

City of Dallas GIS Public REST Services

A Primer on Access City GIS Data Via the Esri REST API

This document provides a very basic overview of how to structure a REST call in URL format to accomplish:

- (1) convert valid street address (e.g., 1500 MARILLA ST) to a coordinate pair (XY)
- (2) pass a coordinate pair to a REST service to capture an attribute

The document also provides a primer for City of Dallas GIS Services (City GIS) public REST services. These services are published to the internet (www) and may be accessed by anyone. However, City GIS does not offer support for the public use of these services and may modify or remove any service at any time without notification. The Standard Disclaimer for City GIS data use applies:

This product is for informational purposes and may not have been prepared for or be suitable for legal, engineering, or surveying purposes. It does not represent an on-the-ground survey and represents only the approximate relative location of property boundaries.¹

Key Concepts

The following concepts are important to the discussion of leveraging REST for coding events with geographic assignments.

Spatial Reference

A spatial reference system (SRS) or coordinate reference system (CRS) is a coordinate-based local, regional or global system used to locate geographical entities"²

The City of Dallas uses data that is projected in the State Plane Coordinate System (SPCS), North American Datum 1983 (NAD83) using feet as the unit of measurement. Geographic Coordinates (GCS) – commonly referred to as latitude and longitude (lat/long) is a global spatial reference system.³

SRS have standard identifiers that are used to denote them. These are referred to as SRID (spatial reference ID). The SRID used in this document are from the European Petroleum Survey Group (EPSG), one of the major authorities for SRID assignments. The SRID for City of Dallas GIS data is 2276. The SRID for GCS (WGS84) is 4326.

Representational State Transfer (REST)

City GIS uses the Esri implementation of REST. From their API help document:

The ArcGIS Server REST API, short for Representational State Transfer, provides a simple, open Web interface to services hosted by ArcGIS Server. All resources and operations exposed by the

¹ Texas Government Code § 2051.102. The full City GIS disclaimer may be found at:

http://gis.dallascityhall.com/documents/COD_DataDisclaimer.pdf

² https://en.wikipedia.org/wiki/Spatial_reference_system

³ Lat/Long is NOT XY. Longitude (Easting) is the equivalent of X and Latitude (Northing) is Y. Thus, long/lat would be the same as XY. This is important when managing coordinates in REST.

REST API are accessible through a hierarchy of endpoints or Uniform Resource Locators (URLs) for each GIS service published with ArcGIS Server.

When using the REST API, you typically start from a well-known endpoint, which represents the server catalog. [sic]⁴

The public REST URL for the City GIS services catalogue is:

`http://gis.dallascityhall.com/wwwgis/rest/services/`

Accessing REST Resources

REST endpoints may be accessed in multiple ways:

- Using a web browser
- Using desktop software (e.g., ArcGIS for Desktop, Quantum GIS)
- Using an application (e.g., a JavaScript web application)

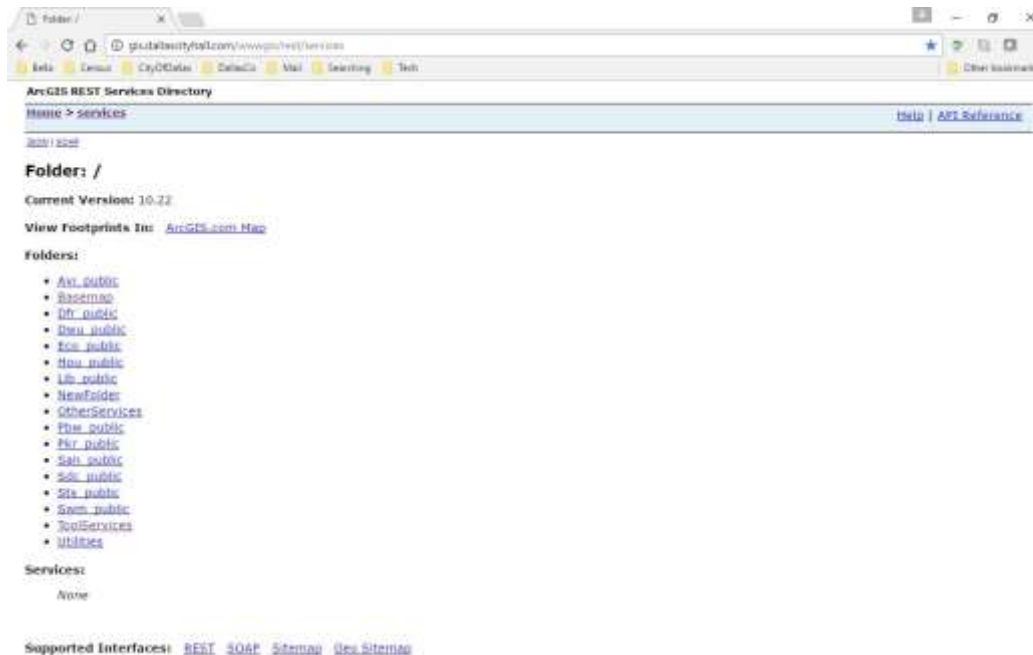
The examples used in this document primarily show web browser navigation and queries. It is assumed that application developers can use the information to write programs to access the REST resources hosted by City GIS.

