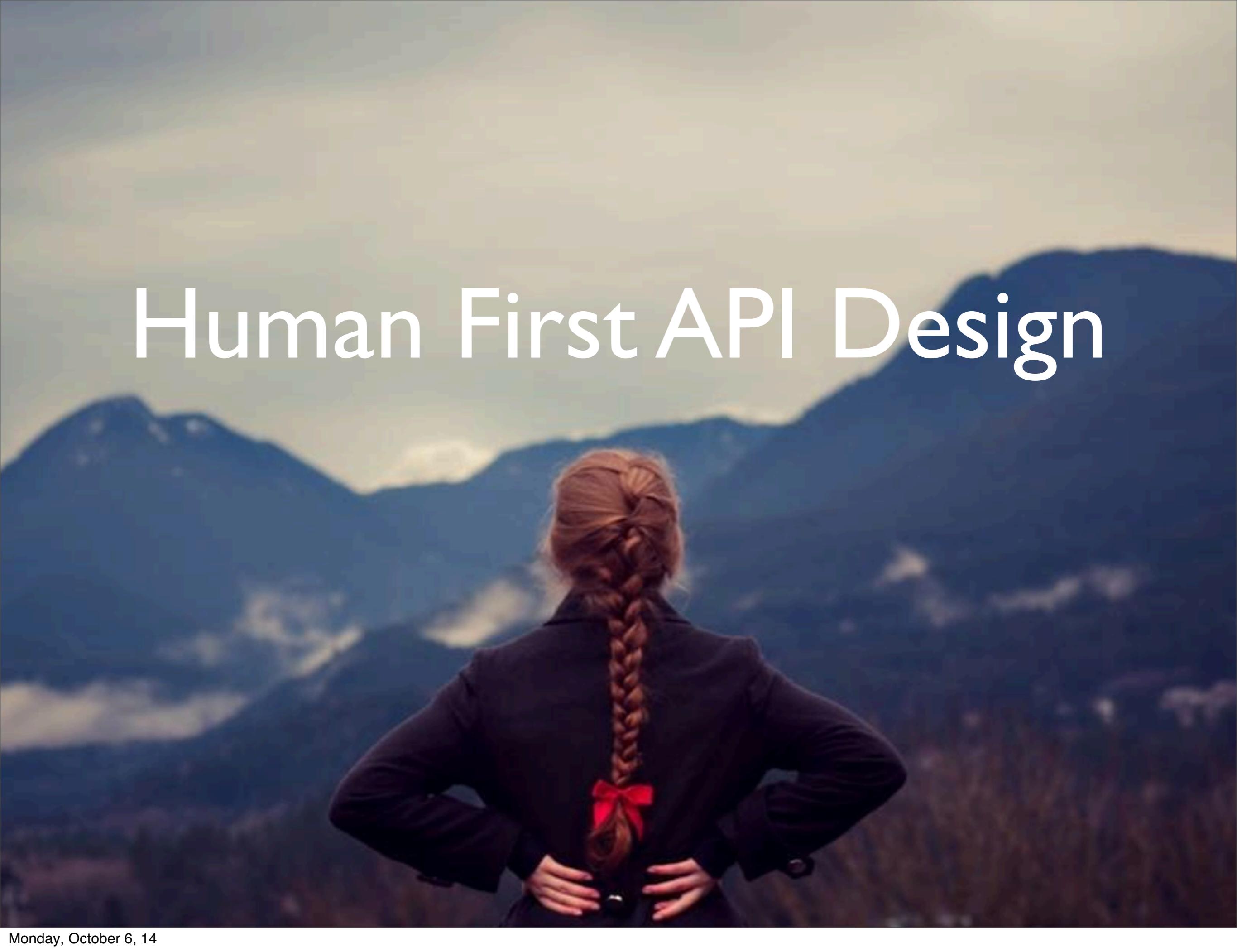


# Human First API Design

A photograph of a woman from behind, looking out over a range of mountains. She has long, reddish-brown hair styled in a braid with a red ribbon. She is wearing a dark, button-down shirt. The mountains in the background are partially obscured by clouds, creating a sense of depth and scale.

