Introduction to Django

AWPUG, July 2017

Quick survey

http://bit.ly/2v2s9O5

Announcements

- Upcoming expo: DeveloperWeek.com/Austin/
- School’s Out Hackathon looking for mentors: www.sohacks.com

The Plan

- HTTP:
  - Fundamentals
  - Requests
  - Responses
- Django overview
- (Brief) installation overview
- Starting a Django app
- Writing our first view
- Working with requests

The Plan

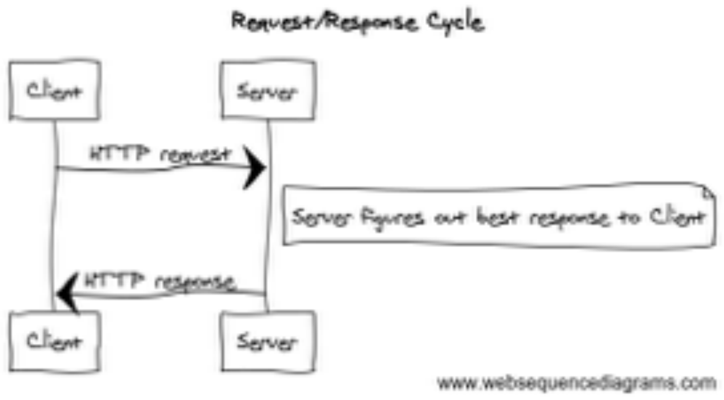
- **HTTP:**
  - **Fundamentals**
  - Requests
  - Responses
- Django overview
- (Brief) installation overview
- Starting a Django app
- Writing our first view
- Working with requests

HTTP

- HTTP: **H**yper**T**ext **T**ransfer **P**rotocol
- TCP-based, simple text protocol
- v0.9 released in 1991, v1.1 released in 1997
  - Wanna read the spec? See RFC 2616

HTTP’s main parts

- Typical function separated into two big parts
- Request
- Response



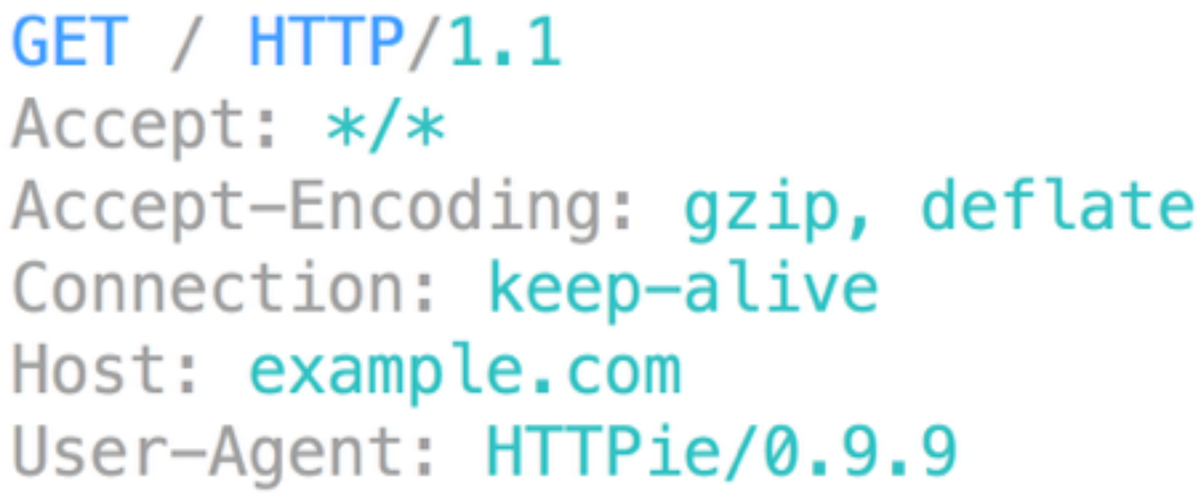
The Plan

- **HTTP:**
  - ~~Fundamentals~~
  - **Requests**
  - Responses
- Django overview
- (Brief) installation overview
- Starting a Django app
- Writing our first view
- Working with requests

Example HTTP request

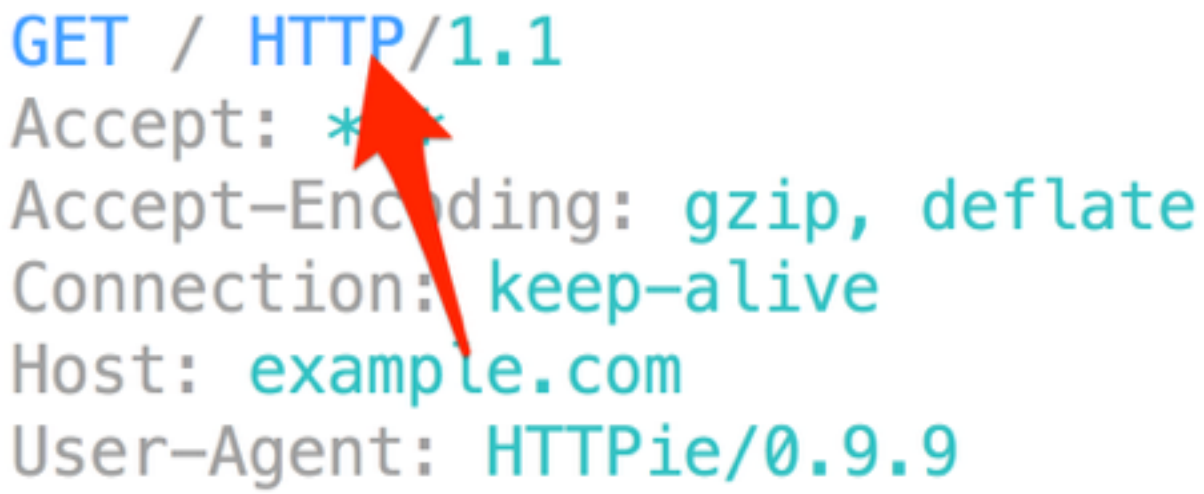
```
$ brew install httpie
...
$ http --verbose example.com
```

Example HTTP request



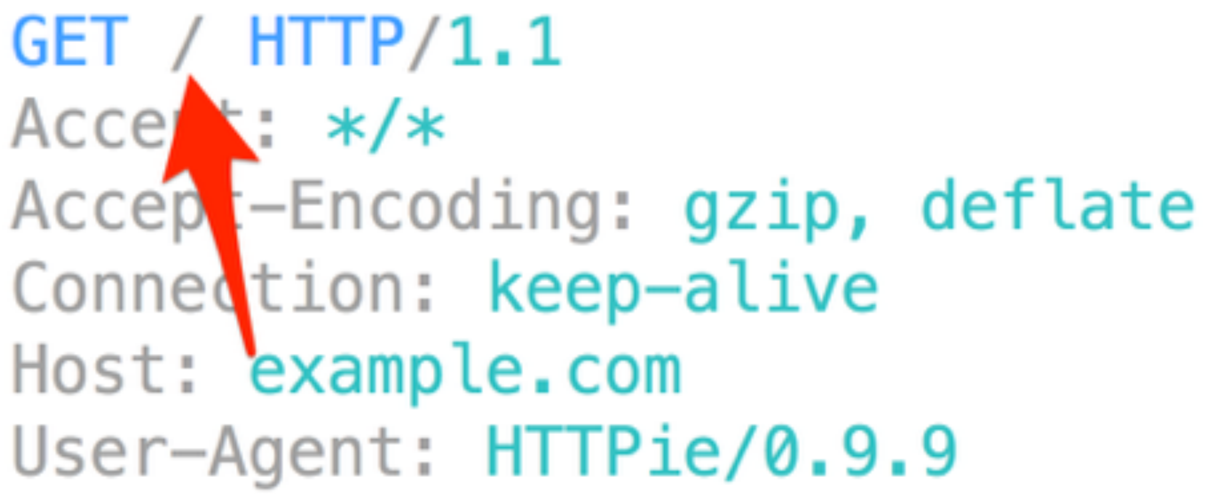
```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: example.com
User-Agent: HTTPie/0.9.9
```

HTTP version



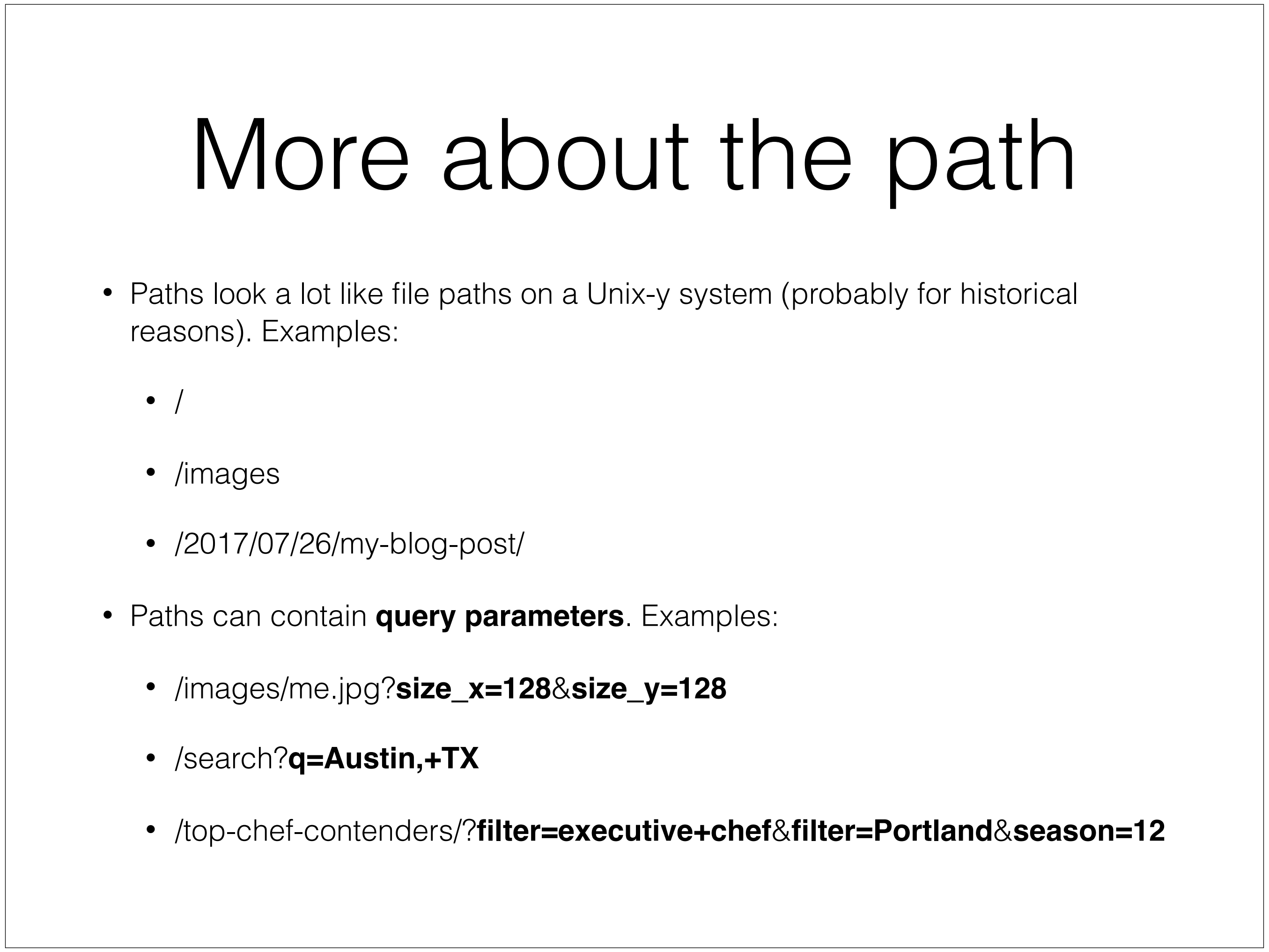
```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: example.com
User-Agent: HTTPie/0.9.9
```

Path



```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: example.com
User-Agent: HTTPie/0.9.9
```

We’ll talk a bit more about paths in a bit.