Syntax

As REST is a highly structured architecture, the syntax is well-defined. Thus, it becomes a matter of passing parameters to build a string that is sent to the REST endpoint and results are returned to the passing agent.

When the well-known endpoint is navigated using a web browser, a directory listing of the server catalogue is displayed.

⁴<http://resources.arcgis.com/en/help/rest/apiref/>



The directory structure organizes the services. For City of Dallas REST services, the primary organizational unit is department. Thus, `Avi_public` is the directory for services that support the Department of Aviation; `Dfr_public` is for Dallas Fire-Rescue; etc.

If you follow (click on) a service directory – like `ToolServices` – the browser URL will add the directory to the path. There is also a navigation bar at the top of the page, which allows the user to jump back any number of steps to a previous stage in the catalogue.

Different types of services have different options. Common types of services are:

- Geocoding service (GeocodeServer)
- Map service (MapServer)

There are other serviced types but these are the two that will be covered in this document and are the two primarily served by City GIS.

Map Service

An example map service will be used to show the structure. This service – `CityInfo` – is a map service used by City GIS for its primary information lookup application and for other uses. As it is heavily used by applications and references data from several departments, it is stored in the `ToolServices` directory.

```
http://gis.dallascityhall.com/wwwgis/rest/services/ToolServices/
CityInfo/MapServer
```

When this directory is navigated to in the browser the layers and other information is displayed.

