<ELEMENT stripeTypeCode (#PCDATA)>
<ELEMENT stripeLeftXValue (#PCDATA)>
<ATTLIST stripeLeftXValue unit CDATA #FIXED "centimeter">
<ELEMENT stripeRightXValue (#PCDATA)>
<ATTLIST stripeRightXValue unit CDATA #FIXED "centimeter">
<ELEMENT stripeDiffuseWidth (#PCDATA)>
<ATTLIST stripeDiffuseWidth unit CDATA #FIXED "centimeter">
<ELEMENT roadlinkAddress ((roadlinkAddressId, location, housenumber, housenumberAddon, housenumberPosition,
housenumberDescription, housenumberCode, beginLifespan, endLifespan, processor))>
<ELEMENT roadlinkAddressId (#PCDATA)>
<ELEMENT housenumber (#PCDATA)>
<ELEMENT housenumberAddon (#PCDATA)>
<ELEMENT housenumberPosition (#PCDATA)>
<ELEMENT housenumberDescription (#PCDATA)>
```

```

<!ELEMENT housenumberCode (#PCDATA)>
<!ELEMENT roadlinkConnectionNodes ((roadlinkConnectionNodesId, location, nodeCode, beginLifespan, endLifespan,
processor))>
<!ELEMENT roadlinkConnectionNodesId (#PCDATA)>
<!ELEMENT nodeCode (#PCDATA)>
<!ELEMENT wkb_geometry (#PCDATA)>
<!ELEMENT functionalroadclassCode (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ATTLIST location unit CDATA #FIXED "meter">
<!ELEMENT beginLifespan (#PCDATA)>
<!ELEMENT endLifespan (#PCDATA)>
<!ELEMENT processor (#PCDATA)>

```

7.1.2. kommsvz_roadlinkbrowser.dtd

```

<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2012 rel. 2 (x64) (http://www.altova.com)-->
<!ELEMENT roadlinkbrowser ((xmlErrors, xmlNumberOfResults, browserlevel*))>
<!ELEMENT xmlErrors ((roadlinkbrowserError*))>
<!ELEMENT roadlinkbrowserError (#PCDATA)>
<!ATTLIST roadlinkbrowserError roadlinkbrowserId CDATA #REQUIRED
errorNumber CDATA #REQUIRED>
<!ELEMENT xmlNumberOfResults (#PCDATA)>
<!ELEMENT browserlevel ((commune* | district* | road* | roadlink*))>
<!ATTLIST browserlevel type CDATA #REQUIRED
departmentCode CDATA #IMPLIED
communeCode CDATA #IMPLIED
districtCode CDATA #IMPLIED
roadCode CDATA #IMPLIED
roadlength CDATA #IMPLIED
roadlinkCnt CDATA #IMPLIED>
<!ELEMENT commune (#PCDATA)>
<!ATTLIST commune communeCode CDATA #REQUIRED>
<!ATTLIST commune length CDATA #REQUIRED>
<!ATTLIST commune roadlinkCnt CDATA #REQUIRED>
<!ELEMENT district (#PCDATA)>
<!ATTLIST district districtPlz CDATA #REQUIRED>
<!ATTLIST district districtCode CDATA #REQUIRED>
<!ATTLIST district length CDATA #REQUIRED>
<!ATTLIST district roadlinkCnt CDATA #REQUIRED>
<!ELEMENT road (#PCDATA)>
<!ATTLIST road roadCode CDATA #REQUIRED>
<!ATTLIST road length CDATA #REQUIRED>
<!ATTLIST road roadlinkCnt CDATA #REQUIRED>
<!ELEMENT roadlink (#PCDATA)>
<!ATTLIST roadlink roadlinkCode CDATA #REQUIRED>
<!ATTLIST roadlink length CDATA #REQUIRED>

```

7.1.3. kommsvz_roadlinksave.dtd

```

<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2012 rel. 2 (x64) (http://www.altova.com)-->
<!ELEMENT roadlinksave ((xmlErrors, xmlNumberOfResults, saveResult))>
<!ELEMENT xmlErrors ((roadlinksaveError*))>
<!ELEMENT roadlinksaveError (#PCDATA)>
<!ATTLIST roadlinksaveError roadlinksaveId CDATA #REQUIRED
errorNumber CDATA #REQUIRED>
<!ELEMENT xmlNumberOfResults (#PCDATA)>
<!ELEMENT saveResult (#PCDATA)>
<!ATTLIST saveResult saveId CDATA #REQUIRED>

```

7.1.4. kommsvz_nextroadlinknumber.dtd

```

<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2012 rel. 2 (x64) (http://www.altova.com)-->
<!ELEMENT nextroadlinknumber ((xmlErrors, xmlNumberOfResults, roadlinkNumber*))>
<!ELEMENT xmlErrors ((nextroadlinknumberError*))>
<!ELEMENT nextroadlinknumberError (#PCDATA)>
<!ATTLIST nextroadlinknumberError nextroadlinknumberId CDATA #REQUIRED
errorNumber CDATA #REQUIRED>
<!ELEMENT xmlNumberOfResults (#PCDATA)>
<!ELEMENT roadlinkNumber (#PCDATA)>

```

7.1.5. kommsvz_roadnodeconnect_roadlink.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2012 rel. 2 (x64) (http://www.altova.com)-->
<ELEMENT roadnodeconnect_roadlink ((xmlErrors, xmlNumberOfResults, roadnodeCode*))>
<ELEMENT xmlErrors ((roadnodeconnect_roadlinkError*))>
<ELEMENT roadnodeconnect_roadlinkError (#PCDATA)>
<ATTLIST roadnodeconnect_roadlinkError roadnodeconnect_roadlinkId CDATA #REQUIRED
      errorNumber CDATA #REQUIRED>
<ELEMENT xmlNumberOfResults (#PCDATA)>
<ELEMENT roadnodeCode ((location, roadlinkCode, countryCode, communeCode, roadCode, roadlinkNumber,
functionalroadclassCode))>
<ATTLIST roadnodeCode code CDATA #REQUIRED>
<ELEMENT location (#PCDATA)>
<ELEMENT roadlinkCode (#PCDATA)>
<ELEMENT countryCode (#PCDATA)>
<ELEMENT communeCode (#PCDATA)>
<ELEMENT roadCode (#PCDATA)>
<ELEMENT roadlinkNumber (#PCDATA)>
<ELEMENT functionalroadclassCode (#PCDATA)>
```

7.1.6. kommsvz_timestamp.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2012 rel. 2 (x64) (http://www.altova.com)-->
<ELEMENT timestamp ((xmlErrors, xmlNumberOfResults, maxsavedate *))*>
<ELEMENT xmlErrors ((timestampError*))>
<ELEMENT timestampError (#PCDATA)>
<ATTLIST timestampError timestampId CDATA #REQUIRED
      errorNumber CDATA #REQUIRED>
<ELEMENT xmlNumberOfResults (#PCDATA)>
<ELEMENT maxsavedate (#PCDATA)>
```

7.1.7. kommsvz_errorxmlcall.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD erstellt mit XMLSpy v2012 rel. 2 (x64) (http://www.altova.com)-->
<ELEMENT errorxmlcall ((xmlErrors))>
<ELEMENT xmlErrors ((errorxmlcallError*))>
<ELEMENT errorxmlcallError (#PCDATA)>
<ATTLIST errorxmlcallError errorxmlcallId CDATA #REQUIRED
      errorNumber CDATA #REQUIRED>
```