GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-15-resources` repo to your computer
2. Open the `09-ajax-apis/starter-code` folder in your code editor

JAVASCRIPT DEVELOPMENT

AJAX & ASYNCHRONOUS JAVASCRIPT

LEARNING OBJECTIVES

At the end of this class, you will be able to

- Use event delegation to manage dynamic content.
- Use implicit iteration to update elements of a jQuery selection
- Identify all the HTTP verbs & their uses.
- Describe APIs and how to make calls and consume API data.
- Access public APIs and get information back.
- Implement an Ajax request with Fetch.
- Create an Ajax request using jQuery.

AGENDA

- APIs
- HTTP
- Ajax using Fetch
- Ajax & jQuery
- Separation of concerns

WEEKLY OVERVIEW

WEEK 5	Events & jQuery / Ajax & APIs
WEEK 6	Asynchronous JS & callbacks / Advanced APIs
WEEK 7	Project 2 lab / Prototypal inheritance

EXIT TICKET QUESTIONS

EXIT TICKET QUESTIONS

1. What is the best practice for knowing when to use libraries like jQuery. Is it a time and money type of decision and what's the current trend.
2. Is there any tips other than knowing syntax, when jquery (i.e. $ sign), JS, or Vanilla JS is being used?
3. There are a lot of new terms and syntax elements that have been introduced over the last few classes. Is there anything we can do to better familiarize with these, aside from coding and general practice?
4. You mentioned that less and less projects are using jQuery. What's the library or the practice that is replacing it?
5. From an optimization perspective is vanilla javascript always faster without jQuery when loading a webpage.

JQUERY BEST PRACTICES

METHOD CHAINING

CHAINING

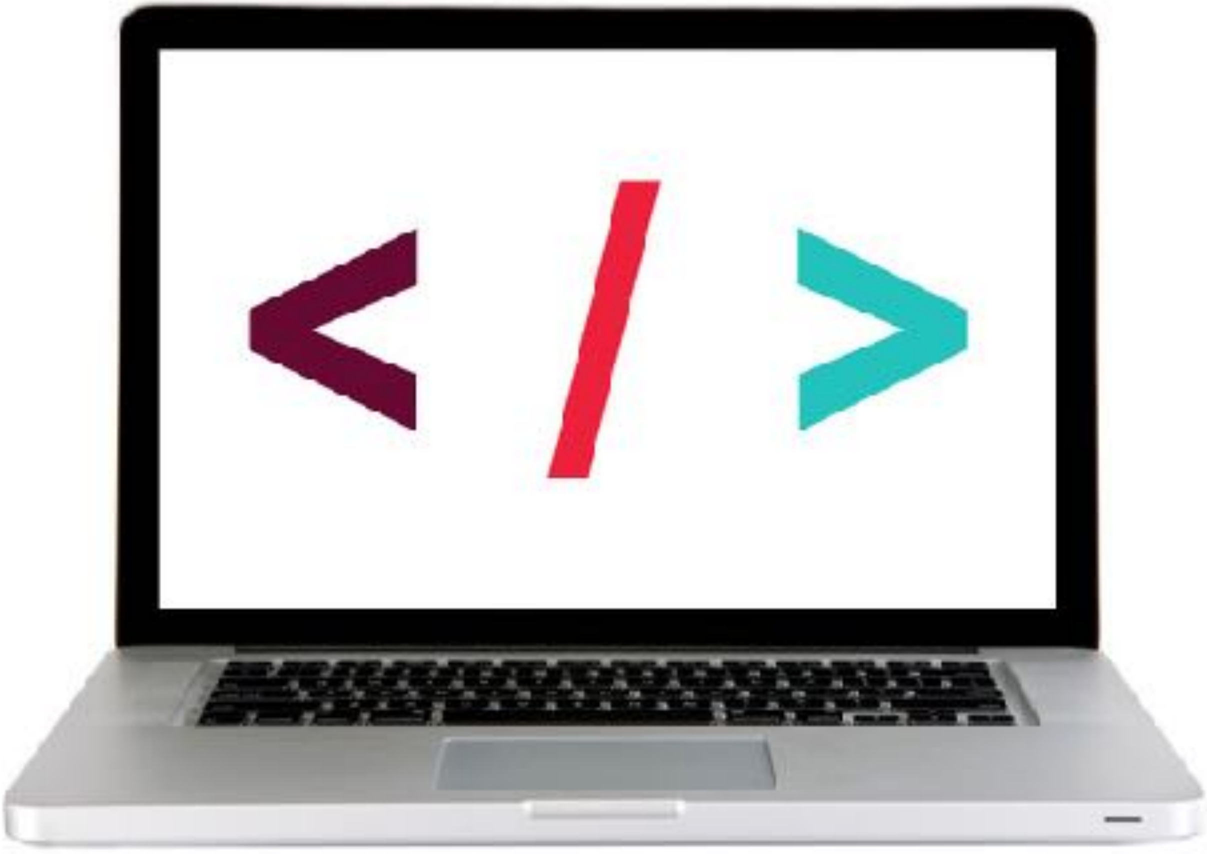
without chaining:

```
let $mainCaption = $('<p>');
let $captionWithText = $mainCaption.text('Today');
let $fullCaption = $captionWithText.addClass('accent');
```

with chaining:

```
let $fullCaption = $('<p>').text('Today').addClass('accent');
```

