

# Using an API



(adamculp)



## ADAM CULP

Awesomeness Creator! SoFloPHP Organizer

### Current Position

Senior Web Developer & Consultant  
**Unique Web Sites, Inc.**



---

**Geographical Location**  
Boca Raton, Florida

**URL**  
<http://www.geekyboy.com>

# Using an API

- Tools
  - HTTPie
  - CURL
  - Charles – <http://www.charlesproxy.com>
  - Firebug – <https://getfirebug.com>



# Using an API

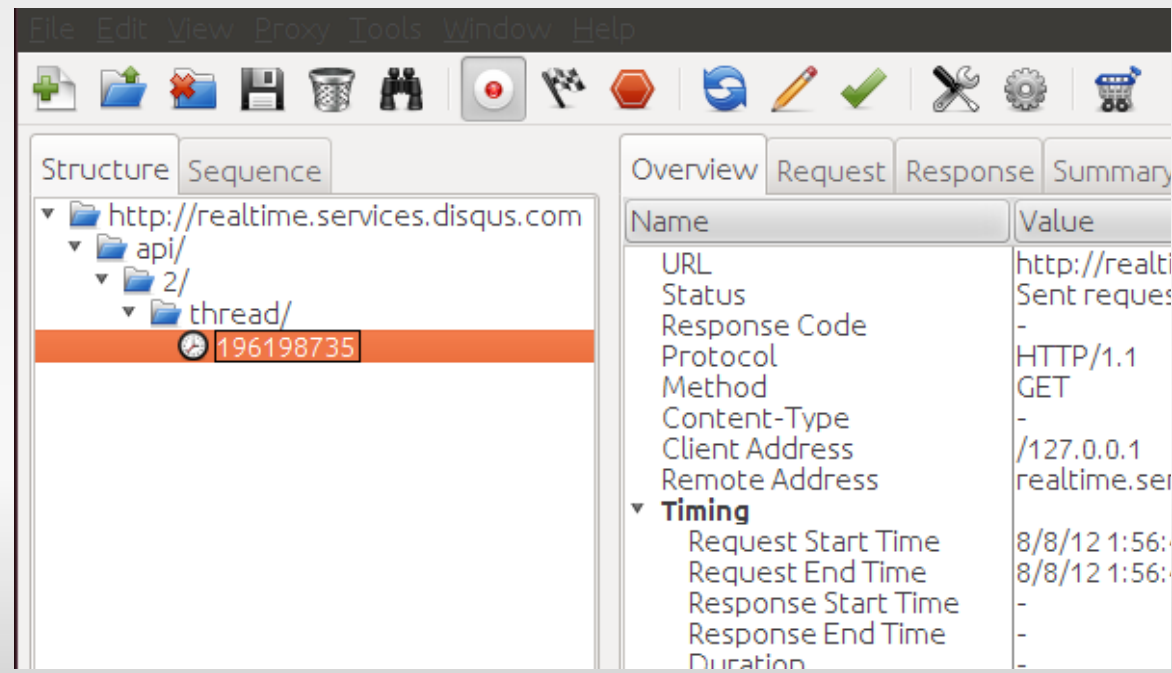
- HTTPie
  - Great for debugging APIs rather than using cURL
  - <http://www.httpie.org>
  - Free
  - Requires Python

```
POST /books HTTP/1.1
HOST: example.com
Content-Type: application/hal+json
Accept-Encoding: identity, deflate, compress, gzip
Accept: application/hal+json
User-Agent: HTTPie/0.2.0

{
  "author": "Stoyan Stefanov",
```

# Using an API

- Charles
  - <http://charlesproxy.com>
  - Great for debugging AJAX and Flash AMF requests
  - Worth the \$50 license fee, but (at least on Ubuntu) you can use it free, though it shuts down after 30 minutes.



# Using an API

- cURL
  - Great to use via PHP, but for command line use HTTPie.
- Firebug
  - Great for debugging from Firefox.
- Others
  - I hear IE has an OK debugging tool like Firebug.
  - I have also heard Chrome has a great Firebug'ish tool also.

# Using an API

- What is an API?
  - Application programming interface
    - A specification intended to be used as an interface by software components to communicate with each other. - wikipedia



# Using an API

- API types (popular)
  - SOAP – Simple Object Access Protocol
  - REST – Representational State Transfer
    - (RFC 2616)
  - Javascript
  - Screen scrape



# Using an API

- SOAP basics
  - Relies on XML as its message format.
  - Uses HTTP, SMTP, TCP, or JMS for transport, but usually HTTP.
  - Generally uses a WSDL (Web Services Description Language) configuration file making it easier to build for or connect to the API.
  - Cross platform and cross language.
  - Very structured – each action is defined (addUser, getUsers, getNewUsersSince, etc.)



# Using an API

- Javascript basics
  - Client side execution
  - Communication from client to server
  - Usually methods are defined on server side (getThis, addThat, etc.)
  - Perception of interactivity (desktop feel)
  - Can be asynchronous
  - Used by Google Maps and others

# Using an API

- Screen scrape basics
  - Not a true client/server api.
  - Relies on regular expression matching of content.
  - Usually done over HTTP using cURL.
  - Breaks when server content changes.
  - Must download all content, so may be slow and resource intensive.

