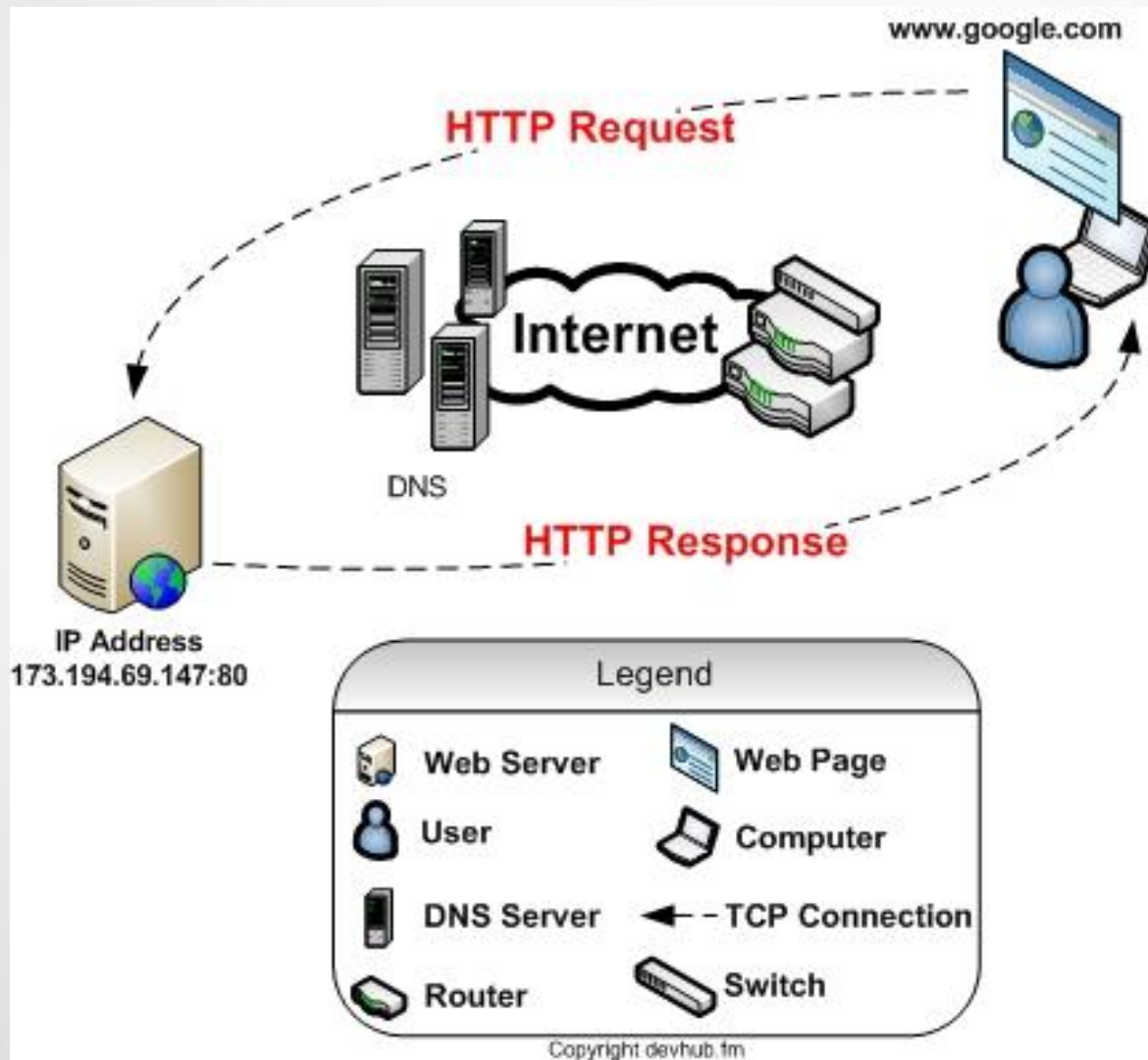


IO Lab: Working with APIs

September 23, 2013

Info 290TA (Information Organization Lab)
Kate Rushton & Raymon Sutedjo-The

HTTP Overview



HTTP Request

<http://web-sniffer.net/>

HTTP Request Header

```
Connect to 170.149.168.130 on port 80 ... ok

GET / HTTP/1.1[CRLF]
Host: www.nytimes.com[CRLF]
Connection: close[CRLF]
User-Agent: Web-sniffer/1.0.46 (+http://web-sniffer.net/) [CRLF]
Accept-Encoding: gzip[CRLF]
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7[CRLF]
Cache-Control: no-cache[CRLF]
Accept-Language: de,en;q=0.7,en-us;q=0.3[CRLF]
Referer: http://web-sniffer.net/[CRLF]
[CRLF]
```

HTTP Response

HTTP Response Header

Name	Value
Status: HTTP/1.1 200 OK	
Date:	Thu, 19 Sep 2013 16:35:11 GMT
Server:	Apache
expires:	Thu, 01 Dec 1994 16:00:00 GMT
cache-control:	no-cache
pragma:	no-cache
Set-Cookie:	RMID=007f01002e39523b27bf0076; Expires=Fri, 19 Sep 2014 16:35:11 GMT; Path=/; Domain=.nyt
Set-cookie:	adxcs=-; path=/; domain=.nytimes.com
Set-cookie:	adxcl=!*35dc6=526f32bf:1; expires=Friday, 19-Sep-2014 16:35:11 GMT; path=/; domain=.nytimes
Vary:	Host
Content-Length:	185138
Connection:	close
Content-Type:	text/html; charset=UTF-8
Content-Encoding:	gzip
Transfer-Encoding:	chunked