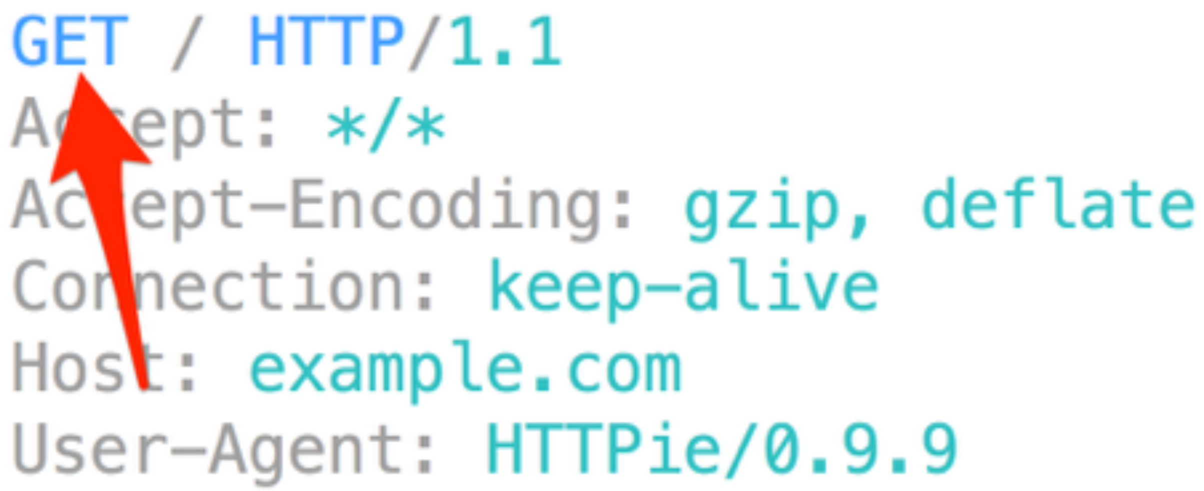


More about the path

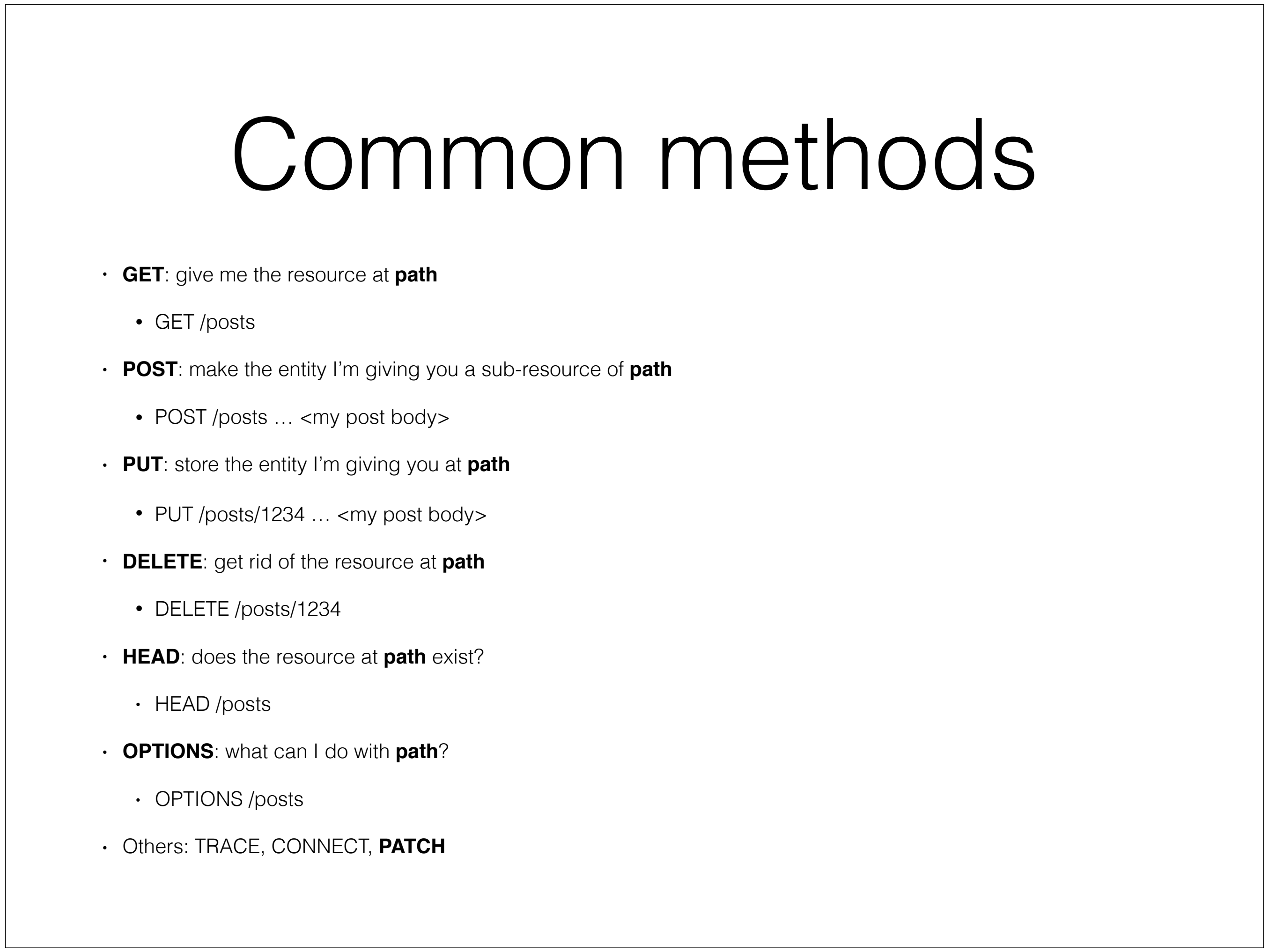
- Paths look a lot like file paths on a Unix-y system (probably for historical reasons). Examples:
  - /
  - /images
  - /2017/07/26/my-blog-post/
- Paths can contain **query parameters**. Examples:
  - /images/me.jpg?**size_x=128**&**size_y=128**
  - /search?**q=Austin,+TX**
  - /top-chef-contenders/?**filter=executive+chef**&**filter=Portland**&**season=12**

There are lots of “religious” battles here: trailing slashes, hyphens vs. camelCase vs. underscores, etc. Use whatever you like.

Method



```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: example.com
User-Agent: HTTPie/0.9.9
```

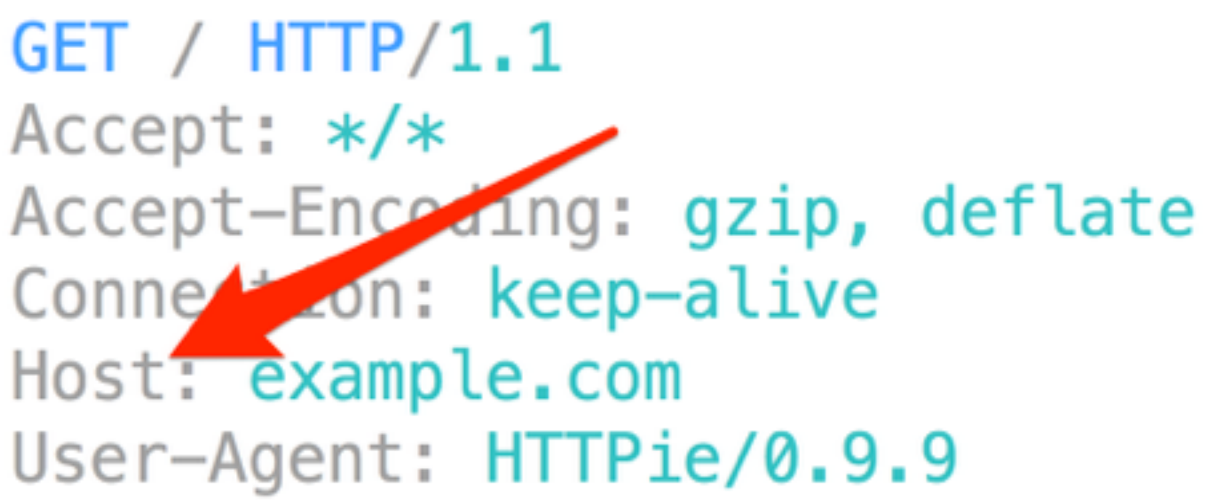


Common methods

- **GET**: give me the resource at **path**
  - GET /posts
- **POST**: make the entity I’m giving you a sub-resource of **path**
  - POST /posts … <my post body>
- **PUT**: store the entity I’m giving you at **path**
  - PUT /posts/1234 … <my post body>
- **DELETE**: get rid of the resource at **path**
  - DELETE /posts/1234
- **HEAD**: does the resource at **path** exist?
  - HEAD /posts
- **OPTIONS**: what can I do with **path**?
  - OPTIONS /posts
- Others: TRACE, CONNECT, **PATCH**

HEAD = GET without a body
OPTIONS = the viable methods on **path**. So this would maybe give us a list: GET, POST, HEAD.
PATCH is an extension method some folks have implemented.

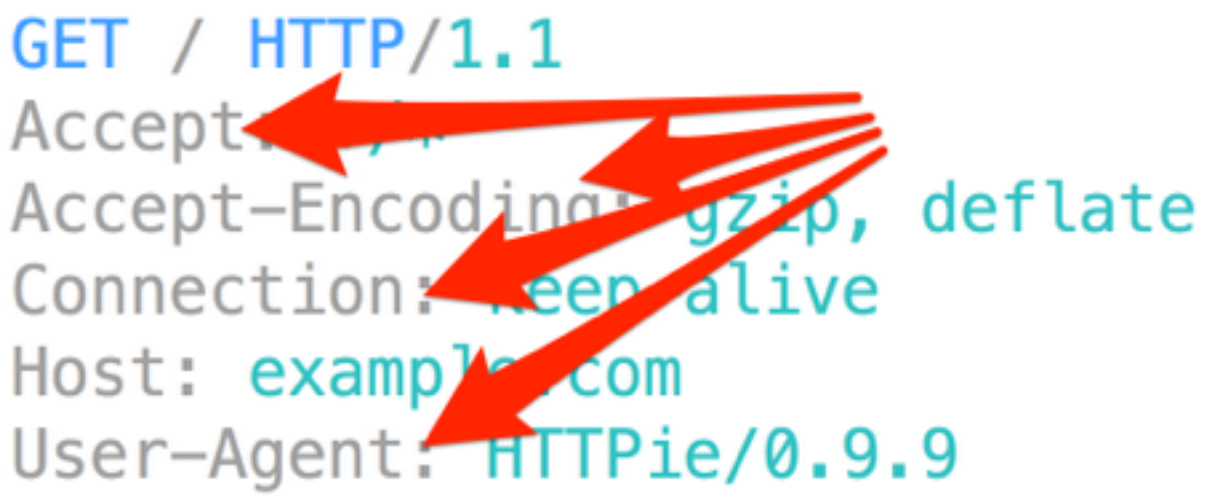
Host header



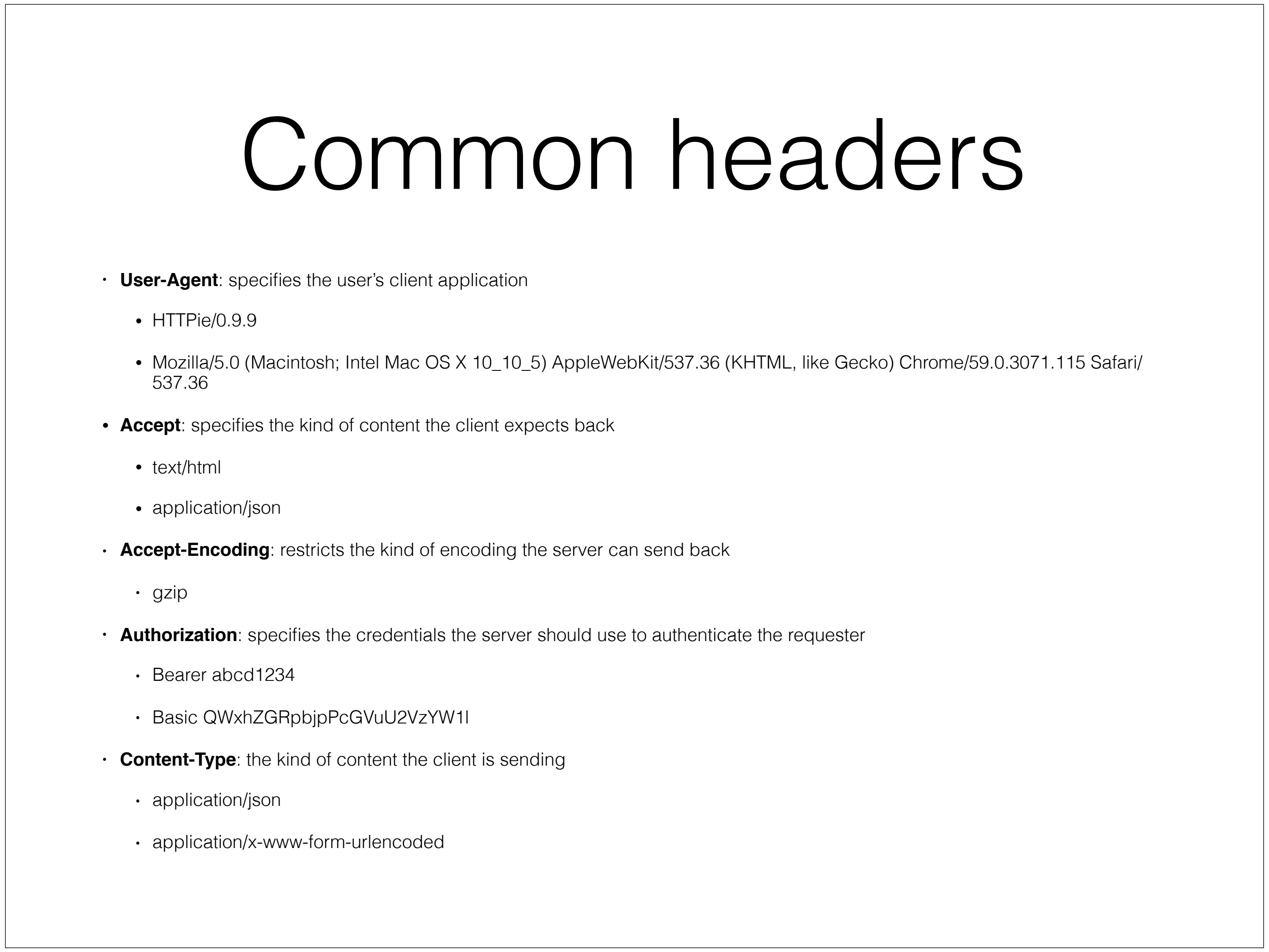
```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: example.com
User-Agent: HTTPie/0.9.9
```

This is required when you’re using HTTP 1.1

Other headers



```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Host: example.com
User-Agent: HTTPie/0.9.9
```



Common headers

- **User-Agent**: specifies the user's client application
  - HTTPie/0.9.9
  - Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
- **Accept**: specifies the kind of content the client expects back
  - text/html
  - application/json
- **Accept-Encoding**: restricts the kind of encoding the server can send back
  - gzip
- **Authorization**: specifies the credentials the server should use to authenticate the requester
  - Bearer abcd1234
  - Basic QWxhZGRpbjpPcGVuU2VzYW1l
- **Content-Type**: the kind of content the client is sending
  - application/json
  - application/x-www-form-urlencoded

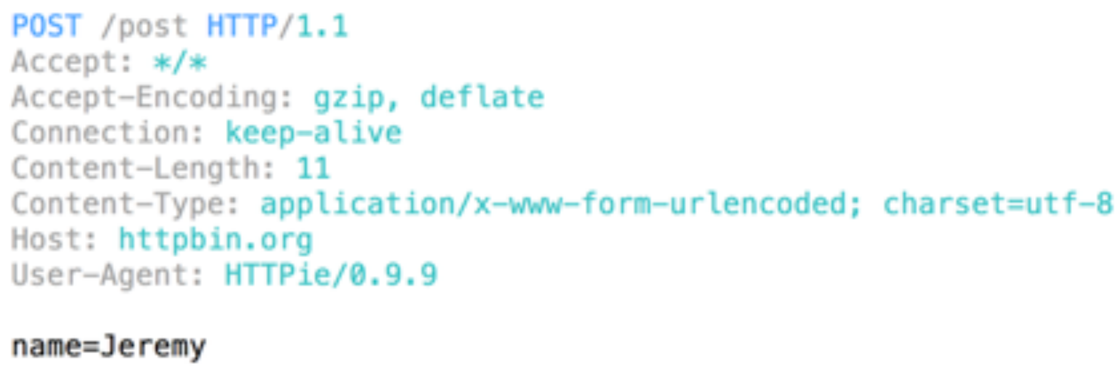
Authorization is poorly named — this is authN, not authZ

Exploring requests

httpbin.org

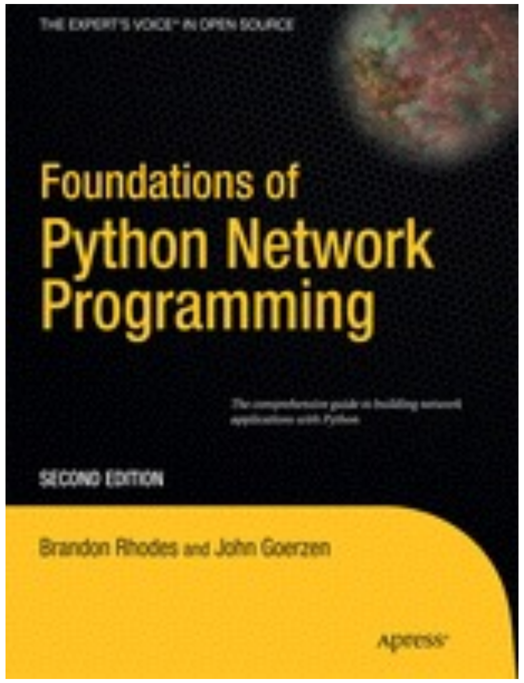
Request bodies

```
http --verbose -f httpbin.org/post name=Jeremy
```



```
POST /post HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Length: 11
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Host: httpbin.org
User-Agent: HTTPie/0.9.9

name=Jeremy
```

External Resources


THE EXPERT'S VOICE® IN OPEN SOURCE
Foundations of
Python Network
Programming
The comprehensive guide to building network applications with Python
SECOND EDITION
Brandon Rhodes and John Goerzen
Apress®

External Resources

- https://www.w3.org/Protocols/rfc2616/rfc2616.html
- https://developer.mozilla.org/en-US/docs/Web/HTTP