# Using an API

- REST basics (main focus for this talk)
  - Client-Server
  - Stateless – each request is complete and self contained (struggle with sessions and cookies)
  - Cacheable
  - Layered – client cannot tell if using end-server, allows load balancers, caches, logging, auth.
  - Uniform interface – decoupled from architecture
  - Methods – **GET, POST, PUT, DELETE**, HEAD, OPTIONS, TRACE, CONNECT

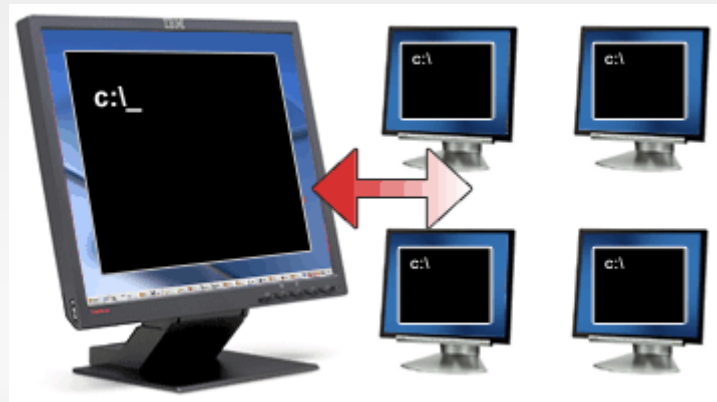
# Using an API

- REST safe methods
  - No side effects on the server
  - GET & HEAD should make no other action than to retrieve.
  - Allows user agents to represent POST, PUT, and DELETE special. (most browsers do not support PUT and DELETE out of the box)



# Using an API

- Idempotence
  - Operations that can be applied multiple times without changing the result. (Ex.-  $N > 0$  identical requests would have the same output)
  - GET, HEAD, PUT, and DELETE share this property
  - OPTIONS and TRACE are inherently idempotent.



# Using an API

- GET
  - Safe and idempotent
  - No effect on the server

```
GET /books/9790482c HTTP/1.1
Host: example.com
Accept-Encoding: identity, deflate, compress, gzip
Accept: application/hal+json
User-Agent: HTTPie/0.2.0
```

# Using an API

- GET response

```
HTTP/1.1 200 OK
Date: Wed, 08 Aug 2012 09:32:42 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.2
ETag: "9790482c-1"
Last-Modified: Wed, 08 Aug 2012 04:14:30 GMT
Content-Length: 254
Content-Type: application/hal+json

{
  "_links": {
    "self": {
      "href": "http://example.com/books/9790482c"
    }
  },
  "author": "Luke Welling, Laura Thomson",
  "id": "9790482c",
  "isbn10": "0672329166",
```

# Using an API

- POST
  - Not safe
  - Not idempotent (same post multiple times could have different effects on server)
  - Used to create resource
  - Is often used to also update a resource, though PUT is more for updating.





# Using an API

- POST example

```
POST /books HTTP/1.1
HOST: example.com
Content-Type: application/hal+json
Accept-Encoding: identity, deflate, compress, gzip
Accept: application/hal+json
User-Agent: HTTPie/0.2.0

{
  "author": "Stoyan Stefanov",
  "isbn10": "1449320198",
  "isbn13": "9781449320195",
  "publisher": "O'Reilly Media",
  "title": "Javascript for PHP Developers",
  "year": 2012
}
```

# Using an API

- Created Response

```
HTTP/1.1 201 Created
Date: Sun, 29 Jul 2012 23:26:49 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.2
Location: http://example.com/books/decd0562
Etag: "decd0562-1"
Last-Modified: Sun, 29 Jul 2012 23:26:49 GMT
Content-Length: 239
Content-Type: application/hal+json

{
  "_links": {
    "self": {
      "href": "http://example.com/books/decd0562"
    }
  },
  "author": "Stoyan Stefanov",
  "id": "decd0562",
  "isbn10": "1449320198",
```

# Using an API

- PUT
  - Not safe
  - But is idempotent
  - Primarily an update action, but can also be used to create though POST should be used for that.



# Using an API

- Sample PUT

```
PUT /books/decd0562 HTTP/1.1
HOST: example.com
Content-Type: application/hal+json
Accept-Encoding: identity, deflate, compress, gzip
If-Match: "decd0562-1"
Accept: application/hal+json
User-Agent: HTTPie/0.2.0

{
  "_links": {
    "self": {
      "href": "http://example.com/books/decd0562"
    }
  }
  "author": "Stoyan Stefanov",
  "isbn10": "1449320198",
```

# Using an API

- Update response

```
HTTP/1.1 201 OK
Date: Sun, 29 Jul 2012 23:26:49 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.2
Location: http://example.com/books/decd0562
Etag: "decd0562-2"
Last-Modified: Sun, 29 Jul 2012 23:26:49 GMT
Content-Length: 270
Content-Type: application/hal+json

{
  "_links": {
    "self": {
      "href": "http://example.com/books/decd0562"
    }
  },
  "author": "Stoyan Stefanov",
  "id": "decd0562",
  "isbn10": "1449320198",
```