ADVANCED JQUERY



LET'S TAKE A CLOSER LOOK

IMPLICIT ITERATION

IMPLICIT ITERATION

explicit iteration

```
$(‘li’).each(function() {
 $(this).removeClass(‘current’);
});
```

jQuery `.each()` method works like a `forEach` loop

✗ not necessary for element collections

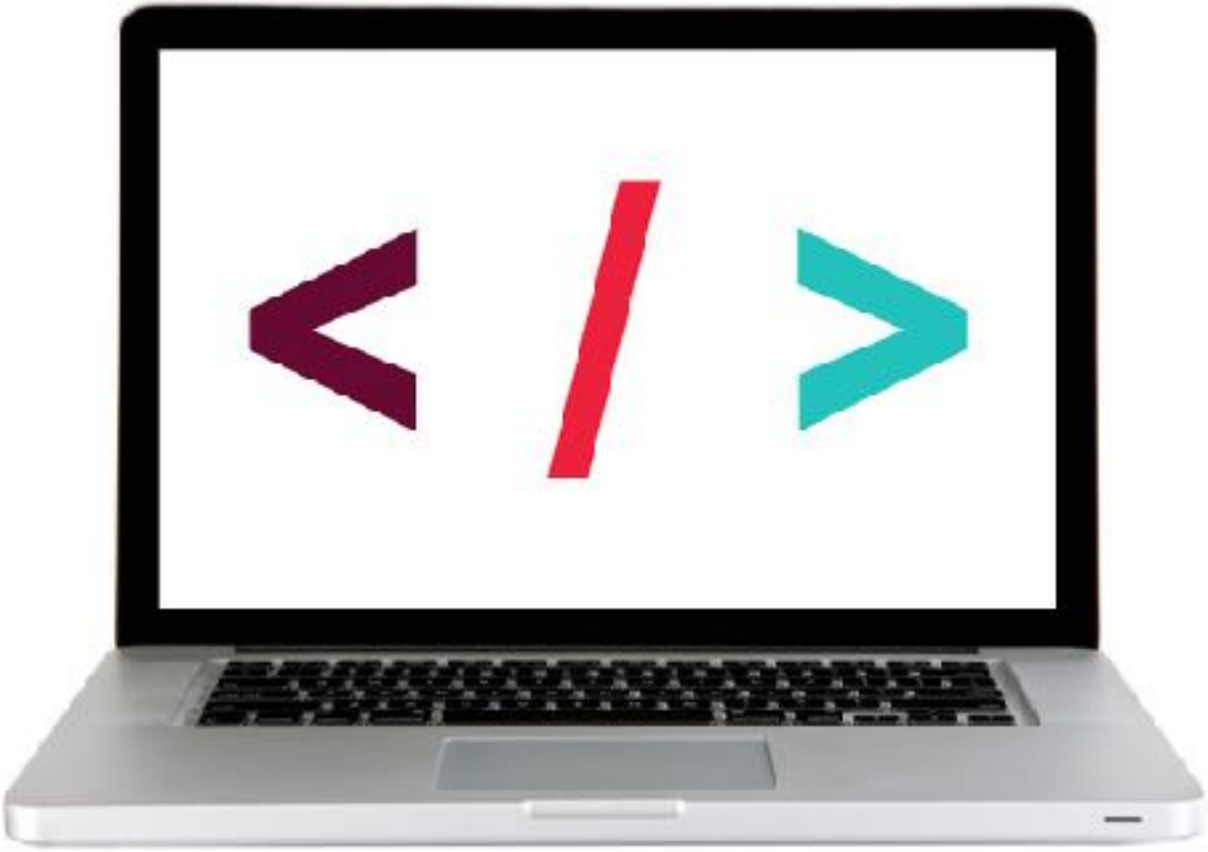
implicit iteration

```
$(‘li’).removeClass(‘current’);
```

applying any method to a jQuery collection iterates through each element!

✓ less code = best practice!

ADVANCED JQUERY



LET'S TAKE A CLOSER LOOK

EXERCISE - IMPLICIT ITERATION

EXERCISE

OBJECTIVES

- Use chaining to place methods on selectors.
- Use implicit iteration to update elements of a jQuery selection.

LOCATION

- `starter-code > 09-best-practices-exercise`

TIMING

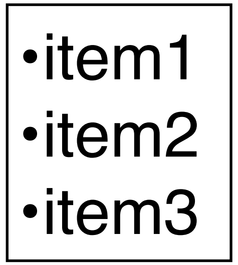
5 min

1. Open main.js in your editor and complete Items 1-3.
2. In your browser, reload index.html and verify that the functionality is unchanged.

EVENT DELEGATION

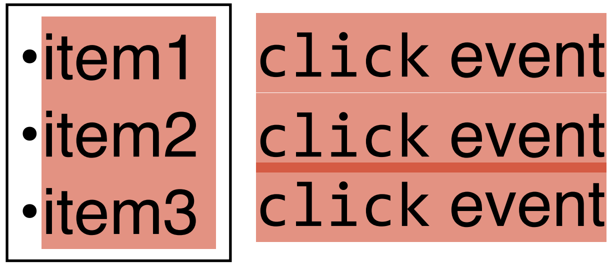
WITHOUT EVENT DELEGATION

1. load page

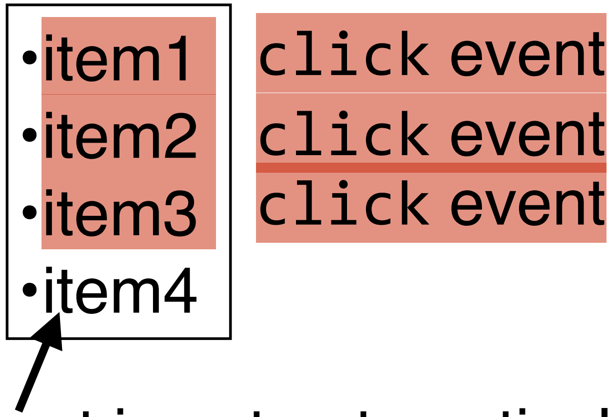


2. set event listener on list items

```
$(‘li’).on(‘click’,function(){
  addClass(‘selected’)
});
```



