

Manuál k XMLRPC/REST interface

Dokument platný od verze 004

Poslední aktualizace dne 6. září 2018

Služba bude poskytována jako web-service typu XML-RPC2 <http://en.wikipedia.org/wiki/XMLrpc> nebo pomocí REST služby https://cs.wikipedia.org/wiki/Representational_State_Transfer

Endpointy interface:

End-pointy služby dle výstupu:

Typ	URI služby	XML Output CHARSET
XML-RPC2	https://[servername]/RPC2	LATIN1 format
XML-RPC2	https://[servername]/RPC2UTF8	UTF8 format
REST	https://[servername]/api/[vystupniformat]/[nazev metody] nebo https://[servername]/api/[nazev metody]	UTF8

XML-RPC2

V případě XML-RPC2 je výstup ve formátu XML.

Způsob práce je následující:

a) nejdřív je nutné zavolat metodu Authenticate, odeslat login a heslo (v hashované podobě). Návratem je sessionid, který je potřeba použít jako součást každého dotazu. Vyjimkou tvoří dotazy, které nepotřebují autentizaci; potom je sessionid rovno 0.

b) provést požadované operace

c) odhlásit se - zavolat metodu Deauthenticate

REST:

U REST je možné volit výstupní formát z XML (vystupniformat=xml), JSON (vystupniformat=json) a GEOJSON (vystupniformat=geojson) a TEXT (vystupniformat=text). Název metody je nutno uvádět

malými písmeny. Všechny parametry metody jsou předány jako parametry HTTP POST požadavku.

Výstupní formát je možné určit následujícím způsobem:

a) pomocí GET parametru f

Příklady - výstup ve formátu XML:

<https://server/api/nazevmetody?f=xml>

<https://server/api/json/nazevmetody?f=xml>

b) pomocí hlavičky HTTP požadavku

Příklady - výstup ve formátu XML:

<https://server/api/nazevmetody>

<https://server/api/json/nazevmetody>

→ v obou případech je v HTTP hlavičkách požadavku zaslána hlavička:

X-EWS-DataFormatType: xml

c) pomocí určení cesty

Příklady - výstup ve formátu:

<https://server/api/nazevmetody>

→ není určen výstupní formát,

implicitní výstupní formát je ve formátu JSON

<https://server/api/xml/nazevmetody>

→ výstupní formát je určen cestou, bude

XML

Pro autentizaci je možné buď použít sessionid obdrženo po volání metody Authenticate, nebo je možné uživatele autentizovat přímo pomocí HTTP Base Auth, kde se použije kombinace uživatelského jména a hesla (ne MD5 hesla!); pak je případný sessionid rovno 0. K debugování/testování je doporučeno používat nástroj Postman (<https://www.getpostman.com/>)

Příklad volání metody Authenticate:

`https://[servername]/api/xml/authenticate`

→ v POST jsou následující parametry:

`username:tester@pawouk.net`

`md5:f5d1278e8109edd94e1e4197e04873b9`

Příklad volání metody GetStoredQueryResult:

`https://[servername]/api/xml/getstoredqueryresult`

→ v POST jsou následující parametry:

```
sessionid:a24f14a447b5b02fe655de789ad433b2
queryid:612
rows:100
offset:0
params:<?xml version="1.0" encoding="UTF-8"?>
<request><params><param><parid>1</parid><content>50</content></param>
</params></request>
```

[https://\[servername\]/api/xml/getstoredqueryresult](https://[servername]/api/xml/getstoredqueryresult) → vrátí výsledek v XML formátu

[https://\[servername\]/api/json/getstoredqueryresult](https://[servername]/api/json/getstoredqueryresult) → vrátí výsledek v JSON formátu

[https://\[servername\]/api/geojson/getstoredqueryresult](https://[servername]/api/geojson/getstoredqueryresult) → vrátí výsledek v geoJSON notaci

Ukázka programového kódu

Javascript / AJAX

```
var form = new FormData();
form.append("username", "moje.emailova@adresa.cz");
form.append("md5", "10b04bce1c6f29203c84fd659ed4cacf");

var settings = {
  "async": true,
  "crossDomain": true,
  "url": "https://servername/api/json/authenticate",
  "method": "POST",
  "processData": false,
  "contentType": false,
  "mimeType": "multipart/form-data",
  "data": form
}

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

CURL

