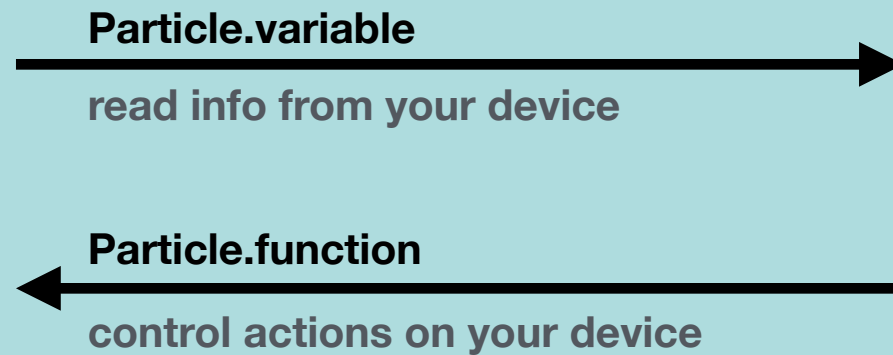
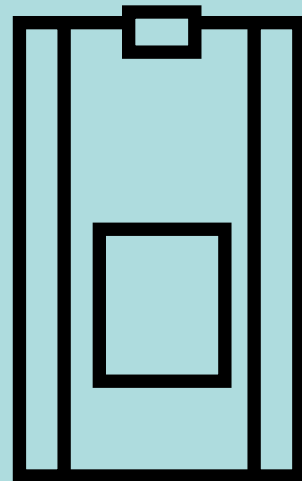
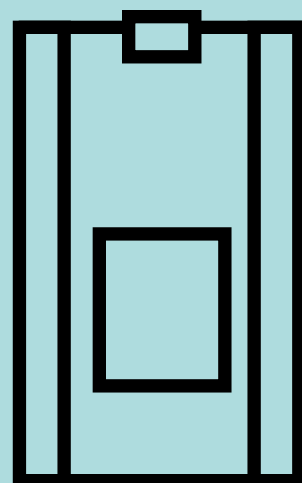


# Working with Webhooks

# Webhooks



**Mobile/Web Apps,  
Services (like IFTTT),  
etc.**



**Web Request  
to online APIs**

# Webhook

**A managed way for your device to easily talk to online services, make request for data and do things with lots of stuff online**

**What can you connect to?**

**Lots**

**What can you connect to?**

**Lots**

**Take a look at this incomplete list:**

**<https://github.com/toddmotto/public-apis>**

**\*Almost every major web application and service out there**

# Public APIs build failing

A collective list of free JSON APIs for use in web development.

For information on contributing to this project, please see the [contributing guide](#).

Please note a passing build status indicates all listed APIs are available since the last update. A failing build status indicates that 1 or more services may be unavailable at the moment.

## Index

- [Animals](#)
- [Anime](#)
- [Anti-Malware](#)
- [Art & Design](#)
- [Books](#)
- [Business](#)
- [Calendar](#)
- [Cloud Storage & File Sharing](#)
- [Currency Exchange](#)
- [Data Access](#)
- [Data Validation](#)
- [Development](#)
- [Documents & Productivity](#)
- [Environment](#)
- [Finance](#)
- [Food & Drink](#)

# What can you connect to?

There are countless services that let you do things like:

- Send an email
- Get a list of non-working days in this calendar year
- Look up currency exchange prices
- Generate random names
- Get AirQuality and weather information
- Search Recipes
- Do Machine learning with [API.ai](#) or [Wit.ai](#)
- Get news links and information from the [NYTimes](#) and [Guardian](#)
- [tronalddump.io](#)
- Look up [CityBikes](#)
- Get your [FitBit](#) data
- Find [NFL](#) arrests
- Get transit and transportation info

All taken from: <https://github.com/toddmotto/public-apis> and yes, it's by no means exhaustive

