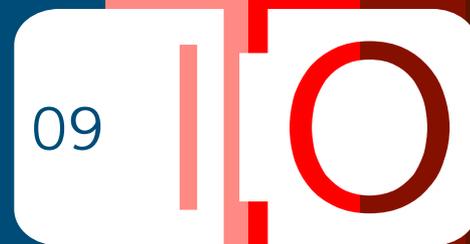


Google™



Building Scalable Geo Web Applications

Mano Marks, Keith Golden, Jeffrey Sambells
May 27th, 2009

Post your questions for this talk on Google Moderator:
code.google.com/events/io/questions



Difficulties in Serving Data

- Client
 - Google takes care of it
- Users
 - The more you have, the harder it is to serve
 - More Users == More Machines
- Machines
 - More Machines ==
 - Load Balancing
 - Maintenance
 - Time
 - Sharding
 - Energy
 - Money

Difficulties in Serving Data

- Multiple format output
- Specialized spatial queries
- Highly Customizable
- Relies on open geo standards

What Google Offers

Building in our cloud

	Multiple output formats	Specialized spatial queries	Customiz-able	Serve large number of requests	Relies on open geo standards
App Engine	✓		✓	✓	
Maps Data API	✓	Soon	Some	✓	✓
Google Base	✓	✓	Some	✓	✓



Google Base API



Google Base

A giant XML store

- Backend for a lot of Google Services
- Easily discoverable
- Bounding box and radius searches
- Google Data API (built on AtomPub)
- Can be accessed as just a feed of XML
- Multiple client libraries
- Standard and customizable attributes
- XML or JSON output
- Upload time can be long

Putting an Item into Base

```
POST /base/feeds/items
Content-Type: application/atom+xml
Authorization: AuthSub token=...
X-Google-Key: key=...

<?xml version='1.0'?>
<entry xmlns='http://www.w3.org/2005/Atom'
  xmlns:g='http://base.google.com/ns/1.0'>
<title type="text">MyDate</title>
  <g:item_type type="text">Personals</g:item_type>
  <g:location>San Francisco, CA</g:location>
  <g:gender>male</g:gender>
  <g:marital_status>single</g:marital_status>
  ...
</entry>
```

Spatial Query

[http://www.google.com/base/feeds/snippets?bq=\[item%20type:personals\]\[location\(location\):@+34-086..@+37-092+%205mi\]&content=geocodes](http://www.google.com/base/feeds/snippets?bq=[item%20type:personals][location(location):@+34-086..@+37-092+%205mi]&content=geocodes)

<http://www.google.com/base/feeds/snippets>

[bq=\[item 20type:personals\]](#)

[\[location\(location\):@+34-086..@+37-092+ 5mi\]&content=geocodes](#)



Google App Engine



Google App Engine

- High performance
- High availability
- Easy Development
- Python or Java environments
- Free to get started
- Built on Google Infrastructure: Work how we work

Advantages

- Built for scaling
 - Build it once and we handle the rest
 - No need (or transparency) for machine management
 - No need for your own servers
- Replicated in data centers around the world
- Use standard Python or Java libraries
- Easy authentication built in
- Highly customizable

Disadvantages for a Spatial Apps

- Very limited spatial support in queries
- Inequality limitations on one property only
 - So no obvious bounding box
- Different (but cooler) way of building applications

Simple Model

```
from google.appengine.ext import db

class myLocation:

    name = db.StringProperty()
    location = db.GeoPtProperty()
```

Simple Query

```
from google.appengine.ext import db
import mylocation

myLocations = MyLocation.gql("WHERE name= :1", 'foo')
```

Geohashing

- Open Source
- Simple radius search
- Uses only one inequality query

Geo Model

- Divide world into 4X4 grid (like a quad-tree but 16)
 - Divide each cell into a 4X4 grid
 - Keep going until you reach maximum number of layers
- Assign your entity to a cell at each zoom level
- Compute cells for each bounding box query
- Find all entities in each cell
- Order them
- Display them

Geo Model Advantages

- Doesn't use inequality query
- Customizable different zoom levels, resolution, grids

Geo Model Disadvantages

- Complex
- Doesn't scale well to other geometry types

Geo Model Calculating Cell

```
while len(cell) < resolution:
    subcell_lon_span = (east - west) / GEOCELL_GRID_SIZE
    subcell_lat_span = (north - south) / GEOCELL_GRID_SIZE
    x = min(int(GEOCELL_GRID_SIZE * (point.lon - west) /
               (east - west)), GEOCELL_GRID_SIZE - 1)
    y = min(int(GEOCELL_GRID_SIZE * (point.lat - south) /
               (north - south)), GEOCELL_GRID_SIZE - 1)
    cell += _subdiv_char((x,y)) # Adds x/y pos of a cell
    south += subcell_lat_span * y
    north = south + subcell_lat_span
    west += subcell_lon_span * x
    east = west + subcell_lon_span
return cell
```



Demo





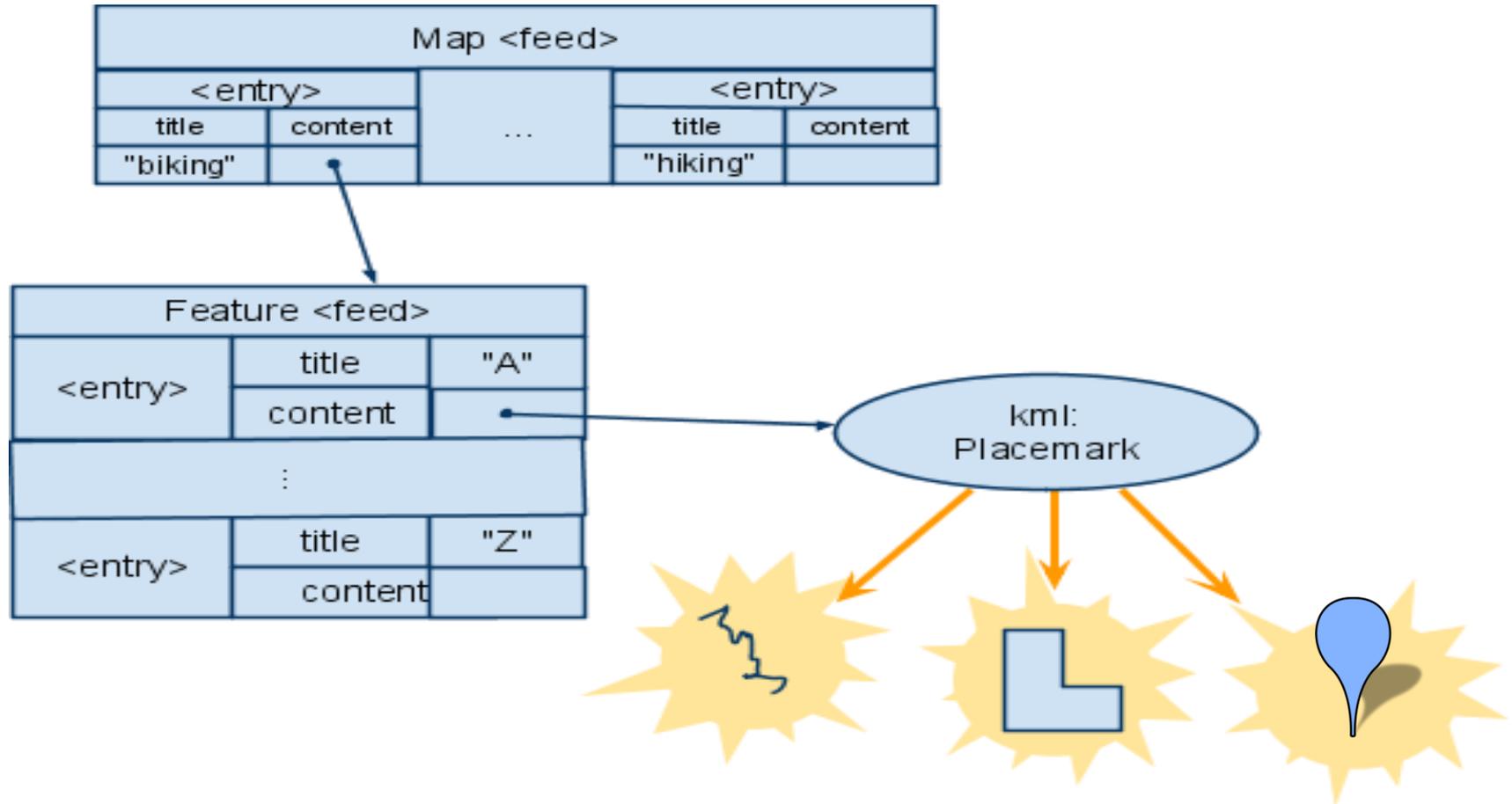
Google Maps Data API



Google Maps Data API

- Google Data API = AtomPub
 - XML/HTTP protocol for reading and writing to data feeds.
 - <http://code.google.com/apis/gdata/>
- Google Maps Data API offers
 - Storage of geospatial data
 - Point, polyline and polygon geometry, with more to come
 - Create and edit My Maps
 - Content optionally indexed and searchable in Google Maps
 - Unlimited user-defined attributes
 - Batch operations
 - Incremental upload and download of changes
 - Client libraries in a variety of languages

Google Maps Data API



Google Maps Data API

- Requests require authentication token in HTTP headers
 - AuthSub, OAuth
 - ClientLogin: service name is "local"
- My maps:
 - <http://maps.google.com/maps/feeds/maps/default/full>

Maps feed

<http://maps.google.com/maps/feeds/maps/default/full>

- GET: list my maps
- POST: add a new map
- PUT: edit the metadata of a map
- DELETE: remove a map

- Set title and summary
- Set public/unlisted (draft) stAccess the feed of features in the map

Features feed

<http://maps.google.com/maps/feeds/features/...>

GET: list the contents of the map

POST: add a new feature to the map

PUT: edit a feature

DELETE: remove a feature from a map

- Set the title
 - Point, line or polygon geometry
 - Embedded `<Style>` to control appearance
 - HTML description
 - Associate custom properties with the feature

Adding a feature to a map

HTTP POST an entry to the edit link of the feature feed

```
<atom:entry xmlns:atom='http://www.w3.org/2005/Atom'
             xmlns:gact="http://schemas.google.com/maps/2008"
             xmlns='http://www.opengis.net/kml/2.2'>
  <atom:title>Random place of the day</atom:title>
  <atom:content type='application/vnd.google-earth.kml+xml'>
    <Placemark>
      <atom:author>
        <atom:name>John Q. Random</atom:name>
      </atom:author>
      <description> Some place </description>
      <Style> ... </Style>
      <Point>
        <coordinates>
          -90.869489,48.254501,0.0
        </coordinates>
      </Point>
    </Placemark>
  </atom:content>
</atom:entry>
```



Controlling which content is indexed

Not all your content need be indexed

```
<entry>
  <category term="http://schemas.google.com/maps/2008#map"
            scheme="http://schemas.google.com/g/2005#kind"/>
  <app:control xmlns:app="http://www.w3.org/2007/app">
    <app:draft>yes</app:draft>
  </app:control>
  <title>MyMapsPoly</title>
  ...
</entry>
```

You decide

Controlling attribution

By default

- Content is attributed to the account it was created under
- Links from search results display the content in Google Maps

If you want different behavior, use KML attribution

- `atom:author` to specify the true author
- `atom:link` to link back to your site

Client Libraries

- No need to talk HTTP directly
- Client libraries in Java, Javascript, Objective-C
- Webready PHP library from We-Create
- Use libkml for KML processing
<http://code.google.com/p/libkml/>

Java example

```
URL queryUrl =
    new URL("http://maps.google.com/maps/feeds/features/"
        + MAPID + "/full");

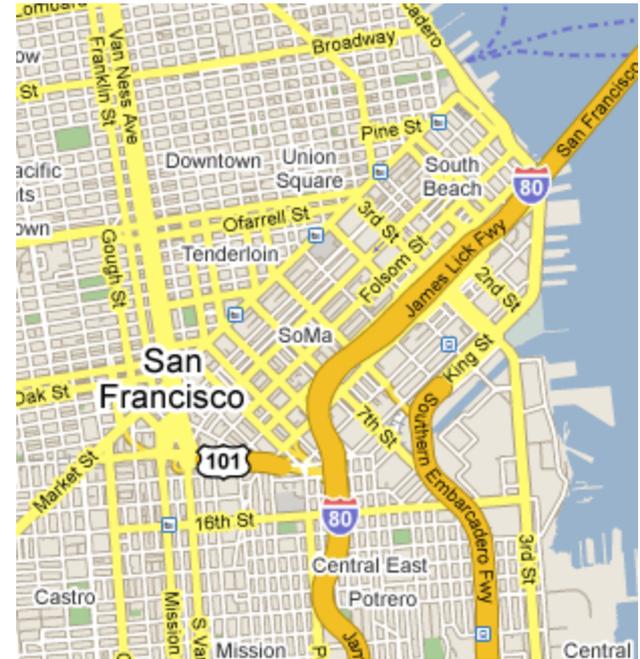
GoogleService myService = new MapsService("exCo-exApp-1");
myService.setUserCredentials(USERNAME, PASSWORD);

FeatureFeed feed = myService.getFeed(queryUrl,
    FeatureFeed.class);

for (FeatureEntry entry : feed.getEntries()) {
    entry.addCustomProperty(
        new CustomProperty("annotation", null, null,
            entry.getTitle()));
    myService.update(new URL(entry.getEditLink().getHref()),
        entry);
}
```

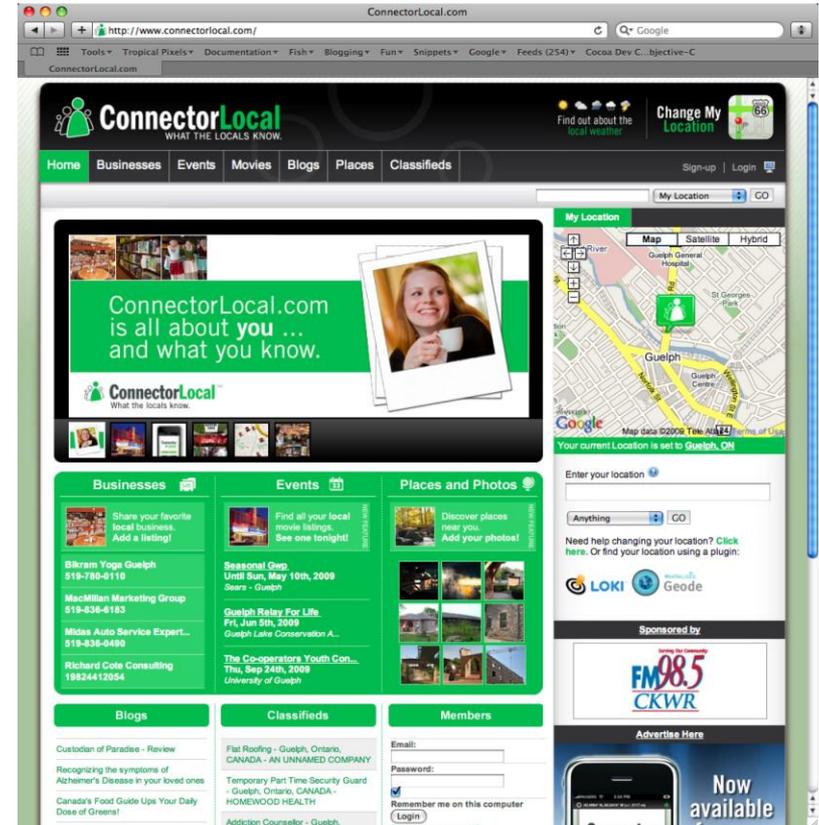
Road Map

- There's more to come
 - Search via the API
 - by keyword
 - by attribute
 - by location
 - Richer content
 - Better Maps API integration
 - More structure
 - New feeds



ConnectorLocal.com

- A service that educates and informs people about the places where they live, work and visit.
- Trusted hyperlocal information from multiple sources
- Users can quickly discover what is engaging in the world around them.



Google Maps Data API

Helping ConnectorLocal Users

- Users can share existing My Maps with ConnectorLocal.
- Users can share their information from ConnectorLocal with My Maps.
- Example: <http://www.connectorlocal.com/places/mymaps>

Google Maps Data API

Helping Our Developers

- All information in ConnectorLocal can be added to the search index with geographic context.
- We can easily access existing geographic information through a common API.

We-Crete Inc. Helping You Code

- Library developed by We-Crete: <http://wecrete.com/ces/webready>
- Built with PHP on top of Zend Framework
- Wraps Google Maps Data API (KML, etc.)
- New BSD License

Authenticating

- Authenticate using Zend_Gdata_AuthSub or Zend_Gdata_ClientLogin and instantiate a new service

```
// Authenticate using the ClientLogin
$client = Zend_Gdata_ClientLogin::getHttpClient(
    'USERNAME',
    'PASSWORD',
    Webready_Gdata_Maps::AUTH_SERVICE_NAME
);

// Connect to the Google Maps Data API
$service = new Webready_Gdata_Maps( $client );
```



Retrieving Maps

```
// Assuming $service is an existing Webready_Gdata_Maps object
```

```
$query = $service->newMapQuery();
```

```
$feed = $service->getMapsMapFeed($query);
```

```
echo "<h2>Available Maps</h2>\n";
```

```
echo "<ul>\n";
```

```
foreach($feed as $entry) {  
    echo "\t<li><a href=\"{"$entry->id}\">  
        {"$entry->title}</a></li>\n";  
}  
echo "</ul>\n";
```



Retrieving Features

```
// Assuming $service is an existing Webready_Gdata_Maps object

$query = $service->newFeatureQuery();

$query->setMapElementId('YOUR_MAP_ID');

$feed = $service->getMapsFeatureFeed($query);

echo "<h2>Available features</h2>\n";
echo "<ul>\n";

foreach($feed as $entry) {
    echo "\t<li><a href=\"{\$entry->id}\">
        {\$entry->title}</a></li>\n";
}

echo "</ul>\n";
```



Creating a New Map

```
// Assuming $service is a Webready_Gdata_Maps object
```

```
$map = $service->newMapEntry();
```

```
$map->newTitle( 'My Google I/O Map' );
```

```
$map->newSummary( 'Hi Everyone! Pretty neat eh?' );
```

```
$newMap = $service->insertMap( $map );
```



Adding Features

```
// Assuming $featureFeed is a Webready_Gdata_Maps_FeatureFeed

$feature = $service->newFeatureEntry();
$feature->newTitle( 'Google I/O 2009 Feature' );

$Placemark = new Webready_Gdata_Extension_Placemark();
$Placemark->newName( 'Moscone Center' );
$Placemark->newPointWithCoordinates(
    '-122.40155,37.78419,0.0'
);

$feature->newContentWithPlacemark( $Placemark );

// Insert the using the feed's POST url
$service->insertMapFeature(
    $feature,
    $featureFeed->getPostUrl()
);
```



Q & A

Post your [questions](#) for this talk on Google Moderator:
code.google.com/events/io/questions

Google™