```
curl --insecure \
  --request POST \
  --url https://servername/api/xml/authenticate \
  --form username=moje.emailova@adresa.cz \
  --form md5=10b04bce1c6f29203c84fd659ed4cacf
```

Seznam:

[METODA Authenticate\(username,md5\);](#)

[METODA Deauthenticate\(sessionid\);](#)

[METODA GetCapabilities\(sessionid\);](#)

[METODA GetSchema\(sessionid\);](#)

[METODA GetComposedQueryResult \(sessionid, xmlrequest\);](#)

[METODA GetStoredQueryResult \(sessionid, queryid, rows, offset, xmlparams\);](#)

METODA Authenticate(username,md5);

Vstupní parametry:	
username	STRING - Identifikátor uživatele (e-mail)
md5	STRING – MD5 hash hesla

Návratová hodnota:	STRING - sessionid – použije se při dalším volání ostatních funkcí
--------------------	--

METODA Deauthenticate(sessionid);

Vstupní parametry:	
sessionid	STRING - Identifikátor session, který je výstupem funkce Authenticate()

Návratová hodnota:	BOOLEAN - true/false – pokud uživatel byl předtím autentizován pod danou sessionID, je vrácena hodnota true, jinak false
--------------------	--

METODA GetCapabilities (sessionid);

Vstupní parametry:	
sessionid	STRING - Identifikátor session, který je výstupem funkce Authenticate(); může být použita hodnota 0 - pak se jedná o anonymní přístup

Návratová hodnota:	STRING - XMLdata – informace o jednotlivých funkcích XMLRPC interface pro daného uživatele
--------------------	--

METODA GetTablesCount (sessionid);

Vstupní parametry:	
sessionid	STRING - Identifikátor session, který je výstupem funkce Authenticate()

Návratová hodnota:	STRING - XMLdata – informace o jednotlivých tabulkách, ke kterým má uživatel přístup
--------------------	--

METODA GetSchema(sessionid);

Tato funkce vrátí v XML datech následující bloky:

- Seznam tzv. plain queries – tzn. předdefinované dotazy, které je možno volat s parametry
- Schéma (schema name, tabulky/pohledy/materializované pohledy, sloupce) ke kterým má uživatel přístup
- Obojí uživatel dostane customizované dle jeho uživatelských práv

Vstupní parametry:	
sessionid	STRING - Identifikátor session, který je výstupem funkce Authenticate(); může být použita hodnota 0 - pak se jedná o anonymní přístup

Návratová hodnota:	STRING - XMLdata – definice viz níže
--------------------	--------------------------------------

Návratová hodnota – XMLDATA:

```
<?xml version="1.0" encoding="UTF-8"?>
<response sitetype="DEV_001" siteversion="DEV" siteexpires="2012-12-01">
  <metadata>
    <user>tester@tester.net</user>
    <created>Thu, 03 May 2012 11:32:52 +0200</created>
    <validuntil>Thu, 03 May 2012 11:32:52 +0200</validuntil>
  </metadata>
  <storedqueries>
    <query id="18">
      <qdescription><![CDATA[Dotaz na uzivatelskou tabulku - vsichni uzivatelé mající
příjmení zvolený řetězec]]></qdescription>
      <qparam id="1" type="NUMERIC">
        <qdescription><![CDATA[ID uživatele větší než (v klauzuli
WHERE)]]></qdescription>
      </qparam>
      <qparam id="2" type="VARCHAR">
        <qdescription><![CDATA[Příjmení uživatele - musí obsahovat
řetězec]]></qdescription>
      </qparam>
      <qparam id="3" type="NUMERIC">
        <qdescription><![CDATA[ID uživatele větší než (v klauzuli
HAVING)]]></qdescription>
      </qparam>
    </query>
  </storedqueries>
  <tables>
    <userschema name="APP_UP">
      <table name="FVU" type="table">
        <cols>
          <c name="OBJECTID" type="NUMBER" length="22"/>
          <c name="ID" type="NUMBER" length="22"/>
          <c name="KODFP" type="NVARCHAR2" length="20"/>
        </cols>
      </table>
    </userschema>
  </tables>
</response>
```

