

INTRO TO HACKATHONS

03 Intro to APIs

0.

Recap

SCHEDULE

3

- | | |
|-------------------|-------------------|
| » Serving The Web | Missed it. |
| » Front End | Missed that too. |
| » APIs | Today! |
| » Deployment | Thursday 22 March |

All at 6pm in Computer Room 4, The Diamond

bit.ly/glitch-hs-18

Open this link in two tabs: one will stay the same as my copy, the other you can edit freely

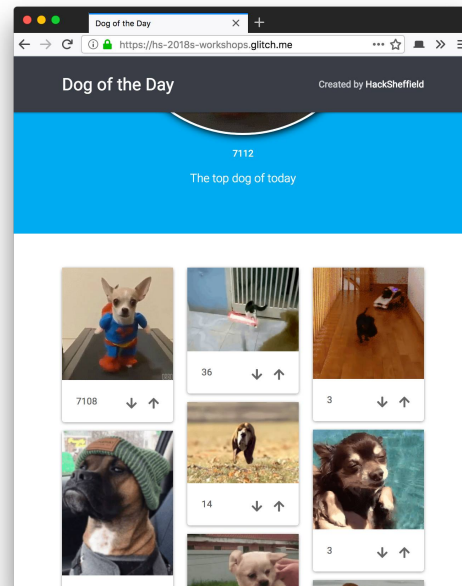
THE PROJECT

5

We currently have this!

But this just has ten dogs...

What if we want lots of new dogs?



- » These workshops are open to all abilities
- » If you don't understand anything, ask away!
- » These workshops are all about learning
- » Slides will be made available

Enough waffle, let's get started!

1.

What's an API?

What's an API?

8

Application Programming Interface

Fundamentally, they allow different systems to communicate...

Let's just show you!

What's JSON?

9

JavaScript Object Notation. It's simply a way to represent objects as a string ("serialisation").

Here's some JSON representing this workshop!

```
{  
  "name": "Intro to APIs",  
  "eventType": "Workshop",  
  "time": "2018-03-15T18:00:00.000Z",  
  "presenters": [  
    "Danny",  
    "Will"  
  ]  
}
```

What's the difference between an API and a data set?

10

An API lets you query specific pieces of data. “I want information about this user”

A data set is a **huge dump** of lots and lots of data. Imagine a super huge spreadsheet. “I want everything there is to know about sunlight intensity in Sheffield per square meter.”

2.

Authenticated APIs

Since APIs are open to the world, they often require that you authenticate with them. This means they can tell you to go away if you upset them (for example, spamming their server with requests).

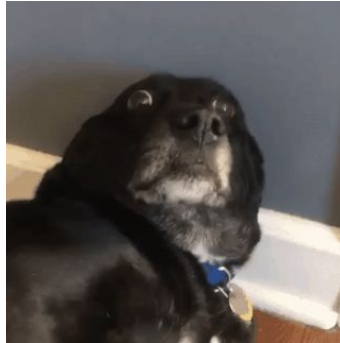
Alternatively, you may be accessing content specific to a user and accessing private data! Obviously, this needs authenticating!

Most authenticated APIs are still simple!

Clarifai Demo

<https://github.com/dannycjones/clarifai-demo>

It can be different depending on the website and can seem quite scary but the process is somewhat like what follows...



1. You need to register an 'app' with **the website**! This gives you some API keys to be able to talk to them.
2. When the user asks you to do something with **the website**, you redirect them to a special OAuth URL (sometimes */oauth/authorize*). Your user will often see a screen asking them to accept the permission you request.

3. After accepting or declining, your user gets redirected back to a URL on your website! They attach a special token as a parameter (for example, `/callback?token=3j4h5jg342hkj45jrfkj`).
4. You go ask to swap this token for an access token. For example, a POST to `/oauth/token`.
5. If everything works out as planned, you'll receive an access token which you can then use to access their API!



OAuth can appear quite complicated - but fear not! People have created some libraries.

- » Want to add login via Google (or FB or...) to your app?
<http://www.passportjs.org/>
- » Twitter stuff? <https://github.com/desmondmorris/node-twitter>

3.

Let's implement the GIPHY API

Map, Filter, Reduce

These are gifts from functional programming!

Map applies a func across all entries of an array!

```
map([🐮, 🍠, 🐔, 🌽], cook)  
=> [🍔, 🍟, 🍗, 🍿]
```

Filter includes only entries which pass a predicate func.

```
filter([🍔, 🍟, 🍗, 🍿], isVegetarian)  
=> [🍟, 🍿]
```

Reduce creates a new result by processing all the entries in the array!

```
reduce([🍔, 🍟, 🍗, 🍿], eat)  
=> 🍌
```

```
map([🐮, 🍌, 🐔, 🌽], cook)  
=> [🍔, 🍟, 🍗, 🍿]
```

```
filter([🍔, 🍟, 🍗, 🍿], isVegetarian)  
=> [🍟, 🍿]
```

```
reduce([🍔, 🍟, 🍗, 🍿], eat)  
=> 🤩
```

Map, Filter, Reduce

developers.giphy.com/dashboard

Go to this link and create an app! Copy your API key.

Or be cheeky and copy my API key:
a9i97NW7vSA7xSYt7CzXexoZLyGpEZJj

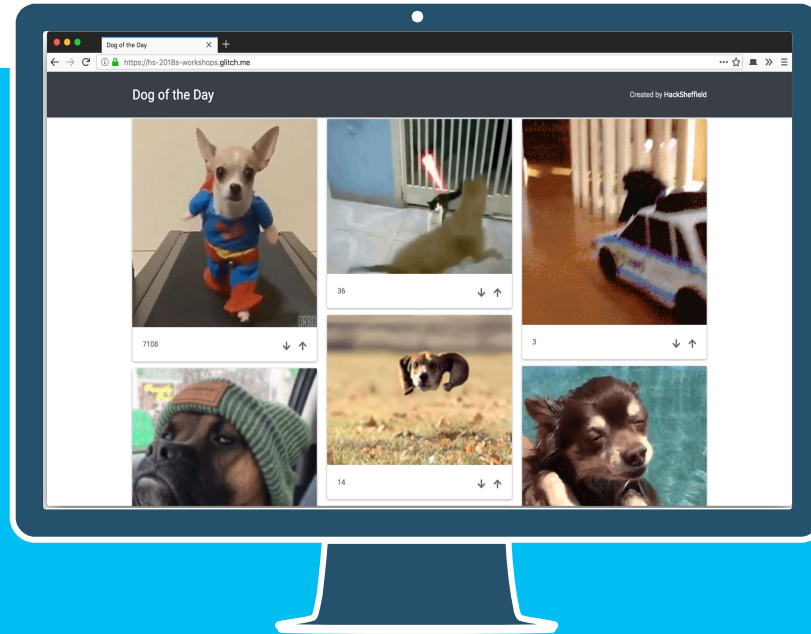
Onto our project...

bit.ly/glitch-hs-18

**THAT'S ALL FOR
TODAY!**

We now have a
working project!

Next week, we
deploy!



THANKS FOR LISTENING!

28

Any questions?

