

What

[Who](#)

[Where](#)

[When](#)



The Premier Ajax Event of 2006
www.theajaxexperience.com

May 10-12th, 2006
San Francisco, CA

[Google APIs in Ajax applications: Where and When](#) Patrick Chanezon

May 12th from 9am to 10:30am
At the Westin St. Francis
335 Powell Street, San Francisco, CA 94102



P@ in a slide

API Evangelist at Google

- Software plumber

Family man: married, 3 small kids

Java geek... in scripting rehab

- Ruby, JavaScript, Groovy

Open Source

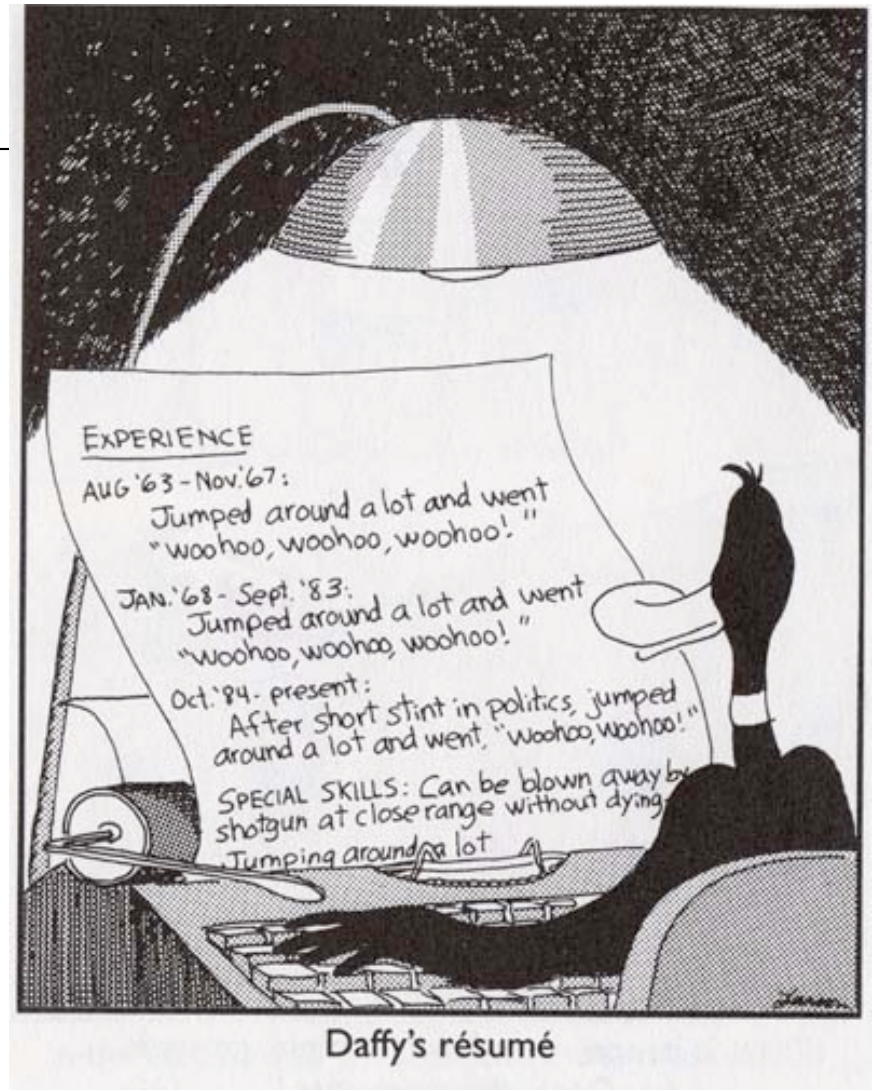
- ROME, adwords4r

Sun

- Blogs, Portals, eCommerce

Netscape/AOL

- LDAP, Calendar, App Servers, CMS, MyNetscape (RSS)



Google APIs

Why are we doing do it?

- Our mission: “Organize the world’s information and make it universally accessible and useful”
 - We can’t organize it all ourselves

14 Google APIs on <http://code.google.com...> and counting!

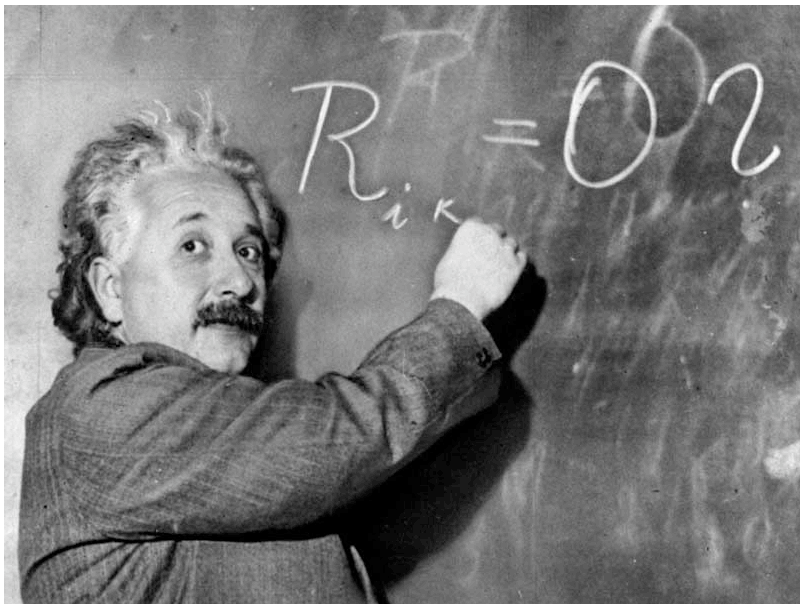
- | | |
|------------------|------------------|
| –Google Data API | –Enterprise APIs |
| –Calendar API | –Homepage API |
| –Maps API | –Related Links |
| –AdWords API | –Sitemaps |
| –Blogger API | –Talk (XMPP) |
| –Data APIs | –Toolbar API |
| –Desktop SDK | –Web Search API |
| –Earth (KML) | |



Where and When?

Space and Time

- 2 important dimensions to organize and present information



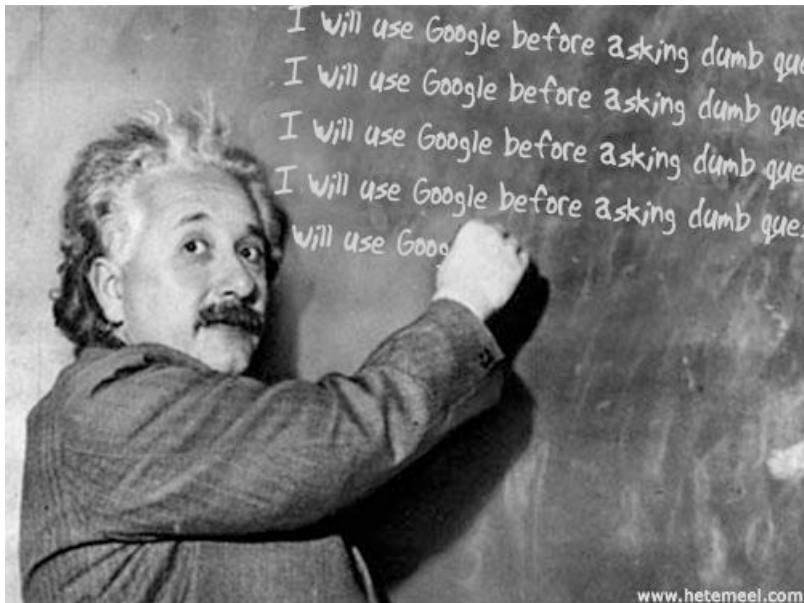
"Space and time are not conditions in which we live; they are simply modes in which we think."

Albert Einstein

How?

This is fine and dandy but how do I use that in my webapp?

- APIs = Plumbing



"If I had my life to live over again, I'd be a plumber."

Albert Einstein

Where and When Google style

Google Maps API

- Triggered the Ajax revolution
- Now everybody's doing it

Google Maps API

- HTML, JavaScript, (XML)

Google Calendar

- An Ajaxy Calendar

Calendar APIs

- REST is the best: HTTP verbs for the rest of us
- Atom syntax: GET
- Atom Publishing Protocol: POST, PUT, DELETE
- Calendar is the first Google Data API



Google Maps and Ajax

AJAX Overview

Google Maps Front End Architecture

Interesting Google Maps AJAX Tricks

- VPages: supporting the back button
- Copyrights: RPC across JavaScript security domains
- API bootstrap: JavaScript API distribution and versioning



What's in a name?

AJAX

- Asynchronous
- JavaScript
- XML

... but

- Only the second part (JavaScript) is really used in Google Maps
- All are completely independent
- All existed in browsers for years before Suggest, Gmail, and Maps
- XML? A data format decision that is completely unimportant relative to the other two
- But a bad name is better than no name!
- Ajax is a great Meme: thanks Jesse James Garrett!

What makes an AJAX Application?

Classic web application

- User clicks on a link / submits a form
- HTTP request
- HTTP response **replaces** the document

AJAX web application

- User triggers an event (mouse click, keyboard click, etc)
- Event invokes scripted event handler
- Event handler *may* initiate a data transfer
- Event handler or data transfer callback **updates** the document

Consequences of AJAX-style development

Sophisticated user interaction

- Display can be partially updated, modified, or animated
- Complex manipulations of page state and UI are possible

Client-side session state

- Transient session state stored on the client
- Persistent user state stored on the server
- *Enables a much more natural architecture for applications*

Technical Components

DOM – Document Object Model

- API for structured text

CSS – Cascading Style Sheets

- Defines visual layout properties, etc.

JavaScript

- Flexible scripting language that is both powerful and fragile

HTTP

- Transport for background data transfer
- Typically done with IFRAME elements or XMLHttpRequest

Transfer Data Format

- JavaScript object literals (JSON) or XML

Illustration: XML vs. JSON – Text Format

XML

```
<data>
<location lon="13.4156" lat="52.5206"/>
</data>
```

JSON

```
var data = {
  location: {
    lon: 13.4156,
    lat: 52.5206
  }
};
```

Illustration: XML vs. JSON – API

XML (DOM)

```
var lon = datanode.firstChild.getAttribute('lon');
```

XML (E4X)

```
var lon = data.location.@lon;
```

JSON

```
var lon = data.location.lon;
```

But E4X only in Firefox today... and Flash as I learnt yesterday from Kevin Lynch's preso!

Standards based technologies? Sort of.

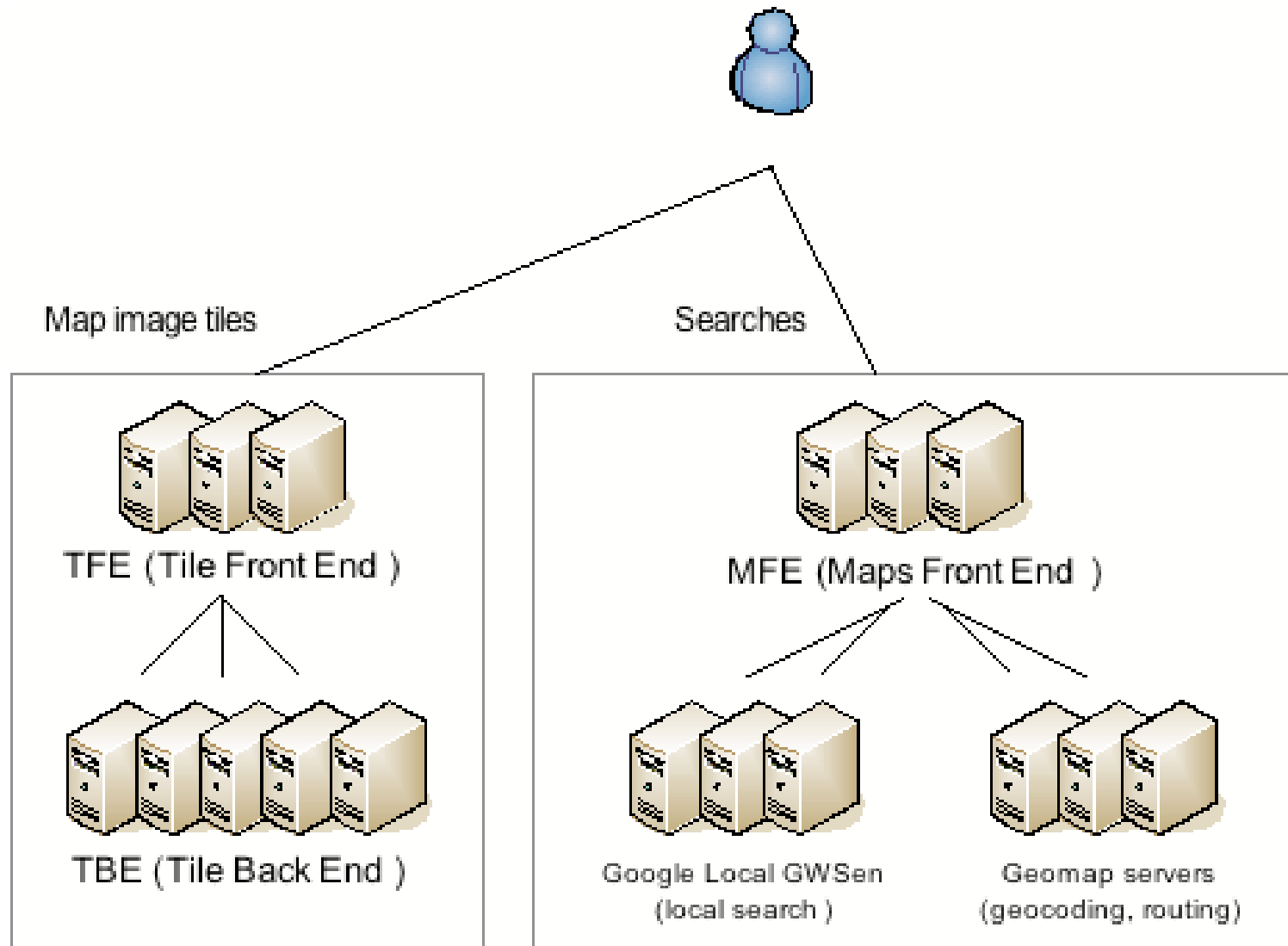
AJAX is a collection of successful technologies being used for things they were never intended for... like the web itself

- The HTML Document Object Model (DOM) was designed for textual documents, not rich graphical interfaces
- XMLHttpRequest was introduced to Internet Explorer by the Microsoft Outlook team to solve a specific problem for Outlook Web Access

Each component has several implementation differences across browsers and platforms

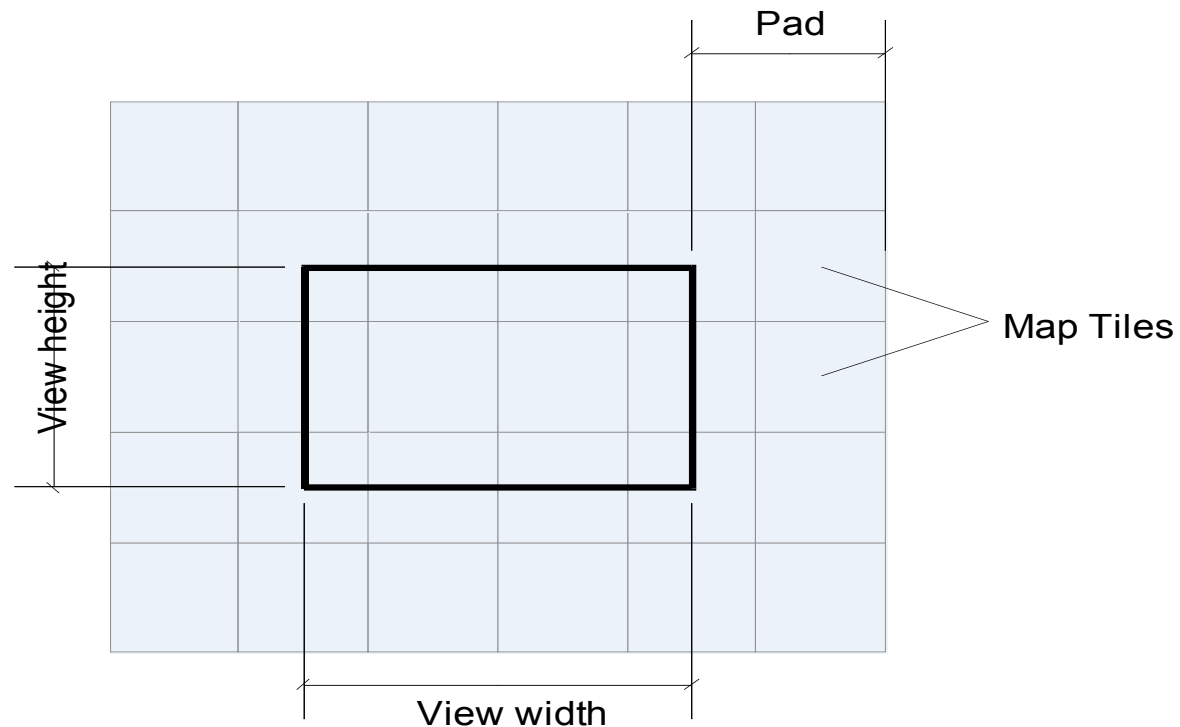
- AJAX Developer is typically synonymous with PhD in Browser Quirks
- See <http://www.quirksmode.org/>

So how does Maps do work?

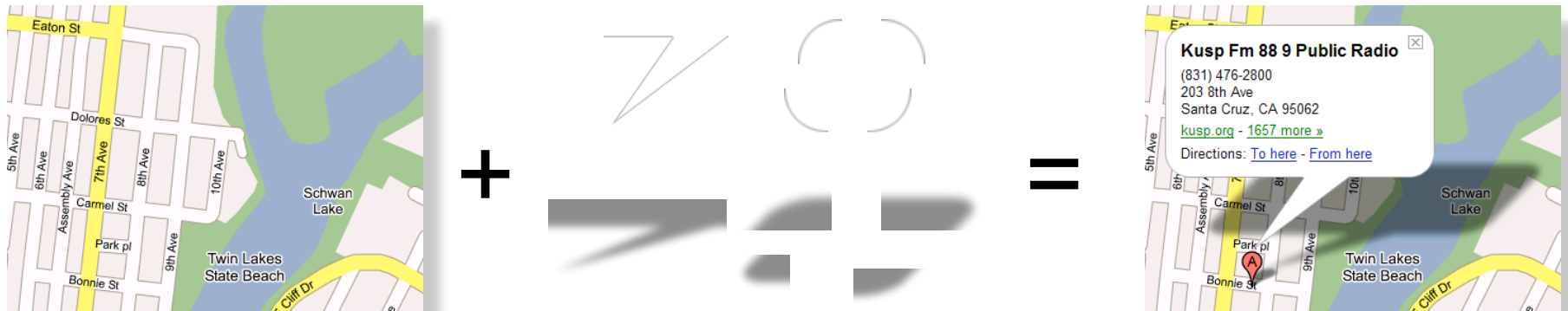


Maps JavaScript architecture

CSS positioned map tiles with a cropped at the map border



Maps JavaScript architecture



Maps Frontend Design

Plenty of crazy hacks to make it work across browsers

- Mozilla/Safari/Opera don't support vector markup (VML, SVG)
 - Draw driving directions lines on the server in a PNG image
- IE doesn't support alpha transparency in PNGs
 - Use ActiveX!
- AJAX interaction typically breaks browser back button
 - Use dummy IFRAMEs to store transient state
- And many, many more...

Trick #1: VPages

Problem: Browsers only update the back button when a new page replaces a current page

- Since AJAX applications typically *update* portions of the page rather than *replacing* the entire page, the back button will never be updated

Solutions for back button updates

- Popular solution: history stored in the hash/anchor portion of the URL
 - Replace the URL <http://myapp.com/app> with <http://myapp.com/app#state> to add a state to the back button
 - Major disadvantage: small state change vs. large state change
 - e.g., dragging the map vs. doing a new search
- Maps solution: dummy IFRAME (“VPage”)
 - Major disadvantage: URL in address bar is not bookmarkable

Trick #2: Copyrights

Problem: JavaScript can only download data from the same host as the HTML document that includes the `<script>` tag

- The Maps API needs to download copyright information as the user drags the map around to update the Google Earth imagery copyrights
 - Maps API sites are not hosted by Google, but the copyright data is hosted at <http://maps.google.com/maps>

Solutions for cross-host RPC

- `` tags can have URLs that are hosted on any server
 - One way communication (client to server):

```
var img = new Image();  
img.src = "http://myserver.com/myapp?a=1&b=2&c=3";
```
- `<script>` tag can include scripts that are hosted on any server

Trick #2: Copyrights

Client script

```
function DoRPC(arg0, arg1) {  
    var script = document.createElement("script");  
    script.src = "http://myserver.com/myapp?arg0=" +  
                escape(arg0) + "&arg1=" + escape(arg1);  
    document.body.appendChild(script);  
}  
  
function HandleRPCResponse(data) {  
    ...  
}
```

<http://myserver.com/myapp?arg0=foo&arg1=bar> response

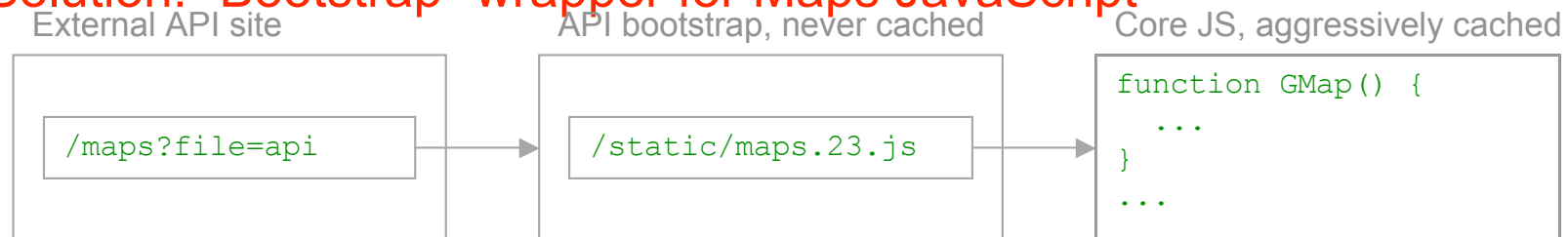
```
HandleRPCResponse("This is response data");
```

Trick #3: API bootstrap

Problem: Maps JavaScript file is over 75K (and it used to be over 160K before January)

- Use name-based versioning to cache file aggressively
 - e.g., maps.1.js has HTTP expiration in 2026, we just name the next update maps.2.js
 - But what about users of the API?
 - API sites need to be able to refer to a single, unchanging URL on their sites

Solution: “Bootstrap” wrapper for Maps JavaScript



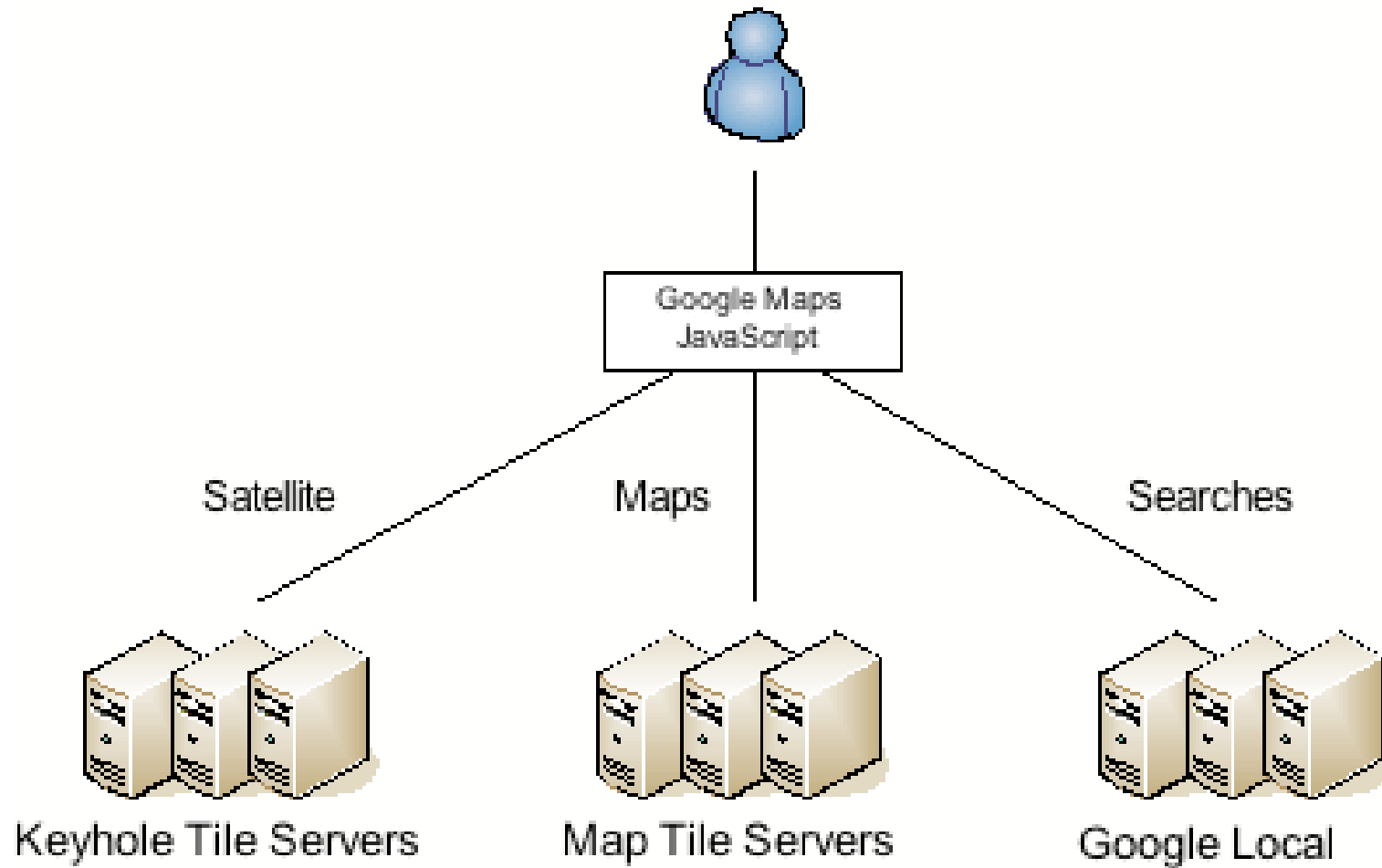
- Others benefits: localized strings, subversioning for developers, e.g., <http://maps.google.com/maps?file=api&v=2.23>

Google Maps API

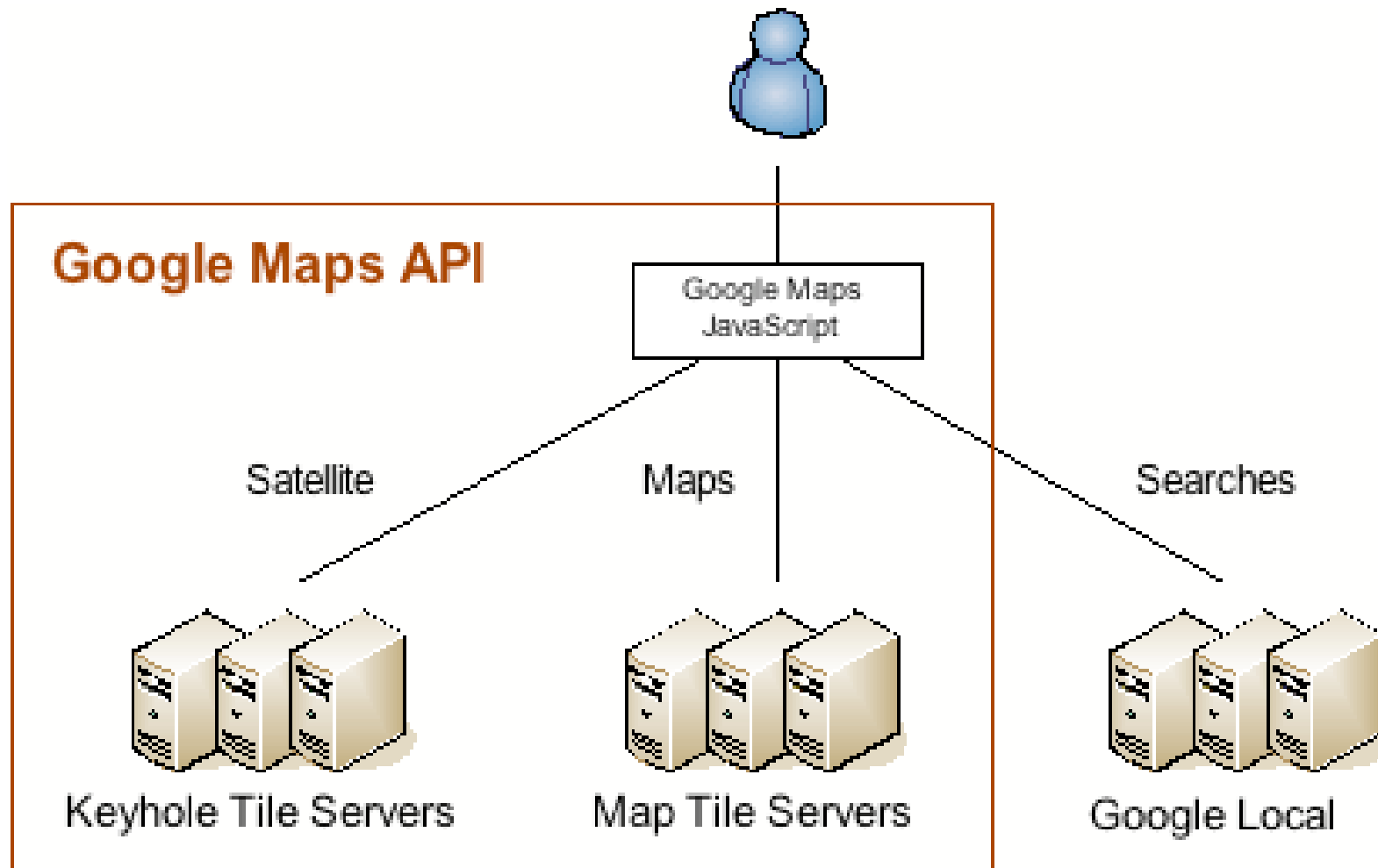
What is it?

- Put Google Maps on your site for free, just like Frappr.com, housingmaps.com and chicagocrime.org
- Overlay custom content, write custom functionality
- Free to all web sites that are free to consumers (including commercial web sites)
- Less than 1 year old: V1 in june 2005, V2 in April 2006

Google Maps Overview



Google Maps Overview



What you can do with the Google Maps API

Switch between maps

- “Map”, “Satellite” and “Hybrid” modes supported by default

Overlays

- Markers
 - Custom icons, shapes, etc
- Polylines
 - e.g., driving directions lines
- Custom overlays

“Info window”

- You know, that cool window with the shadow underneath it

New in Maps API V2: Features

Greater customizability

- Custom map controls, overlays, and map types

Overview map

- New GOverviewMapControl control displays an attractive and collapsible map in the corner of the screen

GLatLng class that is distinct from Gpoint

- You no longer need to reverse your latitude and longitude to construct a geographic point!

Tabbed InfoWindows

- openInfoWindowTabs() and the GInfoWindowTab class

Two additional satellite zoom levels

- Increased imagery resolution and coverage in the U.S., Canada, U.K., Japan, Europe, and almost every major city in the world.
- $\text{newZoom} = 17 - \text{oldZoom}$



New in Maps API V2: Plumbing

Much smaller JavaScript download

- half the size of the old JavaScript file

Fewer memory leaks

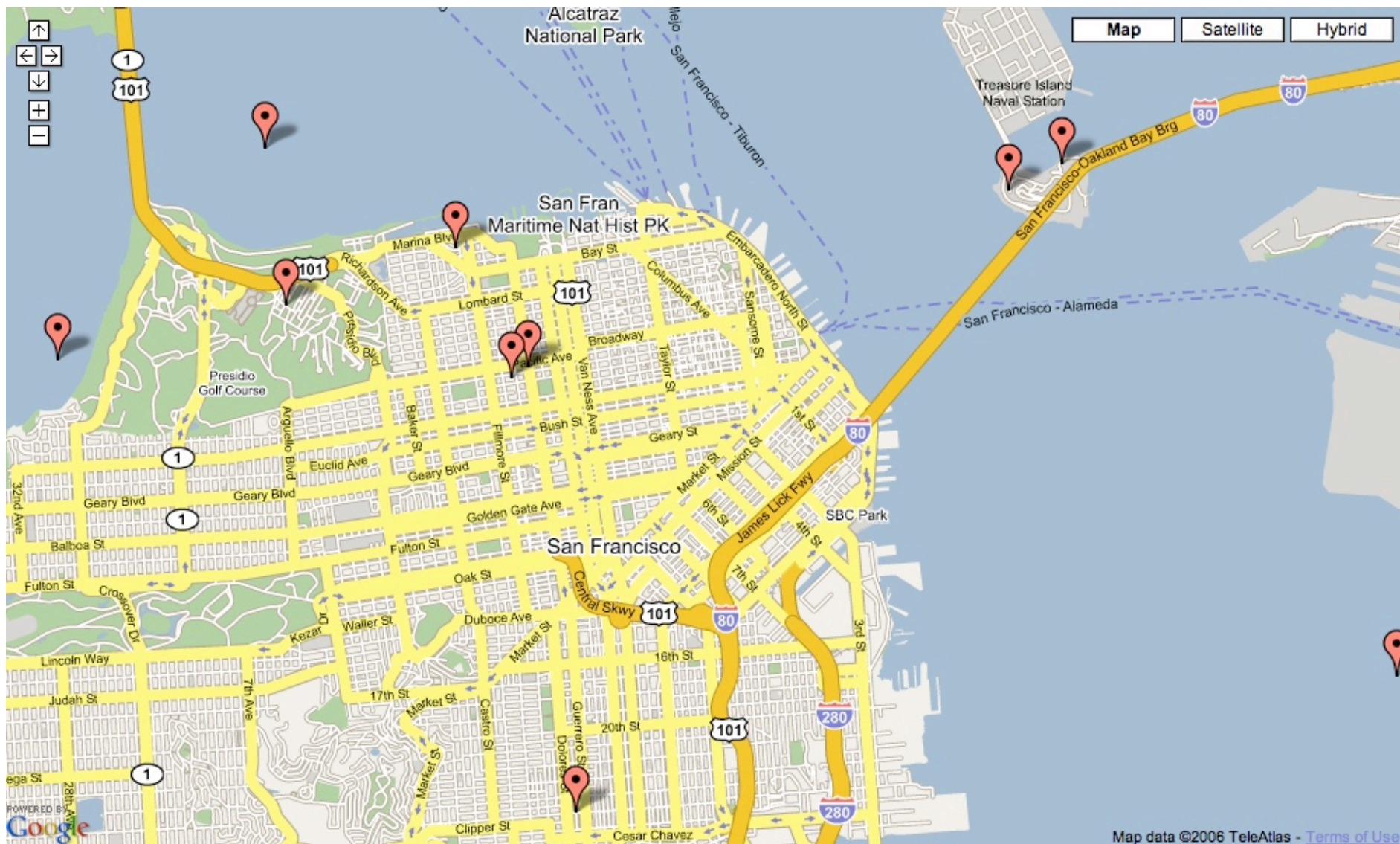
- GUnload method to eliminate memory leaks in IE

Debugging log

- alert() -> GLog.write().

Emphasis on Simplicity

```
// Center the map on Westin Hotel in San Francisco
var map = new GMap2(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());
map.setCenter(new GLatLng(37.787742,-122.408295), 13);
// Add 10 markers to the map at random locations
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
    var point = new GLatLng(southWest.lat() + latSpan *
Math.random(),
    southWest.lng() + lngSpan * Math.random());
    map.addOverlay(new GMarker(point));
}
```



Icons

GIcon class, can copy constructor to customize a base Icon

```
// Create a base icon for all of our markers that specifies the
// shadow, icon dimensions, etc.
var baseIcon = new GIcon();
baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png";
baseIcon.iconSize = new GSize(50, 70);
baseIcon.shadowSize = new GSize(37, 34);
baseIcon.iconAnchor = new GPoint(9, 34);
baseIcon.infoWindowAnchor = new GPoint(42, 50);
baseIcon.infoShadowAnchor = new GPoint(18, 25);
```

Icons

```
// Create P@'s icon based on baseIcon  
var icon = new GIcon(baseIcon);  
icon.image = "http://www.chanezon.com/pat/pat-icon.png";  
icon.shadow = "http://www.chanezon.com/pat/pat-shadow.png"  
icon.shadowSize = new GSize(70, 70);  
  
var marker = new GMarker(map.getCenter(), icon);
```



Tabbed InfoWindow

openInfoWindowTabsHtml (text) and openInfoWindowTabs (DOM)

```
var infoTabs = [  
    new GInfoWindowTab("What", '<a  
href="http://theajaxexperience.com/index.jsp"></a>...'),  
    new GInfoWindowTab("Who", '...'),  
    ...  
];  
  
marker.openInfoWindowTabsHtml(infoTabs);
```

Tabbed InfoWindow Result



Custom Map Controls

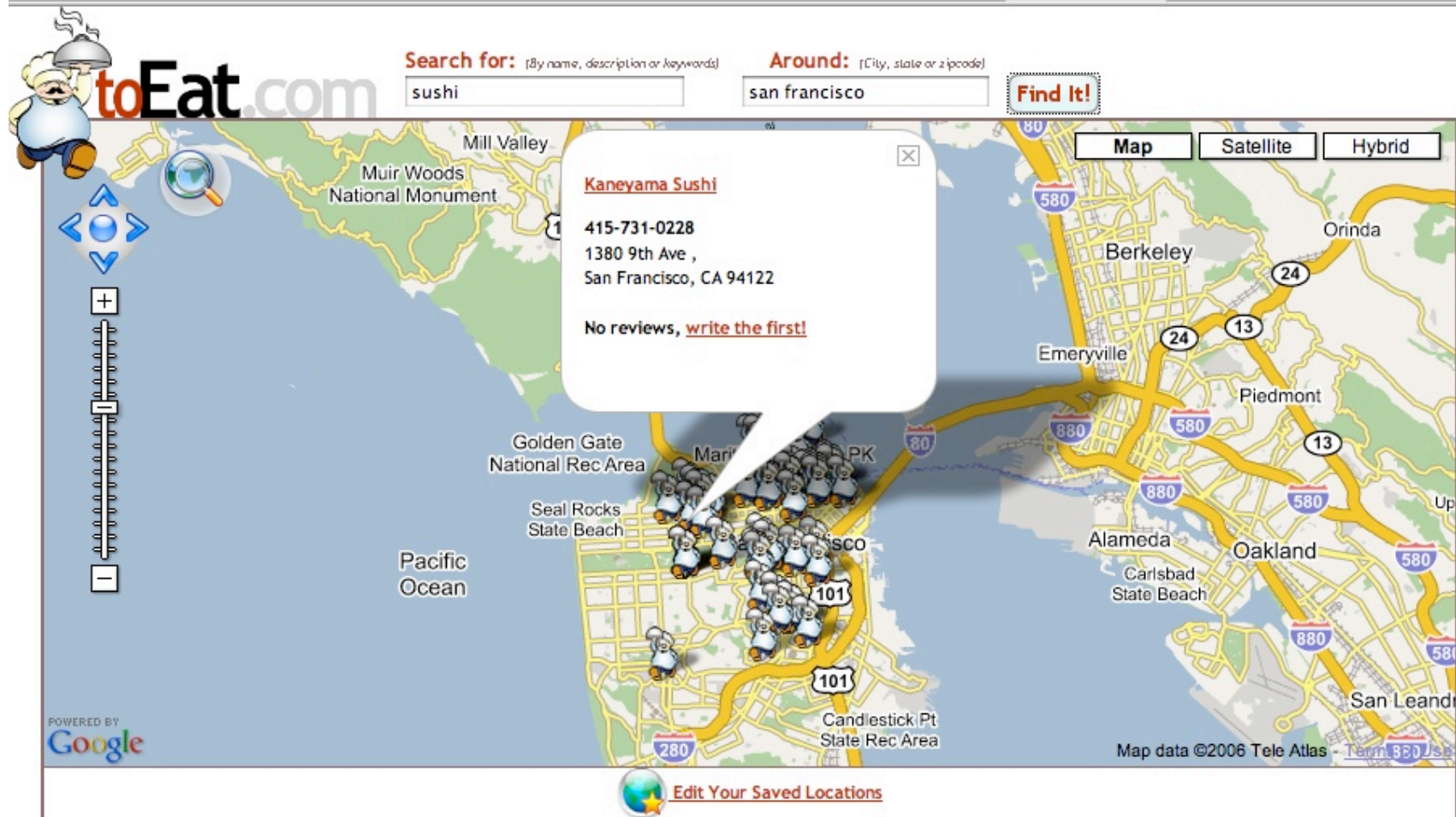
Subclass Gcontrol

- Replace the built-in pan and zoom

Implement GControl interface:

- initialize(): create the DOM elements for your control
- getDefaultPosition(): when none is specified

Custom Map Control Example



Custom Overlays

Subclass GOverlay

- custom map overlays, like the built-in GMarker and Gpolyline
- Add it to the right GMapPane depending on z-index you want
 - From G_MAP_MAP_PANE to G_MAP_FLOAT_PANE

Implement 4 required methods

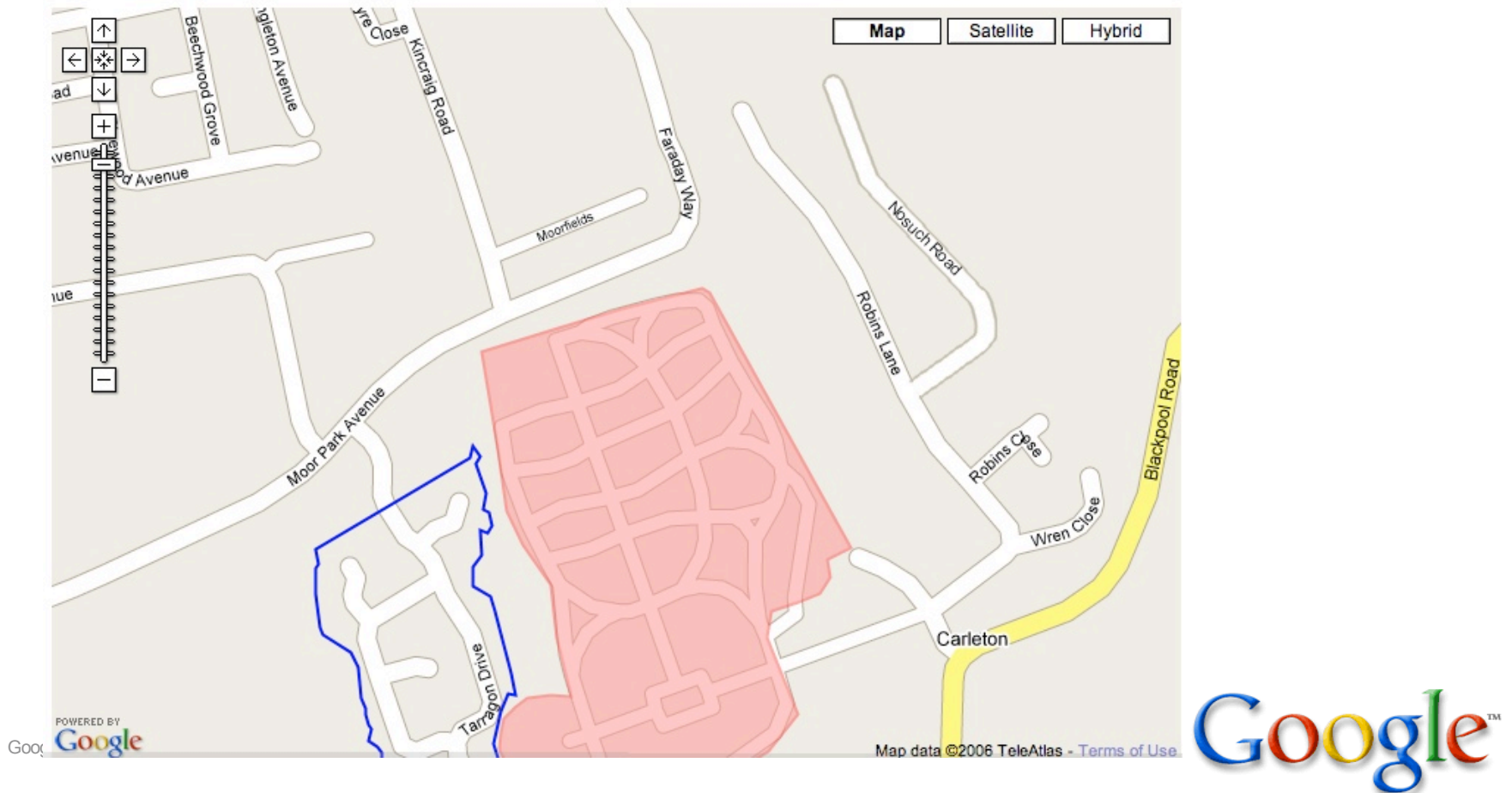
- initialize(), create your DOM
- remove(), ... remove your overlay
- copy(), copies the overlay for use in another map instance
- redraw(), positions and sizes the overlay on the map based on the current projection and zoom level.

Custom Overlay example

Mike William's Einsert extension uses GOverlay

The street labelled "Nosuch Road" does not exist in reality. I've drawn it onto the map with an EInsert.

The blue line and pink polygon are also EInserts.



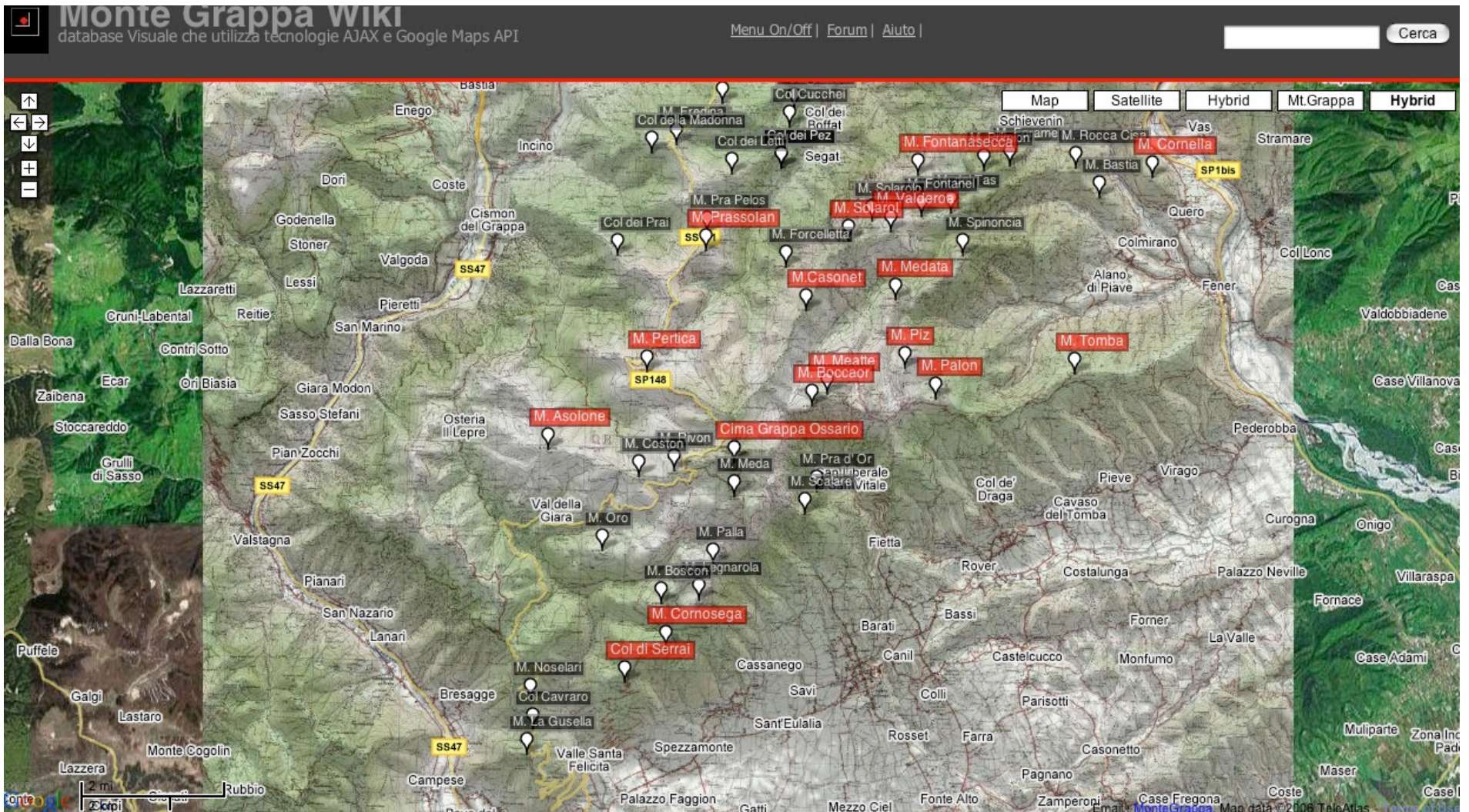
Custom Map Types

Use of subclass GMapType

- From Mariuss “Monte Grappa” map

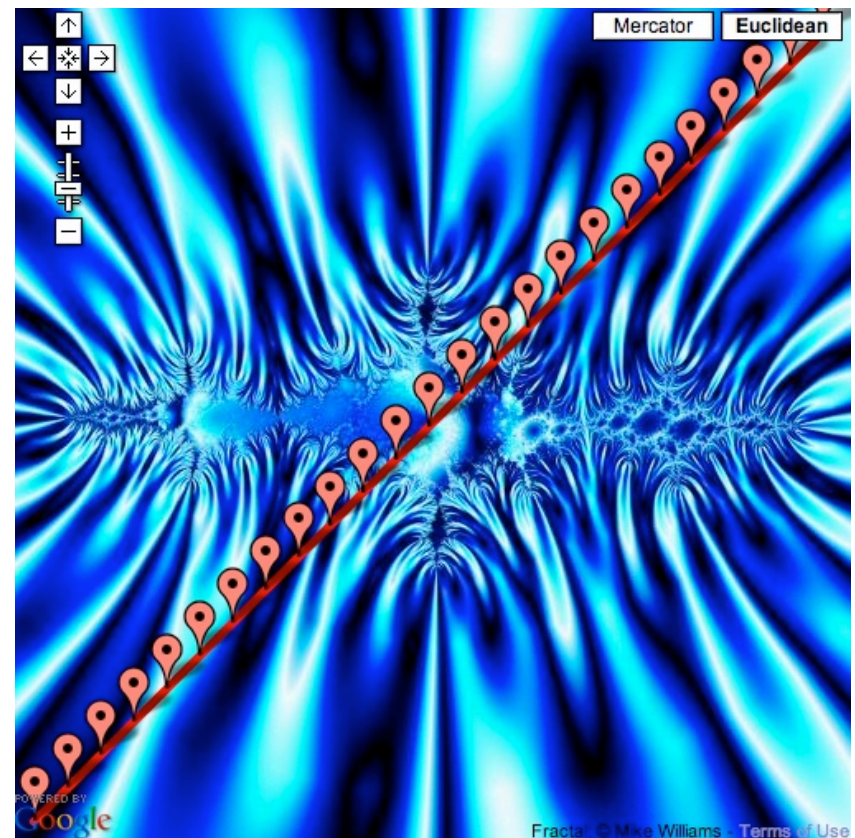
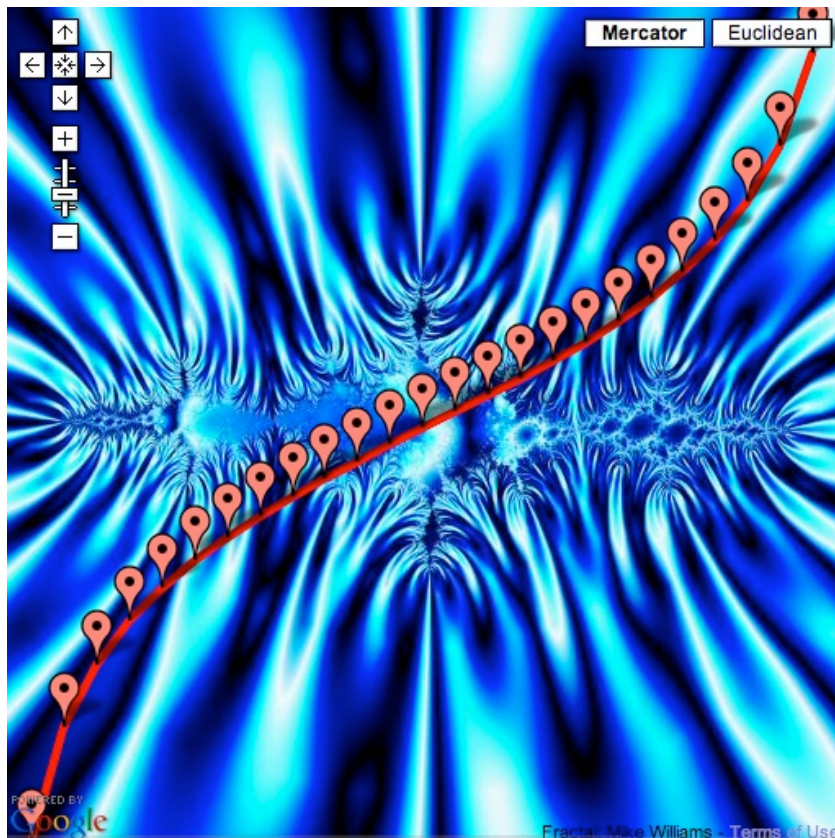
```
// ***** crea 2 nuovi layer*****  
  
var tilelayers = [new GTileLayer(new  
GCopyrightCollection('Mariuss.r@libero.it 2006'),0,19)];  
  
tilelayers[0].getTileUrl= NewTile;  
  
tilelayers[0].getCopyright = function(a,b) {return  
{prefix:'Email :', copyrightTexts:['<a  
href=mailto:montegrappa@altervista.org>MonteGrappa</a>']}};  
  
var custommap = new  
GMapType(tilelayers,G_SATELLITE_MAP.getProjection(),  
'Mt.Grappa',{errorMessage:_mMapError});  
  
map.addMapType(custommap);
```

Custom Map Type Example



Custom Map types with Custom Projections

Mike Williams: Mercator (default) / Euclidian Projection on a dummy Map



A few cool Google Maps applications

Useful/Cool

- <http://del.icio.us/chanezon/ajaxian+google+maps+app+cool>

Technically interesting

- <http://del.icio.us/chanezon/ajaxian+google+maps+app+tech>

Maps API Web Resources

- <http://code.google.com/>
- <http://www.google.com/apis/maps>
- Google-Maps-API Developer Forum:
<http://groups.google.com/group/Google-Maps-API?Ink=li>
- Maps API blog: <http://googlemapsapi.blogspot.com/>
- Mike William's tutorials: <http://www.econym.demon.co.uk/googlemaps/>
- <http://del.icio.us/chanezon/google+maps>

Maps API Developer Forum

Great community

- 1 year, 10 000 members, 33854 messages
- Quick answers, samples, experimentations, add-ons

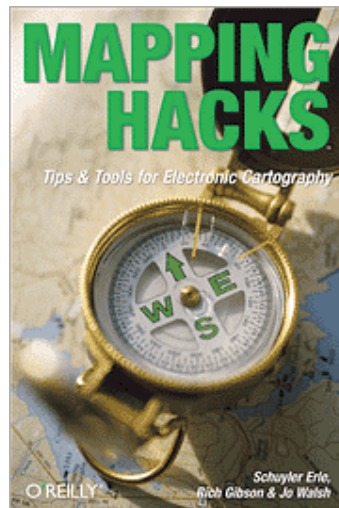
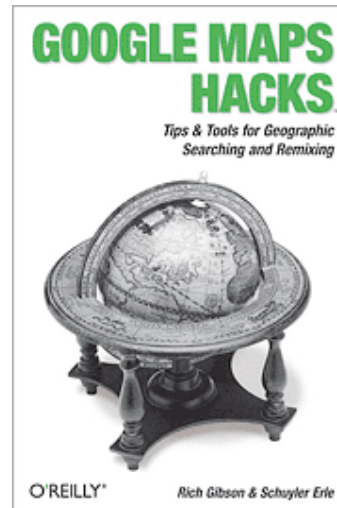
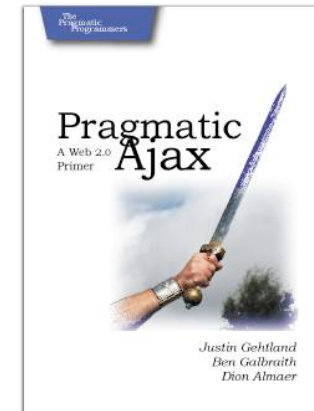
Thank you to the top 8 posters

- They posted 24% of the 33854 messages!
- 2401 Mike Williams, author of a fantastic Maps tutorial
- 1696 John Coryat, from the excellent <http://maps.huge.info>
- 1404 Larry from geocodezip.com
- 810 Doug Henderson
- 779 Will James
- 410 evilc@evilc.com
- 358 Esa
- 318 webmaster@legad.org



Maps API Books

- The Ajaxians book: Pragmatic Ajax
 - implement your own Map UI in a day...
then you have to do the server side:-)
- The O'Reilly ones:
 - Google Maps Hacks
 - Mapping Hacks



Google Data API

REST API, Based on Atom and the Atom Publishing Protocol



*"The release of atom power
has changed everything
except our way of
thinking..."*

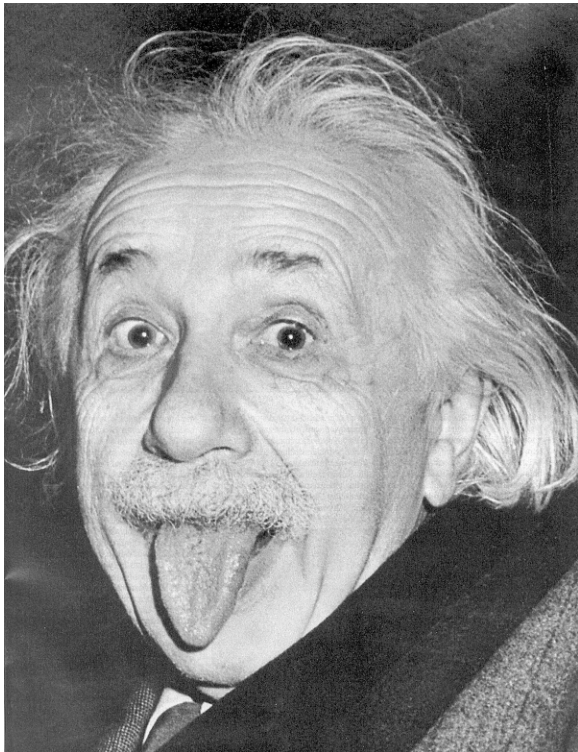
Albert Einstein



First Google Data API: Calendar

GET, POST, PUT, DELETE Calendar Entries

Released in April 2006



"...the solution to this problem lies in the heart of mankind. If only I had known, I should have become a watchmaker."

Albert Einstein

Google Data API

frank mantek (fmantek@google.com)



quotes...



Adam Bosworth on a bike

"'Google Data APIs Protocol' is a horrible name. You can tell that ex-Microsoft employees had a hand in this effort. :)"

Dare Obasanjo
(Microsoft)

more quotes...

“The benefit is that you'd have a single API that could be used to query, update, and index structured data on the web--anywhere on the web. It's a pretty powerful vision and something I didn't expect to see for a couple more years”
- **Jeremy Zawodny (Yahoo)**

That's what I really want to do....



instead.. i learned...

ODBC

OLEDB

(P@: In my case JDBC, EJB and Hibernate... but same feeling)

Remoting technologies

XML

SOAP, WSDL

WS_* specs to your hearts content...



that left me...

confused, stressed, and with a lot of printouts...

when do i use what?

what tool do i use with what technology...?

so, i was still searching...



but...

isn't computing supposed to be simple...?

maybe if I just focus on one vendor?

... that sounds great !

enter

Google, as a random example

- lot's of APIs available
- some are SOAP based (AdWords API)
- some are somewhat REST based
- ... all are somehow different



Now, what do i need?

simple data protocol

query? Sure, i just love queries...

updates - there is no point getting data if you can not mess with it....

simple data protocol

REST kind of design...

data format should be easy to understand

easy to consume by existing tools

and the winner is...

the Syndication format (Atom 1.0 and RSS 2.0).

in the simple case

I just enter a URI, like:

- <http://www.google.com/calendar/user/public/full>
- and get an Atom feed of my calendar data

the result

```
<feed>
  <id>
http://www.google.com/calendar/feeds/user/public/basic
  </id>
  <updated>2006-05-05T14:22:41.000Z</updated>
  <title type="text">Frank Mantek</title>
  <subtitle type="text">Frank Mantek</subtitle>
  <link rel="http://schemas.google.com/g/2005#feed" type="application/atom+xml"
fmantek@gmail.com</email>
  </author>
  <generator version="1.0" uri="http://www.google.com/calendar">Google Calendar</generator>
  <openSearch:itemsPerPage>25</openSearch:itemsPerPage>
</feed>
```

the result, again...

data comes in standard Atom or RSS format

some namespaced extensions are used to identify semantic entities
and data items not mappable in Atom and RSS

extensions

google extensions are in their own namespace

- [xmlns:gd=http://schemas.google.com/g/2005](http://schemas.google.com/g/2005)

we have **types**

and there are **kinds**

- semantical grouping of types

so, what about types?

types can appear anywhere in the document

they do not have semantic meaning by themselves

e.g. a <who> element can appear in any document, but that does not make it a contact

Google kinds...

The Calendar defines

- Contacts
- Events
- Messages

example: an event

```
<entry xmlns:gd="http://schemas.google.com/g/2005">
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event"/>
  <id>http://mycal.example.com/feeds/jo/home/full/e1a2af06df8a563edf9d32ec9fd61e03f7f3b67b</id>
  <published>2005-01-18T21:00:00Z</published>
  <updated>2006-01-01T00:00:00Z</updated>
  <title>Discuss BazMat API</title>
  <content>We will discuss integrating GData with BazMat.</content>
  <author>
    <name>Jo March</name>
    <email>jo@example.com</email>
  </author>
  <gd:when startTime='2005-01-18T21:00:00Z' endTime='2005-01-18T22:00:00Z'>
    <gd:reminder minutes='15'/>
  </gd:when>
  <gd:where valueString='Building 41, Room X'/>
  <gd:eventStatus value="http://schemas.google.com/g/2005#event.confirmed"/>
  <gd:visibility value="http://schemas.google.com/g/2005#event.public"/>
  <gd:transparency value="http://schemas.google.com/g/2005#event.transparent"/>
</entry>
```

example: an event

```
<entry xmlns:gd="http://schemas.google.com/g/2005">
  <category scheme="http://schemas.google.com/g/2005#kind" term="http://schemas.google.com/g/2005#event"/>
  <id>http://mycal.example.com/feeds/jo/home/full/e1a2af06df8a563edf9d32ec9fd61e03f7f3b67b</id>
  <published>2005-01-18T21:00:00Z</published>
  <updated>2006-01-01T00:00:00Z</updated>
  <title>Discuss BazMat API</title>
  <content>We will discuss integrating GData with BazMat.</content>
  <author>
    <name>Jo March</name>
    <email>jo@example.com</email>
  </author>
  <gd:when startTime='2005-01-18T21:00:00Z' endTime='2005-01-18T22:00:00Z'>
    <gd:reminder minutes='15'/>
  </gd:when>
  <gd:where valueString='Building 41, Room X'/>
  <gd:eventStatus value="http://schemas.google.com/g/2005#event.confirmed"/>
  <gd:visibility value="http://schemas.google.com/g/2005#event.public"/>
  <gd:transparency value="http://schemas.google.com/g/2005#event.transparent"/>
</entry>
```

Did someone say Query?

REST filter model

- categories - part of the path

<http://test.com/-/xtech/talks>

Did someone say Query?

parameter query model

- full text - the **q** parameter

<http://test.com?q=GData>

Did someone say Query?

parameter query model

- author - the **author** parameter

<http://test.com?author=Frank ManteK>

Did someone say Query?

parameter query model

- **updated-min/max** parameter

<http://test.com?updated-min=2006-01-01&updated-max=2006-12-31>

query example

[http://test.com/-
/xtech/talks/2006?author=frank
mantek&q=GData&updated-min=2006-01-
01&updated-max=2006-12-31](http://test.com/-/xtech/talks/2006?author=frankmantek&q=GData&updated-min=2006-01-01&updated-max=2006-12-31)

...and direct access

each entry contains a “self” URI

- `<link rel="self" href="http://test.com/feeds/talks/idforthis"/>`

using that gives you just this entity

What about updating?

Updates are based on the Atom Publishing protocol draft

- <http://www.ietf.org/internet-drafts/draft-ietf-atompub-protocol-08.txt>
- that is still in development

Updates are done using optimistic concurrency

Updates...

Insert a new entry in a feed

- POST to the feed's **service.post** URI

Delete an entry

- DELETE to the entries **edit** URI

Update an entry

- PUT to the entries **edit** URI

Why do i care?

Does this actually help me?

Or is it just another....

...road block...



on the way?



Google...

just released one property supporting the Data API

Google calendar was well received since its April 12th launch date

more Google properties will support this API over time



Support?

Language libraries for Java & C#, including documentation

Active community at code.google.com

Support for other languages (PHP, Javascript) is in the works

Demo....



Why do i care again?

This is just the first step

Right now, just another printout, but a really small one
but...

more quotes...

“I think GData is hot. I'd like us to use this as a building block for some of the developer platform stuff we are doing for Windows Live” - **Dare Obasanjo (Microsoft)**

I just love quotes...

“The next logical questions, for me at least, are:

1. Will MySQL add native GData support to the server anytime soon?
2. Should Yahoo begin to enable a GData API on our data stores?

I hope the answer to #1 is "yes, they should" and suspect the answer to #2 is "probably--at least for some of them." - **Jeremy Zawodny**

(Yahoo)



are we there yet?

and other questions....

<http://code.google.com/apis/gdata/index.html>

Google Data Authentication

Now: Account Authentication for Installed Applications

- programmatic login
- CAPTCHA
- But username/password stored by the application: Bad

Soon: Account Authentication Proxy for Web Applications

- Connects the user to a Google service that requests credentials
- The service then returns a token that the web application can use
- Google (rather than the web front end) securely handles and stores the user's credentials.



Applications

Outlook sync

- Daniel Tiles, RemoteCalendars

Cell Phone Sync

- Thomas Oldervoll's Gcalsync, J2ME midlet, he's at the conference!

RDFCalendar XVCD

- By Justsystems Corporation: Google, eventful, upcoming

Resources

- <http://code.google.com/>
- <http://code.google.com/apis/gdata/overview.html>
- <http://code.google.com/apis/gdata/calendar.html>
- Google Calendar Data API Developer Forum:
<http://groups.google.com/group/google-calendar-help-dataapi?lnk=li>
- <http://del.icio.us/chanezon/gdata>

Demo

Conclusion

Maps and Calendar are very complementary

- Time and Space are important dimensions to organize and present information

Calendar is the first Google Data API... not the last

REST Data services + Ajax UI is a nice combination

Google Technical Operations Is hiring!

We're looking for technical wizards with a history of excellence.

Jobs:

- Web Applications Engineers. Technical Managers. Data Reporting Engineer. Product Managers. Spam Fighters. Technical Support. Google Web Analytics Support. Bug Fighters. API Support. And more.

E-mail resume and cover letter to:

Techops-jobs@google.com

Credits

Entertaining you this morning:

- Vocals: Patrick Chanezon
- Guitar on “Maps”: Bret Taylor
- Synthesizer Programming on “Maps”: Steffen Meschkat
- Percussions on “Gdata”: Frank Mantek
- Plumbing/Watchmaking by Einstein

Questions?

GET /ajaxians/questions HTTP/1.1

Accept: application/atom+xml