```
<c name="POPIS" type="NVARCHAR2" length="1000"/>
<c name="TYPUZEMI" type="NVARCHAR2" length="510"/>
<c name="KATEGORIEUZEMI" type="NVARCHAR2" length="510"/>
<c name="TEXT_" type="NCLOB" length="4000"/>
<c name="TEXT_FV" type="NCLOB" length="4000"/>
<c name="TEXT_DV" type="NCLOB" length="4000"/>
<c name="TEXT_VV" type="NCLOB" length="4000"/>
<c name="C_POZN" type="NVARCHAR2" length="20"/>
</cols>
</table>
</userschema>
</tables>
</response>
```

Popis XML:

Metadata – obsahuje informace o XML, které bylo vygenerováno:

- Username – pro kterého uživatele
- Created – kdy bylo vytvořeno
- Valid until – do kdy může být dané schéma považováno za platné (např. na testovacím stroji může být validita velice krátká, kvůli testování. Konzument služby by se měl tímto datumem řídit a upravit případné cachování dat
- StoredQueries - preduložené dotazy a seznam parametrů. Pro použití pomocí funkce *getstoredqueryresult*
- Tables -> seznam tabulek/pohledů/materializovaných pohledů, z kterých uživatel může daný dotaz konstruovat pomocí funkce *getcomposedqueryresult*
- Userschema – odpovídá v databázi názvu SCHEMA
- Table – název tabulky v daném schématu
- Columns – sloupce v dané tabulce
- C -> sloupec v tabulce; jeho název a datový typ a délka

METODA GetComposedQueryResult (sessionid, xmlrequest);

Tato funkce vrátí v XML datech výsledky dotazu na základě vytvořeného dotazu pomocí query builderu. Informace pro query builder jsou získávána pomocí metody GetSchema, v xml datech v XPATH response->tables

Vstupní parametry:	
sessionid	STRING - Identifikátor session, který je výstupem funkce Authenticate(); může být použita hodnota 0 - pak se jedná o anonymní přístup
xmlrequest	STRING – Obsah dotazu formátovaný do XML

XMLREQUEST – formát dat

UKÁZKA - VERZE 1

```
<request>
  <numrows>500</numrows>
  <offset>1000</offset>
  <cols>
    <c>
      <name>nejakynazevsloupce1</name>
      <code>schemaname1.tablename1.columnname1</code>
      <function>DISTINCT</function>
    </c>
    <c>
      <name>nejakynazevsloupce2</name>
      <code>schemaname2.tablename2.columnname2</code>
      <function>SUM</function>
    </c>
    <c>
      <name>nejakynazevsloupce3</name>
      <code>schemaname2.tablename2.columnname3</code>
    </c>
  </cols>
  <tables>
    <table name="schemaname1.tablename1" />
    <table name="schemaname2.tablename2" />
  </tables>
  <joins>
    <join joinedtable="APP_UP.FVU" left="PUP_CUR.PVP_FVU_P.KODFP1_A"
      right="APP_UP.FVU.KODFP" type="LEFT" />
  </joins>
  <groupby>
    <gb column="nejakynazevsloupce1" />
    <gb column="nejakynazevsloupce2" />
  </groupby>
  <orderby>
    <ob column="nejakynazevsloupce1" />
    <ob column="nejakynazevsloupce2" />
  </orderby>
  <where><![CDATA[
schemaname1.tablename2.columnname4>##PARAM1## AND
schemaname2.tablename2.columnname7=##PARAM2##
]]></where>
  <having><![CDATA[
nejakynazevsloupce2 > ##PARAM3##
]]></having>
  <params>
```



```

        <param>
            <parid>1</parid>
            <partype>NUMERIC</partype>
            <content>100</content>
        </param>
        <param>
            <parid>2</parid>
            <partype>VARCHAR</partype>
            <content>somevalue</content>
        </param>
        <param>
            <parid>3</parid>
            <partype>NUMERIC</partype>
            <content>500</content>
        </param>
    </params>
</request>

```

UKÁZKA - VERZE 2

