
Google™
09
I/O

Going Social with the YouTube APIs

Jeff Fisher, Jochen Hartmann
May 29th, 2009

Post your questions for this talk on Google Moderator:
http://tinyurl.com/io-social-youtube

Raise Your Hand

Why Do I Care?

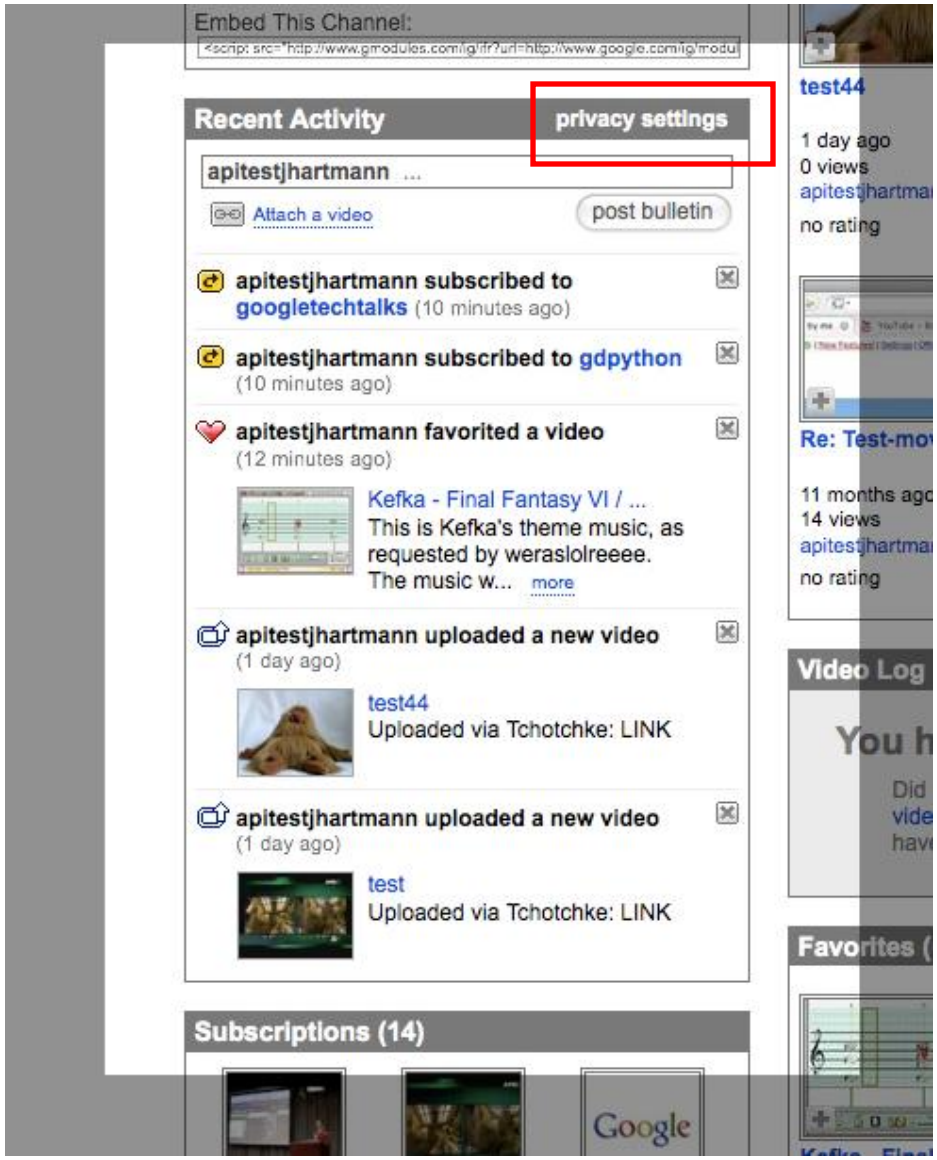
- Bring YouTube activity into
  - Social networks
  - Existing social applications
  - Devices
  - Gadgets
- Create a better YouTube search experience by tracking content that a user's friends like.
- Educate your application about what a user is doing on-site to create a better user experience.

Brief overview of YouTube Data API

- Google Data API based on Atom Publishing Protocol
- Versions (2 > 1)
- Exposes metadata for most things on YouTube
- Allows for programmatic creation / upload
- Use our open-source client libraries
  - .NET, Java, PHP, Python, Objective-C

YouTube Activity Feeds

YouTube Activity Feeds



YouTube Activity Viewer Demo

Activity for apitestjhartmann

Thu Apr 30 2009 18:07:37 GMT-0700 (PDT)

+ apitestjhartmann has subscribed to TheOnion's videos

Wed Apr 29 2009 11:49:43 GMT-0700 (PDT)

+ apitestjhartmann has favorited a video uploaded by HCBailly

Kefka - Final Fantasy VI / III, Mario Paint (video id: 64lSZ1c7ulA)

View count: 13579

Average rating: 4.87013 (rated by 77 users)

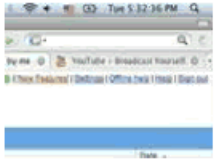
Mon Apr 27 2009 13:41:01 GMT-0700 (PDT)

+ apitestjhartmann has uploaded a video

test44 (video id: mNhSmtvHBnM)

Fri Apr 24 2009 12:09:44 GMT-0700 (PDT)

+ apitestjhartmann has favorited a video uploaded by apitestjhartmann

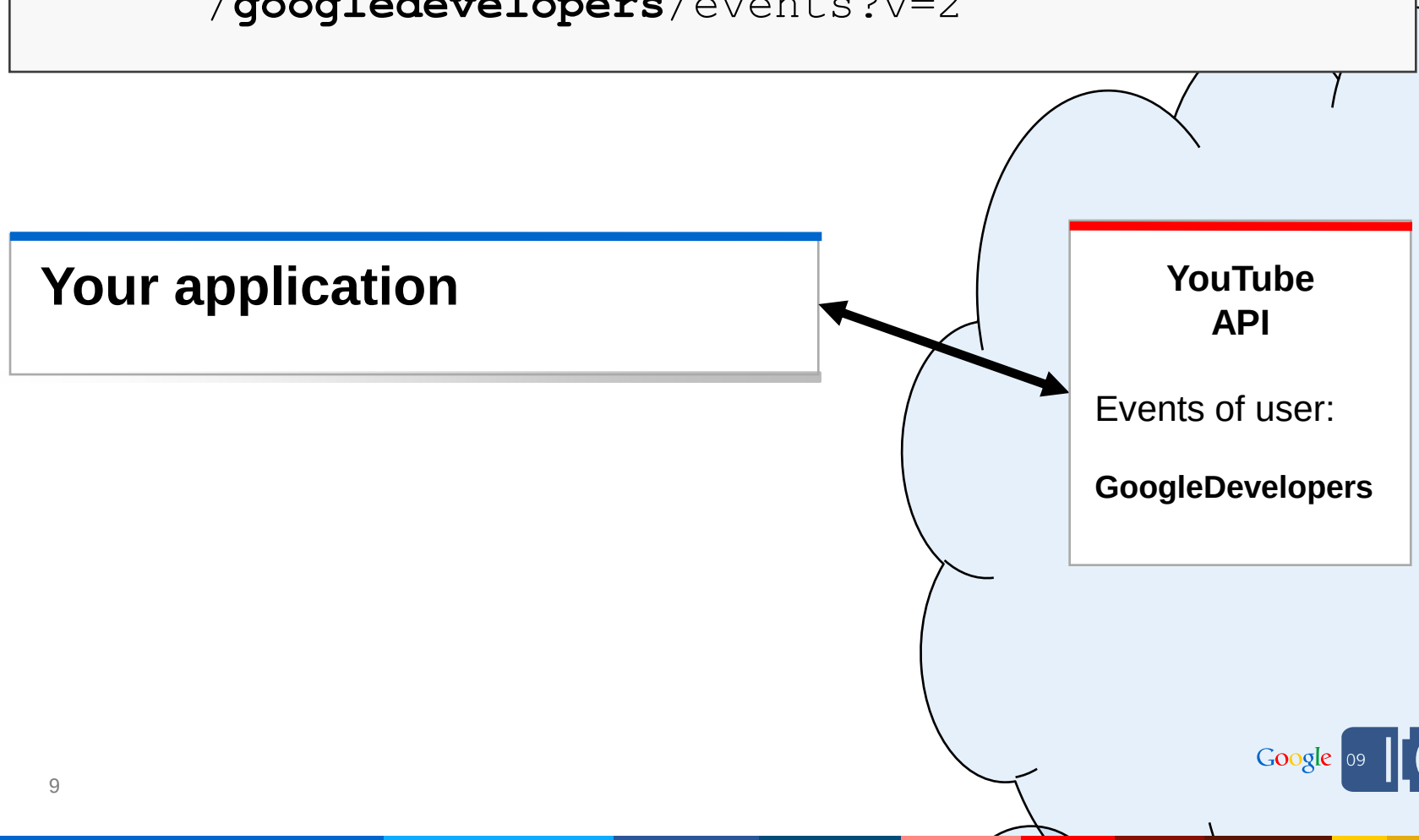


Re: Test-movie (video id: KaWuVYf937Q)

View count: 12

YouTube Activity Feed URL

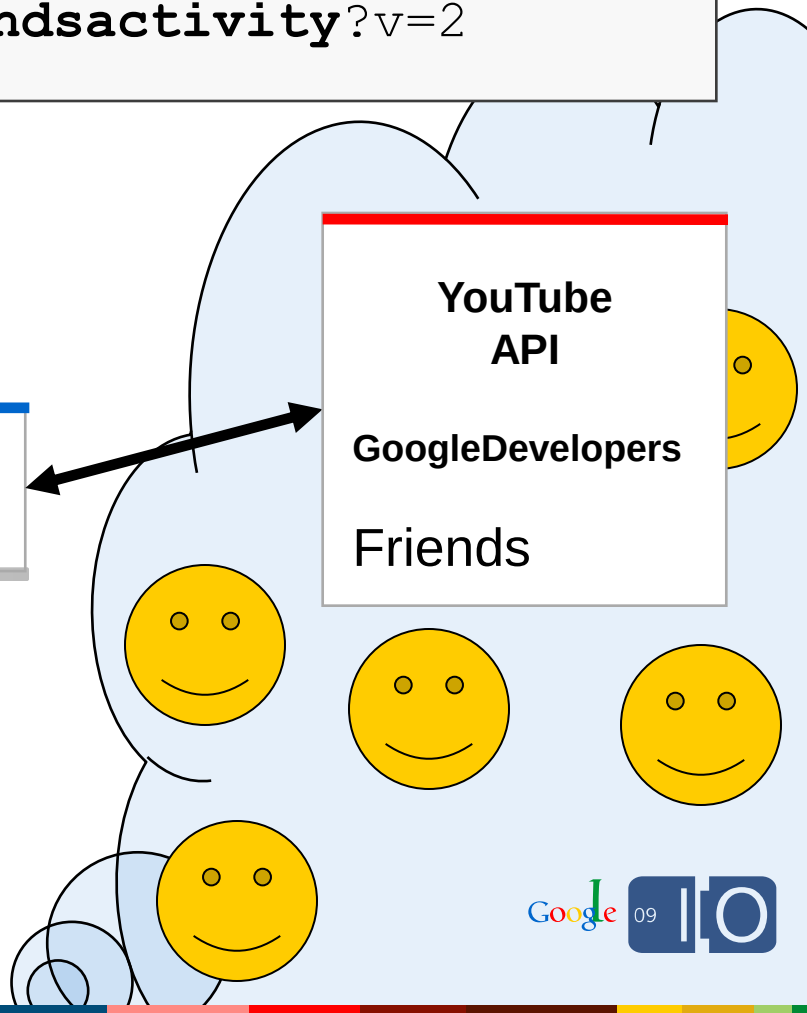
```
http://gdata.youtube.com/feeds/api/users
     /googledevelopers/events?v=2
```



YouTube Activity Feeds: Friend activity

```
http://gdata.youtube.com/feeds/api/users
      /googledevelopers/friendsactivity?v=2
```

Your application



Types of activities in the API

Video activities

- video_rated
- video_shared
- video_favorited
- video_commented
- video_uploaded

User activities

- friend_added
- user_subscription_added

YouTube Activity Feed Entry (Simplified)

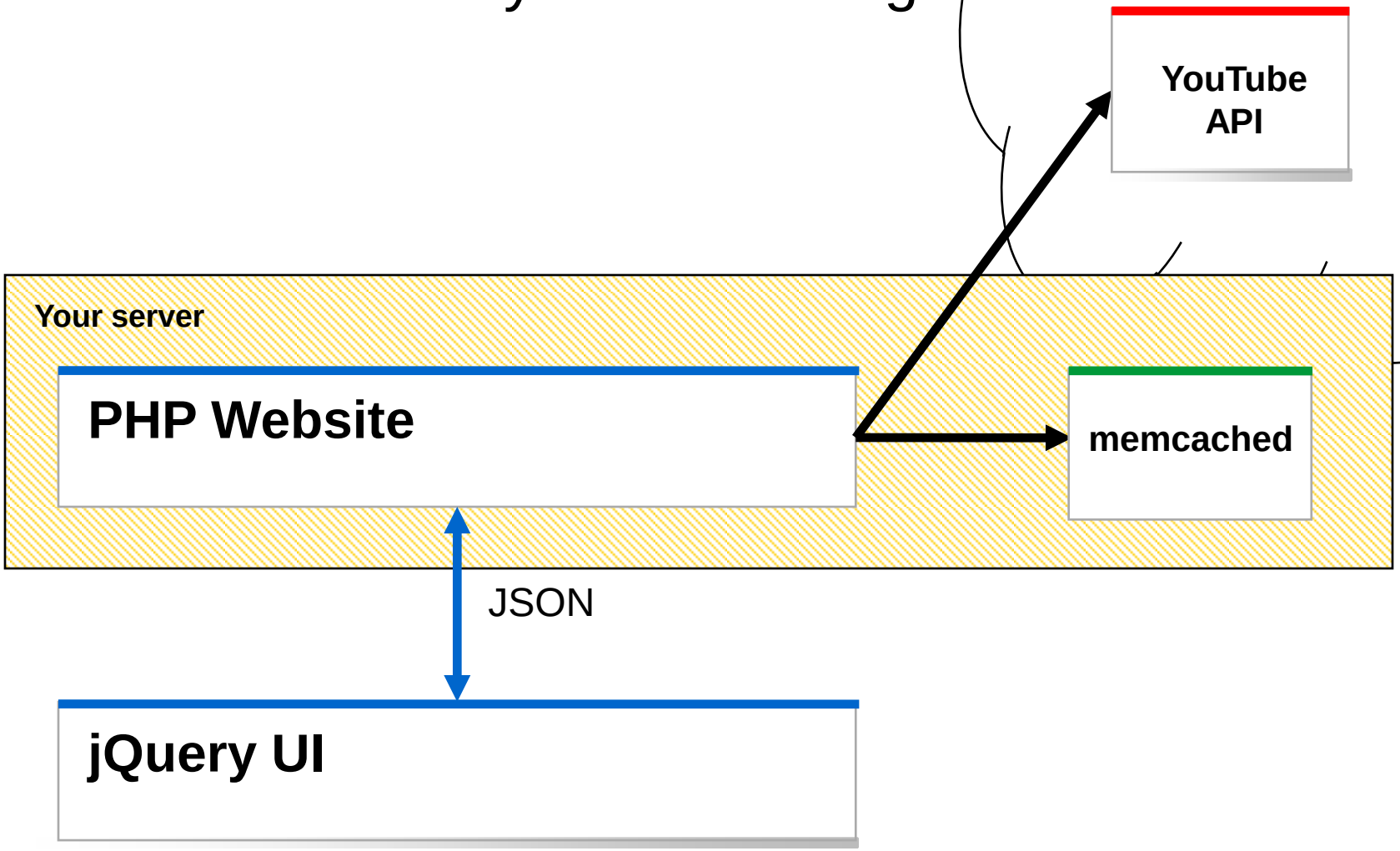
```
<entry>
  <id>tag:youtube.com,2008:user:googledevelopers:event:RvZ...</id>

  <updated>2009-04-30T15:21:36.000Z</updated>
  <category
     scheme='http://gdata.youtube.com/schemas/2007/userevents.cat'
     term='video_uploaded' />

  <title>GoogleDevelopers has uploaded a video</title>
  <author>
    <name>GoogleDevelopers</name>
  </author>

  <yt:videoid>4agYZmIQ3aE</yt:videoid>

 </entry>
```

Full YouTube Activity Feed XML


```
– <feed gd:etag="W/"Dk8NR347eCp7ImA9WxJSEU0."">
    <id>tag:youtube.com,2008:user:googledevelopers:events</id>
    <updated>2009-04-30T15:21:36.000Z</updated>
    <category scheme="http://schemas.google.com/g/2005#kind" term="http://gdata.youtube.com/schemas/2007#userEvent"/>
    <title>Recent activity of googledevelopers</title>
  – <logo>
      http://www.youtube.com/img/pic_youtubelogo_123x63.gif
    </logo>
    <link rel="updates" type="application/json" href="http://gdata.youtube.com/sup?seconds=300#3b47c0e2"/>
    <link rel="http://schemas.google.com/g/2005#feed" type="application/atom+xml" href="http://gdata.youtube.com/feeds/api/
    Google-AtomViewer-mqdfuljp-0"/>
    <link rel="http://schemas.google.com/g/2005#batch" type="application/atom+xml" href="http://gdata.youtube.com/feeds/ap
    /batch?client=ytapi-Google-AtomViewer-mqdfuljp-0"/>
    <link rel="self" type="application/atom+xml" href="http://gdata.youtube.com/feeds/api/users/googledevelopers/events?start
    Google-AtomViewer-mqdfuljp-0"/>
    <link rel="service" type="application/atomsvc+xml" href="http://gdata.youtube.com/feeds/api/users/googledevelopers/event
  – <author>
      <name>YouTube</name>
      <uri>http://www.youtube.com/</uri>
    </author>
    <generator version="2.0" uri="http://gdata.youtube.com/">YouTube data API</generator>
    <openSearch:totalResults>12</openSearch:totalResults>
    <openSearch:startIndex>1</openSearch:startIndex>
    <openSearch:itemsPerPage>25</openSearch:itemsPerPage>
  – <entry gd:etag="W/"Dk8NR347eCp7ImA9WxJSEU0."">
    – <id>
        tag:youtube.com,2008:user:googledevelopers:event:R29vZ2xlRGV2ZWxvcGVyczExMjQxMTA0ODk2MzQ2MzU2NjU
      </id>
      <updated>2009-04-30T15:21:36.000Z</updated>
      <category scheme="http://schemas.google.com/g/2005#kind" term="http://gdata.youtube.com/schemas/2007#userEvent"/
      <category scheme="http://gdata.youtube.com/schemas/2007/userevents.cat" term="video_uploaded"/>
      <title>GoogleDevelopers has uploaded a video</title>
      <link rel="alternate" type="text/html" href="http://www.youtube.com"/>
      <link rel="http://gdata.youtube.com/schemas/2007#video" type="application/atom+xml" href="http://gdata.youtube.com/
      /4agYZmIQ3aE?client=ytapi-Google-AtomViewer-mqdfuljp-0"/>
```

OMG Kittens

YouTube Activity Viewer Design



Be Smart

- Request more metadata for each item
- Video metadata can be retrieved by ID
- User metadata can be retrieved by username
- Request only what you need as you need it

Be Scalable

- Cache the metadata you get back
- Caching trades freshness for speed and scale
- Hitting a local cache is much cheaper than an HTTP request to the API
- memcached is an excellent lightweight caching system

Be Robust

- Errors can happen anytime!
- Be graceful on API service errors
- Provide nice messages to your user in the event of a problem

PHP Website functions

PHP Website

- Use an authenticated **Zend_Gdata_YouTube** service object
- Fetch activity and friend activity feeds for user
- Fetch video metadata from You Tube API
- Create and return JSON feeds
- Store data in memcache

Using the PHP client library

- Getting a reference to the YouTube service object:

```
// create a HTTP client
$httpClient = Zend_Gdata_AuthSub::getHttpClient(
  $_SESSION['sessionToken']);

// create a service
$yt = new Zend_Gdata_YouTube($httpClient, $applicationId,
  $clientId, $devKey);

// set the service's version
$yt->setMajorProtocolVersion(2);
```

Using other client libraries to do the same thing

- Getting a reference to the YouTube service object:

```
// in Java
YouTubeService service = new YouTubeService(clientID,
  developer_key);
service.setAuthSubToken(sessionToken, null);

// in .NET
YouTubeRequestSettings settings = new YouTubeRequestSettings(
  "example app", clientID, developerKey, (String) SessionToken);
YouTubeRequest request = new YouTubeRequest(settings);

// in Python
yt_service = gdata.youtube.service.YouTubeService()
yt_service.SetAuthSubToken(authsub_token)
yt_service.UpgradeToSessionToken()
```

Using the PHP client library

- Getting activity:

```
$activityFeed = $yt->getActivityForUser($username);

$friendActivityFeed = $yt->getFriendActivityForCurrentUser();
```

YouTube Activity Viewer Design: PHP Website

- Getting video metadata

```
$yt = getYtService();
try {
  $videoEntry = $yt->getVideoEntry($videoId);

  // create light-weight array for json encoding
  $video = Array();
  $video['id'] = $videoEntry->getVideoId();
  $video['title'] = $videoEntry->getVideoTitle();

} catch(Zend_Gdata_App_HttpException $e){
  $httpStatus = $e->getResponse()->getStatus();

  if($httpStatus >= 500) {
    $video = 'SERVER_ERROR';
  } else {
    $video = 'NOT_AVAILABLE';
  }
}
```

Try It Yourself

```
http://www.GoogleCodeSamples.com
```

Scaling YouTube Activity Feeds

- What if you have 1,000,000 users?
- What if you have your own friend graph?
- What if you have 1,000,000 users AND your own friend graph?

What's SUP?

Not much, you?

What **is** SUP?

"[The] Simple Update Protocol is a simple and compact 'ping feed' that web services can produce in order to alert the consumers of their feeds when a feed has been updated."

http://code.google.com/p/simpleupdateprotocol

YouTube SUP Feed

```
http://gdata.youtube.com/sup
```

Anatomy of a SUP Feed

```
{ "updated_time": "2009-04-28T21:29:20Z",
  "since_time": "2009-04-28T21:24:19Z",
  "period": 300,
  "available_periods": {
    "300": "http://gdata.youtube.com/sup?seconds=300",
    "900": "http://gdata.youtube.com/sup?seconds=900",
    "600": "http://gdata.youtube.com/sup?seconds=600"
  },
  "updates": [
    ["159aa827", "6e19"],
    ["9559d1d", "6e19"],
    ...
  ]
}
```

Anatomy of a SUP Feed

```
{ "updated_time": "2009-04-28T21:29:20Z",
  "since_time": "2009-04-28T21:24:19Z",
  "period": 300,
  "available_periods": {
    "300": "http://gdata.youtube.com/sup?seconds=300",
    "900": "http://gdata.youtube.com/sup?seconds=900",
    "600": "http://gdata.youtube.com/sup?seconds=600"
  },
  "updates": [
    ["159aa827", "6e19"],
    ["9559d1d", "6e19"],
    ...
  ]
}
```

Discovering User Hashes

```
http://gdata.youtube.com/feeds/api/users/
       googledevelopers/events?v=2
```

```
<link rel='updates' type='application/json'
 href='http://gdata.youtube.com/sup?seconds=300#3b47c0e2' />
```

SUP With You ?
The ***Über*** Activity Viewer

YouTube *Über*Activity Viewer Application Demo

Über Activity Viewer

(add yourself)

« previous page of results next page of results »

bbcworldnews has uploaded a video on 2009-05-11 02:46:23

Duration 2:3
Viewed 32 times so far.

Gordon Brown makes expenses apology
2009-05-11
The prime minister has apologised for the scandal of MPs' expenses claims. ...
Tags: BBC News Gordon Brown makes
Category: News
Ratings so far: Average : 1.0 | NumRaters : 1 |
Watch page: http://www.youtube.com/watch?v=XDYwq2oOo9c

bbcworldnews has uploaded a video on 2009-05-11 02:16:24

Duration 1:41
Viewed 22 times so far.

Every MP should feel 'ashamed' - Duncan
2009-05-11
The shadow leader of the Commons, Alan Duncan, has said every MP should feel "as ...
Tags: BBC News Every MP should
Category: News
Ratings so far: Average : 1.0 | NumRaters : 1 |
Watch page: http://www.youtube.com/watch?v=YbbJXmp4A6I

bbcworldnews has uploaded a video on 2009-05-11 00:16:13

Duration 0:57
Viewed 160 times so far.

US police paraglide on patrol
2009-05-10
A US police department is training officers to fly powered paragliders for search ...
Tags: BBC News US police paraglide
Category: News
Ratings so far: Average : 3.6666667 | NumRaters : 3 |
Watch page: http://www.youtube.com/watch?v=HCt7e-04Mdc

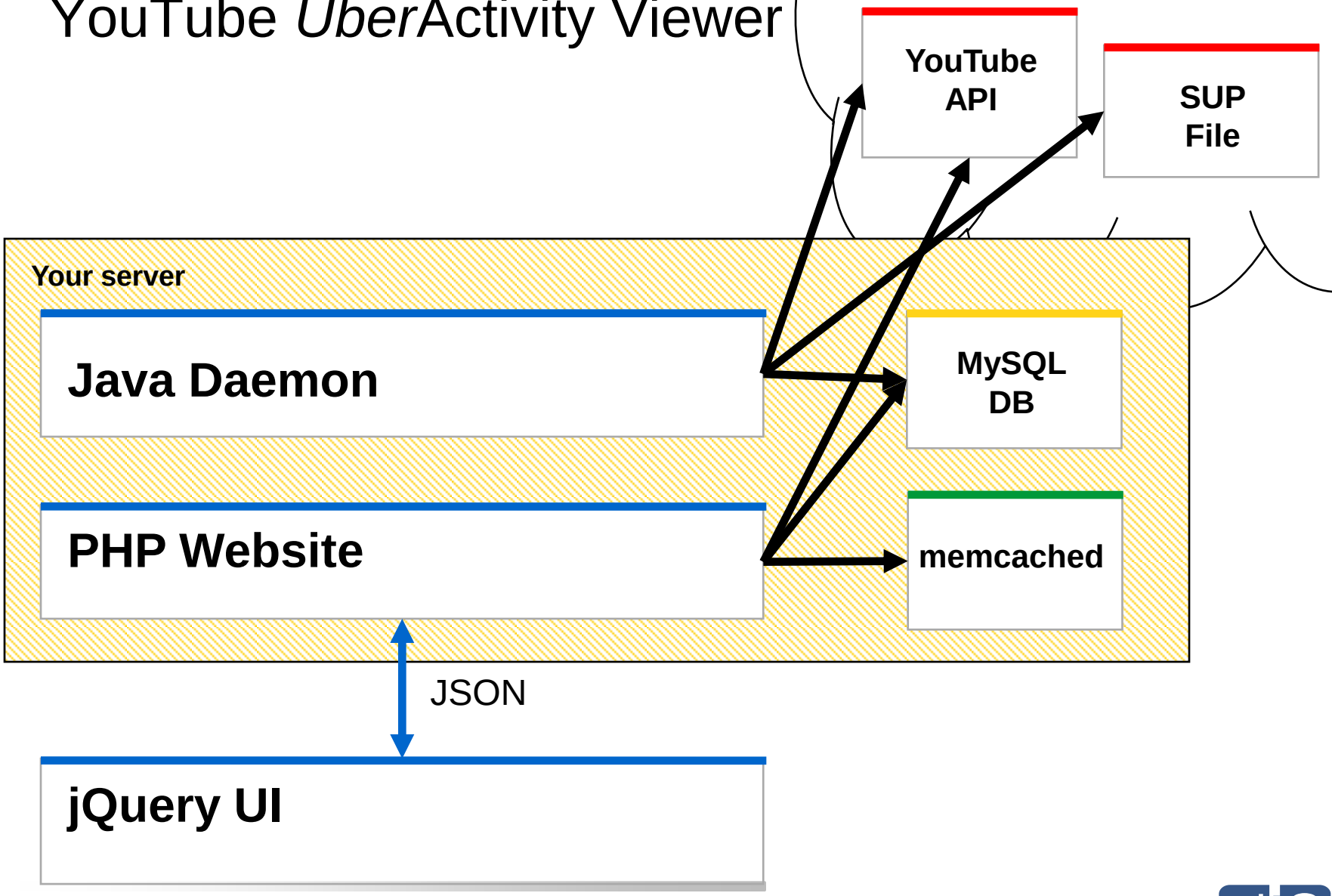
bbcworldnews has uploaded a video on 2009-05-11 00:01:22

Inquiry begins into Australia fires

Try It Yourself

GoogleCodeSamples.com/**uber**viewer

YouTube *Über*Activity Viewer

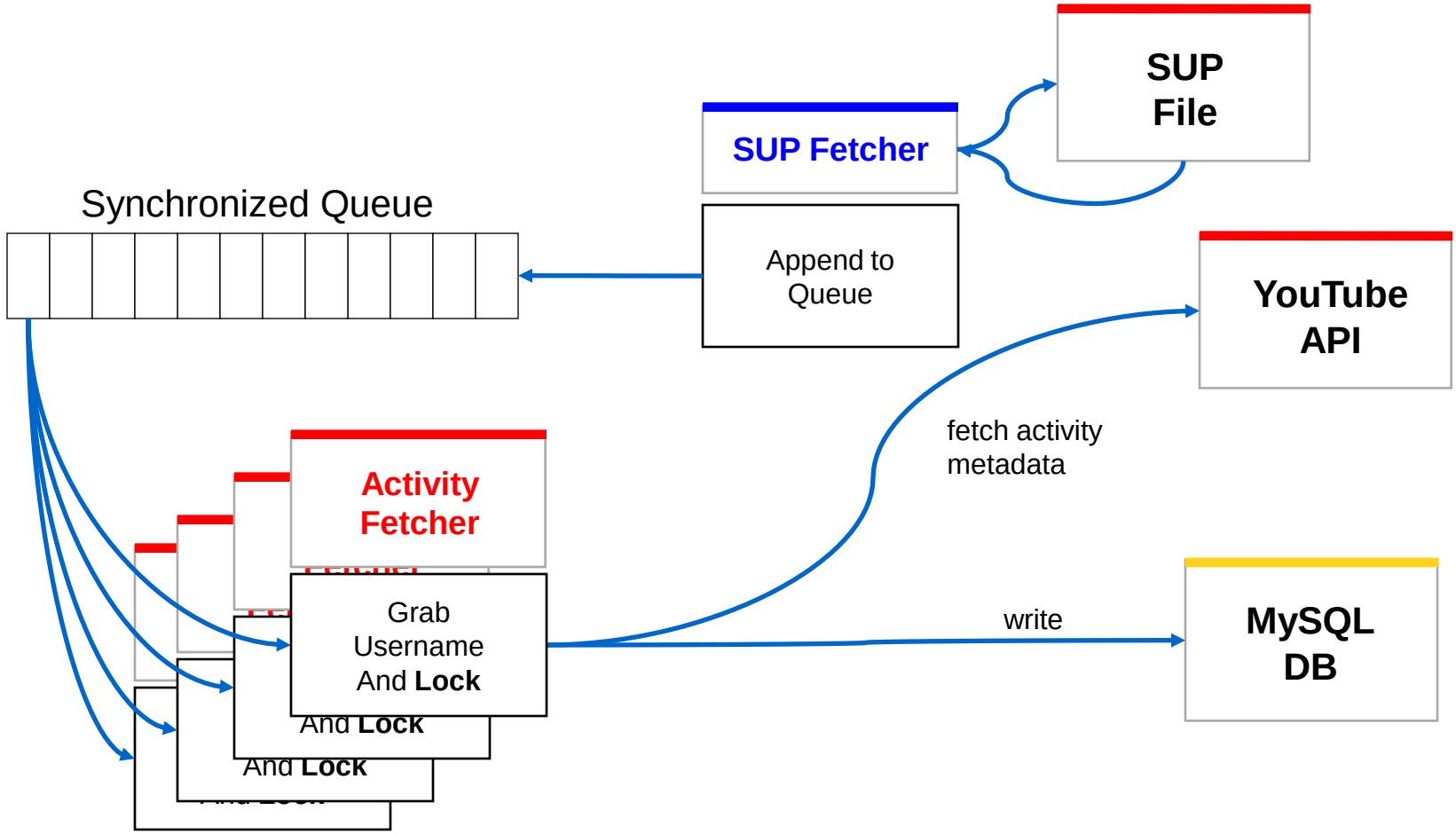


*Über*Activity Viewer Daemon Threads

Java Daemon

- Why Java? Why Threads?
- Main thread
- SUP Fetcher thread
- X Activity Fetcher threads

*Über*Activity Viewer Threading Overview



*Über*Activity Viewer Threading

- Starting threads

```
supFetcher = new Thread(new SupFetcher(config, queue));

supFetcher.start();

for (int i = 0; i < activityThreadCount; i++) {
  activityFetchers[i] = new Thread(
    new ActivityFetcher(config, queue, userLocks));
  activityFetchers[i].start();
}
```

*Über*Activity Viewer Threading

- Closing threads using interrupt

```
for (int i = 0; i < activityThreadCount; i++) {
  activityFetchers[i].interrupt();
}

supFetcher.interrupt();
supFetcher.join();

for (int i = 0; i < activityThreadCount; i++) {
  activityFetchers[i].join();
}
```

*Über*Activity Viewer Threading

- Sharing state between threads

```
// configuration file settings
private ConfigData config;

// queue of usernames to updated, shared between threads
private BlockingQueue<String> queue;

// mutex locks to prevent activity fetchers from colliding
private ConcurrentMap<String, Lock> userLocks;
```

*Über*Activity Viewer **Main Thread**

- Read configuration files
- Set up logging with log4j
- Initialize SUP fetcher and activity fetcher threads
- Listen on shutdown port

```
// Accept connections only on the local loopback (127.0.0.1)

ServerSocket shutdownSocket = new ServerSocket(
  config.getShutdownPort(), 0, InetAddress.getByName(null));
shutdownSocket.accept();
```

*Über*Activity Viewer **SUP Fetcher Thread**

- Retrieve users from DB
- Retrieve SUP feed
- For each found user hash, add their username to a queue

*Über*Activity Viewer Threading

- SUP fetch logic: High level overview

```
// determine next run date
Date nextRun = new Date();

while(true) {
  // if ready to run
  Date now = new Date();
  if(now.after(nextRun)) {
    // Process SUP feed using activity threads
  } else {
    // Sleep until next run time
    Thread.sleep(SLEEP_TIME);
  }
}
```

*Über*Activity Viewer Threading

- SUP fetch logic: Setting the next run time

```
// set next run time
Calendar cal = Calendar.getInstance();
cal.add(Calendar.SECOND, POLL_FREQUENCY);
nextRun = cal.getTime();
```

*Über*Activity Viewer Threading

- SUP fetch logic: Adding usernames of interest to our queue

```
// create map for usernames and data
Map<String, String> userMap;

userMap = db.getUserMap();

// create map for SUP feed data
Map<String, String> supData = getSupFeedData();

//remove all keys that are not being updated right now
userMap.keySet().retainAll(supData.keySet());

// add values into synchronized queue
queue.addAll(userMap.values());
```

*Über*Activity Viewer **Activity Fetcher Thread(s)**

- Grab username off the queue
- Acquire lock on that username
- Retrieve activity feed for user
- Add activity entries to database (ignore duplicates)
- Release lock

*Über*Activity Viewer Threading

- Activity fetcher logic: Get username from queue

```
String username;

username = queue.take();

// ensure one lock per username
Lock newLock = new ReentrantLock();
Lock lock = userLocks.putIfAbsent(username, newLock);

if(lock == null) {
  lock = newLock;
}
```

*Über*Activity Viewer Threading

- Activity fetcher logic: Get Activity for username

```
if(lock.tryLock()) {

  String etag = db.getEtagForUser(username);
  String updated = db.getFeedUpdatedForUser(username);

  UserEventFeed activity = api.getActivityFeed(
    username, etag, updated);

  db.updateFeedUpdatedForUser(
    username, activity.getUpdated().toString());
  db.updateEtagForUser(username, activity.getEtag());
  db.updateUserActivities(activity);

}
```

Daemon Things To Watch Out For

- SUP feed hasn't been updated
- SUP feed or API feeds return temporary service errors
- DB connection has died

In the Real World

- Other daemon processes can filter and aggregate this data
- Notifications can be sent to users or devices
- Learn from user behavior

*Über*Activity Viewer Website

PHP Website

- Same frontend, except:
  - Handle log-in
    - use AuthSub single use token only to verify username
    - retrieve SUP hash (unauthenticated)
    - Write username & hash to DB
  - Read JSON from DB
    - Page in MySQL
    - Add metadata from API
    - Write paged feed, video entry and user entry to memcache

*Über*Activity Viewer Website

- Handle log-in, use AuthSub single use token only

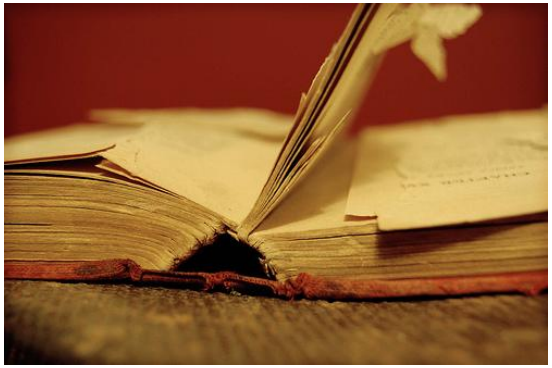
```
// Using authenticated YouTube service,
// get profile for currently authenticated user
$userProfile = $yt->getUserProfile('default');

// Get username and user hash for activity
$username = $userProfile->getUsername();

$feed = $yt->getFeed(
  'http://gdata.youtube.com/feeds/api/users/' . $username .
  '/events');

$hashHref = $feed->getLink('updates')->getHref();
// strip out hash token

// write to DB
```

*Über*Activity Viewer Website

- Get activity feed from DB, add metadata, json_encode

```
$dbFeedArray = getFeedFromDB($page);
$metaDataFeedArray = addMetaData($dbFeedArray);
$jsonOutput = json_encode($metaDataFeedArray);

// user profile metadata
$userProfile = $yt->getUserProfile($username);
$profileData['about_me'] = $userProfile->getAboutMe()->text;
. . .

// video entry metadata
$videoEntry = $yt->getVideoEntry($videoId);
```

*Über*Activity Viewer Website



- Memcache each paged section
- Page in MySQL:

```
// clean the page variable, making sure its integer or
// default to 1

$statement = $mysqli->prepare(
  "SELECT username, updated, json FROM activity ORDER BY " .
  "updated DESC LIMIT ? OFFSET ?");
  $offset = $numItemsPerPage*$page;
  $statement->bind_param('ii', $numItemsPerPage, $offset);
```

Resources

- code.google.com/apis/youtube
- **Office hours**: now - 4 outside!

Q & A

Post your questions for this talk on Google Moderator:
http://tinyurl.com/io-social-youtube

Photography Credits

- Lion image: http://www.flickr.com/photos/**mirsasha**/312674919
- Kittens: http://www.flickr.com/photos/**mathias-erhart**/2562106102
- Scaling: http://www.flickr.com/photos/**roblee**/810410426
- SUP Cat: http://www.flickr.com/photos/**rezsox**/397041105
- Book page: http://www.flickr.com/photos/**just_jump**/2972461681

All images used under the Creative Commons license.


Google™
09
I/O