3. add a new list item



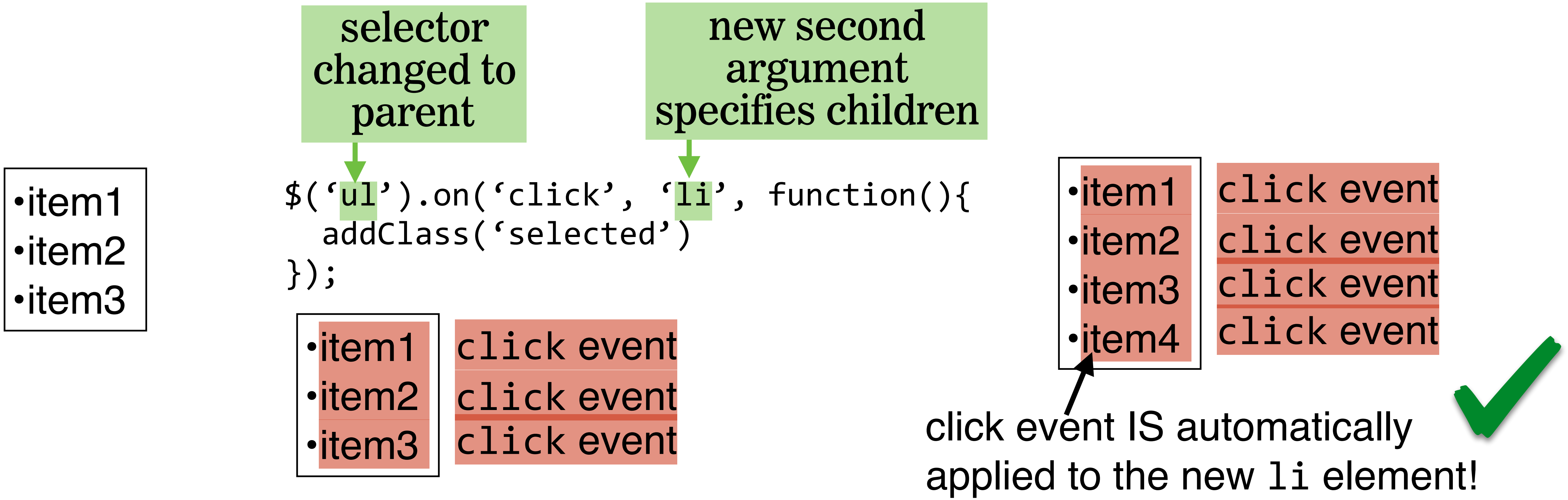
click event is not automatically applied to the new li element ✗

WITH EVENT DELEGATION

1. load page

2. set event listener on *parent of* list items

3. add a new list item



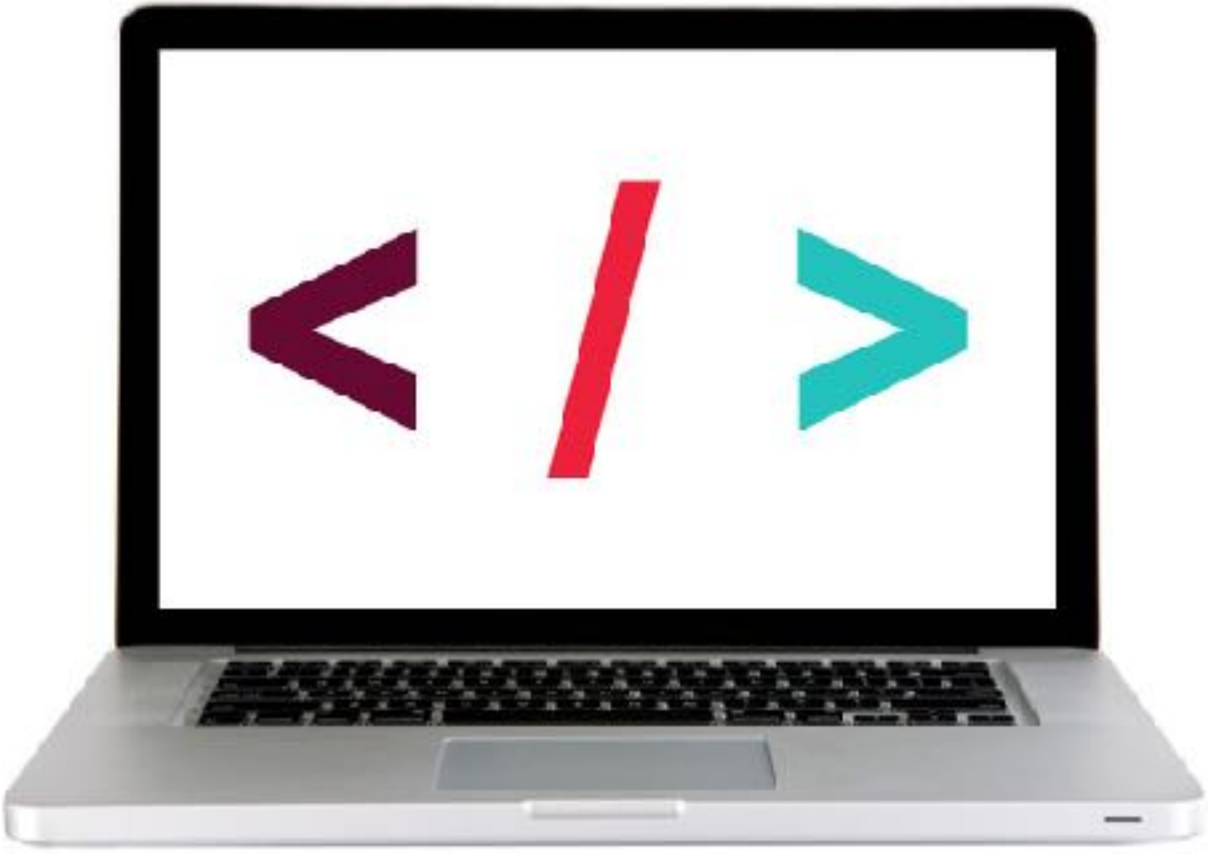
selector changed to parent

new second argument specifies children

```
$('ul').on('click', 'li', function(){
  addClass('selected')
});
```

click event IS automatically applied to the new `li` element!

ADVANCED JQUERY



LET'S TAKE A CLOSER LOOK

EXERCISE - EVENT DELEGATION

EXERCISE

OBJECTIVE

- Use event delegation to manage dynamic content.

LOCATION

- `starter-code > 09-best-practices-exercise`

TIMING

10 min

1. Return to main.js in your editor and complete item 4.
2. In your browser, reload index.html and verify that when you add a new item to the list, its “cross off” link works.
3. BONUS 1: When the user mouses over each item, the item should turn grey. Don't use CSS hovering for this.
4. BONUS 2: Add another link, after each item, that allows you to delete the item.

AJAX & APIS

ACTIVITY

EXERCISE

TYPE OF EXERCISE

- Individual/Partner

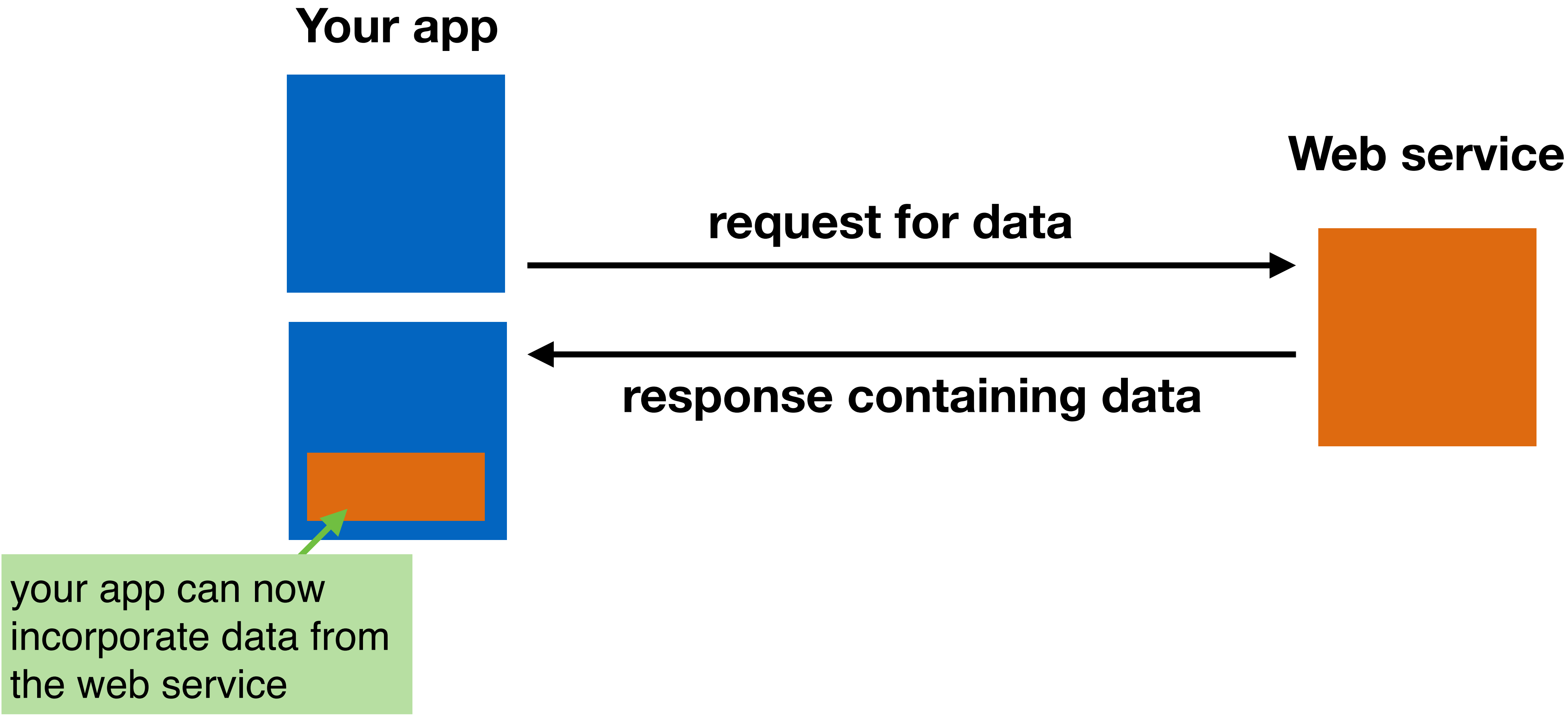
TIMING

3 min

1. Think about how you could use one or more sources of web data in an app.
2. Write a description or sketch a schematic of your app on your desk.

APIs

WEB SERVICES



Your app

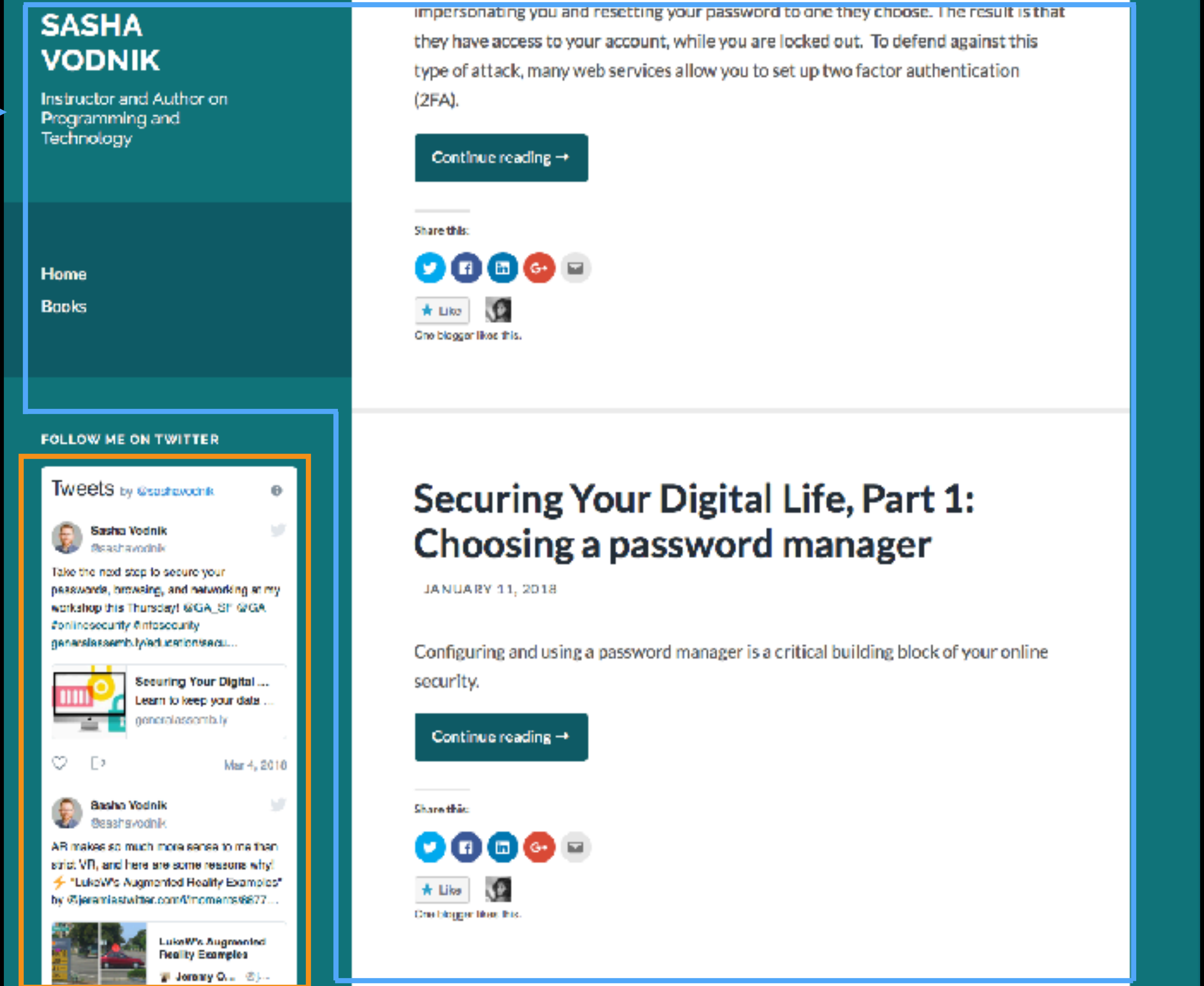
Web service

request for data

response containing data

your app can now incorporate data from the web service my website content

Content from Twitter added using Twitter API



SASHA VODNIK

Instructor and Author on Programming and Technology

Home

Books

FOLLOW ME ON TWITTER

Tweets by @sashavodnik

Sasha Vodnik
@sashavodnik

Take the next step to secure your passwords, browsing, and networking at my workshop this Thursday! @GA_SF @GA #onlinesecurity #infosecurity generalassemb.ly/education/secu…

Securing Your Digital …
Learn to keep your data …
generalassemb.ly

Mar 4, 2018

Sasha Vodnik
@sashavodnik

AR makes so much more sense to me than strict VR, and here are some reasons why! ⚡ "LukeW's Augmented Reality Examples" by @jeremiashtwitter.com/i/moments/9677…

LukeW's Augmented Reality Examples
Jeremy Q…

impersonating you and resetting your password to one they choose. The result is that they have access to your account, while you are locked out. To defend against this type of attack, many web services allow you to set up two factor authentication (2FA).

Continue reading →

Share this:

★ Like

One blogger likes this.

Securing Your Digital Life, Part 1: Choosing a password manager

JANUARY 11, 2018

Configuring and using a password manager is a critical building block of your online security.

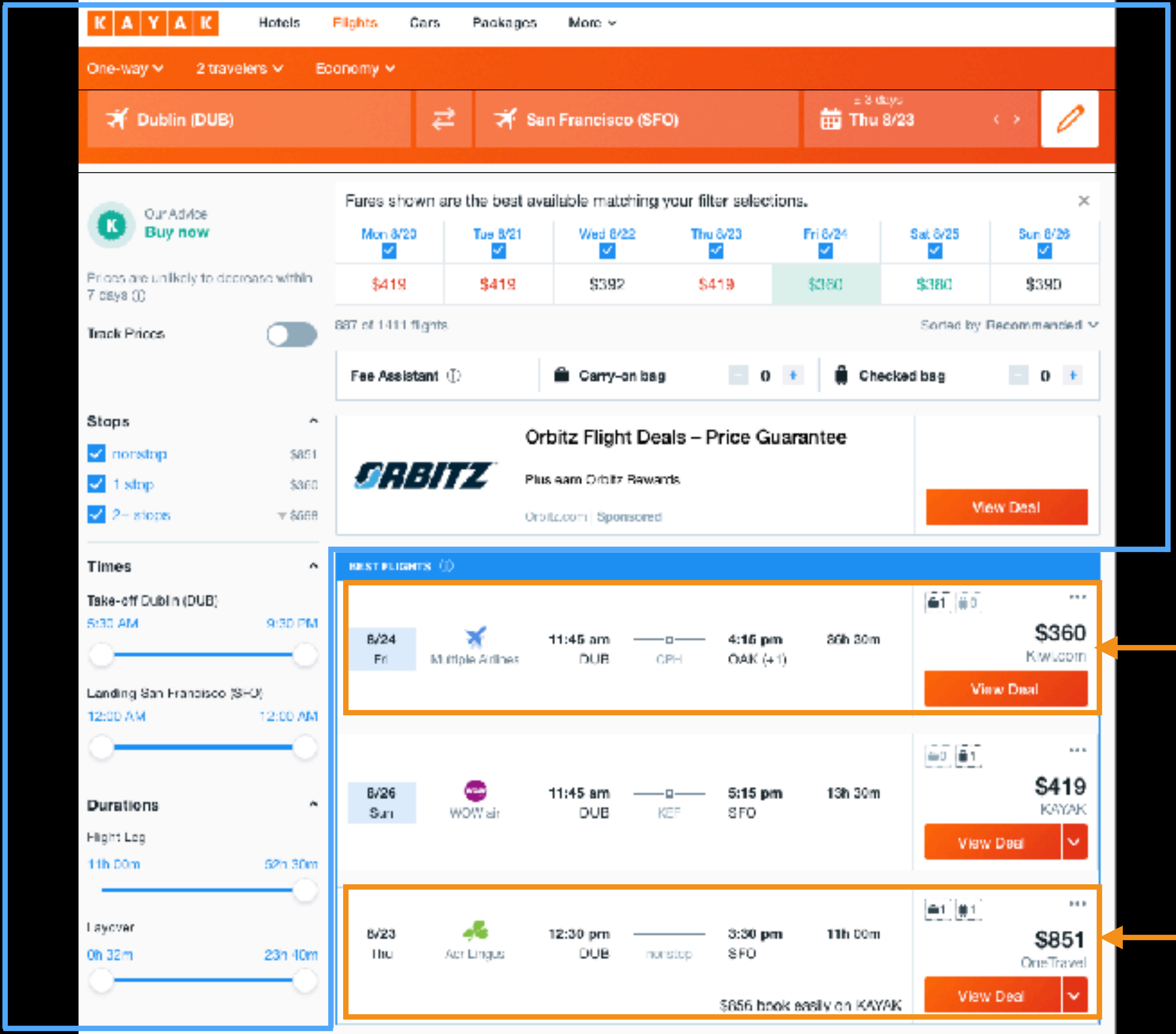
Continue reading →

Share this:

★ Like

One blogger likes this.

Kayak website content



KAYAK Hotels Flights Cars Packages More

One-way 2 travelers Economy

Dublin (DUB) ⇄ San Francisco (SFO) ± 3 days Thu 8/23

Our Advice
Buy now

Prices are unlikely to decrease within 7 days

Track Prices

Fares shown are the best available matching your filter selections. ×

Mon 8/20 ☑	Tue 8/21 ☑	Wed 8/22 ☑	Thu 8/23 ☑	Fri 8/24 ☑	Sat 8/25 ☑	Sun 8/26 ☑
$419	$419	$392	$419	$360	$380	$390

887 of 1411 flights

Sorted by Recommended

Fee Assistant ⓘ Carry-on bag − 0 + Checked bag − 0 +

ORBITZ

Orbitz Flight Deals – Price Guarantee

Plus earn Orbitz Rewards

Orbitz.com | Sponsored

View Deal

Stops

☑ nonstop $851

☑ 1 stop $360

☑ 2+ stops $668

Times

Take-off Dublin (DUB)

5:30 AM 9:30 PM

Landing San Francisco (SFO)

12:00 AM 12:00 AM

Durations

Flight Leg

11h 00m 52h 30m

Layover

0h 32m 23h 40m

BEST FLIGHTS ⓘ

8/24 Fri Multiple Airlines 11:45 am DUB CPH 4:15 pm OAK (+1) 36h 30m 1 0 $360 Kiwi.com View Deal

8/26 Sun WOW air 11:45 am DUB KEF 5:15 pm SFO 13h 30m 0 1 $419 KAYAK View Deal

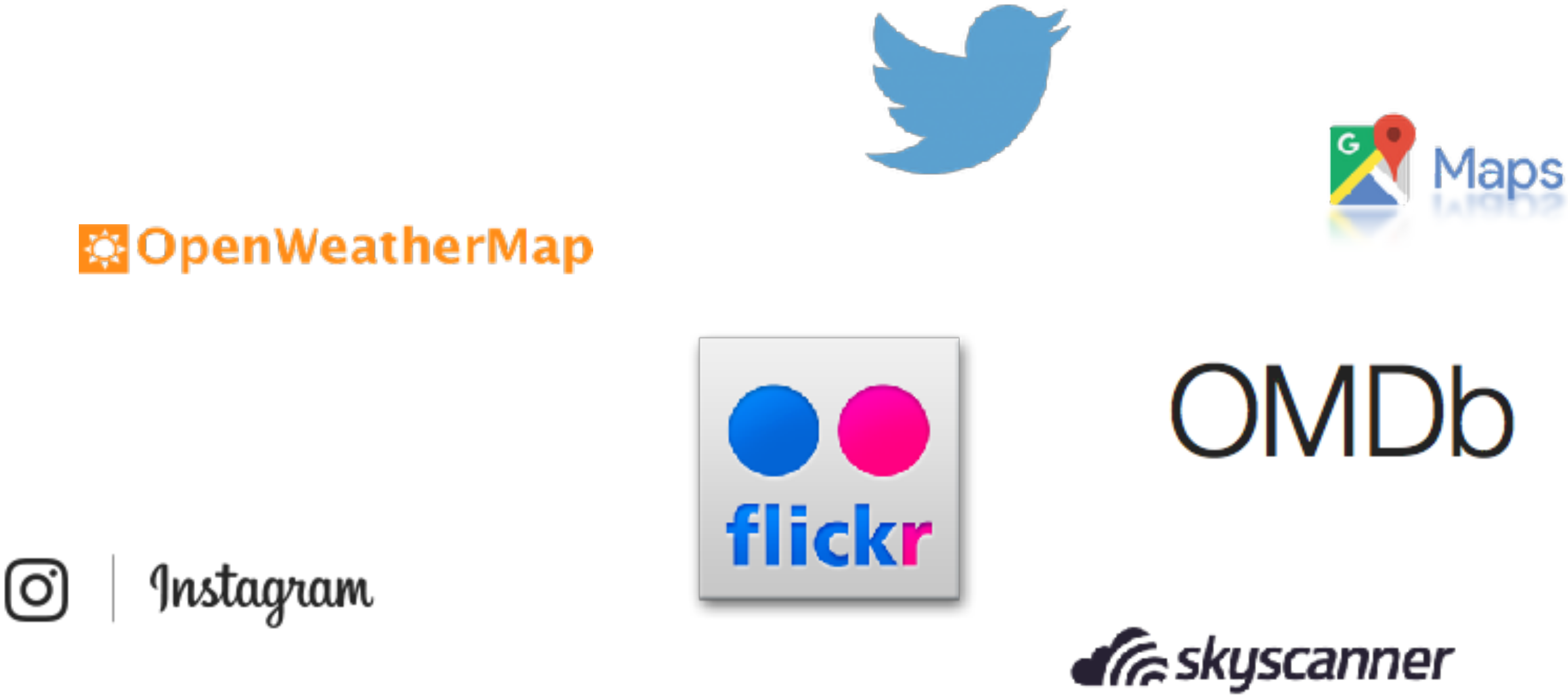
8/23 Thu Aer Lingus 12:30 pm DUB nonstop 3:30 pm SFO 11h 00m 1 1 $851 OneTravel View Deal

$856 book easily on KAYAK

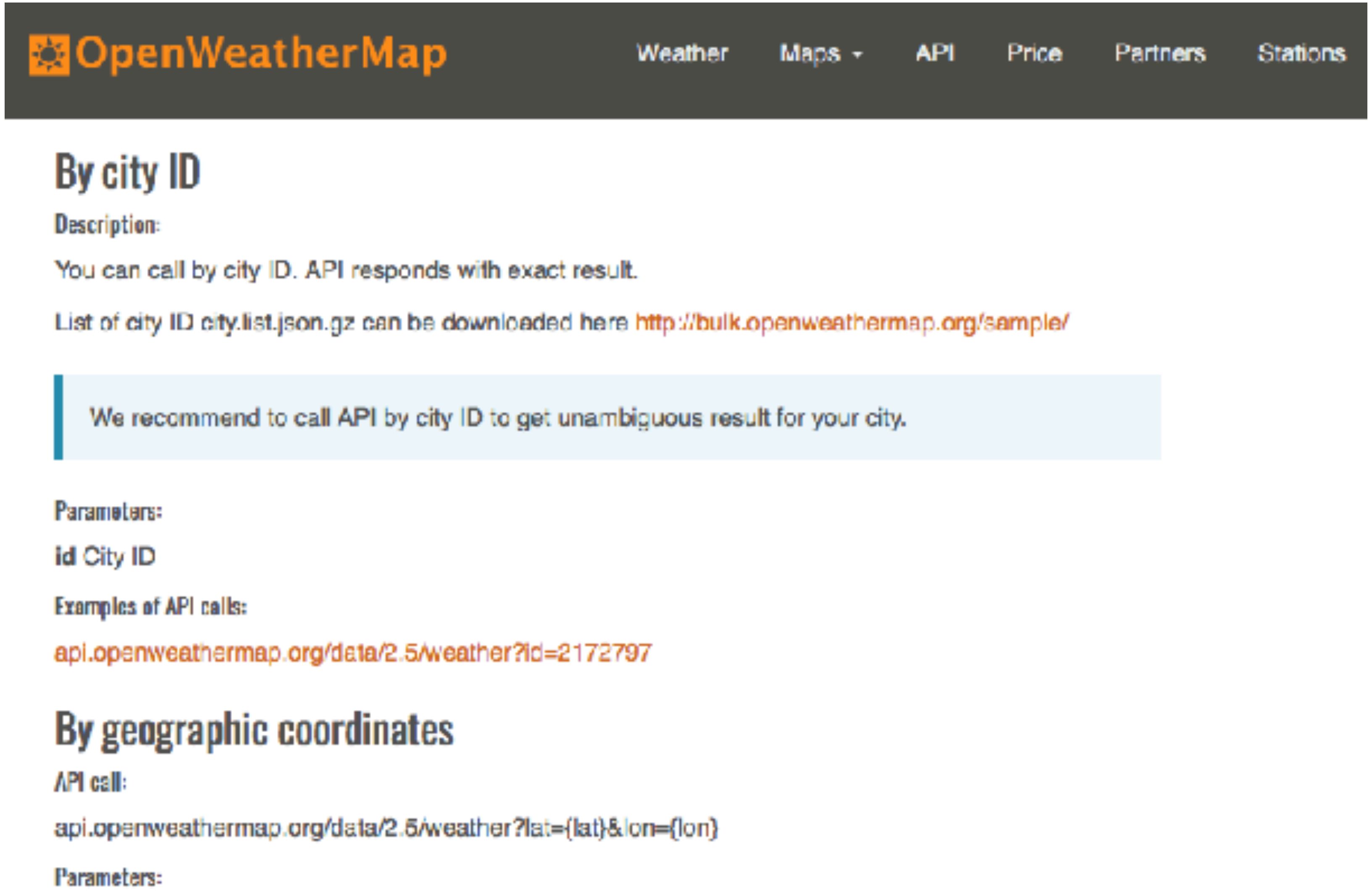
Content from kiwi.com using API

Content from OneTravel using API

WEB SERVICES



API = application programming interface



By city ID

Description:

You can call by city ID. API responds with exact result.

List of city ID city.list.json.gz can be downloaded here http://bulk.openweathermap.org/sample/

> We recommend to call API by city ID to get unambiguous result for your city.

Parameters:

id City ID

Examples of API calls:

api.openweathermap.org/data/2.5/weather?id=2172797

By geographic coordinates

API call:

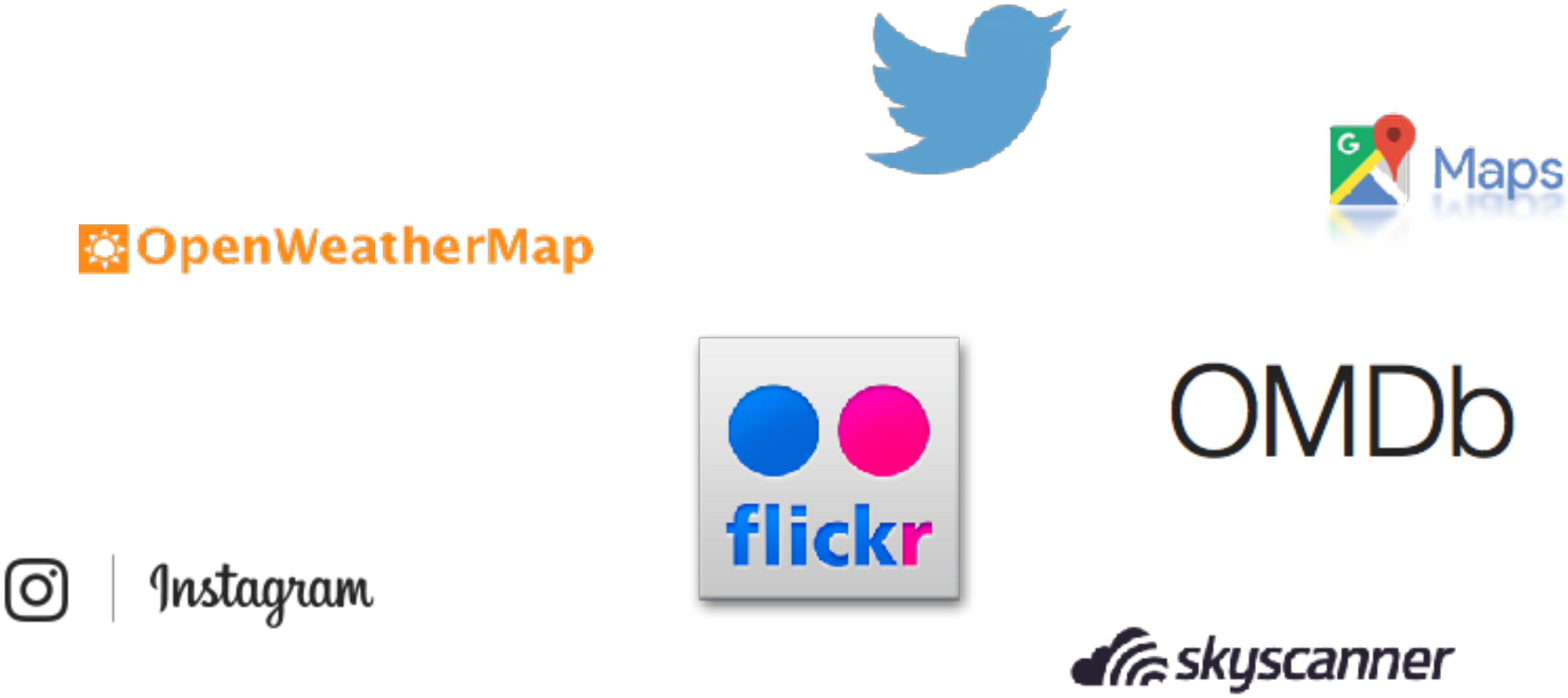
api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}