```

<request>
<query name="main">
<numrows>500</numrows>
<offset>0</offset>
<cols>
<c>
<name>KATASTR</name>
<code>MAP.MAP_KATASTRALNIUZEMI_P.KATUZE_KOD</code>
</c>
<c>
<name>NAZEV</name>
<code>MAP.MAP_KATASTRALNIUZEMI_P.NAZEV</code>
</c>
<c>
<name>SHP_AS_GML</name>
<code>ST_GEOMETRY_TO_GML (MAP.MAP_KATASTRALNIUZEMI_P.SHAPE) </code>
</c>
</cols>
<tables>
<table name="MAP.MAP_KATASTRALNIUZEMI_P" />
</tables>
<joins/>
<groupby/>
<orderby/>
<where><![CDATA[
MAP.MAP_KATASTRALNIUZEMI_P.KATUZE_KOD=##PARAM1##
]]></where>
<having/>
    <params>
        <param>
            <parid>1</parid>
            <description>Katastralni cislo</description>
            <partype>NUMERIC</partype>
            <content>727008</content>
        </param>
    </params>
</query>
<query name="sub1" sub-of="main">
<numrows>500</numrows>
<offset>0</offset>
<cols>
<c>
<name>KAT</name>
<code>DMP_CUR.PARCELAKN_DOKM_P.KATUZE_KOD</code>
</c>
<c>
<name>PARCELA</name>
<code>DMP_CUR.PARCELAKN_DOKM_P.PARCELA</code>
</c>
<c>
<name>NAZEV KATASTRU</name>

```

```

<code>DMP_CUR.PARCELAKN_DOKM_P.NAZEVA_KU</code>
</c>
<c>
<name>NAZEVA_KATASTRU_SUBSTR</name>
<code>SUBSTR(DMP_CUR.PARCELAKN_DOKM_P.NAZEVA_KU,##PARAM4##, ##PARAM5##)</code>
</c>
</cols>
<tables>
<table name="DMP_CUR.PARCELAKN_DOKM_P" />
</tables>
<joins />
<groupby />
<orderby />
<where><![CDATA[DMP_CUR.PARCELAKN_DOKM_P.KATUZE_KOD=##PARAM2## AND
DMP_CUR.PARCELAKN_DOKM_P.PARCELA > ##PARAM3##]]></where>
<having />
<params>
  <param>
    <parid>2</parid>
    <description>Katastralni cislo - pivot z prvni
tabulky</description>
    <partype>NUMERIC</partype>
    <pivot>KATASTR</pivot>
  </param>
  <param>
    <parid>3</parid>
    <description>ID Parcely vetsi nez parametr</description>
    <partype>NUMERIC</partype>
    <content>111</content>
  </param>
  <param>
    <parid>4</parid>
    <description>Nazev katastralniho uzemi - substring - zacatek
retezce</description>
    <partype>NUMERIC</partype>
    <content>0</content>
  </param>
  <param>
    <parid>5</parid>
    <description>Nazev katastralniho uzemi - substring - delka
retezce</description>
    <partype>NUMERIC</partype>
    <content>3</content>
  </param>
</params>
</query>
</request>

```

XMLREQUEST – popis

Obsah XML pro každý dotaz obsahuje tyto dvě informace:

Numrows – počet max požadovaných řádků v resultsetu

Offset – posun o n-řádků

Numrows a offset zajistí postupné procházení v případě potřeby

Cols – seznam sloupců – name (název sloupce – za AS), code – kód dotazu, např. DISTINCT nazevsloupce, SUM(nazevsloupce), nazevsloupce+5 apod

Tables – schema.tabulky, z kterých se bude selectovat; je nutno uvadět všechny tabulky, ne jenom from, ale i ty, které se tykají JOINu

Joins – schema.tabulka.sloupec, podle kterých se bude joinovat – leva a prava strana; type – typ joinu (INNER, OUTER, LEFT, RIGHT, ...); Prvotní tabulku, ke které je join, není třeba uvadět, tzn. tabulku v FROM xxxx klauzuli

DULEZITE: pro cols, tables a joins je vždy třeba uvadět plnou cestu, tzn. schemaname.tablename.colname

Groupby – podle kterých sloupců provádět GROUP BY – stačí název sloupce; interpret

dosadí správný kód

Orderby – podle kterých sloupců provádět ORDER BY – stačí název sloupce; interpreter dosadí správný kód

Where – CDATA – část po WHERE – přímo jako CDATA – umožňuje úplnou customizaci

Having - CDATA – část po WHERE – přímo jako CDATA – umožňuje úplnou customizaci

Params - parametry uvnitř dotazu - jsou zpracovány jako placeholdery v rámci SQL dotazu;

vždy ##PARAMx## kde x je potom parid tag a partype je NUMERIC nebo VARCHAR a content je obsah parametru

VNOŘENÉ DOTAZY - HIERARCHICKÉ STROMY:

Starší verze umožňovala pouze jeden dotaz, novější verze umožňuje dotazy řetěžit do hloubky pomocí pivotálních sloupců - vzniknou tak hierarchické stromy.

Každý dotaz je zabalen do vlastního tagu <query> kde parametry tagu jsou:

name - název dotazu - minimálně 1 musí být "main"

sub-of - jedná se o dotaz vnořený do dotazu, jehož název je stejný jako obsah parametru "sub-of".