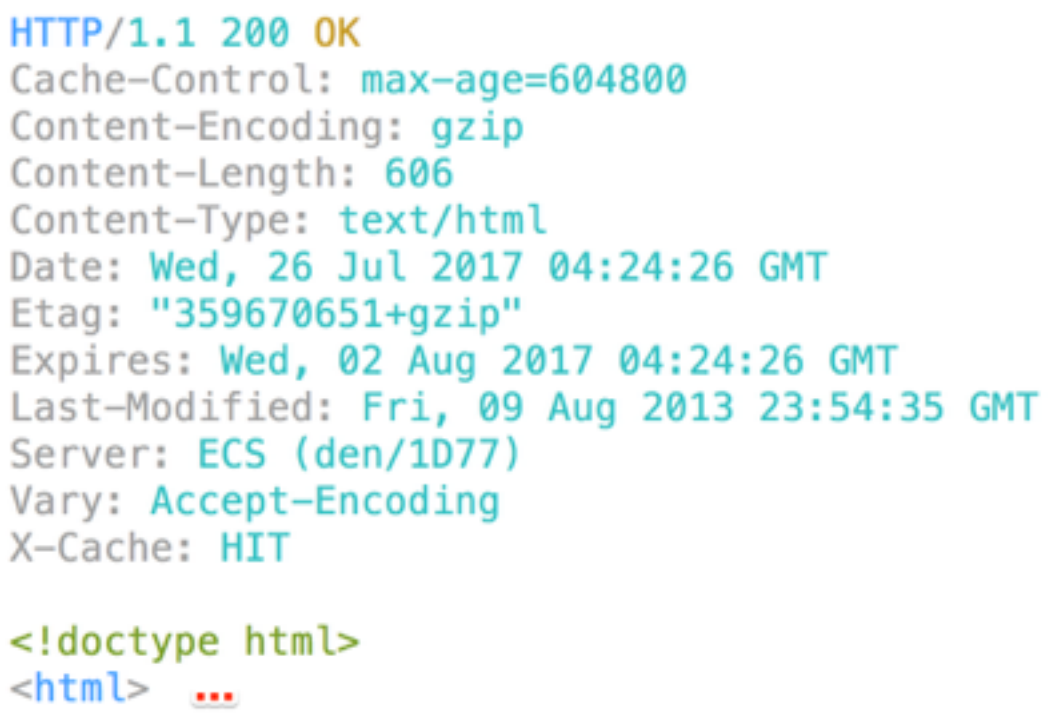
The Plan

- **HTTP:**
  - ~~Fundamentals~~
  - ~~Requests~~
  - **Responses**
- Django overview
- (Brief) installation overview
- Starting a Django app
- Writing our first view
- Working with requests

Example HTTP response

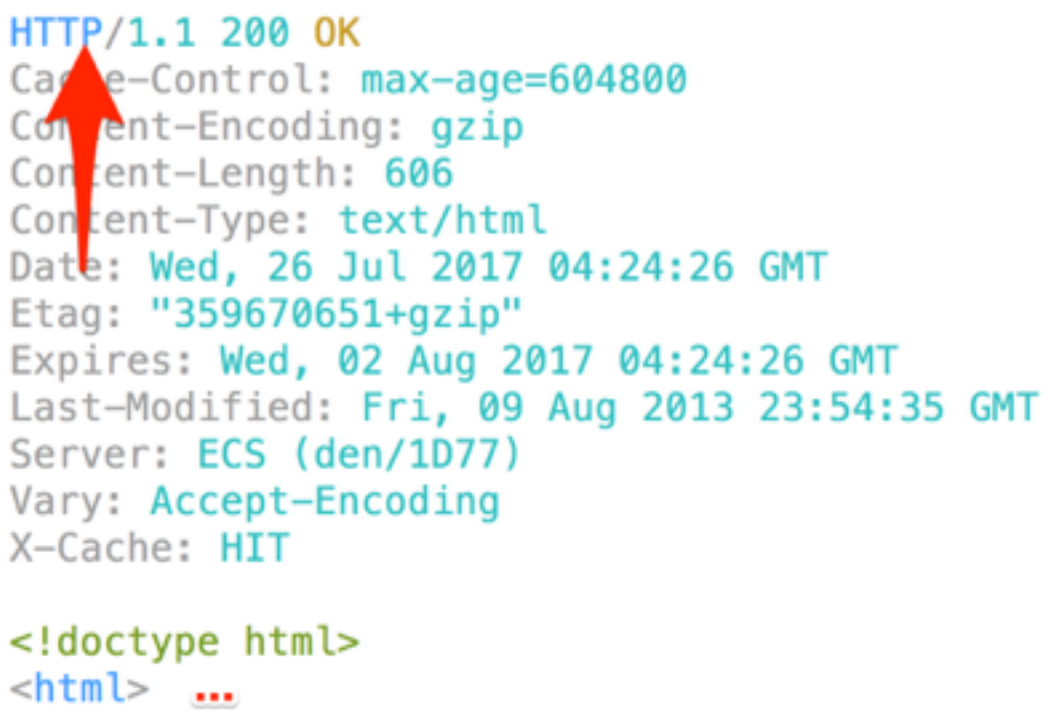
```
$ brew install httpie
...
$ http --verbose example.com
```

Example HTTP response



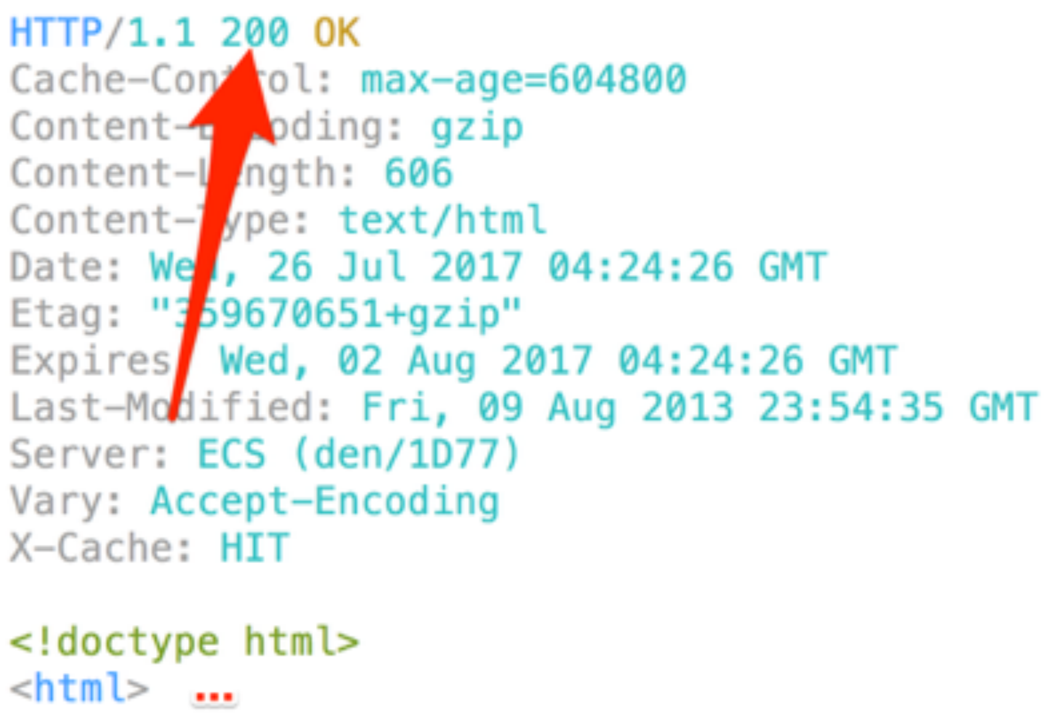
```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html>  ...
```

HTTP version



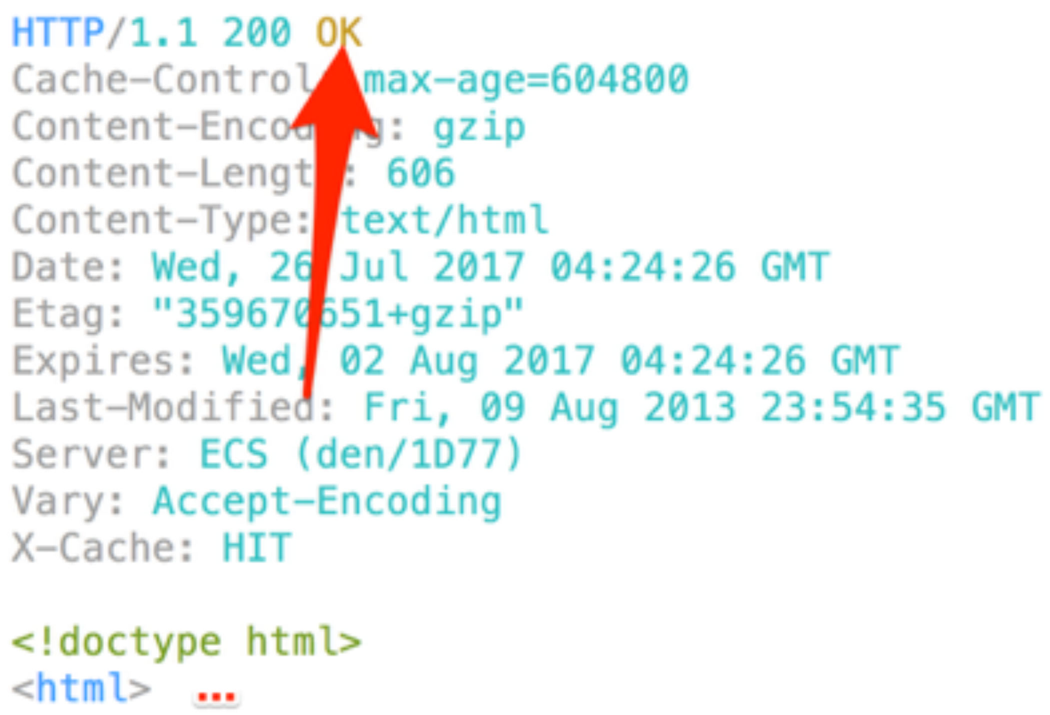
```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html>  ...
```

Status code

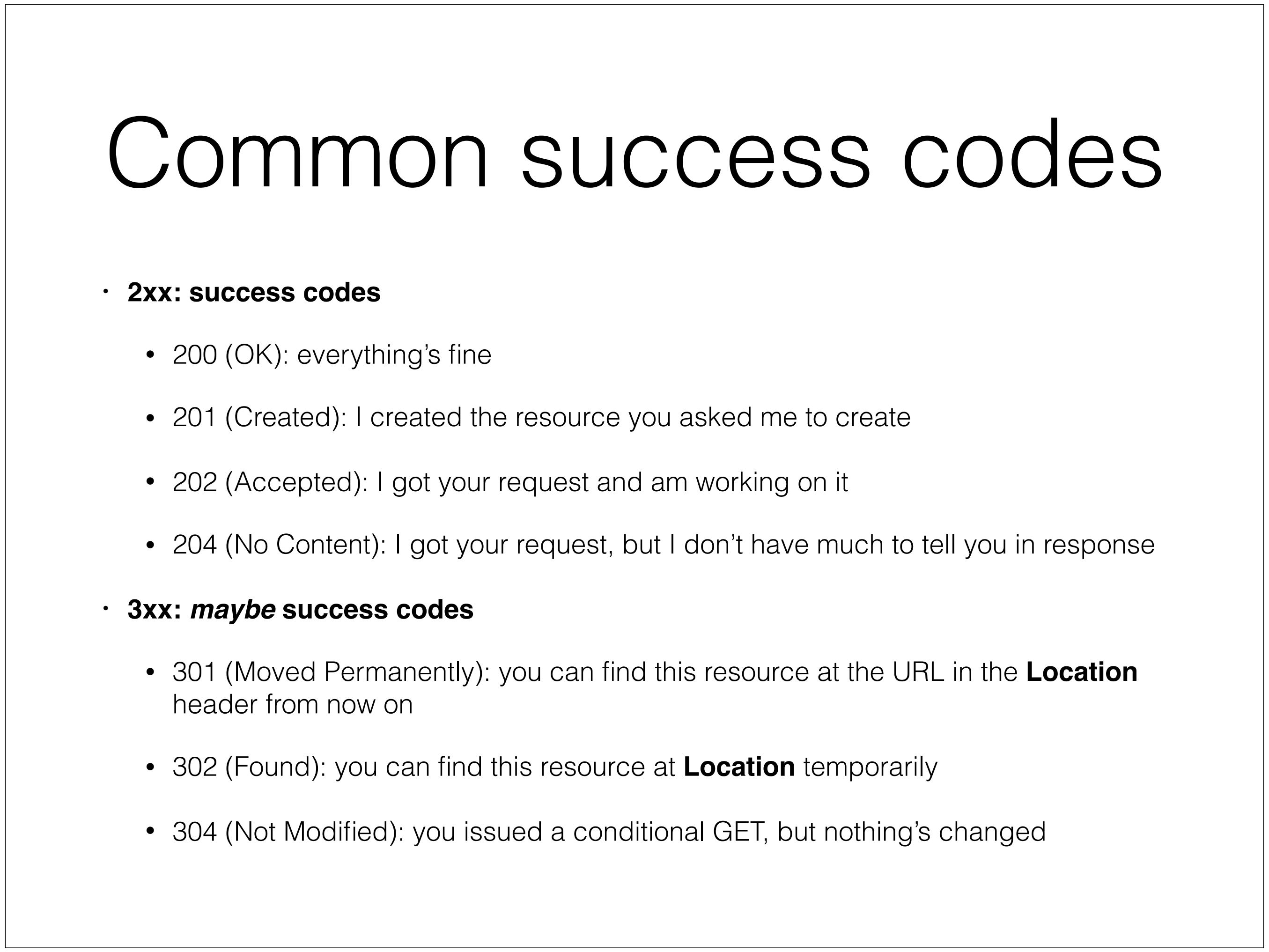


```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```

Reason



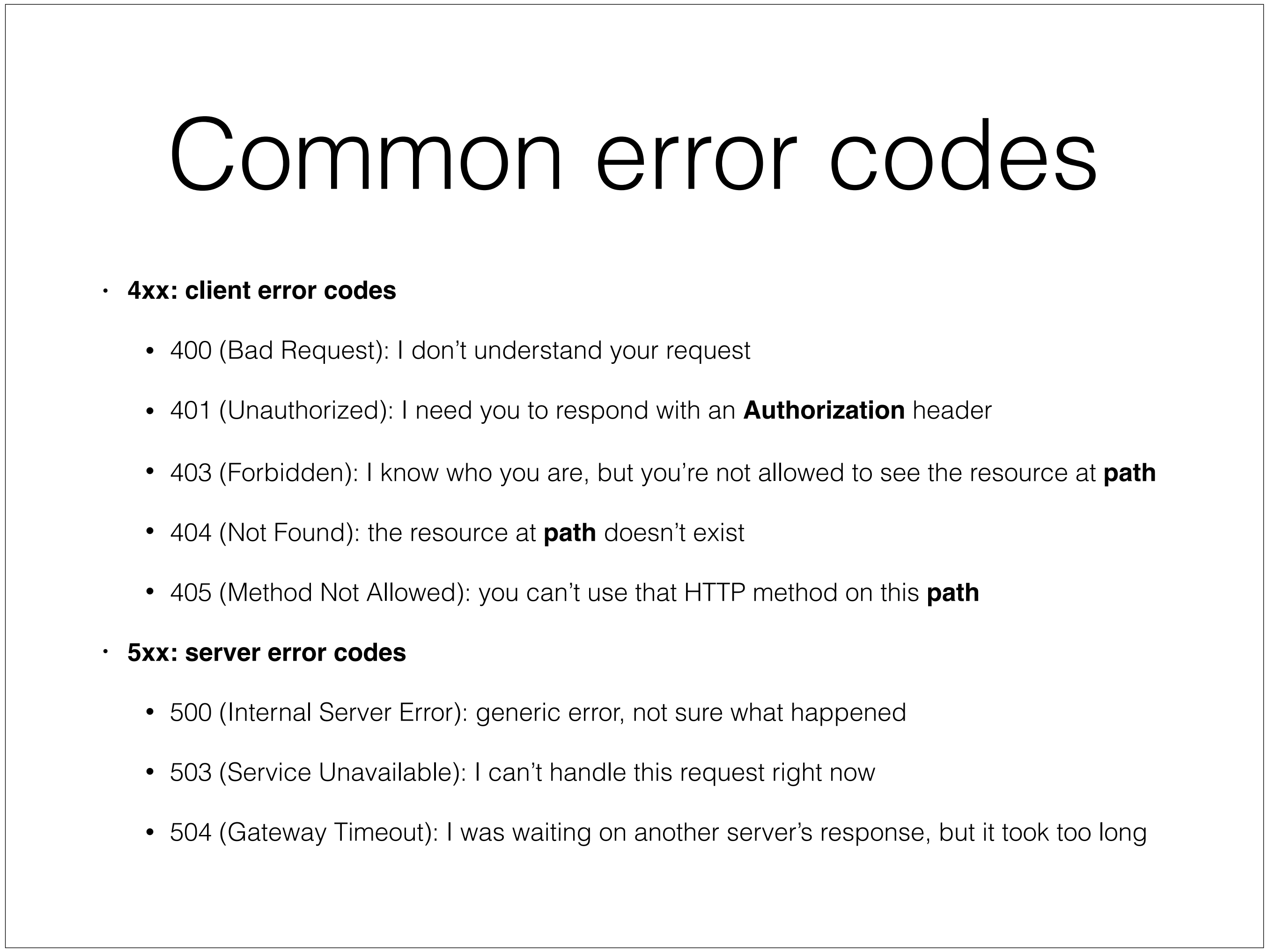
```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```