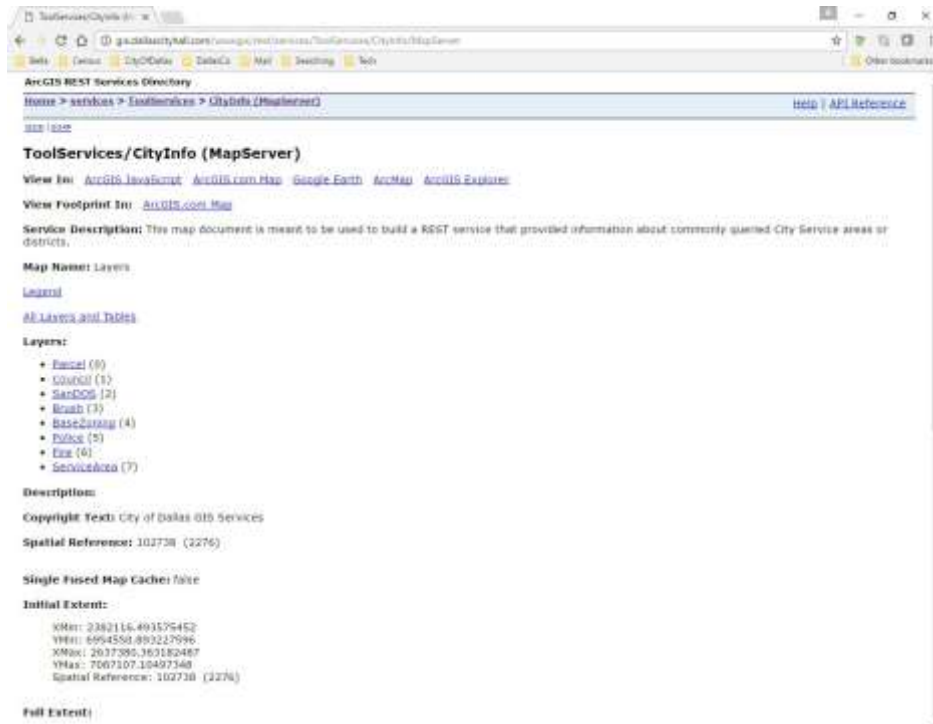


Figure 1: Top of Service Page

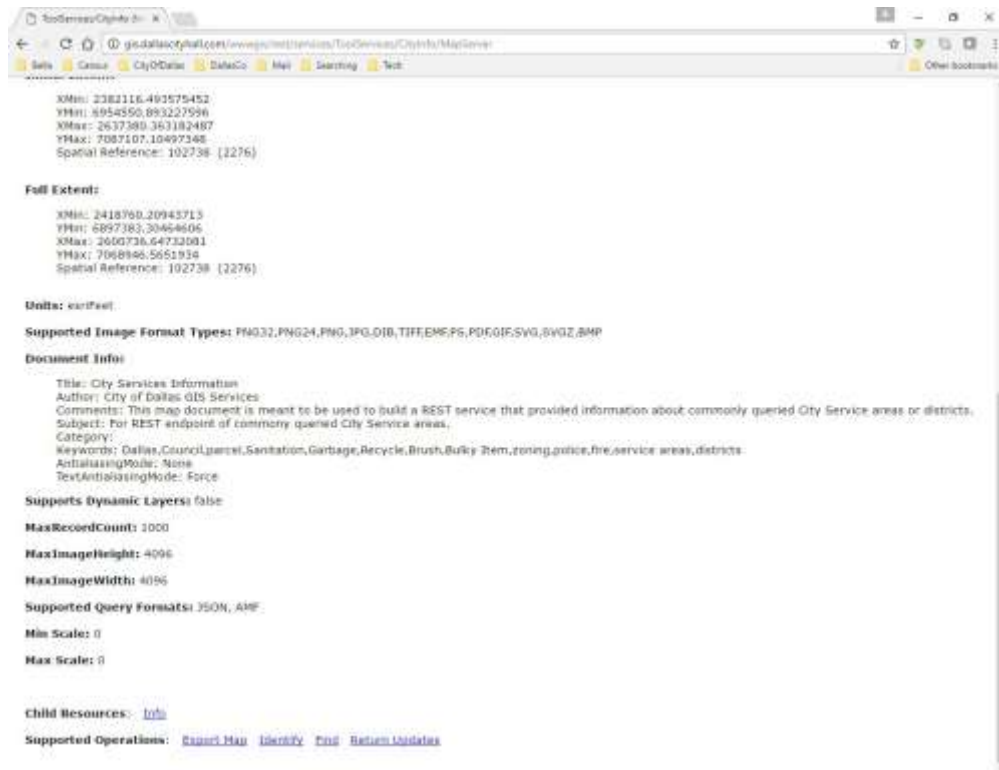


Figure 2: Bottom of service page

Notice that the layers have a reference number, beginning with 0, following the layer name (in parentheses).

Base URL for Find Address Candidates

The DallasStreets locator is the City standard "geocoding" index and is used by CSR to valid Service Request (SR) addresses at intake. The index has been published as a REST services that is publicly available.

```
http://gis.dallascityhall.com/wwwgis/rest/services/ToolServices/  
DallasStreetsLocator/GeocodeServer
```

Find Address Candidates Operation

The Find Address Candidates operation can be used to convert a valid Dallas street address to a coordinate pair (XY). The default coordinate system is SRID 2276 (local State Plane Coordinate System, NAD 1983). However, the output SRID can be specified. However, if all the activity is to occur within the City GIS REST infrastructure, it is not necessary to specific the SRID, as the default will apply to all of the services involved in these examples (below).

The base URL for the Find Address Candidates request to DallasStreetsLocator is:

```
http://gis.dallascityhall.com/wwwgis/rest/services/ToolServices/Dal  
lasStreetsLocator/GeocodeServer/findAddressCandidates/
```

This is the beginning of the request URL. A valid Dallas street address needs to be supplied, using the Street parameter:

```
?Street=
```

The minimum information (recommended – passing additional information can sometimes create a non-match), is address number and street name. For example, 1500 MARILLA ST, could be sent as a string that looks like:

```
1500+marilla+st
```

Notice that the spaces have been replaced with plus signs ("+"). This is useful though not necessarily required. It depends on how the passing application handles strings. This would need to be tested within the passing API.