```
<query name="main">
→ hlavní dotaz
...
</query>
<query name="sub1" sub-of="main">
-> vnořený dotaz v dotazu main
...
</query>
<query name="sub2" sub-of="sub1">
-> první vnořený dotaz v dotazu sub1
...
</query>
<query name="sub3" sub-of="sub1">
-> druhý vnořený dotaz v dotazu sub1
...
</query>
```

U každého podřízeného dotazu je nutné definovat WHERE parametr, podle které bude subset zvolen; tento parametr má pak místo tagu "content" tag "pivot", a jeho obsahem je "name" sloupce v nadřazeném dotazu.

U každého dotazu je též možné definovat offset a počet řádek, parametry apod. Viz ukázka.

Návratová hodnota v případě synchronního volání:

```
<response sitetype="DEV_001" siteversion="DEV" siteexpires="current"
slick-uniqueid="3">
  <query name="main" depth="1" totalrows="0" numrows="32221" offset="500">
    <sql>
      <![CDATA[
        SELECT MAP.KATASTRALNIUZEMI_P.KATUZE_KOD AS KATASTR,
        MAP.KATASTRALNIUZEMI_P.NAZEV AS NAZEV,
        ST_GEOOMETRY_TO_GML(MAP.KATASTRALNIUZEMI_P.SHAPE) AS SHP_AS_GML FROM
        MAP.KATASTRALNIUZEMI_P WHERE
        MAP.KATASTRALNIUZEMI_P.KATUZE_KOD=:sqlparam_1
      ]]>
    </sql>
    <cols>
      <col>
```

```

        <name>KATASTR</name>
        <code>MAP.MAP_KATASTRALNIUZEMI_P.KATUZE_KOD</code>
    </col>
    <col>
        <name>NAZEV</name>
        <code>MAP.MAP_KATASTRALNIUZEMI_P.NAZEV</code>
    </col>
    <col>
        <name>SHP_AS_GML</name>
        <code>
            ST_GEOMETRY_TO_GML (MAP.MAP_KATASTRALNIUZEMI_P.SHAPE)
        </code>
    </col>
</cols>
</query>
<query name="sub1" depth="2">
    <sql>
        <![CDATA[
            SELECT DMP_CUR.PARCELAKN_DOKM_P.KATUZE_KOD AS KAT,
            DMP_CUR.PARCELAKN_DOKM_P.PARCELA AS PARCELA,
            DMP_CUR.PARCELAKN_DOKM_P.NAZEV_KU AS NAZEV_KATASTRU,
            SUBSTR(DMP_CUR.PARCELAKN_DOKM_P.NAZEV_KU,:sqlparam_4, :sqlparam_5) AS
            NAZEV_KATASTRU_SUBSTR FROM DMP_CUR.PARCELAKN_DOKM_P WHERE
            DMP_CUR.PARCELAKN_DOKM_P.KATUZE_KOD=:sqlparam_2 AND
            DMP_CUR.PARCELAKN_DOKM_P.PARCELA > :sqlparam_3
        ]]>
    </sql>
    <cols>
        <col>
            <name>KAT</name>
            <code>DMP_CUR.PARCELAKN_DOKM_P.KATUZE_KOD</code>
        </col>
        <col>
            <name>PARCELA</name>
            <code>DMP_CUR.PARCELAKN_DOKM_P.PARCELA</code>
        </col>
        <col>
            <name>NAZEV_KATASTRU</name>
            <code>DMP_CUR.PARCELAKN_DOKM_P.NAZEV_KU</code>
        </col>
        <col>
            <name>NAZEV_KATASTRU_SUBSTR</name>
            <code>
                SUBSTR(DMP_CUR.PARCELAKN_DOKM_P.NAZEV_KU,##PARAM4##, ##PARAM5##)
            </code>
        </col>
    </cols>
</query>
<rows>
    <r>
        <c>727008</c>
        <c>
            <![CDATA[ Josefov ]]>
        </c>
        <c>
            <![CDATA[
                &lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;gml:FeatureCollection
                xmlns:gml="http://www.opengis.net/gml"
                xmlns:xlink="http://www.w3.org/1999/xlink"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xmlns:fme="http://www.safe.com/gml/fme"
            ]]>
        </c>
    </r>

```