# Using an API

- DELETE
  - Not safe
  - Is idempotent
  - Does not mean to remove the resource, but does mean to remove from public view.

```
DELETE /books/decd0562 HTTP/1.1
Host: example.com
Accept-Encoding: identity, deflate, compress, gzip
Accept: application/hal+json
User-Agent: HTTPie/0.2.0
If-Match: "decd0562-2"
```

# Using an API

- DELETE response

```
HTTP/1.1 204 No Content
Date: Mon, 30 Jul 2012 00:01:44 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.2
Content-Length: 0
Content-Type: application/hal+json
```

# Using an API

- HEAD
  - Same as GET, but doesn't return the body
    - An example of use would be to check content length before actually requesting it.



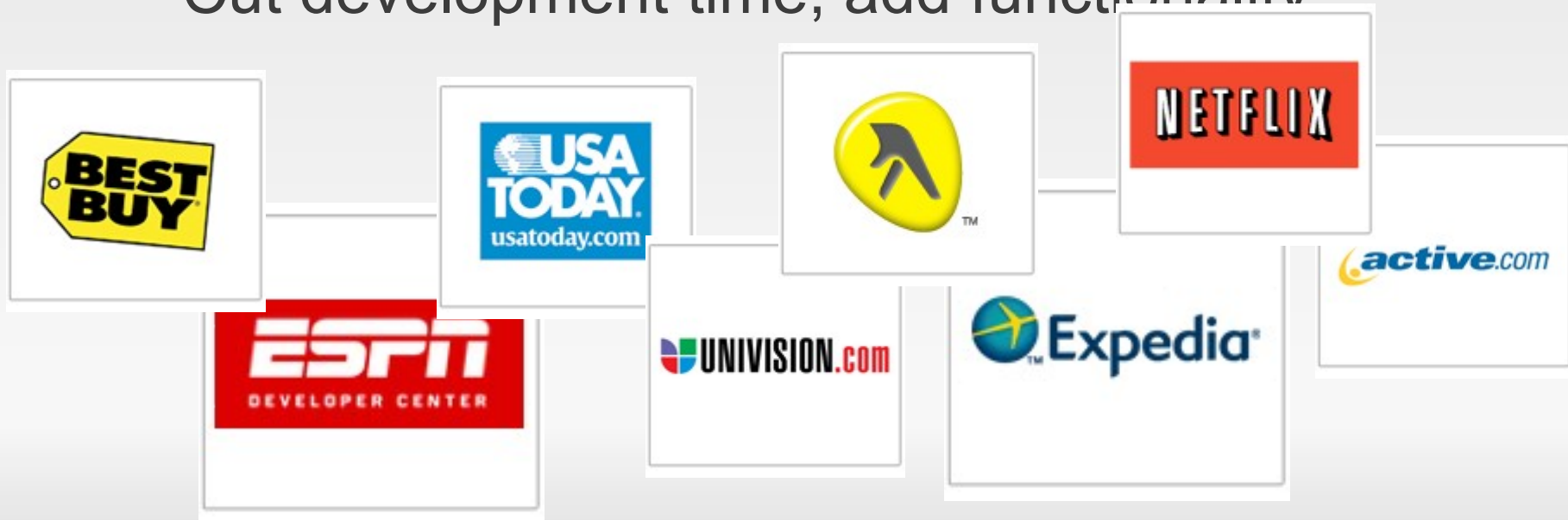
# Using an API

- Status codes
  - Informational (1xx)
  - Successful (2xx)
  - Redirection (3xx)
  - Client error (4xx)
  - Server error (5xx)



# Using an API

- Common usage
  - Most developers aren't building an API
  - Using for sites and application content
  - Leverage resources/services (Google Maps)
  - Cut development time, add functionality



# Using an API

- Resources for great APIs available
  - Mashery.com
    - <https://developer.mashery.com/apis>
  - Apigee.com
    - <https://apigee.com/providers>
  - Google
    - <https://developers.google.com>

**Let's take a look!!!**

# Using an API

- Documentation tools
  - Mashery <http://mashery.github.com>
    - I/O Docs – interactive API documentation
    - I/O Alfred – Alfred app to use some APIs
    - I/O Wraps – API client library generator
    - OAuth Signature Tool
    - OAuth 1.0 JS Testing Tool
    - AppMobi Sample Apps

# Using an API

- Documentation tools
  - apigee
    - Providers list
    - “Console” to experiment with APIs
    - “Console To-Go” to add your own API
    - Usergrid – Backend as a service to get mobile apps running quickly.
    - O-Auth api

# Using an API

- TO THE CODE!!!

# Using an API

- Resources

- <http://www.charlesproxy.com>
- <https://getfirebug.com>
- <http://www.httpie.org>
- <https://developer.mashery.com/apis>
- <https://apigee.com/providers>
- <https://developers.google.com>

# Using an API

- Thank You

Adam Culp  
[@adamculp](mailto:@adamculp)  
<http://geekyboy.com>

Please rate this talk at:  
<http://joind.in/6739>