Content (encoded: 41.86 KiB / decoded: 180.80 KiB)

```
<!DOCTYPE html>  
<!--[if IE]><![endif]-->  
<html lang="en">
```

HTTP Verbs

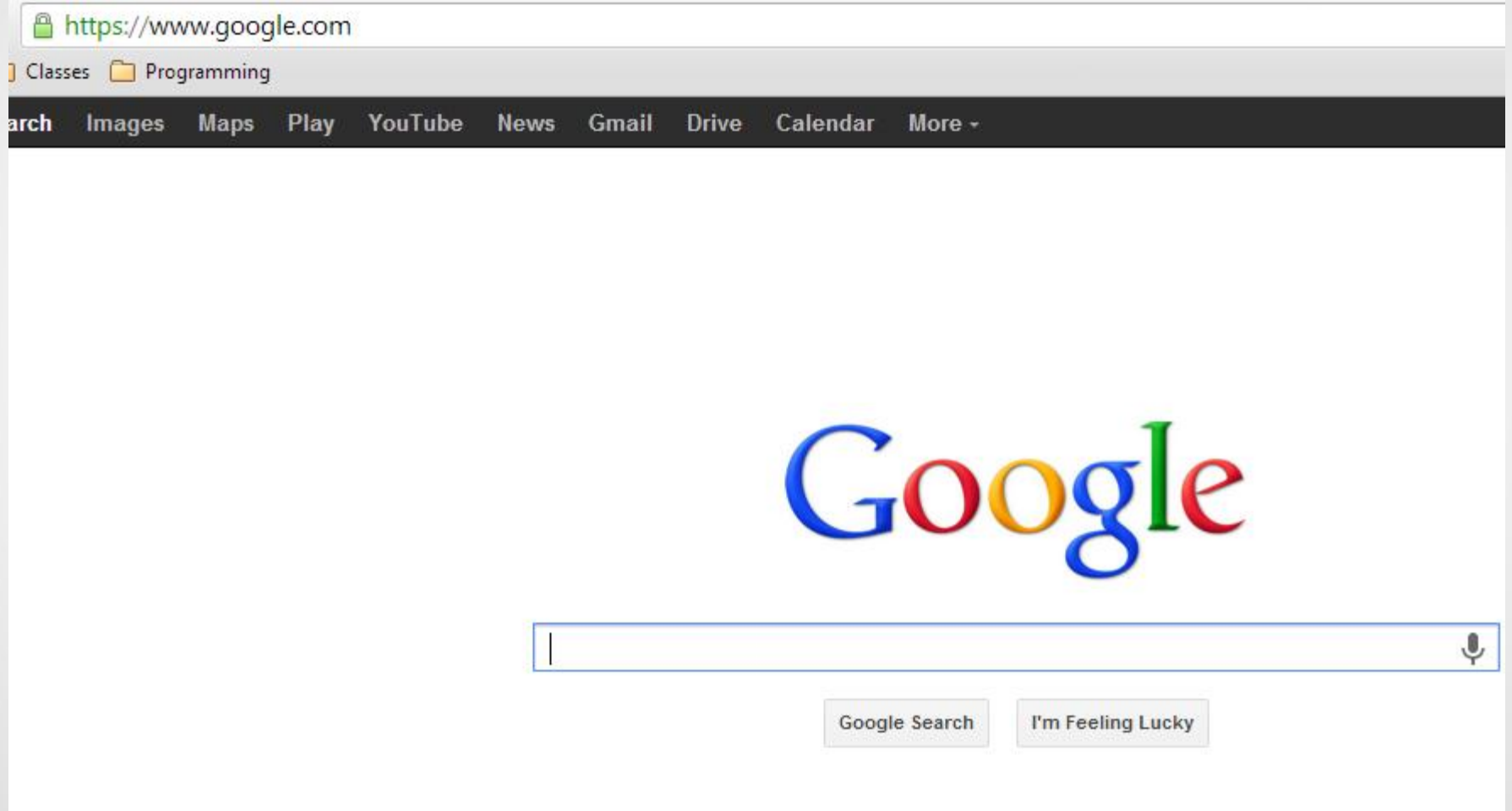
- GET
- POST
- PUT*
- DELETE*

*less common

Which Type of Request?

- Add a book to the collection
- View a list of books
- Check the price of the book with the ID 1
- Remove the book with the ID 1
- Change the price of all books published before 1960

GET Request (Example)



GET Request (Example)

The image shows a screenshot of a Google search page. The address bar contains the URL: `https://www.google.com/search?safe=active&hl=en&q=penguins`. Below the address bar, there are navigation links for 'Classes' and 'Programming'. The main navigation bar includes 'rch', 'Images', 'Maps', 'Play', 'YouTube', 'News', 'Gmail', 'Drive', 'Calendar', and 'More'. The search bar contains the text 'penguins'. Below the search bar, there are tabs for 'Web', 'Images', 'Maps', 'Shopping', 'News', and 'More'. The search results show 'About 63,800,000 results (0.18 seconds)'. The first result is 'Pittsburgh Penguins Scores & Schedule', which includes a table of games. The second result is 'Penguin', which includes a description and classification information.