Common success codes

- **2xx: success codes**
  - 200 (OK): everything's fine
  - 201 (Created): I created the resource you asked me to create
  - 202 (Accepted): I got your request and am working on it
  - 204 (No Content): I got your request, but I don't have much to tell you in response
- **3xx: *maybe* success codes**
  - 301 (Moved Permanently): you can find this resource at the URL in the **Location** header from now on
  - 302 (Found): you can find this resource at **Location** temporarily
  - 304 (Not Modified): you issued a conditional GET, but nothing's changed

204: there's no extra data to tell you, but I might want to update what you think about it
conditional GET: when the client requests a resource but says "only give me a response if it's changed"

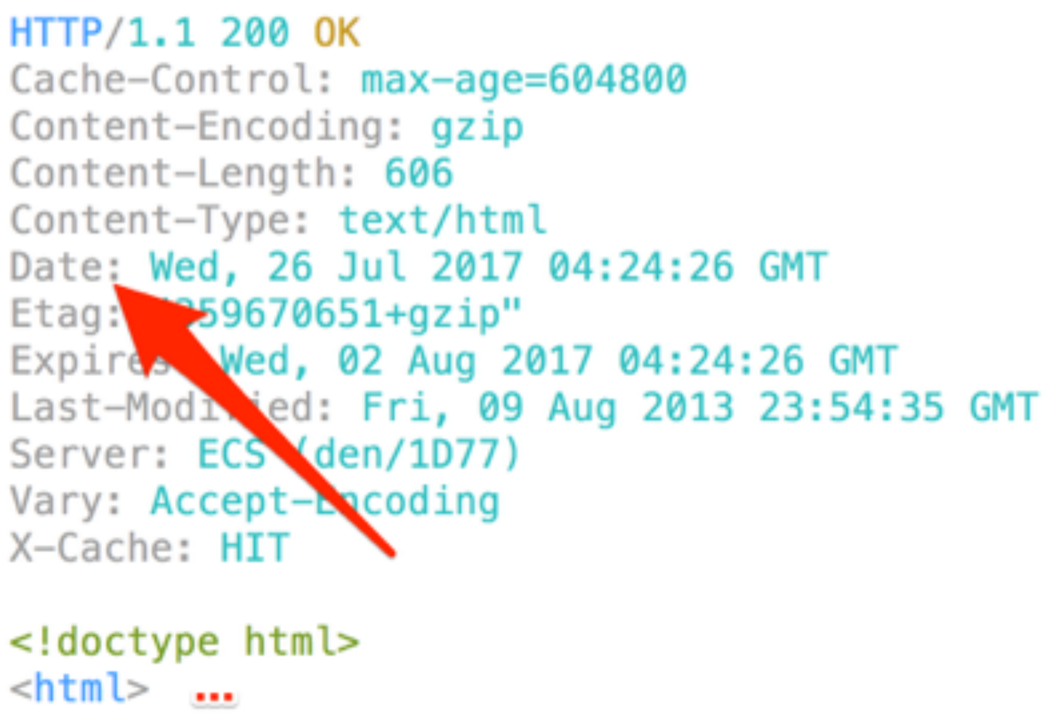


Common error codes

- **4xx: client error codes**
  - 400 (Bad Request): I don’t understand your request
  - 401 (Unauthorized): I need you to respond with an **Authorization** header
  - 403 (Forbidden): I know who you are, but you’re not allowed to see the resource at **path**
  - 404 (Not Found): the resource at **path** doesn’t exist
  - 405 (Method Not Allowed): you can’t use that HTTP method on this **path**
- **5xx: server error codes**
  - 500 (Internal Server Error): generic error, not sure what happened
  - 503 (Service Unavailable): I can’t handle this request right now
  - 504 (Gateway Timeout): I was waiting on another server’s response, but it took too long

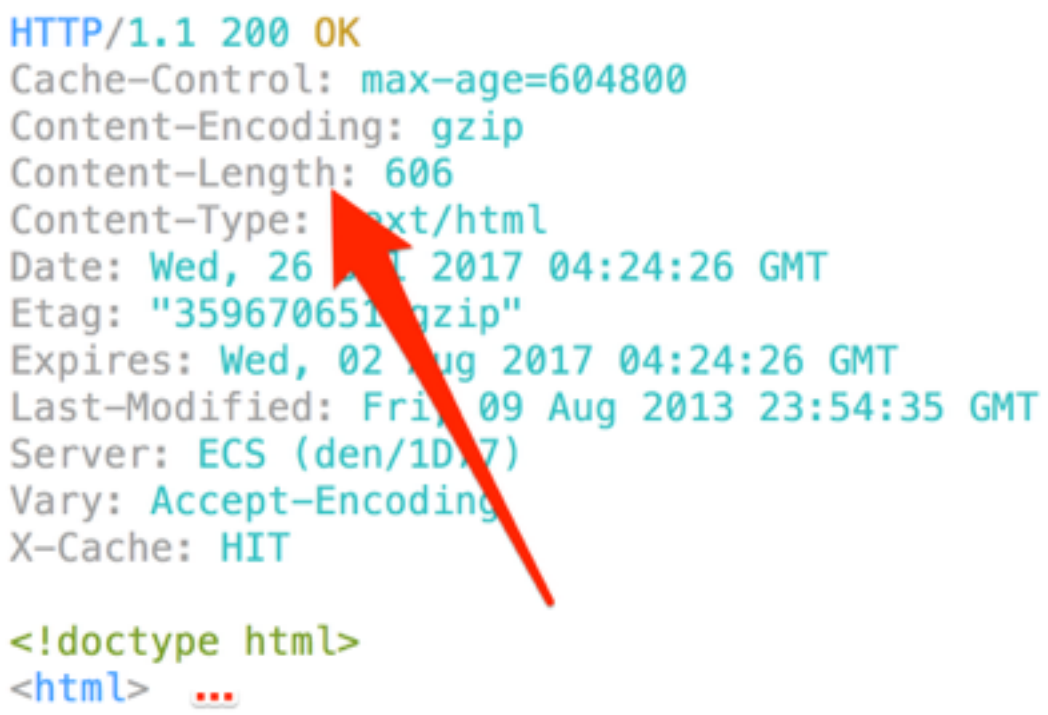
conditional GET: when the client requests a resource but says “only give me a response if it’s changed”

Date



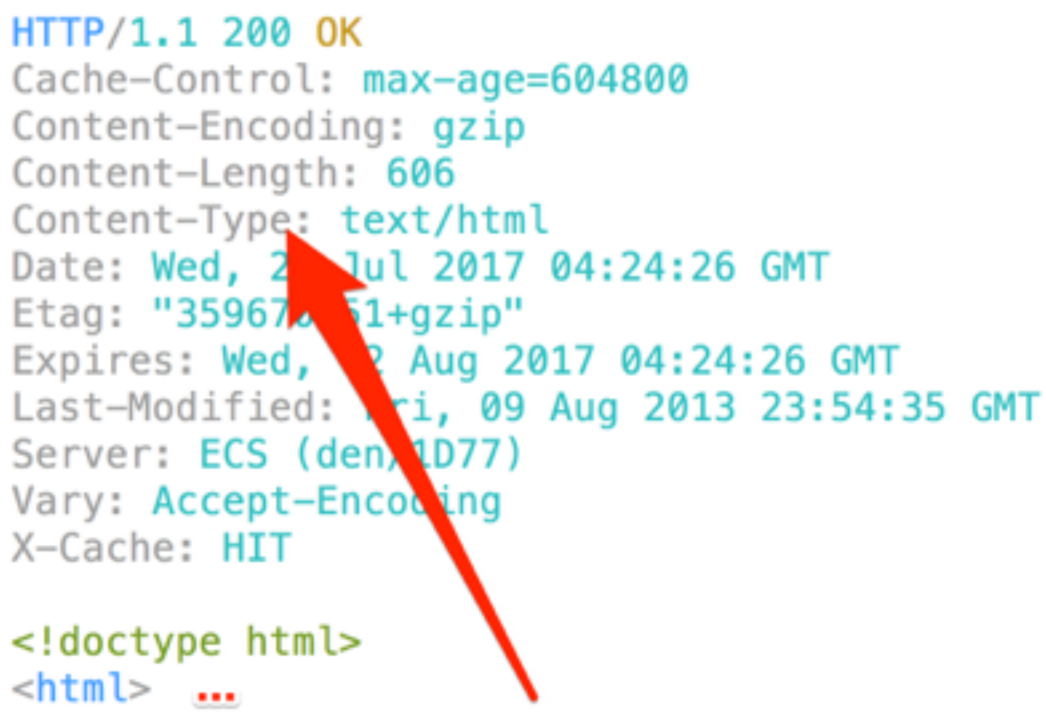
```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```

Content-Length



```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D[illegible]7)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```

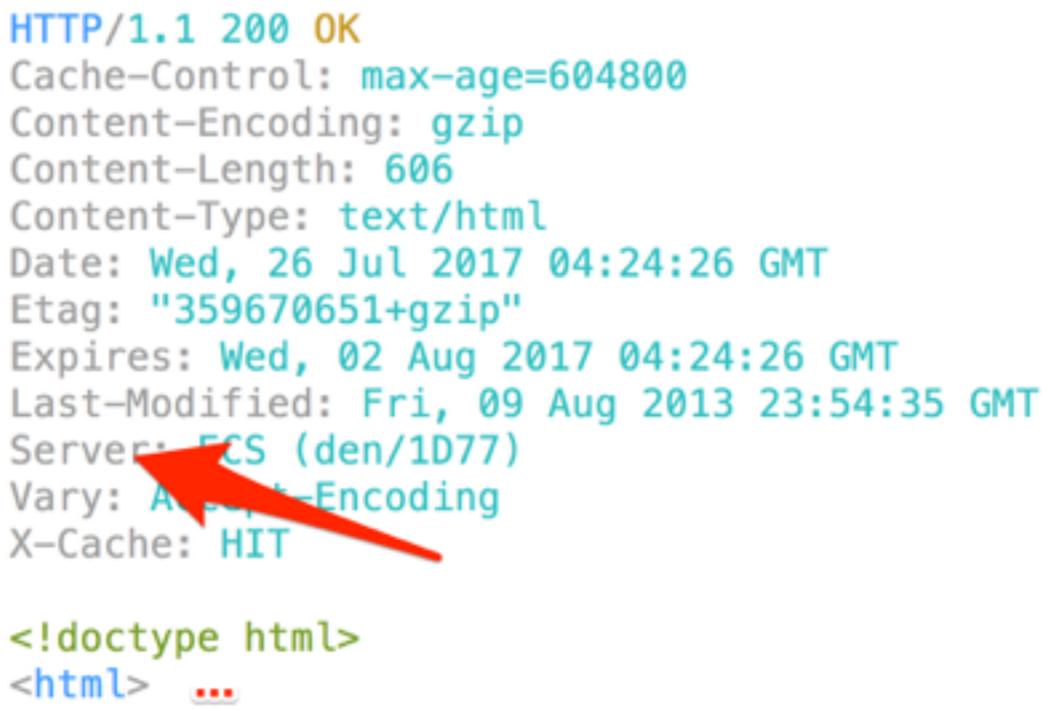
Content-Type



```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```

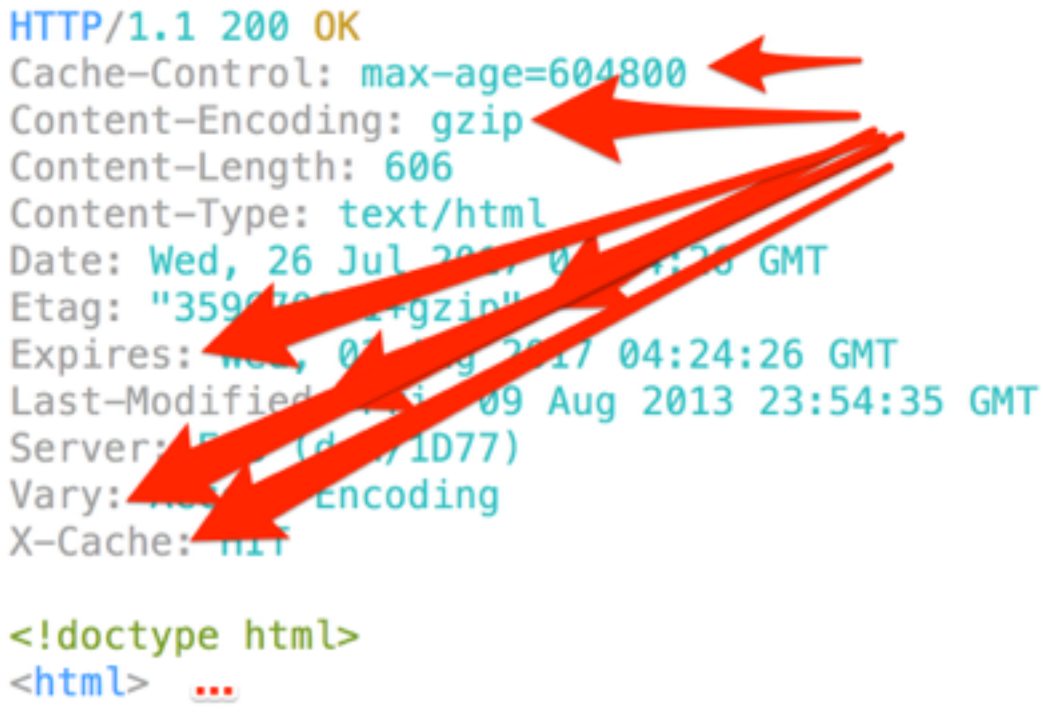
This should match the client’s **Accepted** header.

Server



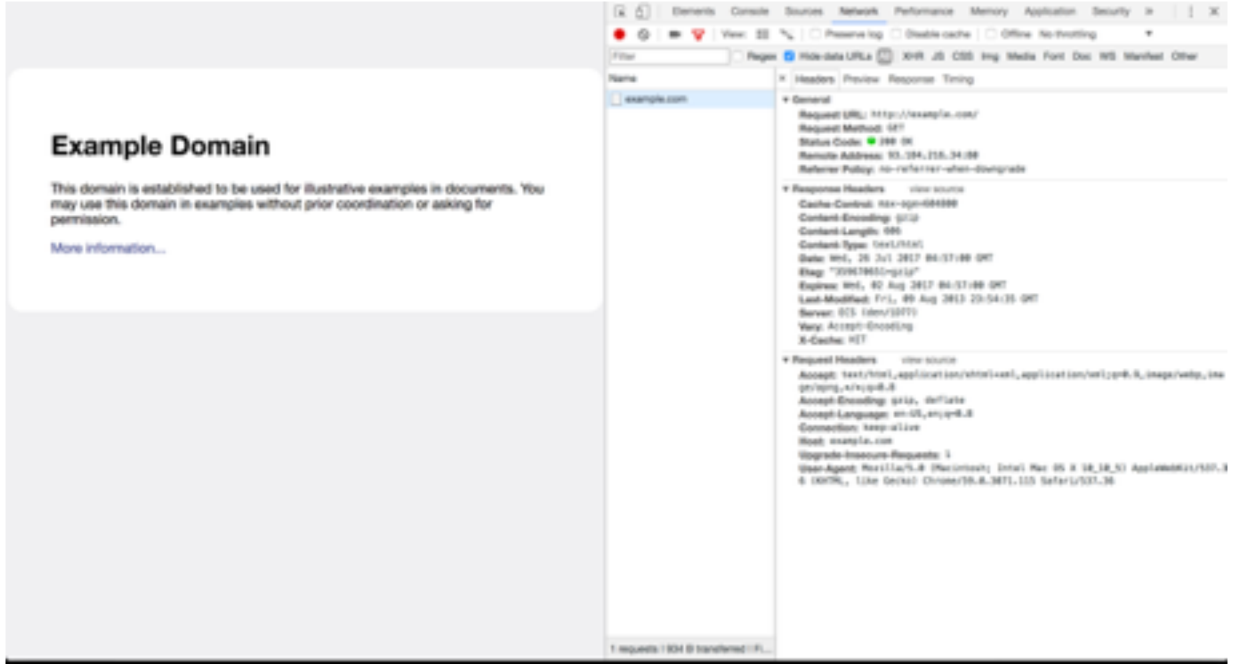
```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (den/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```