```

xsi:schemaLocation="http://www.safe.com/gml/fme GML.xsd">
  <gml:boundedBy> <gml:Envelope srsName="EPSG:4326"
srsDimension="2"> <gml:lowerCorner>-742766,2399
-1042526,5</gml:lowerCorner>
<gml:upperCorner>-743195,470199998
-1042948,5202</gml:upperCorner> </gml:Envelope>
</gml:boundedBy><gml:featureMember> <gml:curveProperty>
<gml:LineString srsName="EPSG:4326" srsDimension="2">
<gml:posList> -742798.87020000 -1042726.15000000 -742766.23990000
-1042712.53990000 -742767.98990000 -1042703.28990000 -742776.22020000
-1042659.07020000 -742808.02020000 -1042659.17980000 -742816.56010000
-1042659.16010000 -742816.85000000 -1042607.57980000 -742826.18990000
-1042608.83990000 -742878.53990000 -1042615.73990000 -742885.56010000
-1042726.15000000</gml:posList> </gml:LineString>
</gml:curveProperty>
</gml:featureMember></gml:FeatureCollection>
]]>
</c>
<subresult n="sub1">
  <rows>
    <r>
      <c>727008</c>
      <c>77</c>
      <c>
        <![CDATA[ Josefov ]]>
      </c>
      <c>
        <![CDATA[ Jos ]]>
      </c>
    </r>
    <r>
      <c>727008</c>
      <c>61</c>
      <c>
        <![CDATA[ Josefov ]]>
      </c>
      <c>
        <![CDATA[ Jos ]]>
      </c>
    </r>
  </rows>
</subresult>
</r>
</rows>
</response>

```

Vrací resultset který obsahuje:

tagy <query>

-> metadata obsahující informace o jednotlivých dotazech

atribut "name" ⇒ název dotazu

atribut "depth" ⇒ hloubka zanoření daného dotazu v rámci hierarchického stromu

atribut "totalrows" ⇒ celkový počet záznamů vyhovující dotazu

atribut "numrows" ⇒ zobrazený počet záznamů (LIMIT)

atribut "offset" ⇒ zobrazený počet přeskočených (OFFSET)

query -> cols ⇒ metadata o sloupcích v dotazu

query -> cols -> col -> name ⇒ název sloupce

query -> cols -> col -> code ⇒ SQL kód sloupce

tagy <rows> obsahují jednotlivé záznamy

Rows -> r – jednotlivé řádky s daty

Rows -> r -> c – sloupce a jejich obsah; pokud se jedná o ne-numerickou hodnotu, je zakódován jako CDATA

Rows -> r -> subresult - V každém řádku s daty (<row>) může být vygenerován subset z poddotazu, např.:

```
<subresult n="sub1">
```

```
<rows>
```

```
....
```

```
</rows>
```

```
</subresult>
```

⇒ atribut *n* je název dotazu, který je použit pro získání dat

METODA GetStoredQueryResult (sessionid, queryid, rows, offset, xmlparams);

Tato funkce vrátí v XML datech výsledky dotazu. Dotaz je již předdefinován správcem aplikace a je možné mu předat parametry. Informace o předdefinovaných dotazech jsou získávána pomocí metody GetSchema, v xml datech v XPATH response->storedqueries

Vstupní parametry:	
sessionid	STRING - Identifikátor session, který je výstupem funkce Authenticate()
queryid	INTEGER - identifikátor query - <query id="xxxx"> v XPATH response->storedqueries->query[id]
rows	INTEGER - Počet řádek k vrácení
offset	INTEGER - Offset
xmlparams	STRING – Obsah dotazu formátovaný do XML

XMLParams – formát dat

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <params>
    <param>
      <parid>1</parid>
      <content>1</content>
    </param>
    <param>
      <parid>2</parid>
      <content>%r%</content>
    </param>
```

```

        <param>
            <parid>3</parid>
            <content>1</content>
        </param>
    </params>
</request>