# Webhooks

**OK, but how does it work?**

**Before we do that... Lets talk about  
APIs**



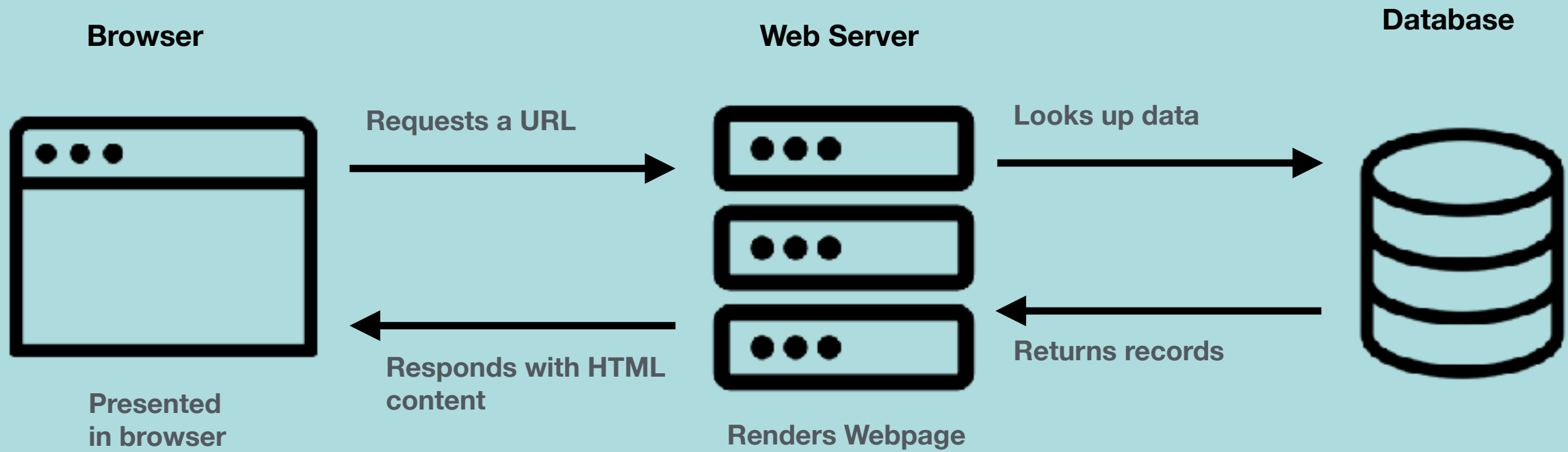
# APIs

**APIs (application programming interfaces) are the web without the UI.**

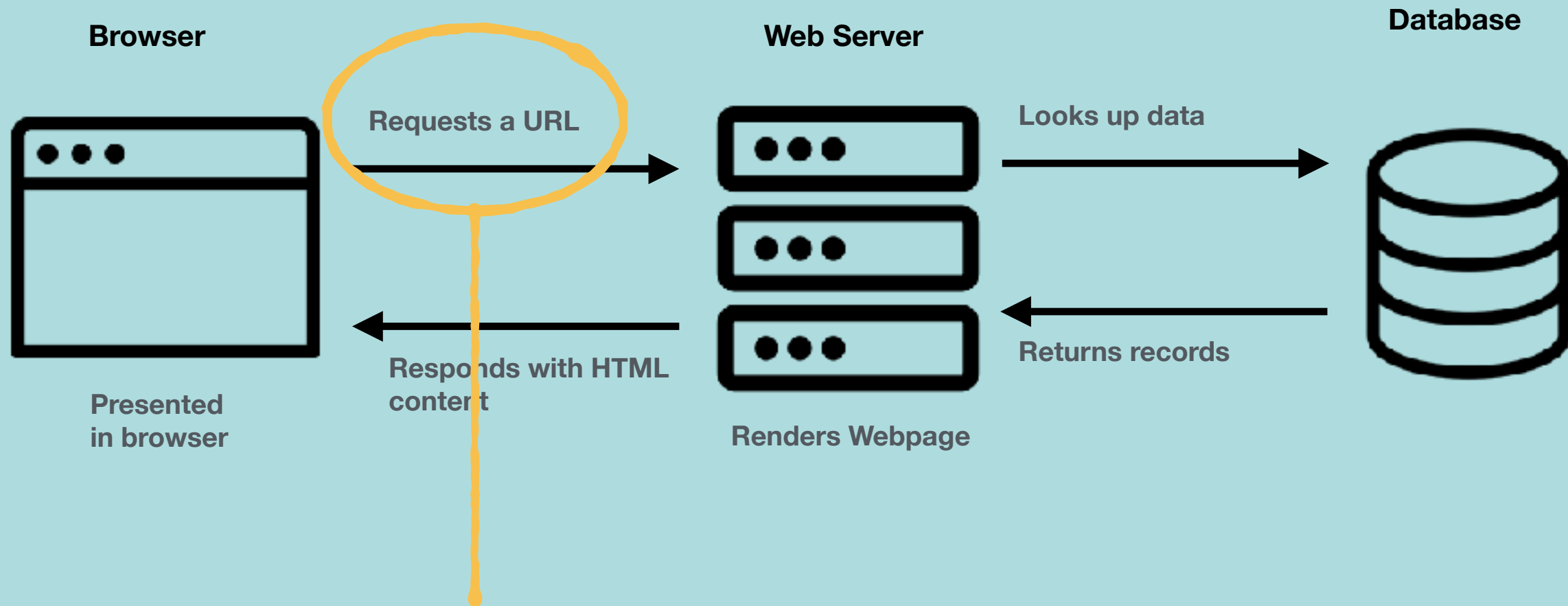
**They're a mainline to the data behind what you see in a browser.**