Classes Programming

rch Images Maps Play YouTube News Gmail Drive Calendar More ▾

le penguins

Web Images Maps Shopping News More ▾ Search tools

About 63,800,000 results (0.18 seconds)

Pittsburgh Penguins
Scores & Schedule

Thu, Oct 3	vs.	Devils	4:00 PM (PT)
Sat, Oct 5	vs.	Sabres	4:00 PM (PT)
Tue, Oct 8	vs.	Hurricanes	4:00 PM (PT)

Penguin
Bird

Penguins are a group of aquat almost exclusively in the Sout especially in Antarctica. [Wikiped](#)

Scientific name: Spheniscid:
Rank: Family
Higher classification: [Sphen](#)
Lower classifications: [Spher](#)

Query string:
`?safe=active&hl=en
&q=penguins`

POST Request (Example)

posting title:

(inner richmond)

posting description:

reply to:

- use craigslist mail relay (recommended) [\[2\]](#)
- show my real email address
- no replies to this email, please

[show on maps \[2\]](#)

street:

optional

cross street:

optional

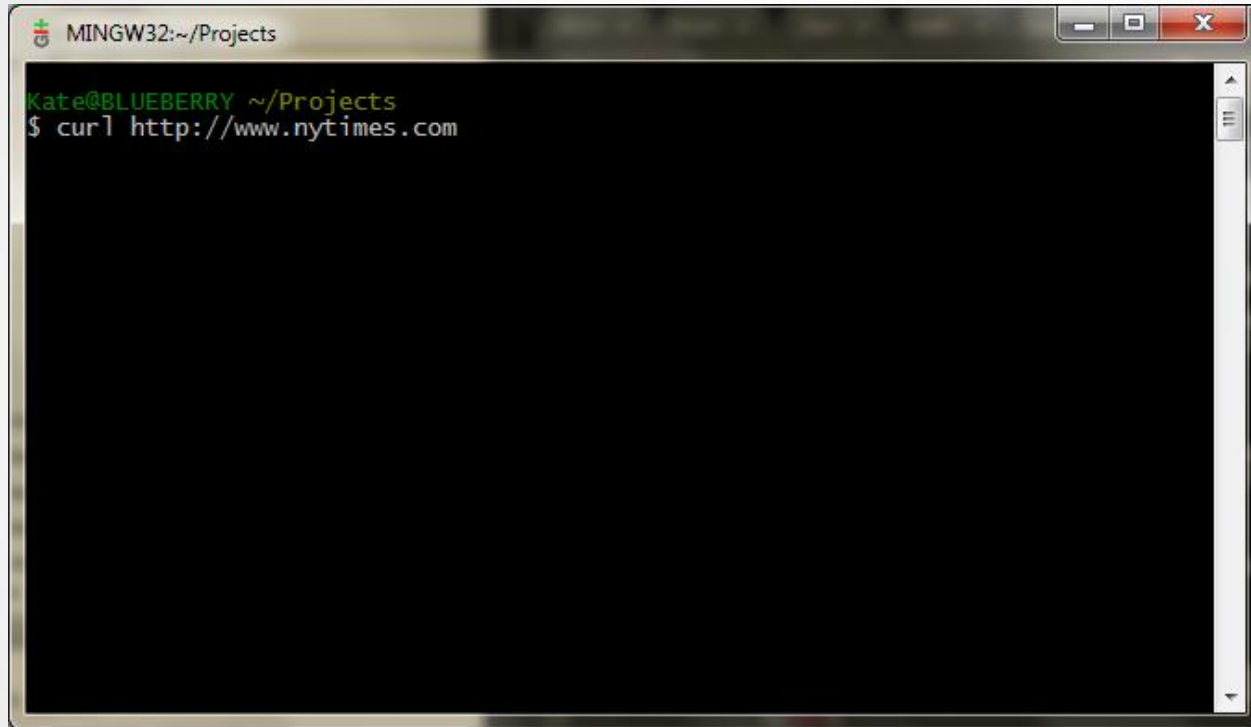
city:

postal:

POST Request (Example)

```
<form name="myForm" action="result.php"  
method="post">  
Posting Title: <input type="text"  
name="title">  
<input type="submit" value="Submit">  
</form>
```