```

XMLParams – popis

XML obsahuje seznam parametrů podle identifikátoru (paramid) a jejich obsahu. Viz možné parametry v response->storedqueries metody GetSchema

Návratová hodnota stejná jako u getComposedQueryResults

Oracle PL-SQL funkce ST_GEOMETRY_TO_GML (G, srsName);

Tato funkce převádí [SDE.ST_GEOMETRY](#) na GML v XML formátu.

UPOZORNĚNÍ: XML výstup z aplikace obsahuje vnořený GML/XML, který je "escapovaný" (např. "<" místo "<" apod.).

Více:

http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references

PHP: <http://www.php.net/manual/en/function.html-entity-decode.php>

.NET

[http://msdn.microsoft.com/en-us/library/System.Web.HttpUtility.HtmlDecode\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/System.Web.HttpUtility.HtmlDecode(v=vs.110).aspx)

Java

[http://commons.apache.org/proper/commons-lang/javadocs/api-2.6/org/apache/commons/lang/StringEscapeUtils.html#unescapeHtml\(java.lang.String\)](http://commons.apache.org/proper/commons-lang/javadocs/api-2.6/org/apache/commons/lang/StringEscapeUtils.html#unescapeHtml(java.lang.String))

Vstupní parametry:	
G	SDE.ST_GEOMETRY
srsName	VARCHAR2 default 'EPSG:4326'

Návratová hodnota:

```
SQL> select ST_GEOMETRY_TO_GML(shape) from gt.fc_test where objectid=1;
```

```

<?xml version="1.0" encoding="utf-8"?>
<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns:fme="http://www.safe.com/gml/fme"
xsi:schemaLocation="http://www.safe.com/gml/fme GML.xsd">
  <gml:boundedBy>
    <gml:Envelope srsName="EPSG:4326" srsDimension="2">
      <gml:lowerCorner>-746571,4399 -1041797,8298</gml:lowerCorner>
      <gml:upperCorner>-746615,0601 -1041856,2</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy><gml:featureMember>
    <gml:curveProperty>
      <gml:Polygon srsName="EPSG:4326" srsDimension="2">
        <gml:posList> -746571.43990000 -1041811.32980000 -746608.66010000
-1041797.82980000 -746613.96010000 -1041820.10000000 -746614.36010000
-1041822.88990000 -746615.06010000 -1041829.57980000 -746614.80000000
-1041835.02980000 -746614.18990000 -1041841.08990000 -746611.63990000
-1041851.87980000 -746609.60000000 -1041854.66010000 -746606.97980000
-1041856.20000000 -746598.27020000 -1041855.68990000 -746598.12020000
-1041853.86010000 -746601.72020000 -1041853.98990000 -746602.01010000
-1041847.25000000 -746598.41010000 -1041847.08990000 -746595.22980000
-1041846.98990000 -746592.21010000 -1041846.87980000 -746589.21010000
-1041846.77980000 -746586.21010000 -1041846.67980000 -746583.97020000
-1041846.61010000 -746571.43990000 -1041811.32980000</gml:posList>
      </gml:Polygon>
    </gml:curveProperty>
  </gml:featureMember></gml:FeatureCollection>

```

Volání funkce s parametrem:

Je potřeba si uvědomit, že datové typy řeší interně aplikace. Tzn. volání jakékoliv PLSQL funkce s parametrem funguje následovně:

```

<where><![CDATA[
UPPER(APP_UAP_V1.V_JEVYGEODATA.GEODATA) like UPPER(##PARAM1##)
]]></where>

```

Není tedy možné použít jednoduché uvozovky:

```

<where><![CDATA[
UPPER(APP_UAP_V1.V_JEVYGEODATA.GEODATA) like UPPER('##PARAM1##')
]]></where>

```