Other headers



```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Encoding: gzip
Content-Length: 606
Content-Type: text/html
Date: Wed, 26 Jul 2017 04:24:26 GMT
Etag: "359670651+gzip"
Expires: Wed, 02 Aug 2017 04:24:26 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (dca/1D77)
Vary: Accept-Encoding
X-Cache: HIT

<!doctype html>
<html> ...
```

Exploring responses



Example Domain

This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.

More information...

```
▼ General
    Request URL: http://example.com/
    Request Method: GET
    Status Code: ● 200 OK
    Remote Address: 93.184.216.34:80
    Referrer Policy: no-referrer-when-downgrade
▼ Response Headers
    Cache-Control: max-age=604800
    Content-Encoding: gzip
    Content-Length: 606
    Content-Type: text/html
    Date: Wed, 26 Jul 2017 04:57:00 GMT
    Etag: "359670651+gzip"
    Expires: Wed, 02 Aug 2017 04:57:00 GMT
    Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
    Server: ECS (den/1D77)
    Vary: Accept-Encoding
    X-Cache: HIT
▼ Request Headers
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,ima
    ge/apng,*/*;q=0.8
    Accept-Encoding: gzip, deflate
    Accept-Language: en-US,en;q=0.8
    Connection: keep-alive
    Host: example.com
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.3
    6 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
```

Just use your Chrome inspector tools, in particular the network tab!

The Plan

- ~~HTTP:~~
  - ~~Fundamentals~~
  - ~~Requests~~
  - ~~Responses~~
- **Django overview**
- (Brief) installation overview
- Starting a Django app
- Writing our first view
- Working with requests

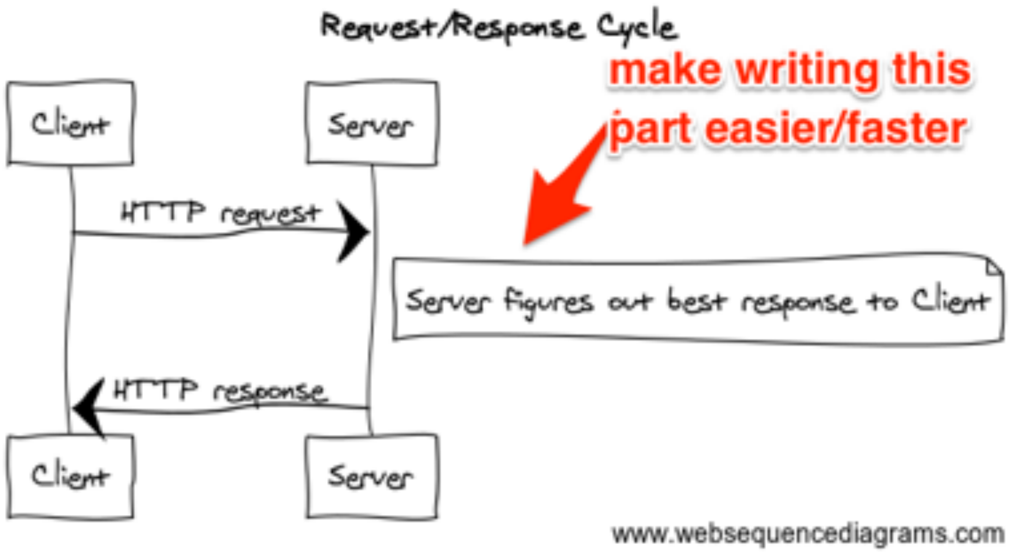
Django overview

- Born at *Lawrence Journal-World* in 2005
    - “The web framework for perfectionists with deadlines.”
- Pure Python
    - Latest version supports 2.7, 3.4-3.6

Notable features

- Easy-to-use Object Relational Mapper (ORM)
- Out-of-the-box web-based admin application
- Pluggable design allows reuse of your (or others') code
- Built-in REPL for debugging, admin tasks, etc.

Django’s role

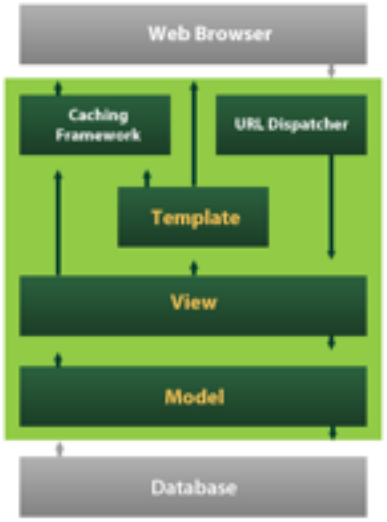


make writing this part easier/faster

django

5. Templates typically return HTML pages. The Django template language offers HTML authors a simple-to-learn syntax while providing all the power needed for presentation logic.

4. After performing any requested tasks, the view returns an HTTP response object (usually after passing the data through a template) to the web browser. Optionally, the view can save a version of the HTTP response object in the caching system for a specified length of time.

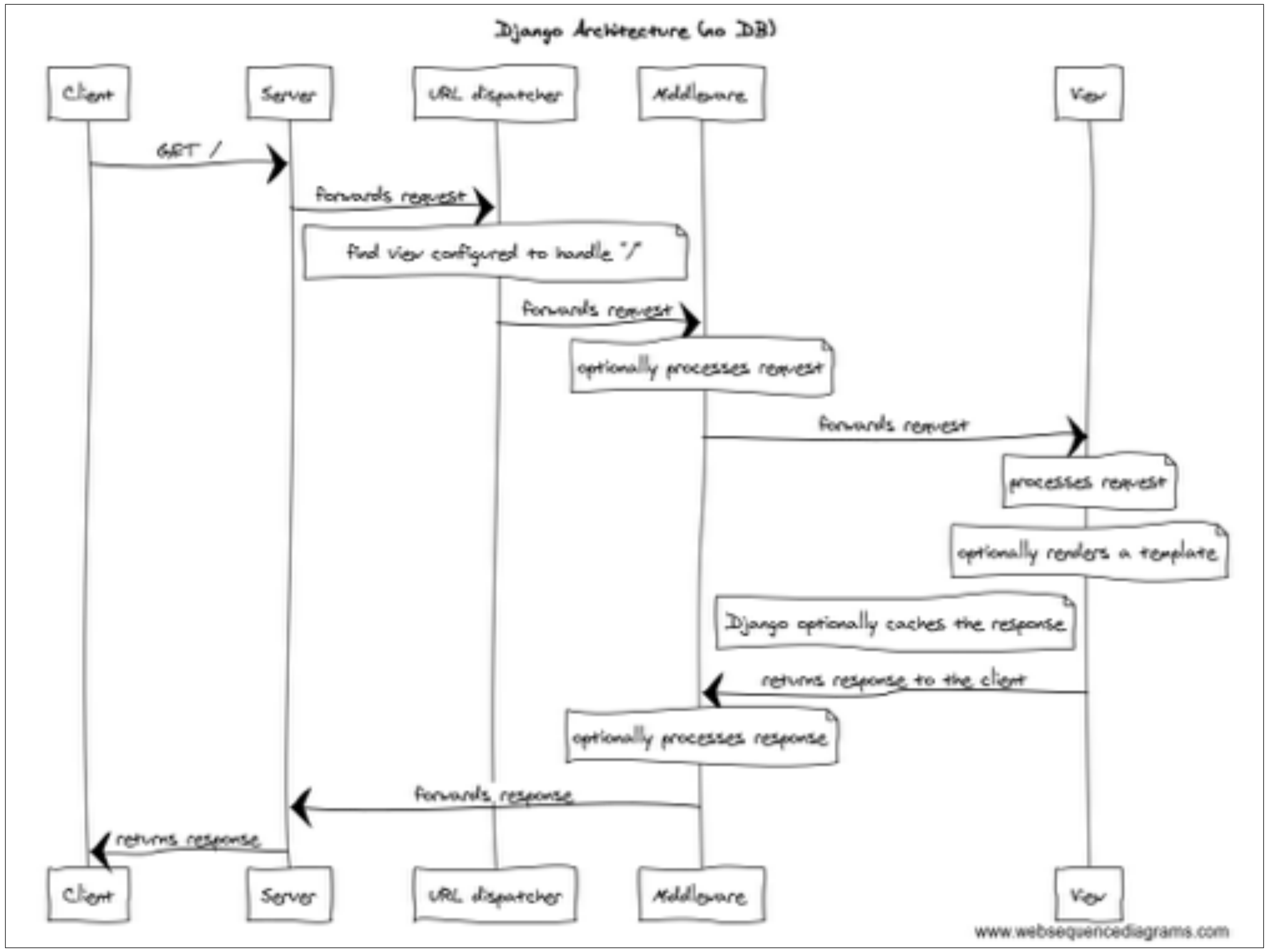


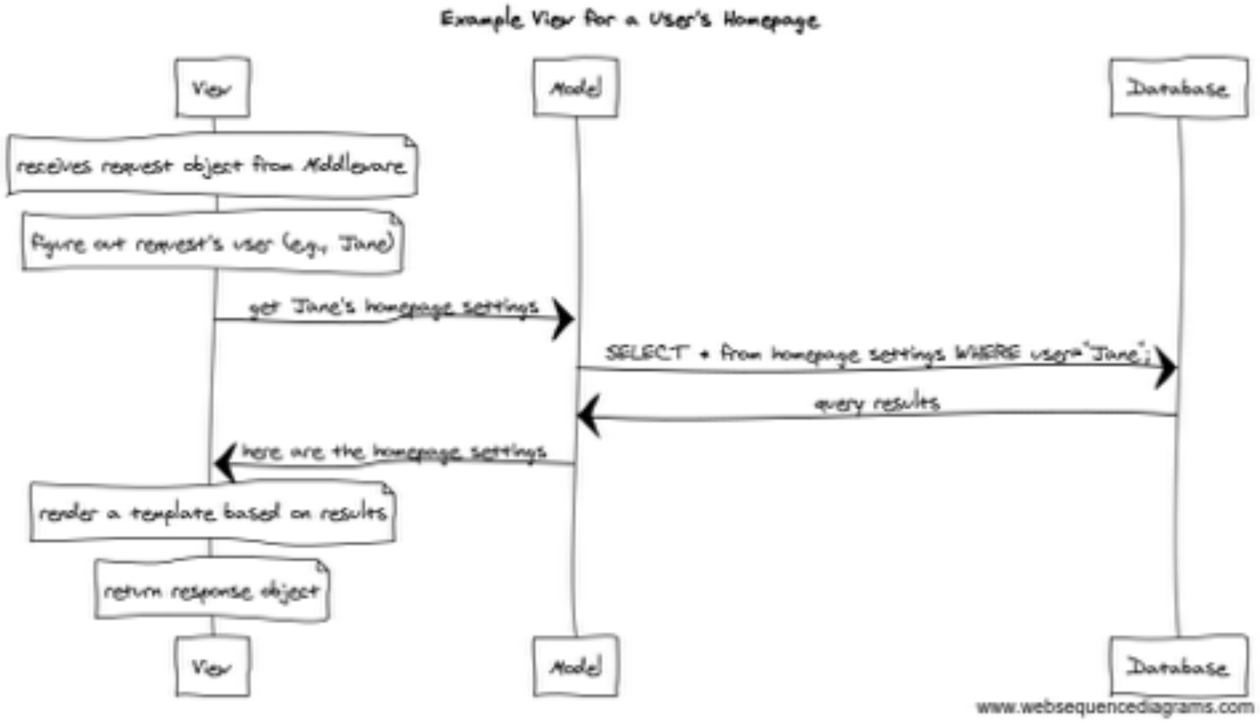
1. The URL dispatcher (urls.py) maps the requested URL to a view function and calls it. If caching is enabled, the view function can check to see if a cached version of the page exists and bypass all further steps, returning the cached version, instead. Note that this page-level caching is only one available caching option in Django. You can cache more granularly, as well.

2. The view function (usually in views.py) performs the requested action, which typically involves reading or writing to the database. It may include other tasks, as well.

3. The model (usually in models.py) defines the data in Python and interacts with it. Although typically contained in a relational database (MySQL, PostgreSQL, SQLite, etc.), other data storage mechanisims are possible as well (XML, text files, LDAP, etc.).

credit: Jeff Croft


Django Architecture (no DB)
Client
Server
URL dispatcher
Middleware
View
GET /
forwards request
find view configured to handle "/"
forwards request
optionally processes request
forwards request
processes request
optionally renders a template
Django optionally caches the response
returns response to the client
optionally processes response
forwards response
returns response
Client
Server
URL dispatcher
Middleware
View
www.websequencediagrams.com


Example View for a User's Homepage
View
Model
Database
receives request object from Middleware
figure out request's user (e.g., Jane)
get Jane's homepage settings
SELECT * from homepage settings WHERE user="Jane";
query results
here are the homepage settings
render a template based on results
return response object
View
Model
Database
www.websequencediagrams.com

The Plan

- ~~HTTP:~~
  - ~~Fundamentals~~
  - ~~Requests~~
  - ~~Responses~~
- ~~Django overview~~
- **(Brief) installation overview**
- Starting a Django app
- Writing our first view
- Working with requests

Installation

bit.ly/2v0VIje

VERY brief installation instructions

```
brew install python3 git
sudo pip3 install virtualenvwrapper

# go create a GitHub repo

mkdir -p ~/Repos && cd ~/Repos
git clone <your GH repo URL>

# mine is called awpug_django_2017
cd awpug_django_2017
touch requirements.txt
echo Django==1.11.3 > requirements.txt
mkvirtualenv -p python3 django-tutorial-env
pip install -Ur requirements.txt
django-admin.py startproject config .
```

More details at `bit.ly/2v0VIje`

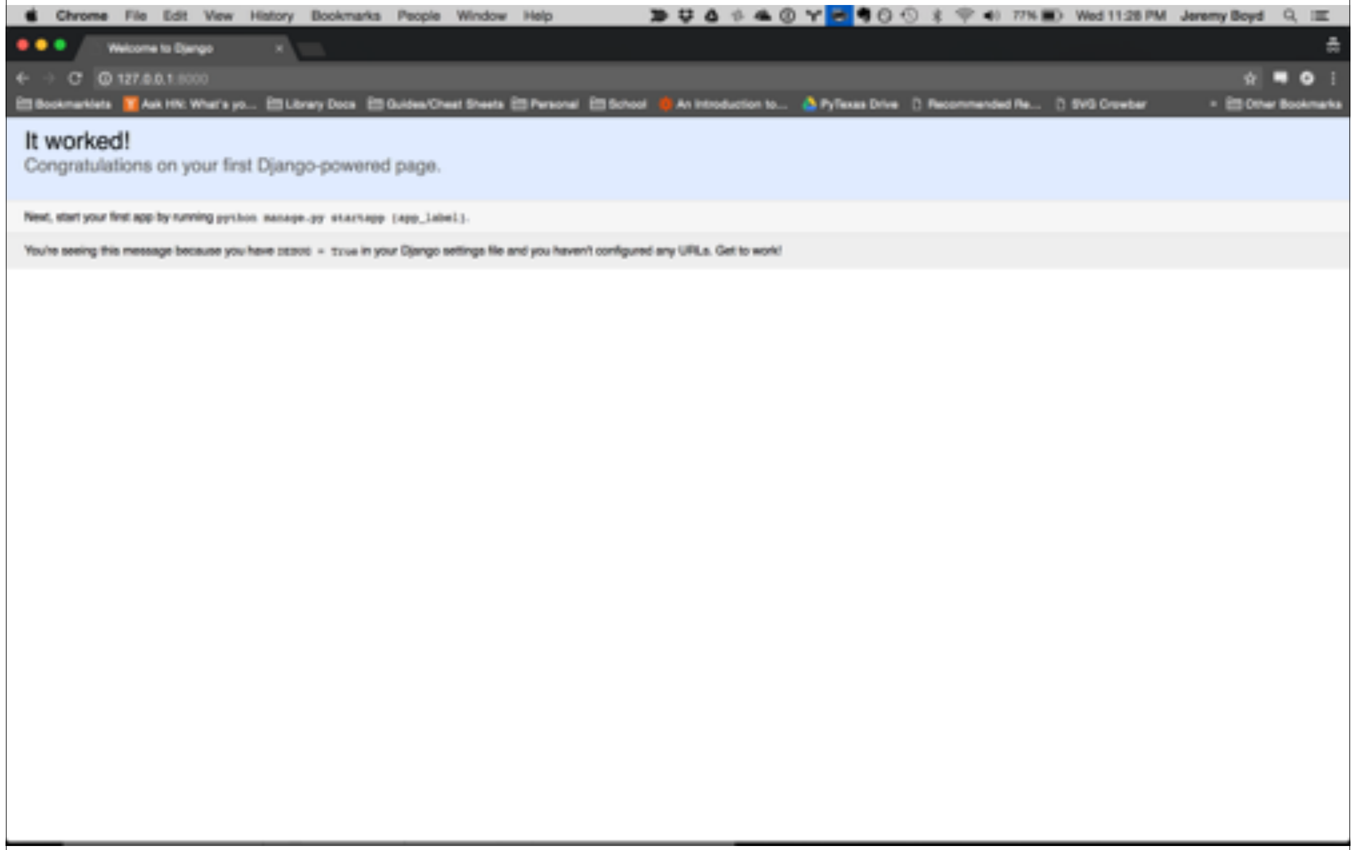
Results

```
(django-tutorial-env) [jboyd@erasmas:awpug-django-2017 (master +)]$ ls -al
total 32
drwxr-xr-x   9 jboyd  staff   306 Jul 26 22:40 .
drwxr-xr-x  28 jboyd  staff   952 Jul 26 22:38 ..
drwxr-xr-x  12 jboyd  staff   408 Jul 26 22:40 .git
-rw-r--r--   1 jboyd  staff  1157 Jul 26 22:38 .gitignore
drwxr-xr-x   9 jboyd  staff   306 Jul 26 22:40 .idea
-rw-r--r--   1 jboyd  staff    19 Jul 26 22:38 README.md
drwxr-xr-x   6 jboyd  staff   204 Jul 26 22:40 config
-rwxr-xr-x   1 jboyd  staff   804 Jul 26 22:40 manage.py
-rw-r--r--   1 jboyd  staff    15 Jul 26 22:39 requirements.txt
```

Run server for the first time

```
python manage.py runserver
```

http://127.0.0.1:8000/



It worked!

Congratulations on your first Django-powered page.

Next, start your first app by running `python manage.py startapp [app_label]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

Make your first commit

```
git add config/ manage.py requirements.txt
git commit –m "Initial project structure”
git push origin HEAD
```

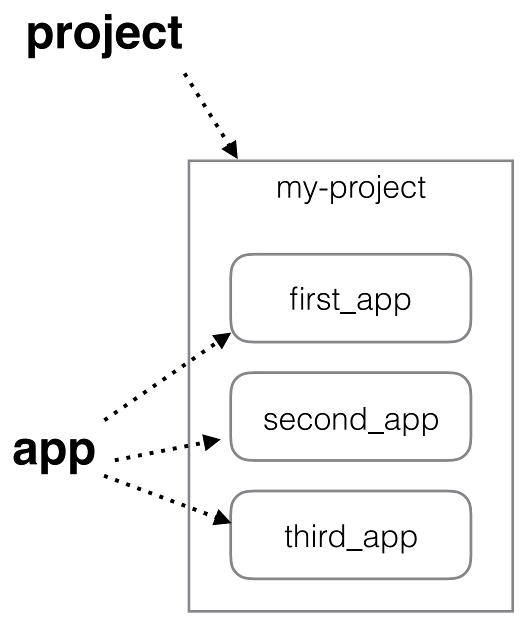
https://github.com/boydjj/awpug-django-2017/releases/tag/v0.1.0

The Plan

- ~~HTTP:~~
  - ~~Fundamentals~~
  - ~~Requests~~
  - ~~Responses~~
- ~~Django overview~~
- ~~(Brief) installation overview~~
- **Starting a Django app**
- Writing our first view
- Working with requests

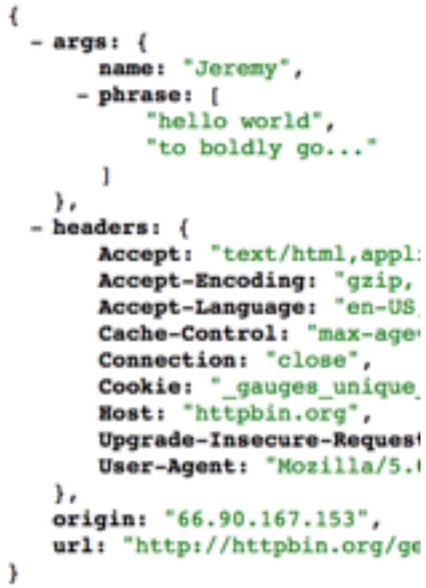
Anatomy of a Django project

- **project**: your whole website, e.g. **instagram.com**
- **app**: a piece of functionality for your site, e.g. **users**
  - each app is a Python package
  - must be named accordingly



Let’s make an app!

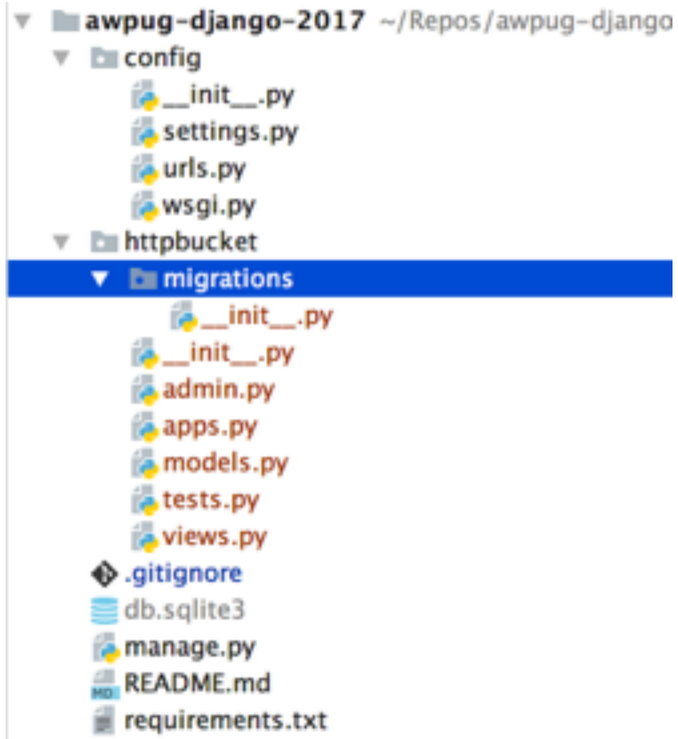
- Let’s build a clone of httpbin.org
  - specifically, the functionality from **/get**
- MVP:
  - we don’t need JSON output
  - can be stateless — we don’t need user info, history, etc.



```
{
  - args: {
        name: "Jeremy",
      - phrase: [
            "hello world",
            "to boldly go..."
        ]
    },
  - headers: {
        Accept: "text/html,appl
        Accept-Encoding: "gzip,
        Accept-Language: "en-US
        Cache-Control: "max-age
        Connection: "close",
        Cookie: "_gauges_unique_
        Host: "httpbin.org",
        Upgrade-Insecure-Reques
        User-Agent: "Mozilla/5.
    },
    origin: "66.90.167.153",
    url: "http://httpbin.org/ge
}
```

Start a new app

```
python manage.py startapp httpbucket
```



```
▼ awpug-django-2017 ~/Repos/awpug-django
  ▼ config
      __init__.py
      settings.py
      urls.py
      wsgi.py
  ▼ httpbucket
    ▼ migrations
        __init__.py
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      views.py
    .gitignore
    db.sqlite3
    manage.py
    README.md
    requirements.txt
```

We don’t need all of this

```
rm admin.py models.py tests.py
rm -rf migrations/
```

Commit your changes

https://github.com/boydjj/awpug-django-2017/releases/tag/v0.2.0

The Plan

- ~~HTTP:~~
    - ~~Fundamentals~~
    - ~~Requests~~
    - ~~Responses~~
- ~~Django overview~~
- ~~(Brief) installation overview~~
- ~~Starting a Django app~~
- **Writing our first view**
- Working with requests

Create a simple view

- A **view** is a **function** that takes in a **request** and returns a **response**

- Let’s write a really simple view in `httpbucket/views.py`

```
from django.http import HttpResponse


def hello_world(request):
    return HttpResponse("Obligatory greeting!")
```

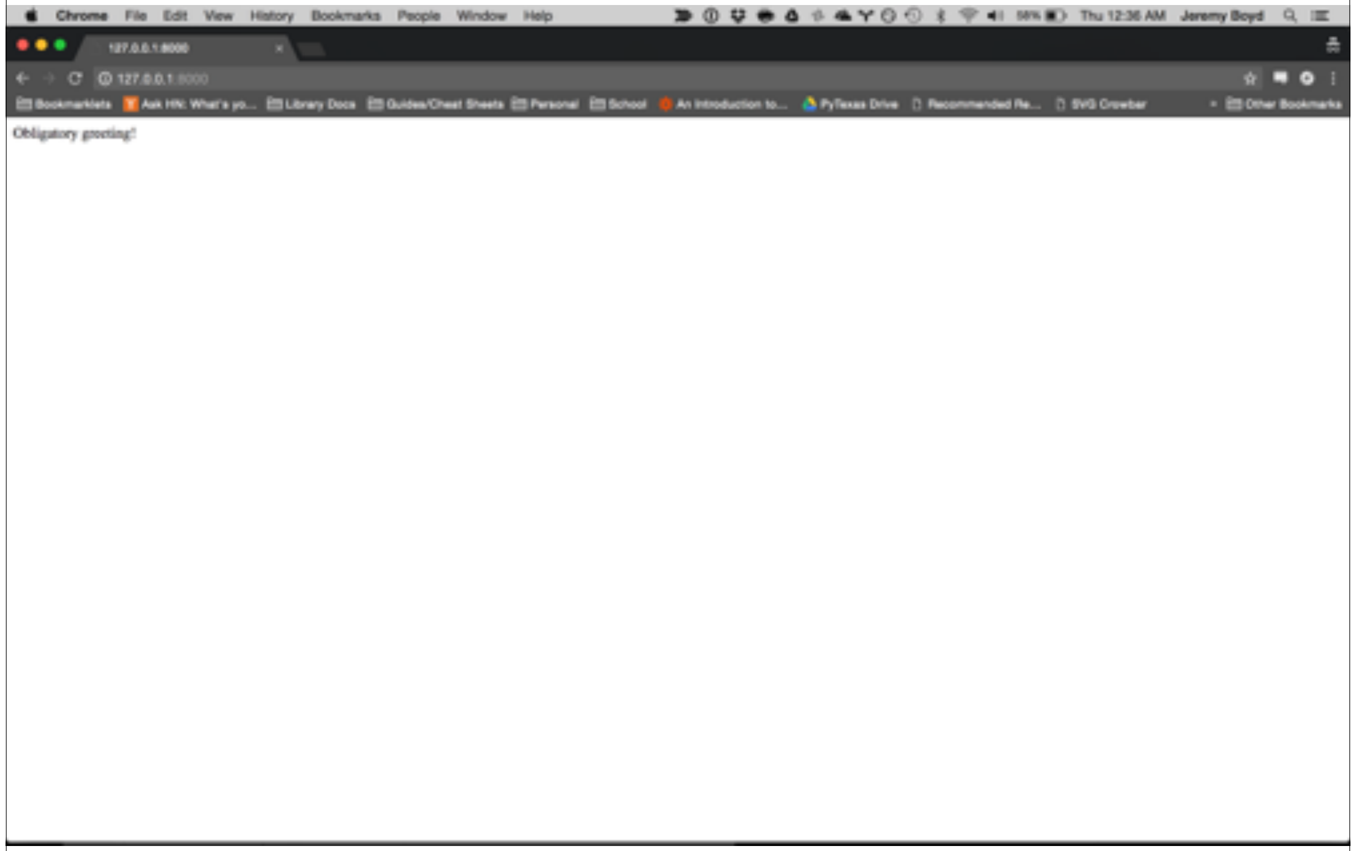
Tell the URL dispatcher about our new view

- Remember: the **URL dispatcher** needs to know how to **route** requests
- Let's hook our new view up to the **/** path in `config/urls.py`

```
from httpbucket import views as httpbucket_views

urlpatterns = [
    url(r'^$', httpbucket_views.hello_world),
    url(r'^admin/', admin.site.urls),
]
```

An aside on URL configs

- Q: In the previous slide we used the regular expression **^$**. This seems weird because we see the view's output when we go to **localhost:8000/**. Isn't that **/** required? Shouldn't the regex be **^/$**?
- A: No. By the time the **path** gets to the URL dispatcher, the *leading* **/** is actually stripped from the path. So if we consider the path **/**, and then we strip away the leading **/**, all we're left with is the empty string, which is what **^$** represents.
- Also worth noting: the the URL dispatcher is matching the config against the request's path *with all the query parameters stripped off*.



Obligatory greeting!

Commit your changes

https://github.com/boydjj/awpug-django-2017/releases/tag/v0.3.0

The Plan

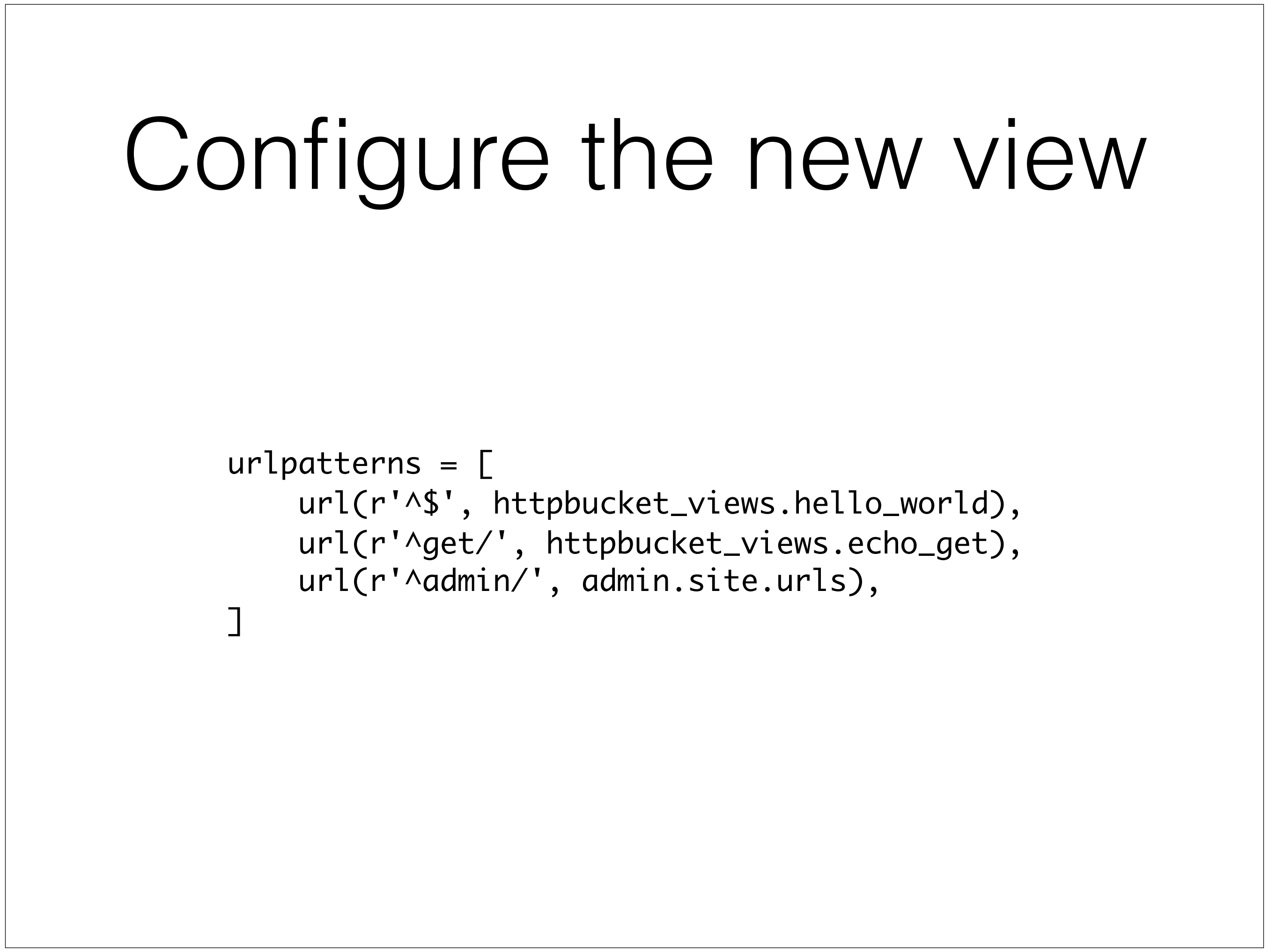
- ~~HTTP:~~
  - ~~Fundamentals~~
  - ~~Requests~~
  - ~~Responses~~
- ~~Django overview~~
- ~~(Brief) installation overview~~
- ~~Starting a Django app~~
- ~~Writing our first view~~
- **Working with requests**

Getting information out of a request

httpbucket/views.py

```
import json

...

def echo_get(request):
    response = {}

    args = {}
    for k, v in request.GET.items():
        args[k] = v
    response['args'] = args

    return HttpResponse(json.dumps(response))
```

We’re using **json.dumps** to convert the **response** dict into a string.

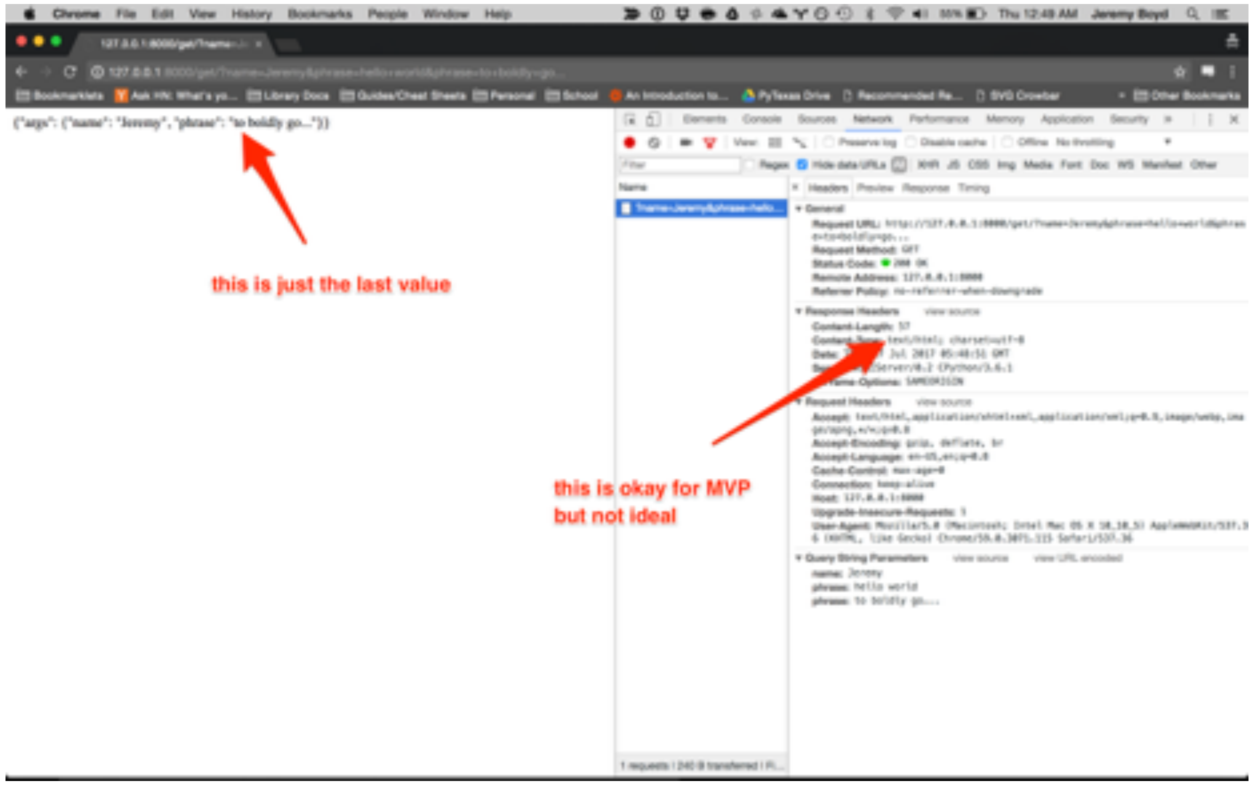


Configure the new view

```
urlpatterns = [
    url(r'^$', httpbucket_views.hello_world),
    url(r'^get/', httpbucket_views.echo_get),
    url(r'^admin/', admin.site.urls),
]
```

The middle line is new.

But this isn’t quite right…



```
{"args": {"name": "Jeremy", "phrase": "to boldly go..."}}
```

this is just the last value

```
▼ General
  Request URL: http://127.0.0.1:8000/get/?name=Jeremy&phrase=hello+world&phrase=to+boldly+go...
  Request Method: GET
  Status Code: ● 200 OK
  Remote Address: 127.0.0.1:8000
  Referrer Policy: no-referrer-when-downgrade
▼ Response Headers    view source
  Content-Length: 57
  Content-Type: text/html; charset=utf-8
  Date: T… Jul 2017 05:48:51 GMT
  Server: …Server/0.2 CPython/3.6.1
  X-Frame-Options: SAMEORIGIN
▼ Request Headers    view source
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
  Accept-Encoding: gzip, deflate, br
  Accept-Language: en-US,en;q=0.8
  Cache-Control: max-age=0
  Connection: keep-alive
  Host: 127.0.0.1:8000
  Upgrade-Insecure-Requests: 1
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
▼ Query String Parameters    view source    view URL encoded
  name: Jeremy
  phrase: hello world
  phrase: to boldly go...
```

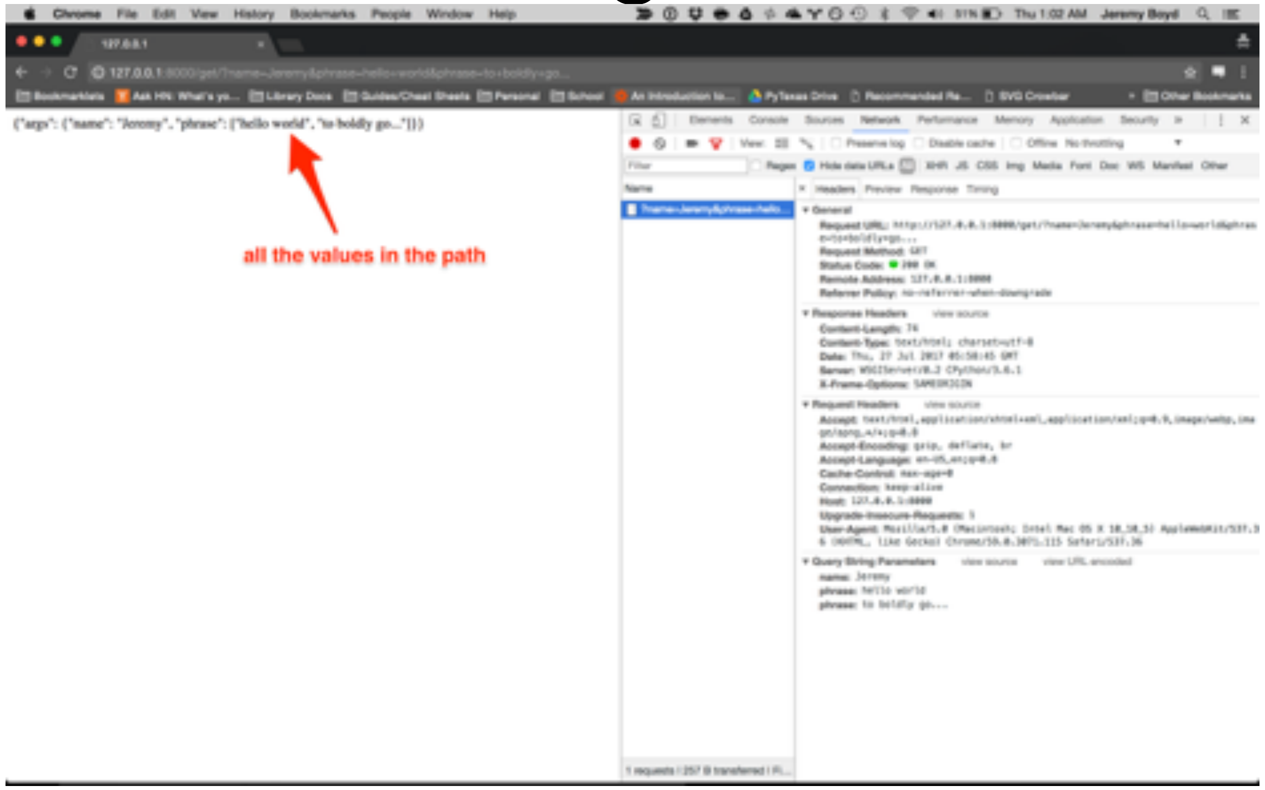
this is okay for MVP
but not ideal

Using multi-valued query params

- **request.GET** has special methods for dealing with multi-valued parameters

```
def echo_get(request):
    response = {}

    args = {}
    for k, v in request.GET.lists():
        if len(v) == 1:
            args[k] = v[0]
        else:
            args[k] = v

    response['args'] = args

    return HttpResponse(json.dumps(response))
```

Looking better



127.0.0.1:8000/get/?name=Jeremy&phrase=hello+world&phrase=to+boldly+go...

```
{"args": {"name": "Jeremy", "phrase": ["hello world", "to boldly go..."]}}
```

all the values in the path

```
▼ General
    Request URL: http://127.0.0.1:8000/get/?name=Jeremy&phrase=hello+world&phras
    e=to+boldly+go...
    Request Method: GET
    Status Code: ● 200 OK
    Remote Address: 127.0.0.1:8000
    Referrer Policy: no-referrer-when-downgrade
▼ Response Headers      view source
    Content-Length: 74
    Content-Type: text/html; charset=utf-8
    Date: Thu, 27 Jul 2017 05:58:45 GMT
    Server: WSGIServer/0.2 CPython/3.6.1
    X-Frame-Options: SAMEORIGIN
▼ Request Headers      view source
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,ima
    ge/apng,*/*;q=0.8
    Accept-Encoding: gzip, deflate, br
    Accept-Language: en-US,en;q=0.8
    Cache-Control: max-age=0
    Connection: keep-alive
    Host: 127.0.0.1:8000
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.3
    6 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
▼ Query String Parameters      view source      view URL encoded
    name: Jeremy
    phrase: hello world
    phrase: to boldly go...
```

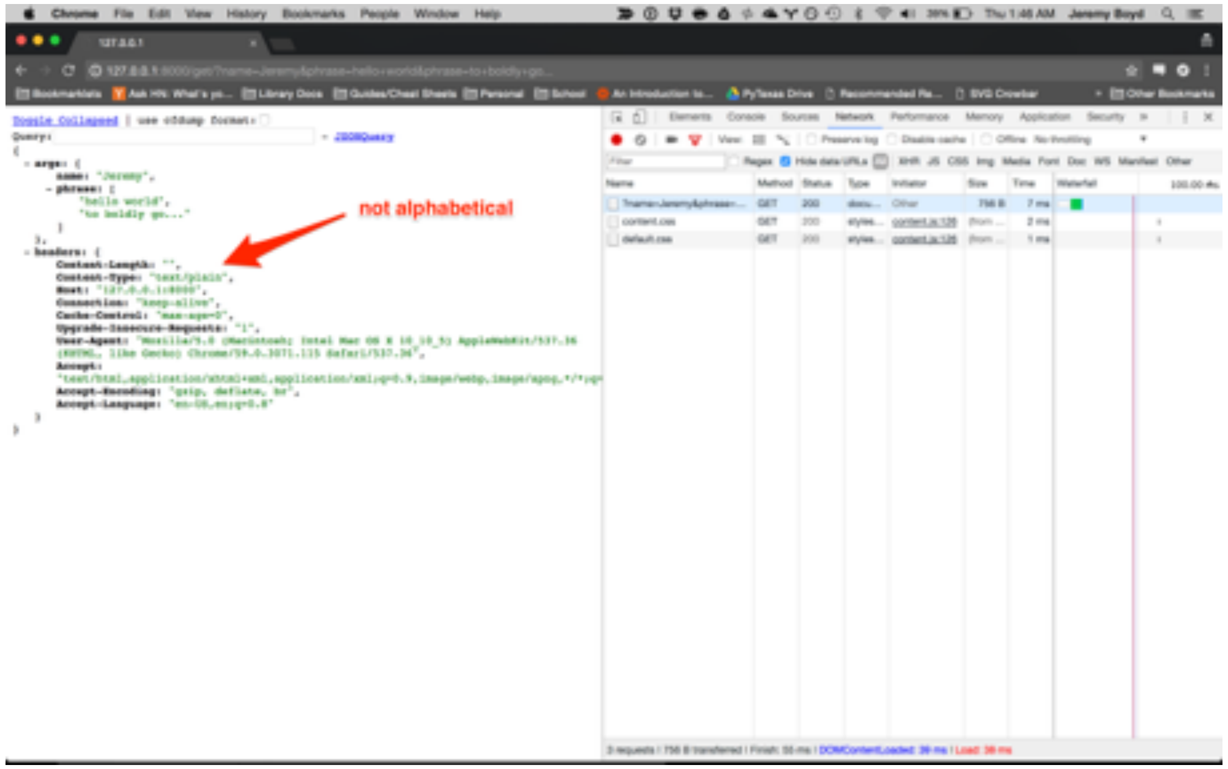
Chrome File Edit View History Bookmarks People Window Help
Thu 1:02 AM Jeremy Boyd
127.0.0.1
Bookmarklets
Ask HN: What's yo...
Library Docs
Guides/Cheat Sheets
Personal
School
An Introduction to...
PyTexas Drive
Recommended Re...
SVG Crowbar
Other Bookmarks
Elements Console Sources Network Performance Memory Application Security
View: Preserve log Disable cache Offline No throttling
Filter Regex Hide data URLs XHR JS CSS Img Media Font Doc WS Manifest Other
Name
?name=Jeremy&phrase=hello...
Headers Preview Response Timing
1 requests | 257 B transferred | Fi...

Adding header info

- **request.META** has the request’s headers
- **Content-*** headers can be used (almost) as-is
- all other request headers start with **HTTP_**

```
headers = {}
for k, v in request.META.items():
    new_key = None
    if k.startswith("CONTENT"):
        new_key = k
    elif k.startswith('HTTP_'):
        new_key = k[5:]
    if new_key is not None:
        headers[new_key.replace('_', '-').title()] = v
response['headers'] = headers
```

Getting there…



Toggle Collapsed | use oldump format: □
Query: ← JSONQuery

```
{
  - args: {
        name: "Jeremy",
      - phrase: [
            "hello world",
            "to boldly go..."
        ]
    },
  - headers: {
        Content-Length: "",
        Content-Type: "text/plain",
        Host: "127.0.0.1:8000",
        Connection: "keep-alive",
        Cache-Control: "max-age=0",
        Upgrade-Insecure-Requests: "1",
        User-Agent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36
        (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36",
        Accept:
        "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=
        Accept-Encoding: "gzip, deflate, br",
        Accept-Language: "en-US,en;q=0.8"
    }
}
```

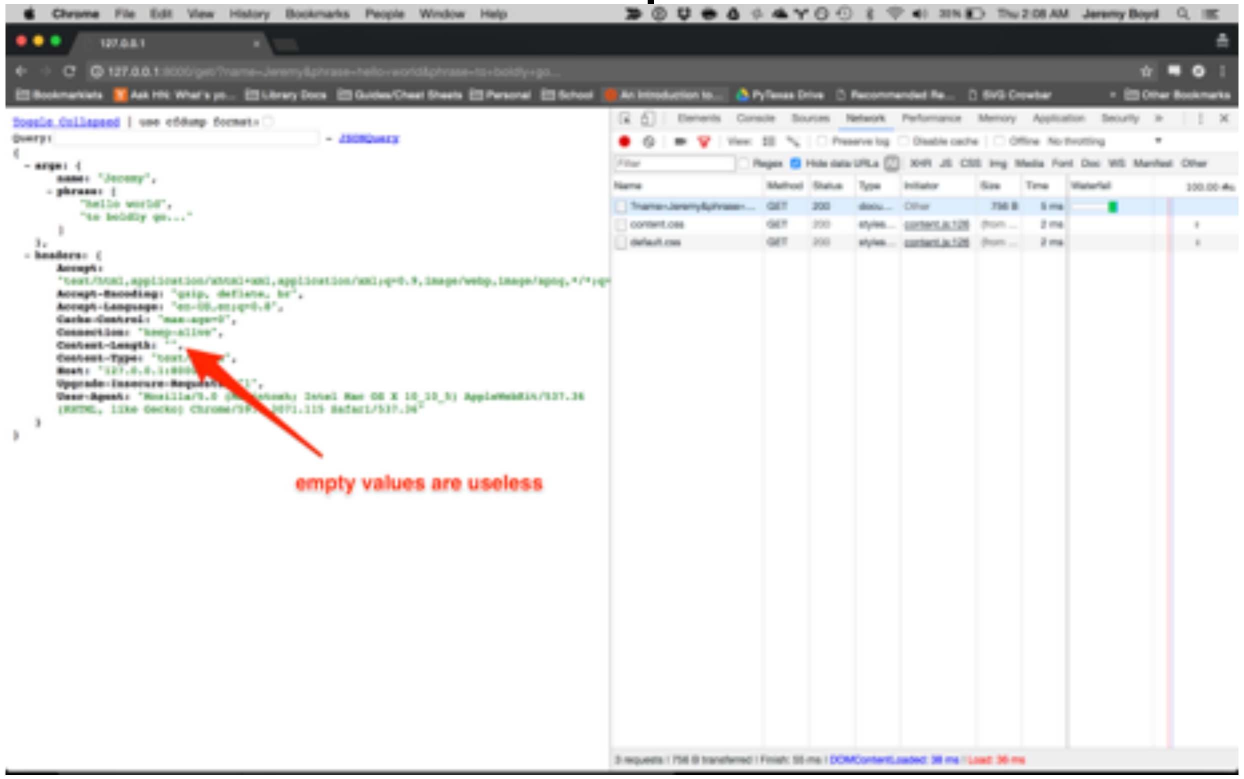
not alphabetical

Chrome File Edit View History Bookmarks People Window Help
Thu 1:46 AM Jeremy Boyd
127.0.0.1
127.0.0.1:8000/get?name=Jeremy&phrase=hello+world&phrase=to+boldly+go...
Elements Console Sources Network Performance Memory Application Security
Name Method Status Type Initiator Size Time Waterfall
?name=Jeremy&phrase=... GET 200 docu... Other 756 B 7 ms
content.css GET 200 styles... content.js:126 (from ... 2 ms
default.css GET 200 styles... content.js:126 (from ... 1 ms
3 requests | 756 B transferred | Finish: 33 ms | DOMContentLoaded: 38 ms | Load: 38 ms

Easy fix

```
return HttpResponse(json.dumps(response, sort_keys=True))
```

Except…



Toggle Collapsed | use cfdump format: □

Query: - JSONQuery

```
{
  - args: {
        name: "Jeremy",
      - phrase: [
            "hello world",
            "to boldly go..."
        ]
    },
  - headers: {
        Accept:
        "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=
        Accept-Encoding: "gzip, deflate, br",
        Accept-Language: "en-US,en;q=0.8",
        Cache-Control: "max-age=0",
        Connection: "keep-alive",
        Content-Length: "",
        Content-Type: "text/[illegible]",
        Host: "127.0.0.1:800[illegible]
        Upgrade-Insecure-Requests: "1",
        User-Agent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36
        (KHTML, like Gecko) Chrome/59.[illegible]3071.115 Safari/537.36"
    }
}
```

empty values are useless

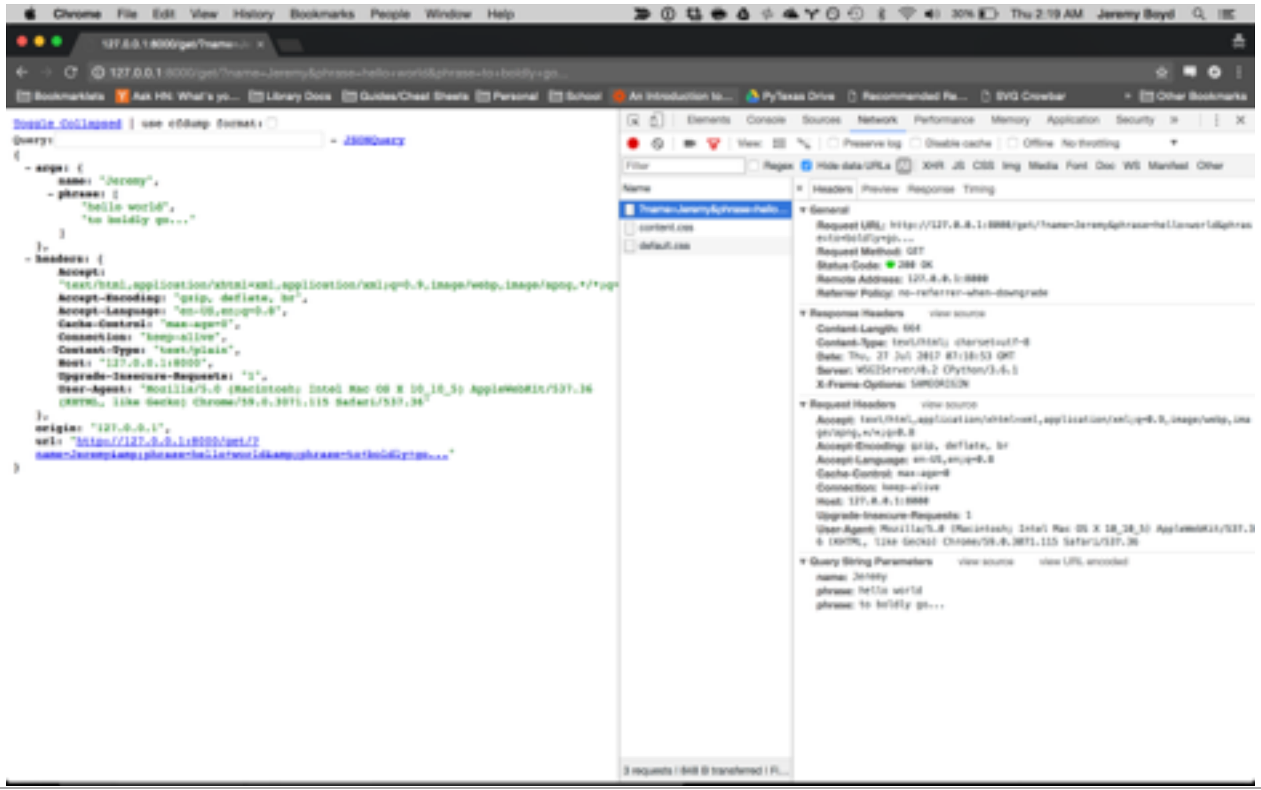
Another easy fix

```
headers = {}
for k, v in request.META.items():
    if not v:
        continue
    new_key = None
    if k.startswith("CONTENT"):
        new_key = k
    elif k.startswith('HTTP_'):
        new_key = k[5:]
    if new_key is not None:
        headers[new_key.replace('_', '-').title()] = v
response['headers'] = headers
```

Add the last couple items

```
response['origin'] = request.META.get('REMOTE_ADDR')
response['url'] = request.get_raw_uri()
```

Feature parity



Toggle Collapsed | use cfdump format: □

Query: - JSONQuery

```
{
  - args: {
      name: "Jeremy",
    - phrase: [
        "hello world",
        "to boldly go..."
      ]
    },
  - headers: {
      Accept:
      "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=
      Accept-Encoding: "gzip, deflate, br",
      Accept-Language: "en-US,en;q=0.8",
      Cache-Control: "max-age=0",
      Connection: "keep-alive",
      Content-Type: "text/plain",
      Host: "127.0.0.1:8000",
      Upgrade-Insecure-Requests: "1",
      User-Agent: "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36
      (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36"
    },
    origin: "127.0.0.1",
    url: "http://127.0.0.1:8000/get/?
    name=Jeremy&amp;phrase=hello+world&amp;phrase=to+boldly+go..."
}
```

```
▼ General
    Request URL: http://127.0.0.1:8000/get/?name=Jeremy&phrase=hello+world&phrase=to+boldly+go...
    Request Method: GET
    Status Code: ● 200 OK
    Remote Address: 127.0.0.1:8000
    Referrer Policy: no-referrer-when-downgrade
▼ Response Headers     view source
    Content-Length: 664
    Content-Type: text/html; charset=utf-8
    Date: Thu, 27 Jul 2017 07:18:53 GMT
    Server: WSGIServer/0.2 CPython/3.6.1
    X-Frame-Options: SAMEORIGIN
▼ Request Headers     view source
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
    Accept-Encoding: gzip, deflate, br
    Accept-Language: en-US,en;q=0.8
    Cache-Control: max-age=0
    Connection: keep-alive
    Host: 127.0.0.1:8000
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36
▼ Query String Parameters     view source     view URL encoded
    name: Jeremy
    phrase: hello world
    phrase: to boldly go...
```

Chrome File Edit View History Bookmarks People Window Help
Thu 2:19 AM Jeremy Boyd
127.0.0.1:8000/get/?name=Jeremy&phrase=hello+world&phrase=to+boldly+go...
Bookmarklets Ask HN: What's yo... Library Docs Guides/Cheat Sheets Personal School An Introduction to... PyTexas Drive Recommended Re... SVG Crowbar Other Bookmarks
Elements Console Sources Network Performance Memory Application Security
View: Preserve log Disable cache Offline No throttling
Filter Regex Hide data URLs XHR JS CSS Img Media Font Doc WS Manifest Other
Name
?name=Jeremy&phrase=hello...
content.css
default.css
Headers Preview Response Timing
3 requests | 848 B transferred | Fi...

Commit your changes

https://github.com/boydjj/awpug-django-2017/releases/tag/v0.4.0

Next steps

- There’s a glaring bug here. Can you find it?
    - Hint: try **http --verbose -f localhost:8000/get/ name=Jeremy**
- Try to implement one of the other httpbin.org URLs

Fin

twitter.com/jeremyjboyd
github.com/boydjj