**It works the same way.**

# APIs



# APIs



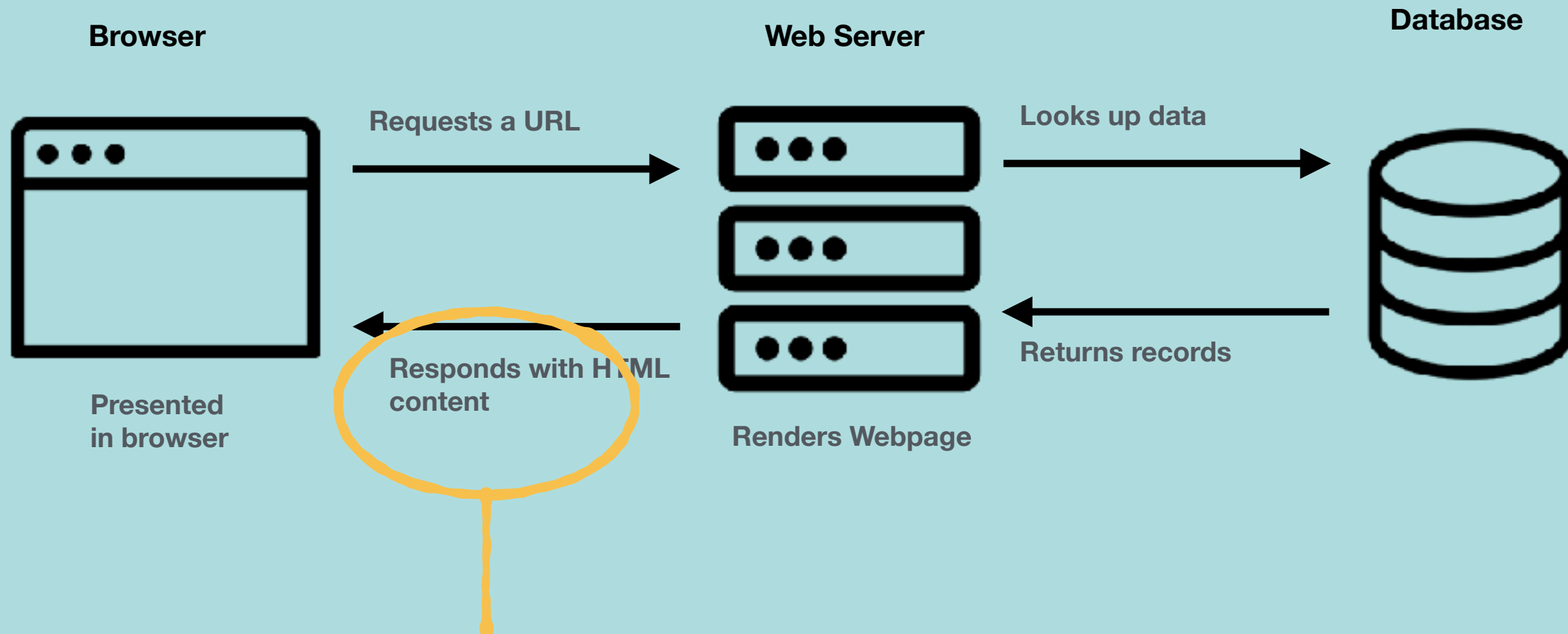
Requests are formed with a method (typically GET or POST), Headers, parameters and a Body

If a request comes from a web browser (typing it in) it's always a GET request

For a get request, parameters are sent in the URL as a query string e.g. [google.com?q=search%20this](https://www.google.com/search?q=search%20this)  
This is good if you want to use the back button and have your page remember what the options were

POST is used when your sending user information to the server. It hides the parameters in the headers so they don't show in the URL. This is good if you're submitting a form!

# APIs



Every request will have a response. It include a body (or the content) but also a status code. The status code lets your browser know if everything went OK. Have you seen the 404 error? Thats a bad status code!

There are four different levels of codes:

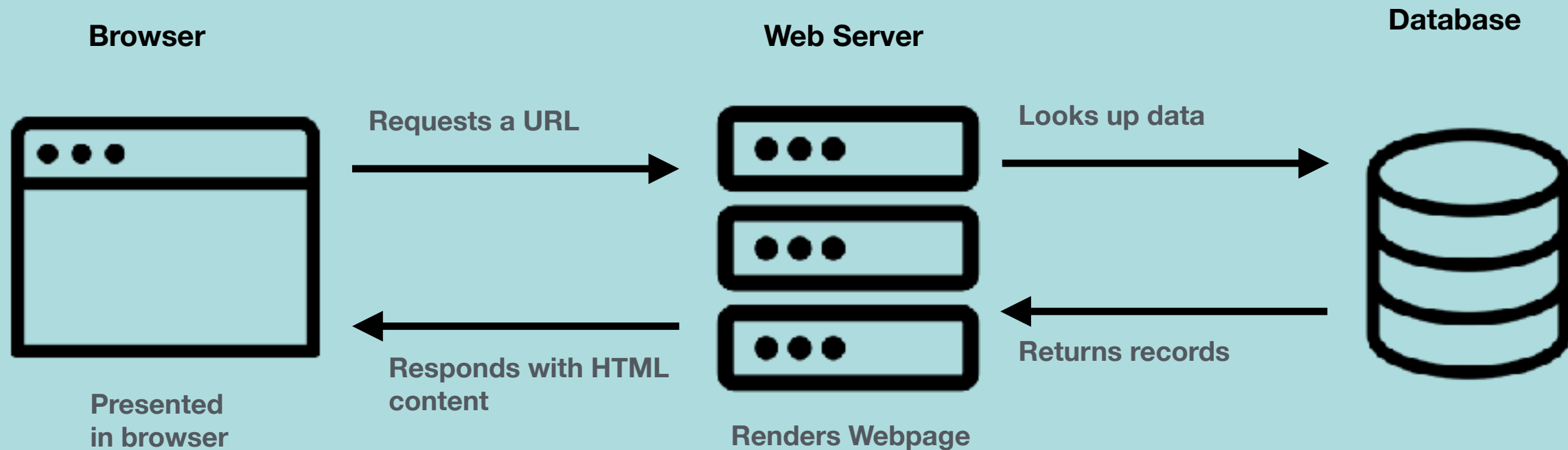
2xx = Success - this is the norm! 200 is the best.

3xx = Redirect - you've probably encountered a 301 Resource Permanently Moved (redirection) at some point

4xx = User error - everyone has definitely come across a 404 error before, right?

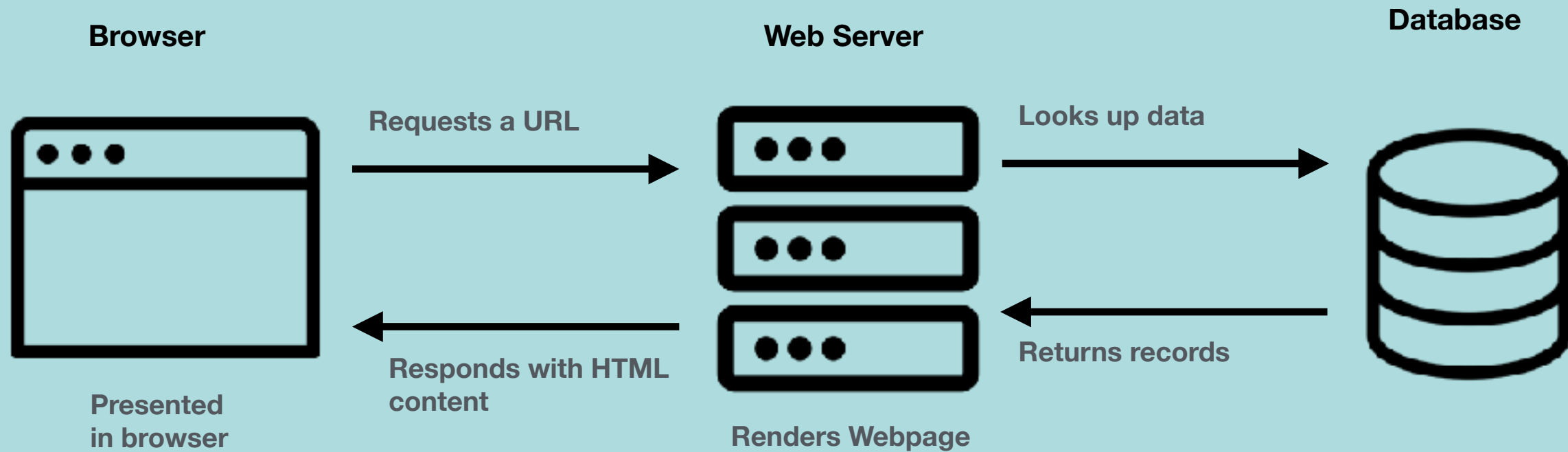
5xx = Server error - this is the bad stuff; it means the server has fallen over.