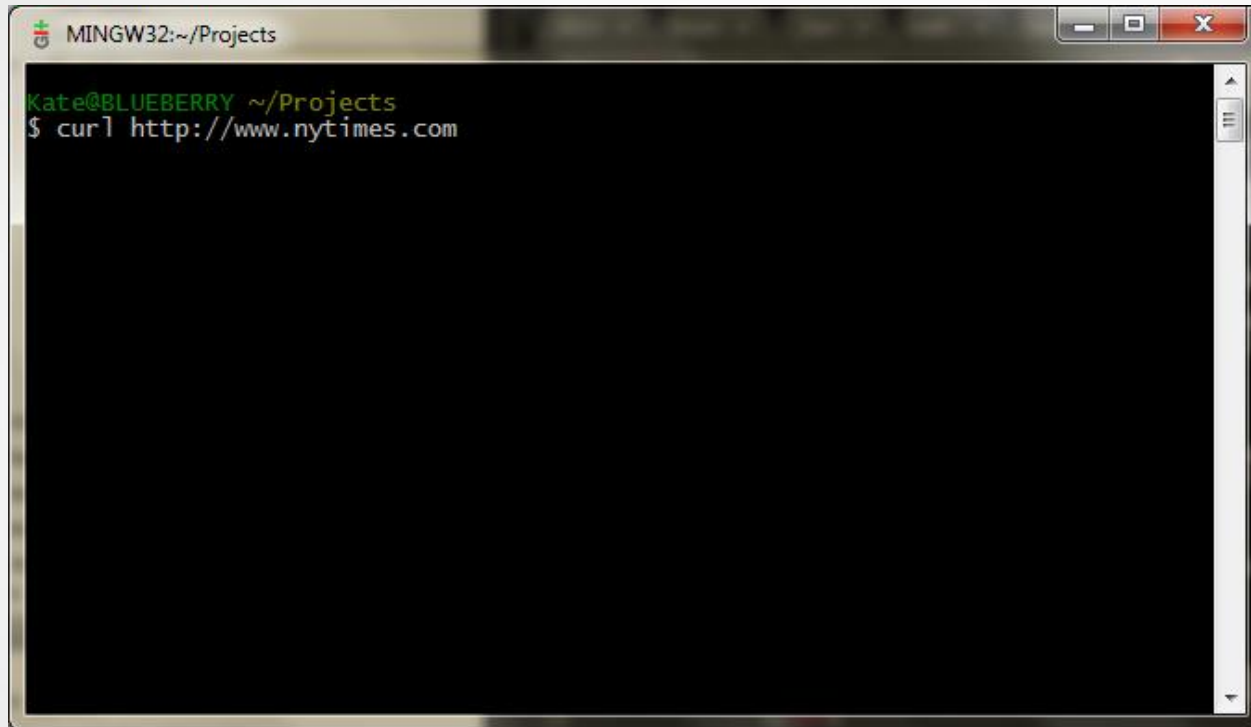
cURL

A screenshot of a terminal window titled "MINGW32:~/Projects". The prompt is "Kate@BLUEBERRY ~/Projects" and the command entered is "\$ curl http://www.nytimes.com". The terminal output is currently empty.

```
MINGW32:~/Projects  
Kate@BLUEBERRY ~/Projects  
$ curl http://www.nytimes.com
```

- **GET**
\$ curl http://www.myurl.com

cURL

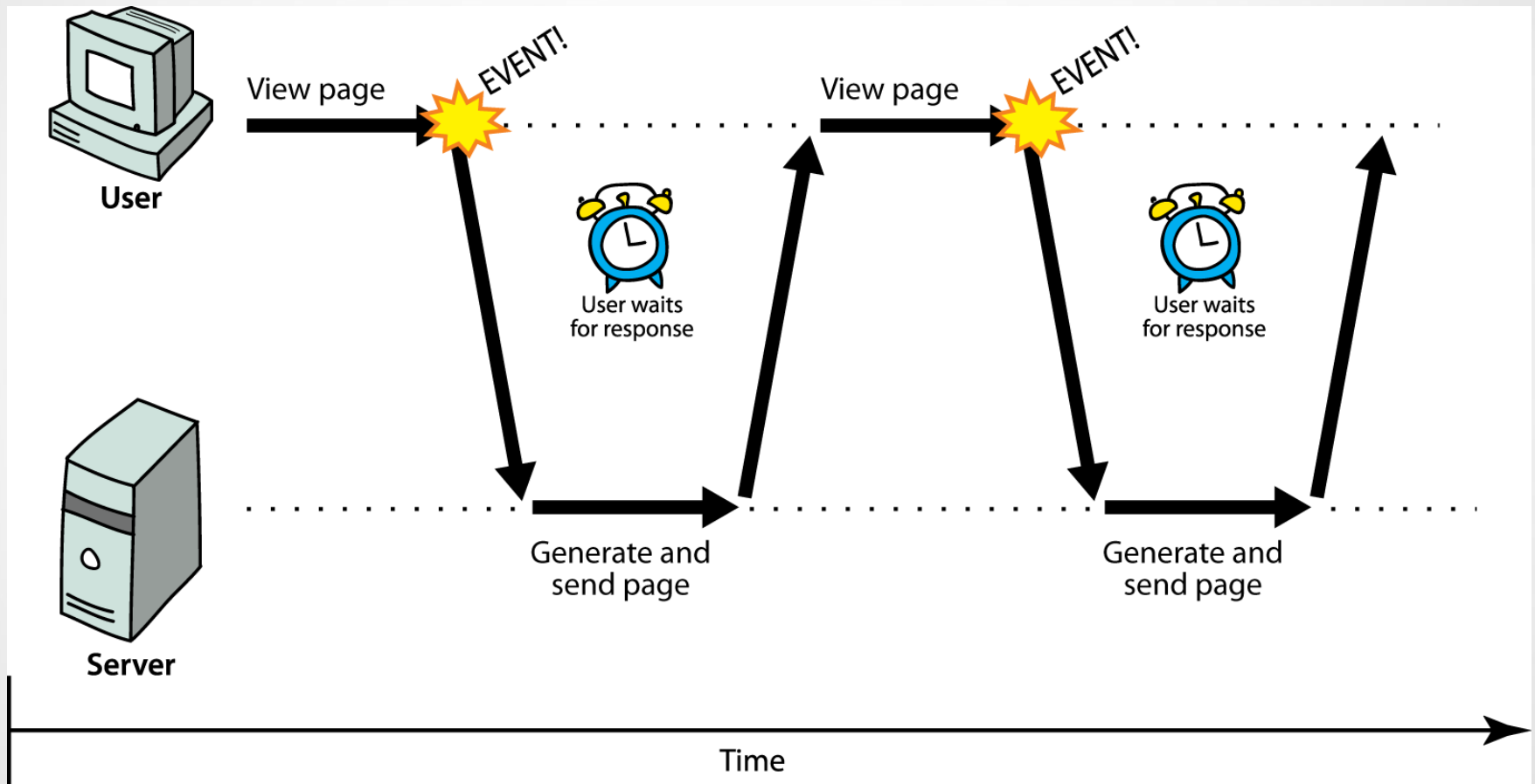
A screenshot of a terminal window titled "MINGW32:~/Projects". The prompt is "Kate@BLUEBERRY ~/Projects" and the command entered is "\$ curl http://www.nytimes.com". The terminal output is currently empty.

```
MINGW32:~/Projects
Kate@BLUEBERRY ~/Projects
$ curl http://www.nytimes.com
```

- **POST**

```
$ curl --data "key1=val1&key2=val2"
http://myurl.com
```

Synchronous



Source: <http://courses.cs.washington.edu/courses/cse190m/>

Synchronous

The screenshot shows the OskiCat UCB Library Catalog search results page. At the top, there is a blue header with the OskiCat logo and the text "UCB Library Catalog". Below the header, there are several navigation buttons: "Quick Search", "Advanced Keyword Search", "Find Articles", "Course Reserves", "Renew Items", and "Questions?". A "Start Over" link is also visible in the top right corner.