The last thing to include in the URL is a limit on the number of locations returned. For a single address, one typically only wants the match with the highest "score", or the "first" returned location. The `maxLocations` parameter accomplishes this. After that, the format is set with the `f` parameter to insure that the data is returned in JSON format.

```
&maxLocations=1&f=json
```

Concatenate all of this together and pass the URL to the REST endpoint. This accomplishes a GET request. The string would look like:

```
http://gis.dallascityhall.com/wwwgis/rest/services/ToolServices/Dal  
lasStreetsLocator/GeocodeServer/findAddressCandidates/?Street=1500+  
marilla+st&maxLocations=1&f=json
```

The results of the request are returned as a JSON string. The request example returns:

```

{
  "spatialReference": {
    "wkid": 102738,
    "latestWkid": 2276
  },
  "candidates": [
    {
      "address": "1500 MARILLA ST, DALLAS, 75201",
      "location": {
        "x": 2491521.7919378052,
        "y": 6969611.0765757067
      },
      "score": 100,
      "attributes": {
        }
      }
    ]
  }
}

```

Notice that a standardized address string ("address") and X and Y values are returned. These values may be captured and used in other processes. For the next example, the X and Y values will be captured and passed to another REST service to capture information about this location.

GET Feature Attribute Using Geometry (Query Operation)

With a valid geometry, such as a point coordinate pair (XY), it is possible to determine what/which feature(s) intersect the geometry using the Query function of a REST layer. For example, in the Geocode exercise, above, the location was defined by its XY. If this information is used with a Query operation against a Council District layer in a REST map service, one can determine the Council District assignment (DISTRICT) for this location.

Breaking Down the Query Operation URL

Here is an example URL to query the XY from the previous example against the Council District layer in the CityInfo map service:

```

http://gis.dallascityhall.com/wwwgis/rest/services/ToolServices/
CityInfo/MapServer/1/query?&geometry=2491521.7919378052%2C696961
1.0765757067&geometryType=esriGeometryPoint&spatialRel=esriSpati
alRelIntersects&returnGeometry=false&f=pjson

```

If you look at the query URL in the web browser, you'll see many options. Not all of them are needed and the URL, above, excludes the unused parameters (options). To understand the parameters, the URL is broken down into its components.

Base URL

This is the catalogue level URL for the REST endpoint.

```

http://gis.dallascityhall.com/wwwgis/rest/services

```

Service Directory

Remember, services are organized first by directory.

```

/ToolServices

```

Service Name

The service name is followed by a reference to the service type. The same service can have different types (capabilities).

```
/CityInfo/MapServer
```

Layer Reference

Layers within a service are referenced by the layer number (ID), from 0 to n (last layer).

```
/1
```

Function

In this URL, the query function is being accessed. It needs to be followed by a question mark to start defining options.

```
/query?
```

Query Parameters

Query parameters are added and each one is prefaced with an ampersand (&). You don't have to include every parameter. If you leave off a parameter, the default value will be used. See API documentation for a full list of layer query parameter.⁵

Geometry

For point geometry, the parameter is provided as a coordinate separated by a comma (X,Y), like:

```
&geometry=2491521.7919378052%2C6969611.0765757067
```

Notice that the browser converts the comma to %2C in the URL path in the address window. It may not be necessary to perform this translation when sending parameters as a URL component but may be necessary when using an application to pass parameters, depending on the programming environment.

Geometry Type

It is necessary to indicate what type of geometry is being passed. For point geometry, `esriGeometryPoint` is the type for Esri REST. The full list of geometry types is found in the API reference.

```
&geometryType=esriGeometryPoint
```

Spatial Relationship

The spatial relationship that is applied to the input geometry must be declared. Here, the `intersect` relationship is used. See the API reference for the full list of relationships that can be used.

```
&spatialRel=esriSpatialRelIntersects
```

Output Format

Typically, one would want the query results as a JSON string. This is declared with the `format` parameter. See the API reference for other options.

```
&f=json
```

⁵ <http://resources.arcgis.com/en/help/rest/apiref/query.html>

Results

The results of the query are returned as a JSON string. It looks like:

```
{
  "displayFieldName": "DISTRICT",
  "fieldAliases": {
    "DISTRICT": "DISTRICT"
  },
  "fields": [
    {
      "name": "DISTRICT",
      "type": "esriFieldTypeString",
      "alias": "DISTRICT",
      "length": 15
    }
  ],
  "features": [
    {
      "attributes": {
        "DISTRICT": "2"
      }
    }
  ]
}
```

If using an application to capture the results, the attributes value of the features object is what would be desired. In this case the DISTRICT attribute value "2" would be captured to answer the question: "What Council District does the XY value intersect?"