# APIs



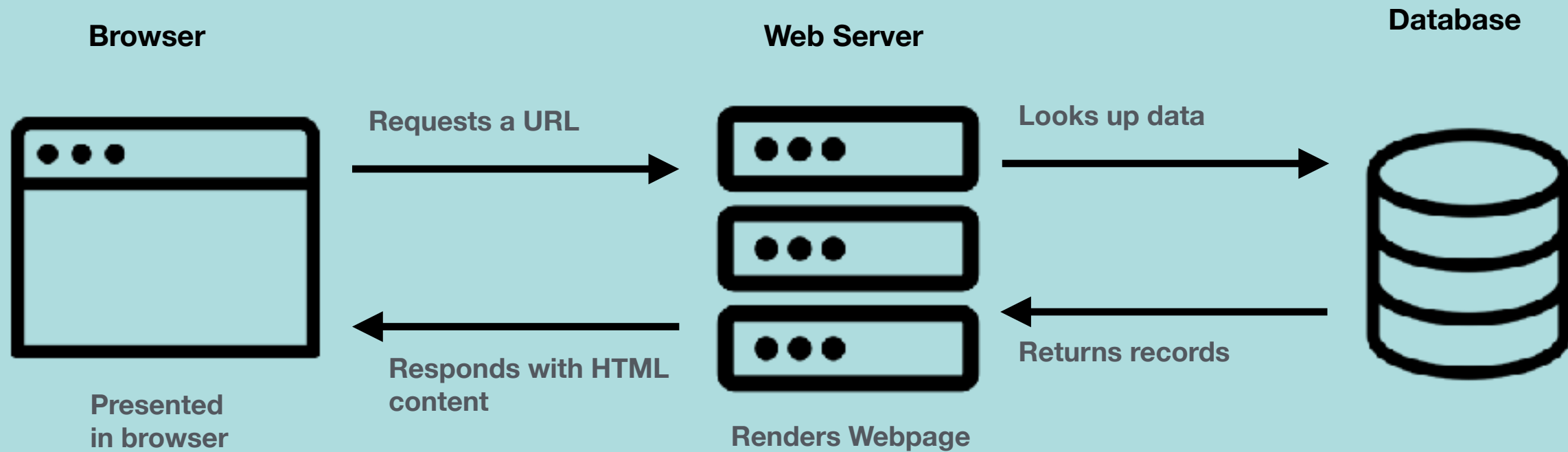
**This is the basics of what happens every time you type in a url like <https://www.cmu.edu/integrated-innovation/>**

# APIs



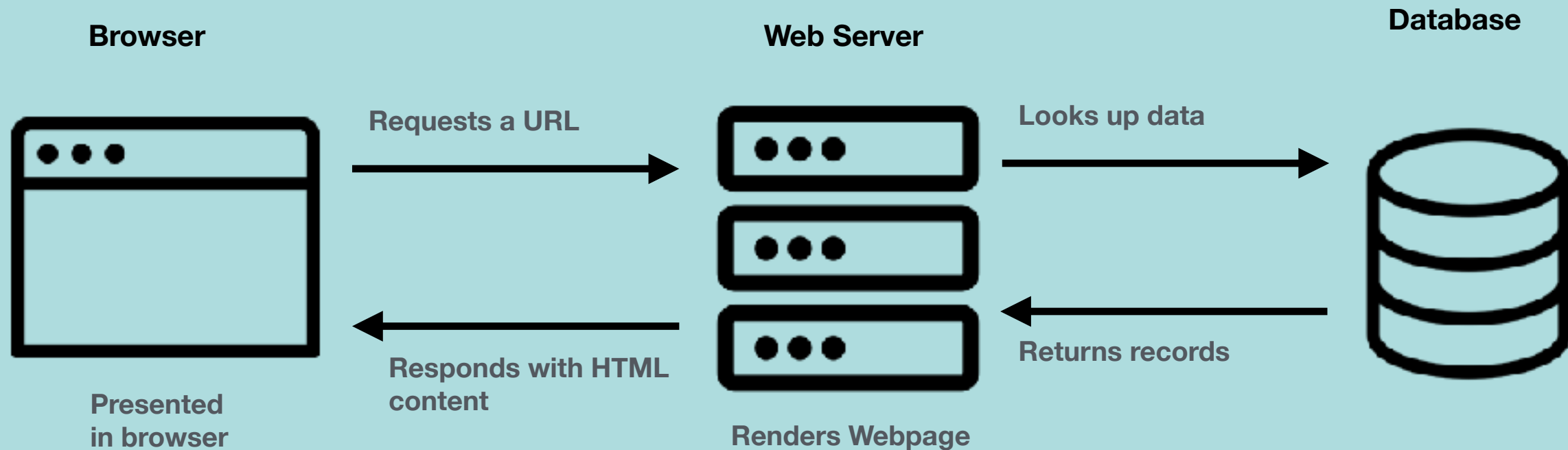
**This is what an API does, but without the HTML**

# APIs



**This is what an API does, but without the HTML**

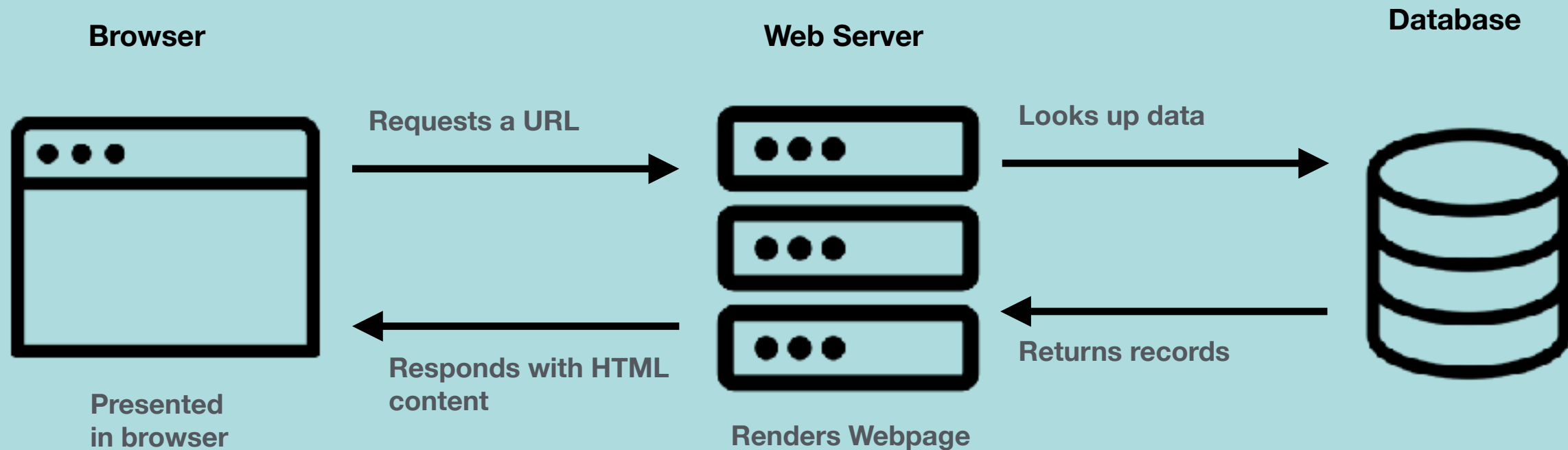
# APIs



**This is what an API does, but  
without the HTML  
Well it does one more thing.**



# APIs



**This is what an API does, but  
without the HTML  
Well it does one more thing.**

## Look at these URLs

<https://github.com/daraghbyrne/advancediot2017>

[https://www.instagram.com/particle\\_io/](https://www.instagram.com/particle_io/)

<https://twitter.com/kottke/status/857290685543829519>

## Look at these URLs

<https://github.com/daraghbyrne/advancediot2017>

[https://www.instagram.com/particle\\_io/](https://www.instagram.com/particle_io/)

































<https://twitter.com/kottke/status/857290685543829519>

They say something about the resource we're looking for  
e.g. show me tweet with the id '857290685543829519'  
that was posted by the account @kottke

**Look at these URLs**

**APIs often exploit structured URLs to make requests for data related to an online application**

# Twitter API

Select an API method			
<input type="text" value="Search methods..."/>			
Timelines			
	GET	/statuses/mentions_timeline.json	
	GET	/statuses/user_timeline.json	
	GET	/statuses/home_timeline.json	
Tweets			
	GET	/statuses/retweets/{id}.json	
	GET	/statuses/show/{id}.json	
	POST	/statuses/destroy/{id}.json	
	POST	/statuses/update.json	
	POST	/statuses/retweet/{id}.json	
	POST	/statuses/update_with_media.json	
	GET	/statuses/oembed.json	
Search			
	GET	/search/tweets.json	
Help			
	GET	/help/configuration.json	
	GET	/help/languages.json	
	GET	/help/privacy.json	
	GET	/help/tos.json	
	GET	/application/rate_limit_status.json	

# Twitter API

Select an API method			
<input type="text" value="Search methods..."/>			
Timelines			
	GET	/statuses/mentions_timeline.json	
	GET	/statuses/user_timeline.json	
	GET	/statuses/home_timeline.json	
Tweets			
	GET	/statuses/retweets/{id}.json	
	GET	/statuses/show/{id}.json	
	POST	/statuses/destroy/{id}.json	
	POST	/statuses/update.json	
	POST	/statuses/retweet/{id}.json	
	POST	/statuses/update_with_media.json	
	GET	/statuses/oembed.json	
Search			
	GET	/search/tweets.json	
Help			
	GET	/help/configuration.json	
	GET	/help/languages.json	
	GET	/help/privacy.json	
	GET	/help/tos.json	
	GET	/application/rate_limit_status.json	

MOST APIs  
Repond with  
JSON

# Webhooks

**Data is typically in JSON.**

**But what is JSON?**

# JSON

**It's JavaScript Object Notation**

**A lightweight, compact way of sharing data**

```
{  
  key: "value",  
  num: 0.34,  
  isTrue: false,  
  anArray: [1,123,23,93,"strings"],  
  anObject: { id: 1, name: "Bob" }  
}
```



# Web API

## Web Page Response

GET <https://twitter.com/kottke/status/857290685543829519>



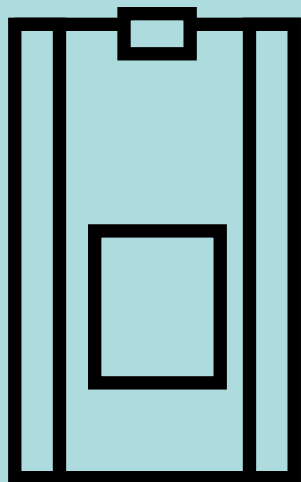
## Web API Response

GET <https://api.twitter.com/1.1/statuses/show.json?id=857290685543829519>

```
{
  "coordinates": null,
  "favorited": false,
  "truncated": false,
  "created_at": "Wed Jun 06 20:07:10 +0000 2012",
  "id_str": "210462857140252672",
  "in_reply_to_user_id_str": null,
  "contributors": [
    14927800
  ],
  "text": "Along with our new #Twitterbird, we've also updated our",
  "retweet_count": 66,
  "in_reply_to_status_id_str": null,
  "id": 210462857140252672,
  "geo": null,
  "retweeted": true,
  "possibly_sensitive": false,
  "in_reply_to_user_id": null,
  "place": null,
  "user": {
    "profile_sidebar_fill_color": "DDEEF6",
    "profile_sidebar_border_color": "C0DEED",
    "profile_background_tile": false,
    "name": "Twitter API",
    "profile_image_url": "http://a0.twimg.com/profile_images/228417",
    "created_at": "Wed May 23 06:01:13 +0000 2007",
    "location": "San Francisco, CA",
    "follow_request_sent": false,
    "profile_link_color": "0084B4",
    "is_translator": false
  }
}
```

# Webhooks

OK, but how does it work?



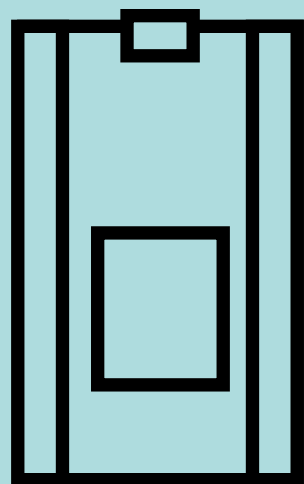
????