The search results section shows a search for "penguins" in the "Keyword(s)" field. The search is limited to the "Entire Collection" and the results are sorted by "relevance". There are 226 results found. The search results are displayed in a table with columns for "Location", "Call No.", and "Status".

Buttons for "START OVER", "MODIFY SEARCH", and "ANOTHER SEARCH" are located above the search input fields. A "Search History" link is also present.

Navigation buttons for "SAVE SELECTED RECORDS", "SAVE ALL", and "SAVE TO MY LIST" are located below the search results. The "Result Page" navigation shows page 1 selected, with options for pages 2, 3, 4, 5, and "Next".

The search results are displayed in a table with columns for "Location", "Call No.", and "Status".

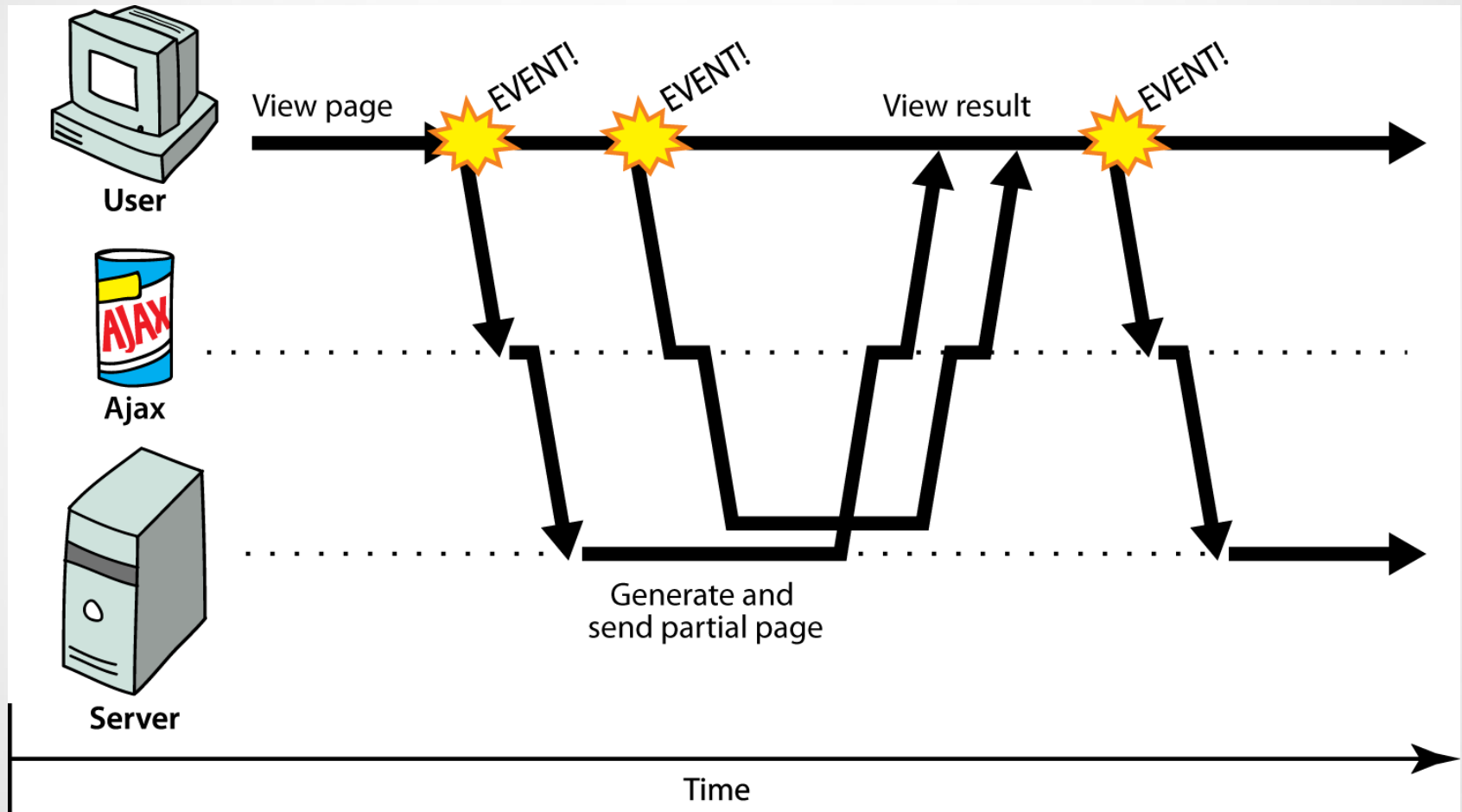
	Location	Call No.	Status
1	Bioscience & Natural Resources	QL696.S473 P4527 2013	AVAILABLE
2			

Most relevant titles entries 1-64

1 [Penguins : natural history and conservation / edited by Pablo Garcia Borboroglu and P. Dee Boersma.](#)
Seattle : University of Washington Press, [2013]

2 [Penguins of the world / text and photographs by Wayne Lynch.](#)
Lynch, Wayne.
Richmond Hill, Ont. : Firefly Books, c2007.
[Request](#)

Asynchronous



Source: <http://courses.cs.washington.edu/courses/cse190m/>

Asynchronous

The image shows a browser window with the URL <https://www.google.com>. The search bar contains the text "penguins". A dropdown menu is visible below the search bar, listing suggestions: "penguins", "penguins – Pittsburgh Penguins, Ice hockey team", "penguins – Bird", "penguins **of madagascar**", "penguins **schedule**", and "penguins **and pajamas**".

Below the search bar, the search results are displayed. The first result is "Pittsburgh Penguins Scores & Schedule". The second result is "Penguin", which is a knowledge panel for the bird species.

Date	vs.	Team	Time
Thu, Oct 3	vs.	Devils	4:00 PM (PT)
Sat, Oct 5	vs.	Sabres	4:00 PM (PT)
Tue, Oct 8	vs.	Hurricanes	4:00 PM (PT)

Penguin
Bird

Penguins are a group of almost exclusively in the especially in Antarctica

Scientific name: Sph
Rank: Family
Higher classification:
Lower classifications

AJAX

- Asynchronous
- JavaScript
- And
- XML (or HTML, JSON, etc.)

AJAX

- Not a programming language
- A particular way of using JavaScript
- The client (web browser) downloads data in the background and dynamically updates the page
- Avoids the click-wait-refresh cycle

Web Applications

CMC Final Paper ☆ 📁

File Edit View Insert Format Tools Table Help Last edit was made on May 22 by zhaoyang

100% Normal text Times Ne... 14 **B** *I* U A ↺ 📄 | ☰ ☰ ☰ ☰ | ↕ ☰ ☰ | ☰ ☰ ☰ ☰ ☰

1 2 3 4 5 6 7


Personality Embodiment in Multicharacter Online Games


Divya Anand, Evie Phan, Kate Rushton, Taeil Kwak, Yang Zhao



Introduction

According to industry reports, seventy-two percent of American households play computer or video games. Many of the top-selling games are in the action, role-playing, shooter, and strategy genres, which often require the player to create or select a character at the start of play (ESA 2011). From that


Web Applications

Search 

Pinterest Add 











Pinners you follow · Everything  · Videos · Discussions · Gifts 

People You May Know


 **Patricia Clark** Follow

View All Follow All (8)


Recent activity


-  **Nikki Steinke** liked your pin.
-  **Stephanie Lord** is now following you.
-  **TsuZenne** <3 liked your pin.
-  **Abbey S** liked your pin.
-  **Holly Leclair** liked your pin.
-  **Brandy Dodd** liked your pin.
-  **Gina Burpee** liked your pin.
-  **Stephanie Torres and Brooke Stacey** liked your pin.
-  **nordish girl** and 5 others are now following your board.
-  **Cortney Porter** liked your pin.


14 Repin Like Comment




{ pumpkin creme brulee }


 **Melissa Kristofferson** onto beautiful food.






metal grate display board


 **Rachel Damiano** via **Katie Hart** onto DIY Interiors







1. TIED FABRIC STRIPS
2. PAPER FLAGS
3. YARN POM POMS
4. SCRAP FABRIC BOWS
5. VINTAGE WALLPAPER


Garlands / 10 Ways

 **Vandi Mellon** via **Laurel Becker** onto Let's make it!




laura lombardi
1 like

 **Cici Absolutely white** via **Luce de Luce** onto Jewelry



Grey White Square Stone Adjustable Ring

 **Cici Absolutely white** onto Gifts

XMLHttpRequest

- JavaScript object for performing AJAX requests
- Problem: Cross-browsing compatibility annoyances
- Problem: Ugly syntax

XMLHttpRequest

```
var xhr = new XMLHttpRequest();  
xhr.onreadystatechange = function() {  
    if (xhr.readyState == 4) {  
        alert(xhr.responseText);  
    }  
}  
  
xhr.open('GET', 'http://example.com',  
true);  
  
xhr.send(null);
```

jQuery + AJAX

- jQuery to the rescue!

```
$.get('http://example.com',  
function(responseText) {  
    alert(responseText);  
});
```

jQuery AJAX Request

1. user clicks, invoking an event handler function in your code
2. the handler creates an AJAX request (XMLHttpRequest)
3. the browser sends the request to the server
4. when the server responds, jQuery fires another event (callback)
5. The callback function updates the page

jQuery AJAX Request

```
<div>  
  <div id='output'></div>  
  <button id='loadButton'>Load</button>  
</div>
```

jQuery AJAX Request

```
$( '#loadButton' ).click(function() {  
    $.get( 'http://myurl.com',  
        function(data) {  
            $( '#output' ).text(data);  
        }  
    }  
});
```

Example 1

jQuery AJAX Variations

- Default: `$.ajax()`
- Shorthand: `$.get()`
`$.post()`
`$.getJSON()`
`$.load()`

<http://api.jquery.com/category/ajax/shorthand-methods/>

jQuery AJAX Variations

```
$.ajax({  
    url: 'http://mywebsite.com/search',  
    type: 'GET',  
    data: {'q': 'penguins'},  
    dataType: 'json',  
    success: function(data) {  
        //do something with data  
    }  
});
```

Example 2

XSS (Cross-Site Scripting)

- In general, browsers must adhere to the *same origin policy*
- AJAX requests to a server must come from the same place – the domain, protocol, and port must match
- If they do not, your AJAX request will fail with an error.

Why?