@motdotla



# What is an API?

# Application Programming Interface

*Generically*

A set of programming  
instructions for  
accessing a Web-based  
software application.

*Colloquially*

[https://api.twitter.com/  
1.1/search/tweets.json?  
q=USC](https://api.twitter.com/1.1/search/tweets.json?q=USC)

*In Practice*

# History of APIs





1990s

In the beginning, there was not the web. It's the year 1990 and it has not yet come about. Sir Tim Berners-Lee has not yet given it to us.

By Christmas 1990, Sir Tim Berners-Lee had given us HTTP, HTML, and the first web browser (named WorldWideWeb).

# First API!

- \* GET request and HTML response.
- \* Human friendly.

A dynamic space battle scene from Star Trek. In the foreground, a large Federation starship is under attack, with several orange energy beams hitting its hull and causing explosions. Behind it, another starship is shown from a low angle, also under fire. The background is filled with numerous other starships of various sizes and colors, some engaged in combat and others moving through the dark space. The overall atmosphere is one of intense conflict and action.

# Browser Wars

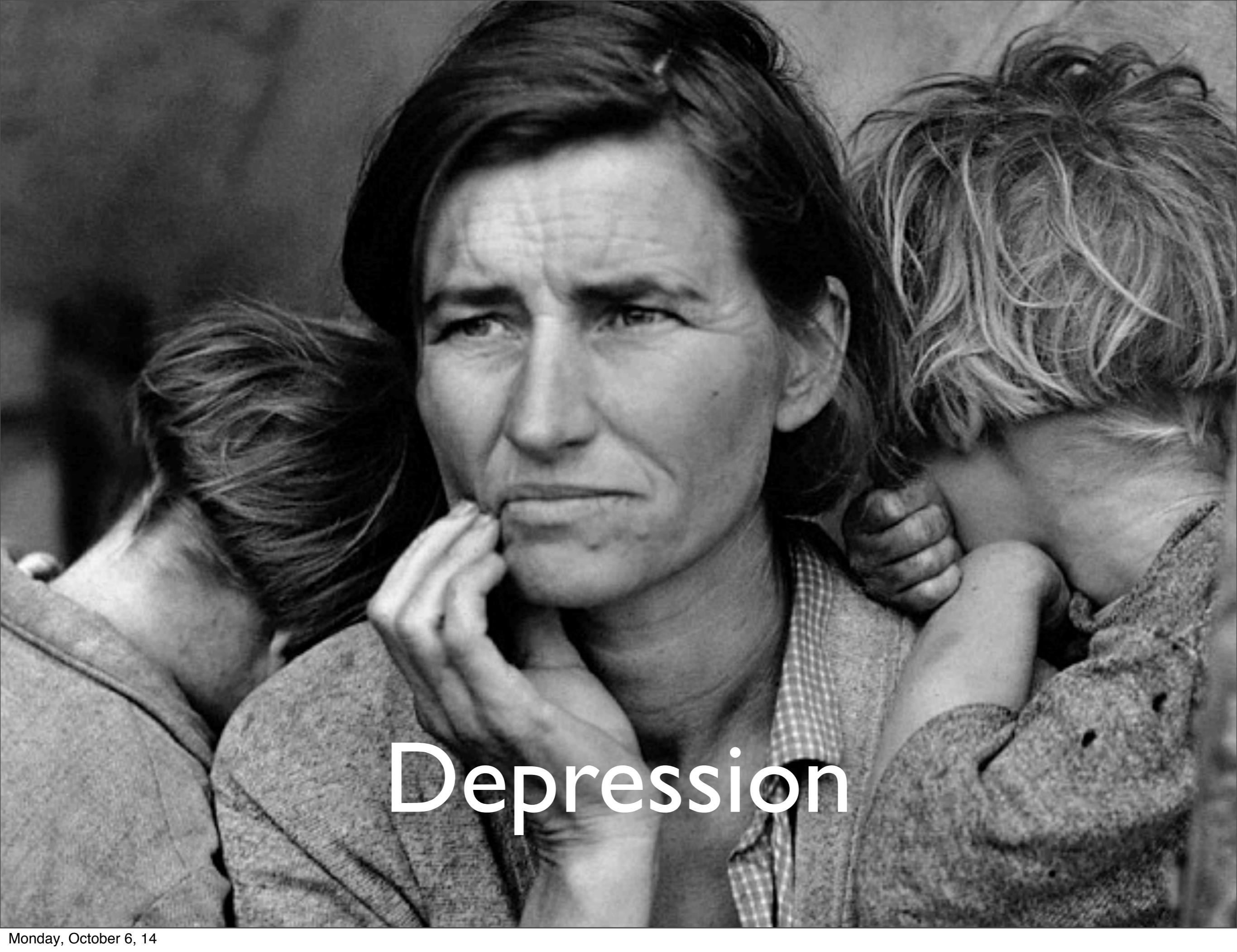
WorldWideWeb  
Mosaic  
Netscape  
Internet Explorer

More than HTML.  
Images, JavaScript, CSS, and more.



Roaring 2000

Startup empires are built.  
Ebay, PayPal, Salesforce, and many other  
giants come out of the late 1990s and the  
year 2000.



# Depression

The bubble bursts. Enter the API.

BizDev 2.0  
February 7th, 2000. Salesforce.  
November 20th, 2000. Ebay.

APIs are wild.  
No accepted rules.

A photograph of two young men in a military or training facility. They are both wearing white t-shirts and dark shorts. They are positioned side-by-side, facing forward, performing push-ups on a dark, polished floor. In the background, there is a large, dark wooden cabinet with a sign that is partially visible, showing the letters 'SOAP'. The lighting is dramatic, with strong shadows and highlights on their skin and uniforms.

SOAP

**SOAP rises to the forefront as a standard -  
by Microsoft and the W3C.**



UNREST

The rule of SOAP causes problems. It leads to an *inflation* of code. The *cost to fuel* one's application becomes prohibitive. Bugs pile up, todos pile up in *long lines*. There just *isn't* enough resources to offset the problems SOAP introduces.

Incidentally, there are still people living in those SOAP days. I weep for them.

A close-up portrait of a man with short brown hair, looking slightly to his left with a neutral expression. He is wearing a dark blue t-shirt.

REST

# Roy Fielding

Dissertation:“Architectural Styles and  
the Design of Network-based Software  
Architectures”.

Removes the ‘UN’ from ‘UNREST’

# Renaissance



Creativity unleashed. Developers create APIs quickly, and mashups from there.

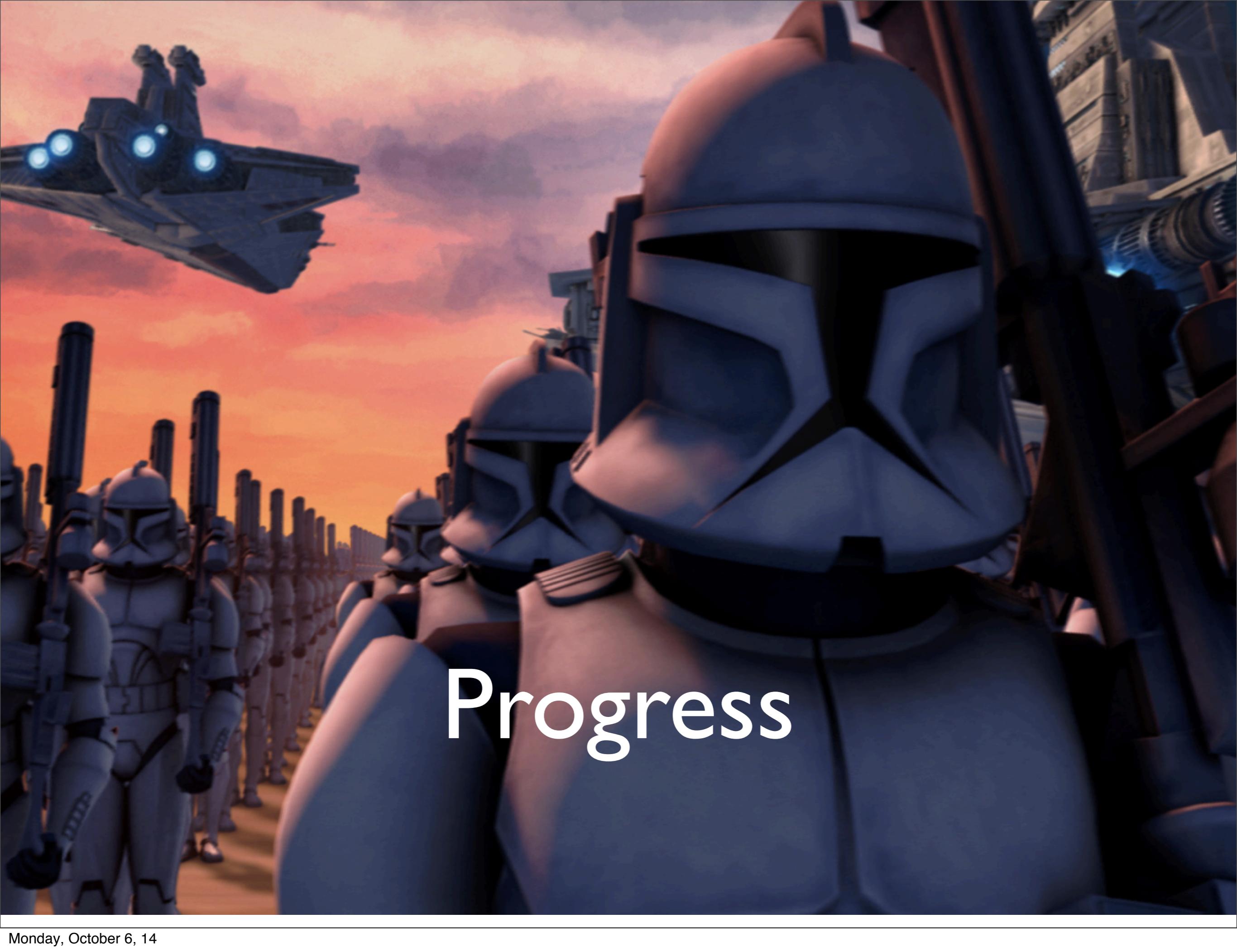
# 2005. Programmable Web.

2006. Ruby on Rails. Popularizes REST.

API first companies like Twilio & SendGrid thrive.

They generally follow REST principles with a tinge of pragmatism. This leads to useful and beautiful things.

The developer is loved during this time.  
The focus is the developer - not machines.

A cinematic shot from Star Wars: The Clone Wars. In the foreground, a long line of clone troopers in white armor march in formation across a sandy ground. In the background, a massive, dark grey starship with glowing blue lights on its hull is docked at a large industrial facility with tall, cylindrical structures. The sky is a dramatic orange and yellow sunset. The word "Progress" is overlaid in large, white, sans-serif letters.

Progress

New empires are built. But slowly REST loses it's focus on the developer - on the human.

REST becomes increasingly cold and complex.

Use the correct verb.

New verbs are introduced like PATCH and HEAD.

Complex OAuth authentication becomes  
the accepted dogma for authenticating an  
API.

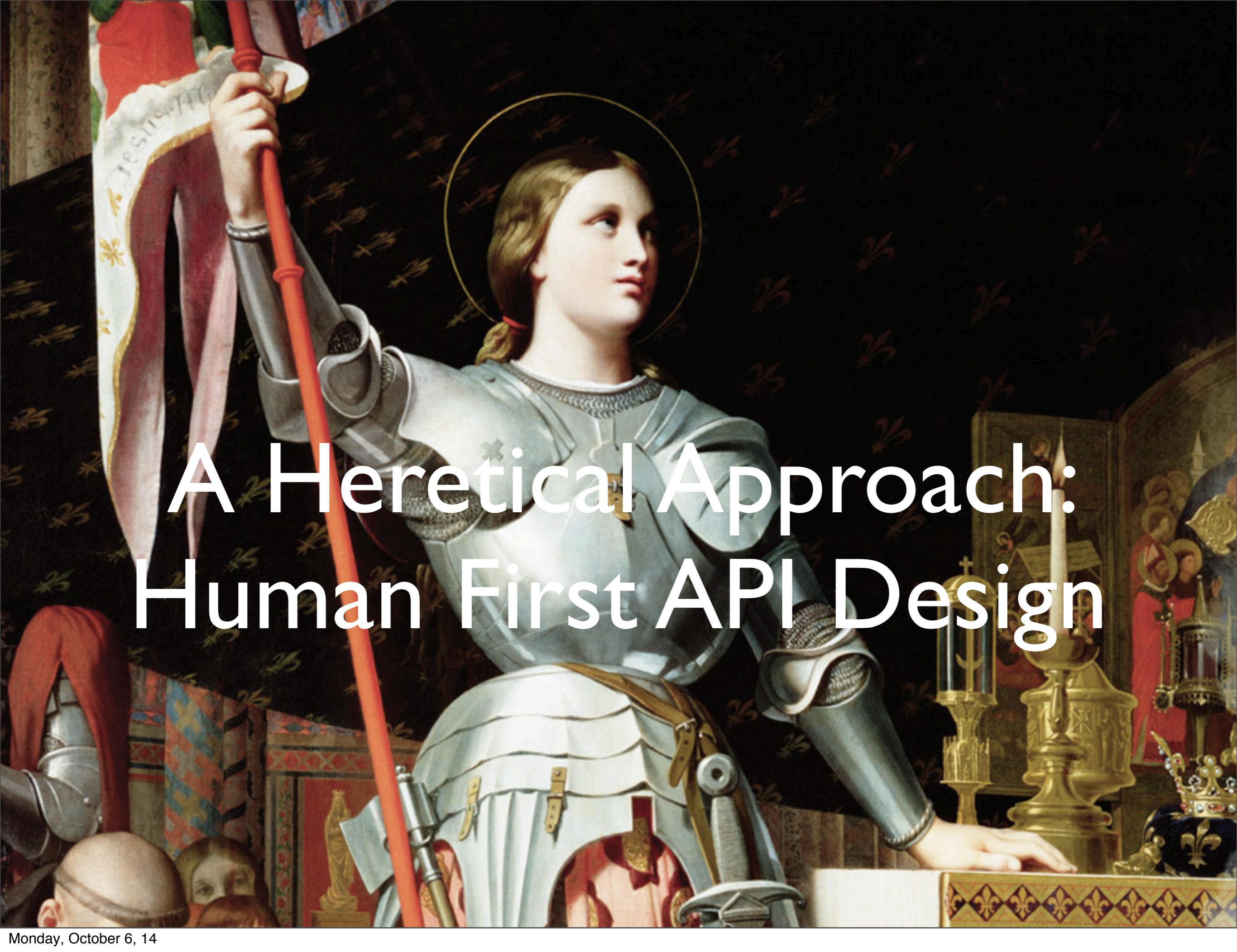
# Strict status code usage.

# Settings in the headers.

Hypermedia is introduced - adding more complexity.

People preach the REST way and scorn  
alternative thinking.

The focus is shifting to the machine, and  
humans are being left out in the cold.

A Heretical Approach:  
Human First API Design

# Guiding Beliefs

Easy as a URL.

[https://carte-api.herokuapp.com/api/v0/accounts/  
create.json?email=\[email\]&api\\_key=\[api\\_key\]](https://carte-api.herokuapp.com/api/v0/accounts/create.json?email=[email]&api_key=[api_key])

# Authorization key in the URL.

```
/accounts/create.json?  
api_key=[api_key]&email=[  
email]
```

# Verb (action) in the URL.

```
/accounts/create.json?  
api_key=[api_key]&email=[  
email]
```

Allow for POST body instead of URL.

```
curl -X POST  
\\accounts/create.json  
\\-d “api_key=[api_key]”  
\\-d “email=[email]”
```

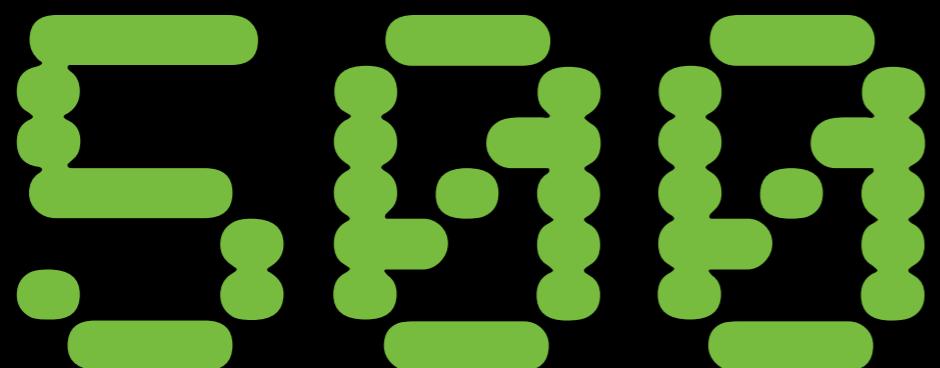
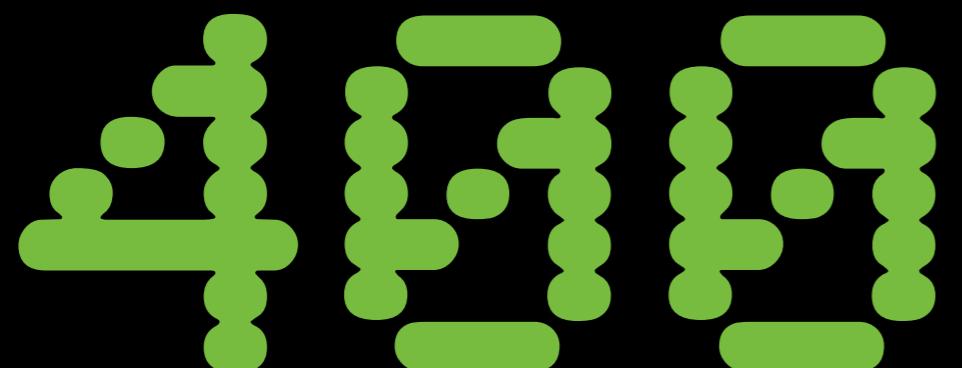
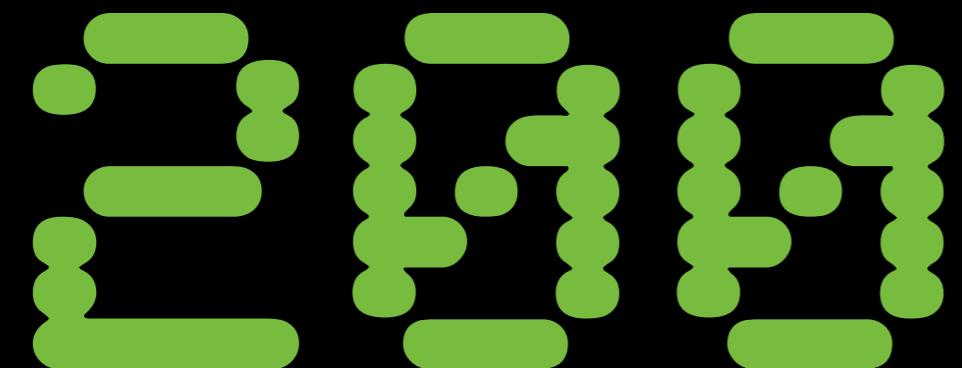
Any verb.

GET

POST

PUT

There are three things a developer wants to know. Did it work, did I mess up, or did the service mess up. Start with those three - 200, 400, and 500. Get more granular as necessary.



Output should be JSON and probably recommended format as json:api.

# { json:api }

**A standard for building APIs in JSON.**

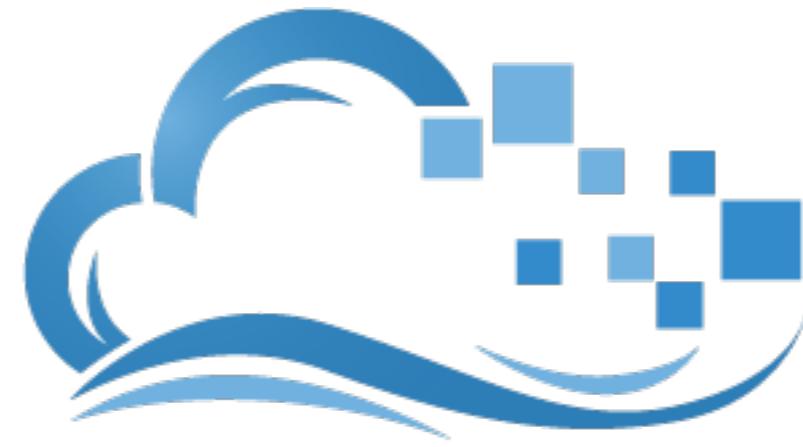
```
{  
  "links": {  
    "posts.author": {  
      "href": "http://example.com/people/{posts.author}",  
      "type": "people"  
    },  
    "posts.comments": {  
      "href": "http://example.com/comments/{posts.comments}",  
      "type": "comments"  
    }  
  },  
  "posts": [{  
    "id": "1",  
    "title": "Rails is Omakase",  
    "links": {  
      "author": "9",  
      "comments": [ "5", "12", "17", "20" ]  
    }  
  }]  
}
```

# Human First Examples



[https://api.sendgrid.com/api/mail.send.json?](https://api.sendgrid.com/api/mail.send.json)  
api\_user=[api\_user]&  
api\_key=[api\_key]&  
to=[to]&  
from=[from]&  
subject=[subject]&  
text=[text]





DigitalOcean

[https://api.digitalocean.com/v1/droplets/new?](https://api.digitalocean.com/v1/droplets/new?client_id=[client_id]&api_key=[api_key]&name=[droplet_name]&size_id=[size_id]&image_id=[image_id]&region_id=[region_id]&ssh_key_ids=[ssh_key_id1],[ssh_key_id2])



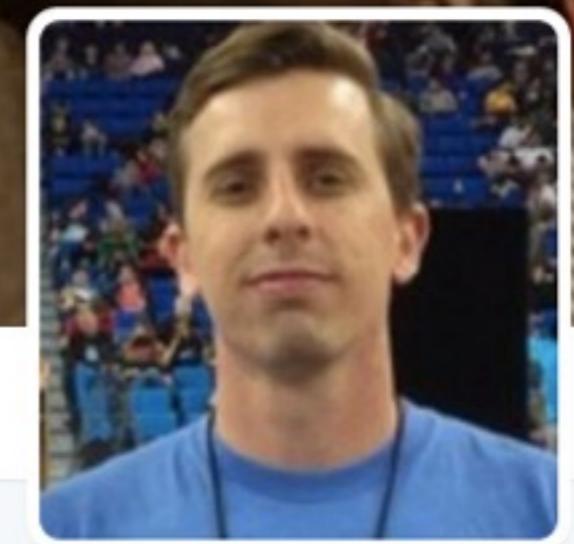
Next time you build an API consider if you  
should design it for humans first.

# Resources

- <http://jsonapi.org/>
- <http://history.apievangelist.com/>
- <http://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencingI.htm>
- <https://sendgrid.com>
- <https://digitalocean.com>



# Thanks



**mot**

@motdotla

Hacker. [@SendGrid](#). The Future is Bright.  
Follow me on [github.com/motdotla](#)

📍 LA

🔗 [mot.la](#)

⌚ Joined March 2009