**Web Request  
to online APIs**

# Webhooks

OK, but how does it work?



MAGIC!



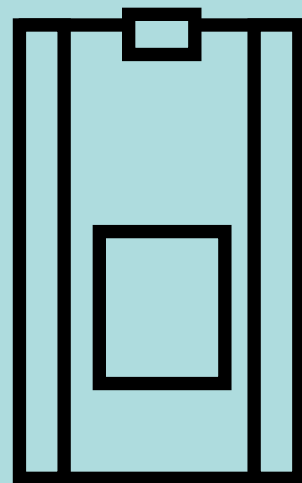
**Web Request  
to online APIs**

# Webhooks

```
Particle.subscribe(  
  "hook-response/webhookname",  
  webhookHandler)
```

Matches the web hook  
Triggers a request

Instagram!

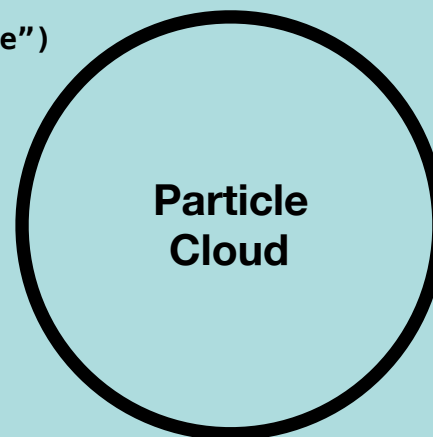


Particle.publish("webhookname")

An event that matches  
your web hook name  
+ data



response event



Requests your URL



Responds with JSON



```
void webhookHandler(...){  
  //handle response  
}
```

Filters the response  
and sends it back down to  
the

# Web Requests

**Your photon will talk to other services through the Particle cloud.**

**The Particle cloud acts as a middleman:**

- 1. To be helpful and responsible (avoids you hammering someones server!)**
- 2. Some APIs return a lot of data; it lets you filter it**
- 3. To allow you to reuse the same web hook across lots of devices**

# Webhooks Examples

**Geolocating your Particle (converting WiFi signals to GPS location)**

**Send a message to Slack**

**Get the current weather or AirQuality and Show on a NeoPixel**

**Send an SMS**



## Devices



ID	Type	Name	Last Connection	
2a002...	Photon	McLuhan	Apr 24th 2017 at 7:55 pm	
34002...	Photon	Magenta	Apr 6th 2017 at 4:58 pm	
3f006...	Photon	Lanier	Apr 6th 2017 at 4:30 pm	
36003...	Photon	Weir	Apr 6th 2017 at 4:24 pm	
1f002...	Photon	marsstudio5	Mar 30th 2017 at 12:05 am	
1e003...	Photon	Swatch	Mar 29th 2017 at 10:27 pm	
54ff6...	Core	Holo	Feb 11th 2017 at 3:41 pm	
41003...	Photon	Pantone	Sep 19th 2016 at 7:02 pm	
54ff7...	Core	Norman	Sep 19th 2016 at 5:53 pm	
3b003...	Photon	marsstudio8	May 2nd 2016 at 6:09 pm	

# Webhooks Examples

**Send a message to Slack**





## Devices



ID	Type	Name	Last Connection	
● 2a002...	Photon	McLuhian	Apr 24th 2017 at 7:55 pm	...
● 34002...	Photon	Magenta	Apr 6th 2017 at 4:58 pm	...
● 3f006...	Photon	Lanier	Apr 6th 2017 at 4:30 pm	...
● 36003...	Photon	Weir	Apr 6th 2017 at 4:24 pm	...
● 1f002...	Photon	marsstudio5	Mar 30th 2017 at 12:05 am	...
● 1e003...	Photon	Swatch	Mar 29th 2017 at 10:27 pm	...
● 54ff6...	Core	Holo	Feb 11th 2017 at 3:41 pm	...
● 41003...	Photon	Pantone	Sep 19th 2016 at 7:02 pm	...
● 54ff7...	Core	Norman	Sep 19th 2016 at 5:53 pm	...
● 3b003...	Photon	marsstudio8	May 2nd 2016 at 6:09 pm	...



# Webhooks

**Getting the weather**

**Webhooks**

**Send an SMS**