Workarounds

- JSONP
- Cross-Origin Resource Sharing (CORS)
- Proxy Server

JSONP

- Requires explicit server support
- Kind of hacky – dynamically generates a `<script>` tag that loads JavaScript from a remote server
- The remote server's code in turn calls a function in your code (the “callback” function)

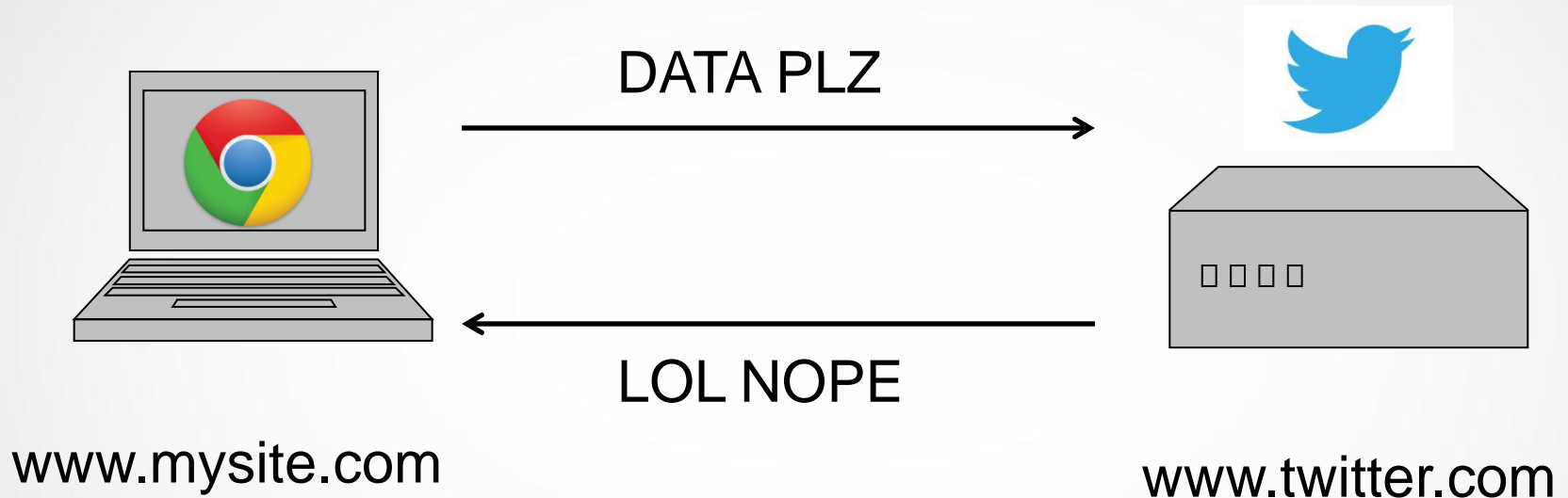
CORS

- Requires explicit server support
- Server-side code can set the Access-Control-Allow-Origin header to explicitly allow a cross-domain request
- Either return the name of the server if it is approved, or “*” for all domains (only use with public resources)
- Does not work in IE ☹️

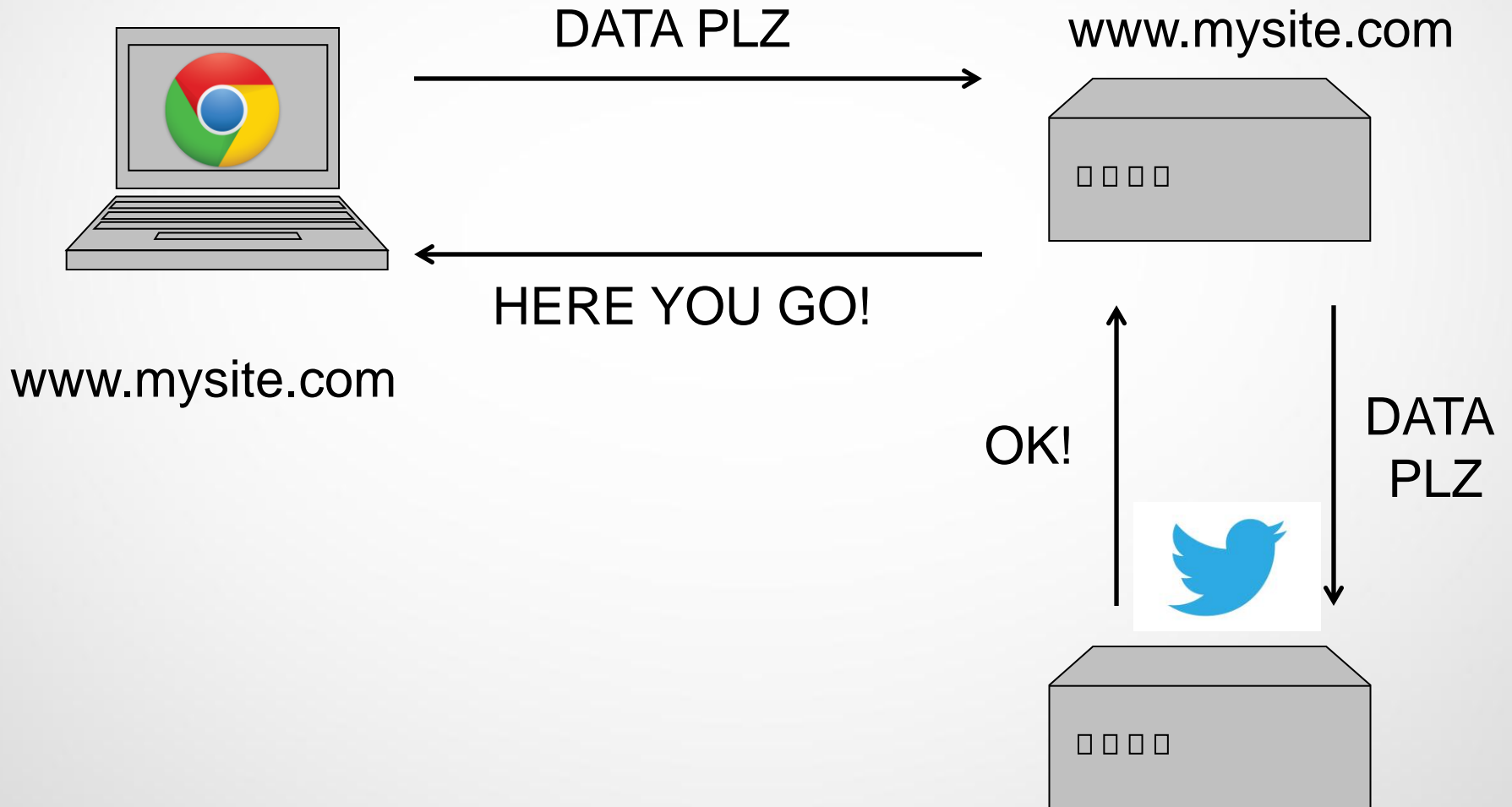
Proxy Server

- Does not require any special behavior from the remote server
- AJAX requests are made to the same origin web server
- The server-side code communicates to the remote server on behalf of the client, then returns the requested data

AJAX Request (Client Only)



AJAX Request (Via Proxy)



So. Much. Data.



Source: <http://www.ischool.berkeley.edu/>

APIs

- A way for your code to interact with other software/systems
- Web API: An intentionally arranged collection of web resources and the interactions they support
- Accessed through HTTP requests

APIs

- Basically, an API (and its documentation) defines for developers:
 - What resources are available?
 - What properties do those resources have?
 - How do I access them?
 - How do I create my own?

Last.fm API

API Methods

Album

- Album.addTags
- Album.getBuylinks
- Album.getInfo
- Album.getShouts
- Album.getTags
- Album.getTopTags
- Album.removeTag
- Album.search
- Album.share

Artist

- Artist.addTags
- Artist.getCorrection
- Artist.getEvents
- Artist.getInfo
- Artist.getPastEvents
- Artist.getPodcast
- Artist.getShouts
- Artist.getSimilar
- Artist.getTags
- Artist.getTopAlbums
- Artist.getTopFans
- Artist.getTopTags
- Artist.getTopTracks
- Artist.removeTag
- Artist.search

Geo

- Geo.getEvents
- Geo.getMetroArtistChart
- Geo.getMetroHypeArtistChart
- Geo.getMetroHypeTrackChart
- Geo.getMetroTrackChart
- Geo.getMetroUniqueArtistChart
- Geo.getMetroUniqueTrackChart
- Geo.getMetroWeeklyChartlist
- Geo.getMetros
- Geo.getTopArtists
- Geo.getTopTracks

Group

- Group.getHype
- Group.getMembers
- Group.getWeeklyAlbumChart
- Group.getWeeklyArtistChart
- Group.getWeeklyChartList
- Group.getWeeklyTrackChart

Library

- Library.addAlbum
- Library.addArtist
- Library.addTrack
- Library.getAlbums
- Library.getArtists

Track

- Track.addTags
- Track.ban
- Track.getBuylinks
- Track.getCorrection
- Track.getFingerprintMetadata
- Track.getInfo
- Track.getShouts
- Track.getSimilar
- Track.getTags
- Track.getTopFans
- Track.getTopTags
- Track.love
- Track.removeTag
- Track.scrobble
- Track.search
- Track.share
- Track.unban
- Track.unlove
- Track.updateNowPlaying

User

- User.getArtistTracks
- User.getBannedTracks
- User.getEvents
- User.getFriends
- User.getInfo

album.getInfo

is
lists API
s
T requests
-RPC requests
r codes
is of Service

Methods

im
um.addTags
um.getBuylinks
um.getInfo
um.getShouts
um.getTags
um.getTopTags
um.removeTag
um.search
um.share

it
ist.addTags
ist.getCorrection
ist.getEvents
ist.getInfo
ist.getPastEvents
ist.getPodcast
ist.getShouts

Params

artist (Required (unless mbid)) : The artist name
album (Required (unless mbid)) : The album name
mbid (Optional) : The musicbrainz id for the album
autocorrect[0|1] (Optional) : Transform misspelled artist names into correct artist names, returning the correct version instead. The corrected artist name will be returned in the response.
username (Optional) : The username for the context of the request. If supplied, the user's playcount for this album is included in the response.
lang (Optional) : The language to return the biography in, expressed as an ISO 639 alpha-2 code.
api_key (Required) : A Last.fm API key.

Auth

This service does **not** require authentication.

Sample Response

```
<album>
  <name>Believe</name>
  <artist>Cher</artist>
  <id>2026126</id>
  <mbid>61bf0388-b8a9-48f4-81d1-7eb02706dfb0</mbid>
  <url>http://www.last.fm/music/Cher/Believe</url>
  <releasedate>6 Apr 1999, 00:00</releasedate>
  <image size="small">...</image>
  <image size="medium">...</image>
  <image size="large">...</image>
  <listeners>47602</listeners>
```

album.addTags

Params

artist (Required) : The artist name

album (Required) : The album name

tags (Required) : A comma delimited list of user supplied tags to apply to this album. Accepts a maximum of 10 tags.

api_key (Required) : A Last.fm API key.

api_sig (Required) : A Last.fm method signature. See [authentication](#) for more information.

sk (Required) : A session key generated by authenticating a user via the authentication protocol.

Auth

This service requires authentication. Please see our [authentication](#) how-to.

This is a **write** service and must be accessed with an HTTP POST request. All parameters should be sent in the POST body, including the 'method' parameter. See [rest requests](#) for more information.

Sample Response

```
<lfm status="ok">
</lfm>
```

Errors

- 2 : Invalid service - This service does not exist
- 3 : Invalid Method - No method with that name in this package
- 4 : Authentication Failed - You do not have permissions to access the service
- 5 : Invalid format - This service doesn't exist in that format
- 6 : Invalid parameters - Your request is missing a required parameter

Delicious API

<https://delicious.com/developers>

Flickr Gallery

<https://github.com/krushton/iolab-lecture6>