Using a Web Browser to Perform a Layer Query Operation

The query operation may also be performed using the REST browser interface. The key parameters are supplied by typing them in the appropriate input boxes and then the Query (GET) button is clicked. If the Format is set to HTML, the results are displayed at the bottom of the page.

Query: Council (ID: 1)

Where:

Text:

Object IDs:

Time:

Input Geometry:

Geometry Type:

Input Spatial Reference:

Spatial Relationship:

Relation:

Out Fields:

Return Geometry: True False

Max Allowable Offset:

Geometry Precision:

Output Spatial Reference:

Return IDs Only: True False

Return Count Only: True False

Order By Fields:

Group By Fields (For Statistics):

Output Statistics:

ReturnZ: True False

ReturnM: True False

Geodatabase Version Name:

Return Distinct Values: True False

Format:

records: 1

DISTRICT: 2

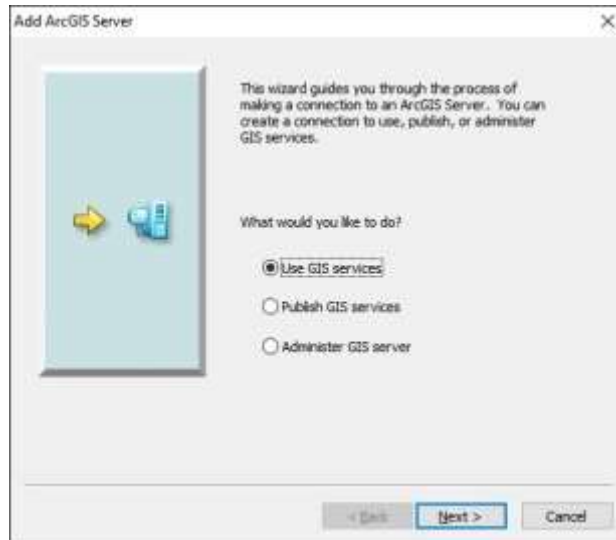
Notice how the entries correspond with the full URL in the address box.

Accessing City GIS REST with Desktop GIS Software

REST web services may be used to display data in desktop GIS software. Unless there are Feature services available, the symbology is not adjustable and there are limits to using the data within the software. Using REST web services is not meant to replace direct feature class access but rather to provide a quick, low memory impact way to access GIS information or to publicly share data that is normally not available outside of the City network.

ArcGIS for Desktop

In ArcCatalog, double-click "Add ArcGIS Server" in the GIS Servers directory in the Catalog Tree. Select "Use GIS Services" and click Next.



In the Server URL box, enter:

```
http://gis.dallascityhall.com/wwwgis
```

Click OK to add the ArcGIS Server connection. An entry will be added to the GIS Servers directory. The directory structure will be the same as the one in the first example where the REST endpoint was navigated using a web browser.

The layers in the various services may be added to an ArcMap session using the Add Data operation.

See the user documentation for ArcGIS for Desktop for more information on using web services in a map document.

Quantum GIS⁶

Quantum GIS or QGIS is a free GIS desktop software application. Visit the QGIS website for instructions on downloading and using the software. Version 2.0 or higher is needed to access Esri REST map service (MapServer) layers.

⁶ <http://www.qgis.org/en/site/>

Resources

GeoServices REST Specification Version 1.0

An Esri® White Paper, September 2010

<https://www.esri.com/library/whitepapers/pdfs/geoservices-rest-spec.pdf>

ArcGIS REST API

Resource Center

<http://resources.arcgis.com/en/help/arcgis-rest-api/>

ArcGIS Server REST API

API Reference

<http://resources.arcgis.com/en/help/rest/apiref/>

PhD Dissertation by Roy Fielding

2000 UC Irvine Dissertation (Chapter 5) which introduced REST

http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm