GeoDa Center - Dynamic Tile Mapper Web Service - Documentation

Author: Charles R. Schmidt, charles.r.schmidt@asu.edu
API Version: 0.2 Alpha Release
Document Date: August 23rd 2010

Introduction
This documentation describes the Dynamic Tile Mapper (DTM) Web Service. The documentation is subject to rapid change until we reach a 1.0 release, at which point we will stop making backwards incompatible changes to the APIs without sufficient notice. We greatly value your feedback and suggestions, which can be sent to Charles.R.Schmidt@asu.edu. "Dynamic Tile Mapper" may be referred to as "DynTM" or "DTM" throughout this documentation. The Dynamic Tile Mapper (DTM) Web Service is currently under going a major refactoring. During this process the API is being overhauled and greatly expanded. As well the service is in the process of being ported to Amazon's Elastic Compute Cloud (EC2). During this transition expect outages and disruptions. We will do our best to ensure that previous versions continue to operate as described. More information on the DTM service can be found in the working paper located at, http://geodacenter.asu.edu/node/175.

Changes since v0.1:
- Documentation for various web services was split into separate documents.
    - This document covers only the Dynamic Tile Mapper Web Service.
- Authentication:
    - DynTM now requires authentication. The authentication scheme will be described in depth below.
- TileSets
    - You can now create new TileSets (Shapefiles) via the API.
- Region IDs
    - Regions are no longer assigned unique ids, they are simply kept in the same order as the original shapefile
    - Future versions will likely bring region ids back and allow for multiple sets of ids. In the meantine application developers are responsible for maintaining record order.

Dynamic Tile Mapper Web Service
- Overview:
    - This service provides map tiles that can be used in OpenLayers, Google Maps API and other tile based web maps.
    - DTM consists of three primary data models.
        - TileSets: A TileSet describes a set of geographic regions. It is the rough equivalent to an ESRI Shapefile.
        - Classifications: A Classification is associate with a particular Tile Set and describes a set of classes and the regions that belong to them. The max number of classes is 250.
        - ColorSchemes: A Color Scheme is simply a list of ***n*** colors and can be associated with any classification that contains ***n*** number of classes.
    - Four service end-points will be described:
        - The first end-point allows you to create, list and delete TileSets.
        - The seconds allows manipulating Classifications
        - The third allows manipulating ColorSchemes
        - The fourth allows you to access the Tiles.

1 DynTM Authentication

1.1 Accounts

Please contact *charles.r.schmidt@asu.edu* to request an account.

1.2 AccessKeys

The Access Key interface is not yet exposed, in the meantime contact *charles.r.schmidt@asu.edu* to request new Access Keys.

1.3 Signing Requests

The DynTM web services now requires most requests to be authenticated. The only requests that do not require authentication are requests to view individual tiles. A DynTM user has permission to create, list and delete data. User's have a username and password which they can use to create AccessKeys. AccessKeys are shared secrets identified by an AccessKeyID. The user will sign each request with the shared secret. DynTM will compare the signature with one calculated by the server. If the signatures match the request is allowed to proceed. A user may create multiple AccessKeys, allowing keys to be rotated in and out of service without disrupting your applications. Any requests with a valid signature will be processed with the permissions of the user who owns the AccessKey. (All of your AccessKeys carry the same permissions).

Requests are signed by setting the HTTP "`Authorization`" header to "`GeoDaWS AccessKeyID:Signature`" where,

```
Signature = Base64( HMAC-SHA1( UTF-8-Encoding-Of( AccessKey,StringToSign ) )
)

StringToSign = HTTP-Verb + "\n" +
               Content-MD5 + "\n" +
               Content-Type + "\n" +
               Date + "\n" +
               ResourceSelector

ResourceSelector = "/path/to/resource"

Example of StringToSign: "GET\n\ntext/plain\nThu, 01 Apr 2010 01:23:10 GMT\n/
dict/user"
```

This scheme is based on Amazon Web Services REST Authentication. More information can be found here: http://docs.amazonwebservices.com/AmazonS3/latest/index.html?RESTAuthentication.html

2 Dynamic Tile Mapper Service Interfaces

2.1 TileSets Interface

Background
TileSets describe a set of geographic regions and are the rough equivalent to an ESRI Shapefile. Currently DynTM only supports creating TileSets from an ESRI shapefile. Future releases may support other file formats and input types.

2.1.1 List TileSets

Service Address
http://apps.geodacenter.org/dyntm.json/tileset/

Authentication
Required

Description
Returns a list of tile sets owned by the authenticated user.

Request Method
GET

Parameters
None.

Example
Request: `http://apps.geodacenter.org/dyntm.json/tileset`
Response: `{"tilesets": ["charlie:CBC583666DF594EBC6641C89A5434B5A"]}`

2.1.2 Describe a TileSet

Service Address
http://apps.geodacenter.org/dyntm.json/tileset/*TileSetID*

Authentication
Required

Description
Return metadata about the requested tileset.

Request Method
GET

Parameters
TileSetID (required) - The ID of the TileSet to be described.

Example
Request: `http://apps.geodacenter.org/dyntm.json/tileset/charlie:CBC583666DF594EBC6641C89A5434B5A`
Response: `{'owner': 'charlie', 'date': 1279326159.0, 'shpfile': 'CBC583666DF594EBC6641C89A5434B5A', 'numregions': 3111, 'public': True}`

2.1.3 Create TileSet From ShapeFile

Service Address
http://apps.geodacenter.org/dyntm.json/tileset

Authentication

Required

Description
Create a new TileSet from an ESRI Shapefile. To create a tileset you need to read in the ".shp" and ".shx" files of your shapefile as binary strings. The binary strings need to be compressed with standard zlib compression, than base64 encoded. The TileSetID returned will be USENAME:MD5_OF_SHAPEFILE. You can verify transmission by comparing the MD5 of the uncompress binary string of the '.shp' file to the MD5 in the TileSetID.

Request Method
POST

Parameters
```
shp = Base64(zlib_compress(shp))
shx = Base64(zlib_compress(shx))
```

Example
Request: POST `shp` and `shx` to `http://apps.geodacenter.org/dyntm.json/tileset`
Response: `{"TileSetID":"charlie:CBC583666DF594EBC6641C89A5434B5A"}`

2.1.4 Delete a TileSet

Service Address
http://apps.geodacenter.org/dyntm.json/tileset/*TileSetID*

Authentication
Required

Description
Remove a TileSet.

Request Method
DELETE

Parameters
TileSetID (required) - The ID of the TileSet to be deleted.

Example
Request: DELETE `http://apps.geodacenter.org/dyntm.json/tileset/charlie:CBC583666DF594EBC6641C89A5434B5A`
Response: `true`

2.2 Classification Interface

Background
Classifications describe a set of classes and the geographic regions which belong to each class. A classification is associated with a particular TileSet. Each class is identified by an integer between 2 and 255, thus the maximum number of classes is 253. Class 0 represents the background. Class 1 represents the borders. You CAN assign regions to background border classes, which can have interesting cartographic effects. The number of classes in your classification is calculated as the 1+max(class id). This allows gaps in the classification, which results in unused colors in the color scheme.

For example...

```
a = [4,4,2,3,4,2]
b = [4,4,2,2,4,2]
n = 1+max(a)
n == 1+max(b)
```

...both classification `a` and `b` are considered to contain 5 (`n`) classes. In classification `b`, class 3 is empty. Both classifications will require a color scheme that contains `n` colors (background + border + class 2 + class 3 + class 4).

Classifications are the foundation of DynTM and the most basic way to manipulate a map. Classifications should be considered cheap and disposable, create as many as you need. Our cost to store 10,000 different classifications of the US Census tracts (~65,000 regions) is approximately $0.25 per month stored.

2.2.1 List Classifications

Service Address
D

Authentication
Required

Description
Returns a list of classifications owned by the authenticated user.

Request Method
GET

Parameters
None.

Example
Request: `http://apps.geodacenter.org/dyntm.json/cl`
Response: `{"classifications": ["charlie:474f117491ba503bde2b1eb2b9a36b3a"]}`

2.2.2 Describe Classification

Service Address
http://apps.geodacenter.org/dyntm.json/cl/*ClassificationID*

Authentication
Required

Description
Return metadata about the requested classification.

Request Method
GET

Parameters
ClassificationID (required) - The ID of the Classification to be described.

Example
Request: `http://apps.geodacenter.org/dyntm.json/cl/charlie:474f117491ba503bde2b1eb2b9a36b3a`
Response: `{'tileset': 'charlie:B9C118EB30A9EAE19C93902E9F01EF8A', 'expires': False, 'N': 80, 'date': 1279756798.0, 'owner': 'charlie', 'n': 4, 'public': True}`

2.2.3 Create a Classification

Service Address
http://apps.geodacenter.org/dyntm.json/cl

Authentication
Required

Description
Classifications are submitted as ordered lists. The length of the list must equal the number of regions in the TileSet (shapefile). The order of the list must match the order of the regions in the original shapefile. The i^{th} entry in the classification is the class id of the i^{th} region in the shapefile.

The list must be properly backed to transmit to the web service. Currently the following two formats are supported:

`csv` -- Comma Separated Values, eg. "`4,4,2,3,4,2`"
`b64zlib` -- base64_encode(zlib_compress(array of unsigned bytes))

The `b64zlib` format can reduce the payload size by as much as 75% when compared `csv`.

Request Method
POST

Parameters
format (required) -- "`csv`" or "`b64zlib`"
dat (required) -- Either the encoded array or comma separated list.
tsid (required) -- ID of the associated TileSet
background (optional) -- Defaults to 0, experimental
borders (optional) -- Defaults to 1, experimental

Example
Request: POST `format,dat,tsid` to `http://apps.geodacenter.org/dyntm.json/cl/`
Response: `{"TileSetID":"charlie:474f117491ba503bde2b1eb2b9a36b3a"}`

2.2.4 Delete a Classification

Service Address
http://apps.geodacenter.org/dyntm.json/cl/*ClassificationID*

Authentication
Required

Description
Remove a classification.

Request Method
DELETE

Parameters
ClassificationID (required) - The ID of the Classification to be deleted.

Example
Request: DELETE `http://apps.geodacenter.org/dyntm.json/cl/charlie:474f117491ba503bde2b1eb2b9a36b3a`
Response: `true`

2.3 Color Scheme Interface

Background
Color Schemes map class ids to colors. Colors are values are represented as RGB hex triplets and may optionally be prepended with the '#' symbol.
For more details see, <http://en.wikipedia.org/wiki/Web_colors#Hex_triplet>. The colors should be packed into a comma separated list. For Example,

```
colors = "#FF0000,#00FF00,#0000FF"
```

2.3.1 List Color Schemes
Service Address
http://apps.geodacenter.org/dyntm.json/colors

Authentication
Required

Description
Returns a list of all color schemes.

Request Method
GET

Parameters
numclasses (options) -- return only color schemes with this many colors (including background and border).

Example
Request: `http://apps.geodacenter.org/dyntm.json/colors`
Response: `{"colorschemes":['charlie:54332b0be258fcb80279eb79612504e2', 'charlie:a2d1bec3b7ba89bf832165c64382ef5d']}`

2.3.2 Describe a Color Scheme
Service Address
http://apps.geodacenter.org/dyntm.json/colors/*ColorSchemeID*

Authentication
Required

Description

Return metadata about the requested color scheme.

Request Method
GET

Parameters
ColorSchemeID (required) - The ID of the ColorScheme to be described.

Example
Request: `http://apps.geodacenter.org/dyntm.json/colors/charlie:54332b0be258fcb80279eb79612504e2`
Response: `'{"alphas": [0], "expires": false, "n": 9, "date": 1279678832.0, "colors": ["#000000", "#000001", "#B2182B", "#EF8A62", "#FDDBC7", "#F7F7F7", "#D1E5F0", "#67A9CF", "#2166AC"], "owner": "charlie", "public": true}'`

2.3.3 Create a Color Scheme
Service Address
http://apps.geodacenter.org/dyntm.json/colors

Authentication
Required

Description
Create a color scheme.

Request Method
POST

Parameters
colors (required) - A comma separated list of RGB hex triplets.
background (optional) - Background color, defaults to transparent.
borders (optional) - Border color, defaults to black.

Example
Request: POST `colors,background,border` to `http://apps.geodacenter.org/dyntm.json/colors`
Response: `{"ColorSchemeID":"charlie:54332b0be258fcb80279eb79612504e2"}`

2.3.4 Delete a Color Scheme
Service Address
http://apps.geodacenter.org/dyntm.json/colors/ColorSchemeID

Authentication
Required

Description
Remove a color scheme.

Request Method
DELETE

Parameters
ColorSchemeID (required) - The ID of the ColorScheme to be deleted.

Example
Request: DELETE `http://apps.geodacenter.org/dyntm.json/colors/charlie:54332b0be258fcb80279eb79612504e2`
Response: `true`

3 Public Interfaces

3.1 Tile Interface

3.1.1 GetTile

Service Address
http://apps.geodacenter.org/dyntm/t

Authentication
Required

Description
Get a Tile.

Request Method
GET

Parameters
ts (required) -- TileSetID
cl (optional) -- ClassificationID
cs (optional) -- ColorSchemeID
z (required) -- Zoom Level
x (required) -- X coordinate
y (required) -- Y coordinate

Example
Request:
```
http://apps.geodacenter.org/dyntm/t/
?ts=charlie:242229211A58F2C60FFF42055EDB3D4B&

 cl=charlie:f80cf153cad07e8426dffdc7e0652f02&

 cs=charlie:43a7a81d4f47831930f95a7cc37c83a1&
                                          x=3&y=6&z=4
```
Response: