Media APIs: Tips, Code Samples, Explanations, and Downloads

Jason A. Clark
Head of Digital Access and Web Services
Montana State University Libraries
jaclark@montana.edu
twitter.com/jaclark

Resources and Tools

- oEmbed guide http://www.oembed.com/
- YouTube Data API scratchpad http://stage.gdata.youtube.com/demo/index.html
- Google Code Playground http://code.google.com/apis/ajax/playground/
- Yahoo Query Language Console http://developer.yahoo.com/yql/console/
- Flickr API Explorer http://www.flickr.com/services/api/explore/?method=flickr.photos.search
- iTunes and App Store API http://www.apple.com/itunes/affiliates/resources/documentation/itunes-store-web-service-search-api.html

Media Web Services – Sample Apps

- oEmbed: Flickr, Youtube, etc.
- YouTube Basic
- blip.tv Basic
- Vimeo Basic
- blip.tv Widget (PHP, jQuery)
- Search iTunes using Google's Ajax Search API
- New Yorker Fiction Podcast Mashup
- Library Lifestream
- Flickr Feed using Tags Search
- Flickr Search using Yahoo Query Language

View samples and download code at http://www.lib.montana.edu/~jason/files.php

Code Sample #1: oEmbed: Flickr, YouTube, etc.

HTML source:

```
...
<div><a href="http://vimeo.com/16607768" class="oembed">Vimeo Video</
a></div>
<div><a href="http://www.youtube.com/watch?v=3IHzHCZtJVg"
class="oembed">YouTube Video</a></div>
<div><a href="http://www.flickr.com/photos/14516334@N00/345009210/"
class="oembed">Flickr Image</a></div
...
```

Javascript source:

```
<script type="text/javascript">
     $(document).ready(function() {
          $(".oembed").oembed(null,
          {
            embedMethod: "append",
            maxWidth: 1024,
            maxHeight: 768,
            vimeo: { autoplay: false, maxWidth: 200, maxHeight:
     200},
            youtube: { autoplay: true, maxWidth: 200, maxHeight:
     200}
          });
       });
     </script>
```

HTML and Javascript Explanation:

- Create links with `class="oembed"`
- Bring in jQuery and jQuery oEmbed plugin
- Tell script to look for "oembed" class and then bring in new markup
  `$(".oembed").oembed`

Code Sample #2: blip.tv Widget (PHP, jQuery)

HTML and CSS source:

```
#videos{width:309px;padding:12px 0px 0px 0px;background-
color:#fff;height:340px;}
.left{float:left;width:100px;margin-right:10px;}
.right{float:left;width:150px;}
.clear{clear:both;}
.vids{float:left;margin-right:2px;width:300px;padding:4px;}
.video-title{margin-bottom:5px;font-size:13px;}
.video-title a:hover{color:#666;}
.video-title a{color:#000;font-weight:700;padding:2px;text-
decoration:underline;}
.odd{background-color:#ccc;border-bottom:1px solid #999;border-
top:1px solid #999;}
.even{background-color:#dedede;}
#pagination{margin-bottom:5px;padding:5px;text-align:right;}
#pagination a{color:#333;text-decoration:underline;}
.qp_counter{margin-left:5px;margin-right:5px;color:#000;}
.nojs{background-
color:#FFCC00;padding:20px;margin:10px;border:1px solid
#FF9900;}
```

```
#title{font-size:20px;margin-left:5px;margin-bottom:5px;}

<div id="videos">
<div id="title">mediaHub @ MSU</div>
  <div id="content">
     <div class="nojs" align="center">You need to enable
JavaScript to view the video feed</div>
  </div>
  <div class="clear"></div>
  <div id="pagination"></div>
</div>
```

HTML and Javascript Explanation:

- Set up styles and layout for widget
- Create markup for widget to write into `<div id="videos">`
- Bring in jQuery and jQuery Pagination plugin

Javascript source:

```
<script type="text/javascript" src="http://ajax.googleapis.com/
ajax/libs/jquery/1.4.3/jquery.min.js"></script>
<script src="quickpaginate.js" type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function(){
     $('#content').hide();
     $('#content').load('feed.php', function(){
     $('#content').fadeIn('slow');
     $('.vids').quickpaginate({ perpage: 2, pager :
$("#pagination") });
});
});
</script>
```

Javascript Explanation:

- Hide content on load of page
- Bring in PHP script to parse XML file `.load('feed.php'`
- Set up pagination `$('.vids').quickpaginate`

PHP source:

```
…
function parseRSS($xml) {
               foreach($xml->channel->item as $entry) {
                    $namespaces = $entry->getNameSpaces(true);
                    $media = $entry-
>children($namespaces['media']);
                    $blip = $entry->children($namespaces['blip']
```

```
);
                    $description = $blip->puredescription;
                    $duration = date("i:s",intval($blip->runtime)
);
               $length = strlen($description);
                    if($length > 20){
                         $description = substr($description, 0,
20)."...";
                    }


                    if ($entry->title) {
                         if($odd = $i%2){$class='odd';}
else{$class='even';}
                         echo "<div class='vids ".$class."'>";
                         echo '<div class="left"><a class="ms-
video" target="_top" href="'.$entry->link.'"><img width="100"
border="0" src="'.$media->thumbnail->attributes()->url.'"
alt="'.$entry->title.'" /></a></div>';
                         echo '<div class="right"><div
class="video-title"><a class="ms-video" target="_top"
href="'.$entry->link.'"><span>'.$media->title.'</span></a></
div>';
                         echo '<div class="video-
description">'.$description.'</div>';
                        echo  $duration.' mins</div>';
                         echo '<div class="clear"></div>';
                         echo "</div>";
                    }

                    $i++;
               //end foreach
               }

          }
```

PHP Explanation:

- Use PHP to “cherrypick” values from RSS feed
- PHP script passed in by Javascript

Code Sample #3: Search iTunes using Google's Ajax Search API

HTML (some PHP) Markup

```
<form id="searchForm" name="searchForm" action="<?php echo
basename(__FILE__); ?>" method="get">
```

```
<h2 class="mainHeading">

<ol<?php echo @$_GET['start'] > 0 ? ' start="' .($_GET['start']
+ 1) .'"' : '' ?>>

<ul class="pages">
```

HTML Explanation:

- Form markup
- Search Results markup
- Count Number of results

PHP File (part 1)

```
//set URL for the Google Ajax Search API
$url = 'http://ajax.googleapis.com/ajax/services/search/web?
v=1.0&q='.urlencode($query).'%20site:'.$site.'&rsz=large&start='.$star
t;

//build request and send to Google Ajax Search API
$request = file_get_contents($url);

//decode json object(s) out of response from Google Ajax Search API
$body = json_decode($request);
```

PHP Explanation

- Make Request to Google Ajax Search API `file_get_contents($url);`
- Store returned values in JSON object `json_decode($request);`

PHP File (part 2)

```
//if(isset($search_results['results'])) {
…
foreach ($search_results['results'] as $sr) {
```

PHP (part 2) Explanation

- IF there are search results loop through items
- Print out title and url values for each search result item

Code Sample #4: Flickr API - Display Photos (JSON)

The URL Request

http://api.flickr.com/services/feeds/photos_public.gne?tags=cil2008&format=json

URL Request: Explanation

- HTTP Request to Flickr API: http://www.flickr.com/services/api/

- API provides data as XML feeds (RSS, ATOM)
- Requesting "/feeds/" with a "format" of JSON (Javascript Object Notation)
- Querying API for all public photos tagged "cil2008" with the "tags" parameter

The URL Request in Javascript

```
<!-- use script tag to make request to flickr api, specify json format
and tag to search -->
<script type="text/javascript" src="http://api.flickr.com/services/
feeds/photos_public.gne?tags=cil2008&format=json">
</script>
```

The URL Request in Javascript - Explanation

- JSON is actually javascript and to make JSON output available we must call it on the page via the <script> tag
- After <script> tag is run, JSON output exists as javascript object ready to be parsed

JSON Response

```
jsonFlickrFeed({
"title": "Photos from everyone tagged cil2008",
"link": "http://www.flickr.com/photos/tags/cil2008/",
"description": "",
"modified": "2008-04-07T18:43:16Z",
"generator": "http://www.flickr.com/",
"items":
[
{
"title": "So many floors",
"link": "http://www.flickr.com/photos/nengard/2395908509/",
"media": {"m":"http://farm4.static.flickr.com/3182/
2395908509_d6452e2d56_m.jpg"},
"date_taken": "2008-04-07T13:07:53-08:00",
"description": "So many floors",
"published": "2008-04-07T18:43:16Z",
"author": "nobody@flickr.com (nengard)",
"author_id": "10137764@N00",
"tags": "hyatt cil2008 cil08"
},
…
]
})
```

JSON Response - Explanation

- More structured data ready to be parsed
- We'll extract the values and format for display using the second javascript

Parse and display with Javascript

```
<script type="text/javascript">
     //run function to parse json response, grab title, link, and
media values - place in html tags
     function jsonFlickrFeed(fr) {
          var container = document.getElementById("feed");
          var markup = '<h1>' + '<a href="' + fr.link+ '">' +
fr.title + '</a>'+ '</h1>';
          for (var i = 0; i < fr.items.length; i++) {
               markup += '<a title="' + fr.items[i].title + '"
href="' + fr.items[i].link + '"><img src="' + fr.items[i].media.m + '"
alt="' + fr.items[i].title + '"></a>';
          }
          container.innerHTML = markup;
     }
</script>
```

Parse and display with Javascript - Explanation

- Create javascript function “jsonFlickrFeed” to parse JSON response returned from first javascript
- Loop statement: “for (var i = 0; i < fr.items.length;i++)” runs through all JSON data nodes
- “`container.innerHTML`” – native javascript function prints out values from JSON in xHTML markup