Parameters:

APIS IN THE REAL WORLD

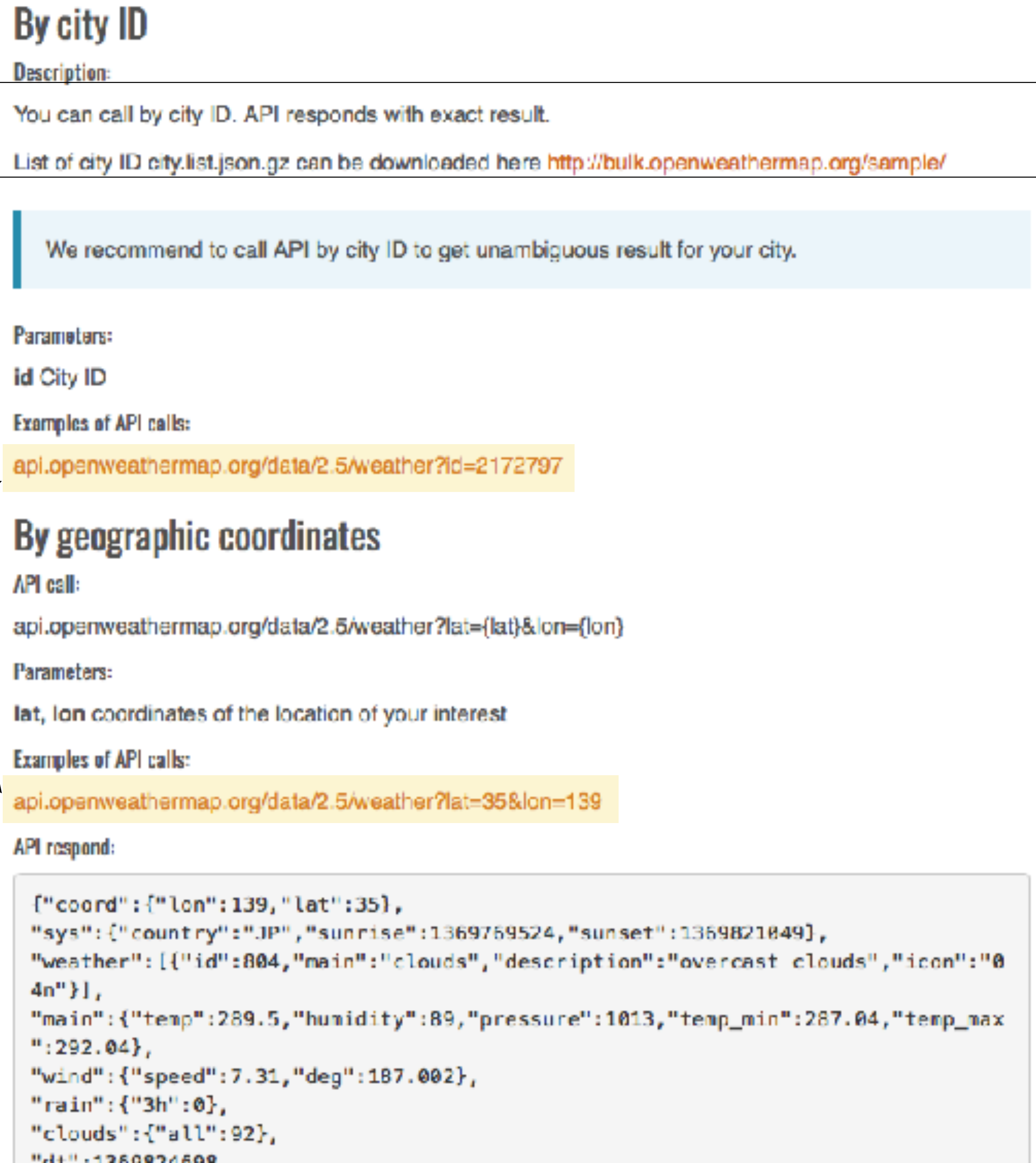
- Most APIs are unique, like separate languages
- APIs for
  - devices (iPhone)
  - operating systems (macOS)
  - JavaScript libraries (jQuery API)
  - services (Slack)

WEB SERVICES



ENDPOINTS

- Addresses (URLs) that return data (JSON) instead of markup (HTML)



By city ID

Description:

You can call by city ID. API responds with exact result.

List of city ID city.list.json.gz can be downloaded here http://bulk.openweathermap.org/sample/

> We recommend to call API by city ID to get unambiguous result for your city.

Parameters:

id City ID

Examples of API calls:

api.openweathermap.org/data/2.5/weather?id=2172797

By geographic coordinates

API call:

api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}

Parameters:

lat, lon coordinates of the location of your interest

Examples of API calls:

api.openweathermap.org/data/2.5/weather?lat=35&lon=139

API respond:

```
{"coord":{"lon":139,"lat":35},
"sys":{"country":"JP","sunrise":1369769524,"sunset":1369821049},
"weather":[{"id":804,"main":"clouds","description":"overcast clouds","icon":"0
4n"}],
"main":{"temp":289.5,"humidity":89,"pressure":1013,"temp_min":287.04,"temp_max
":292.04},
"wind":{"speed":7.31,"deg":187.002},
"rain":{"3h":0},
"clouds":{"all":92},
"dt":1369824698
```

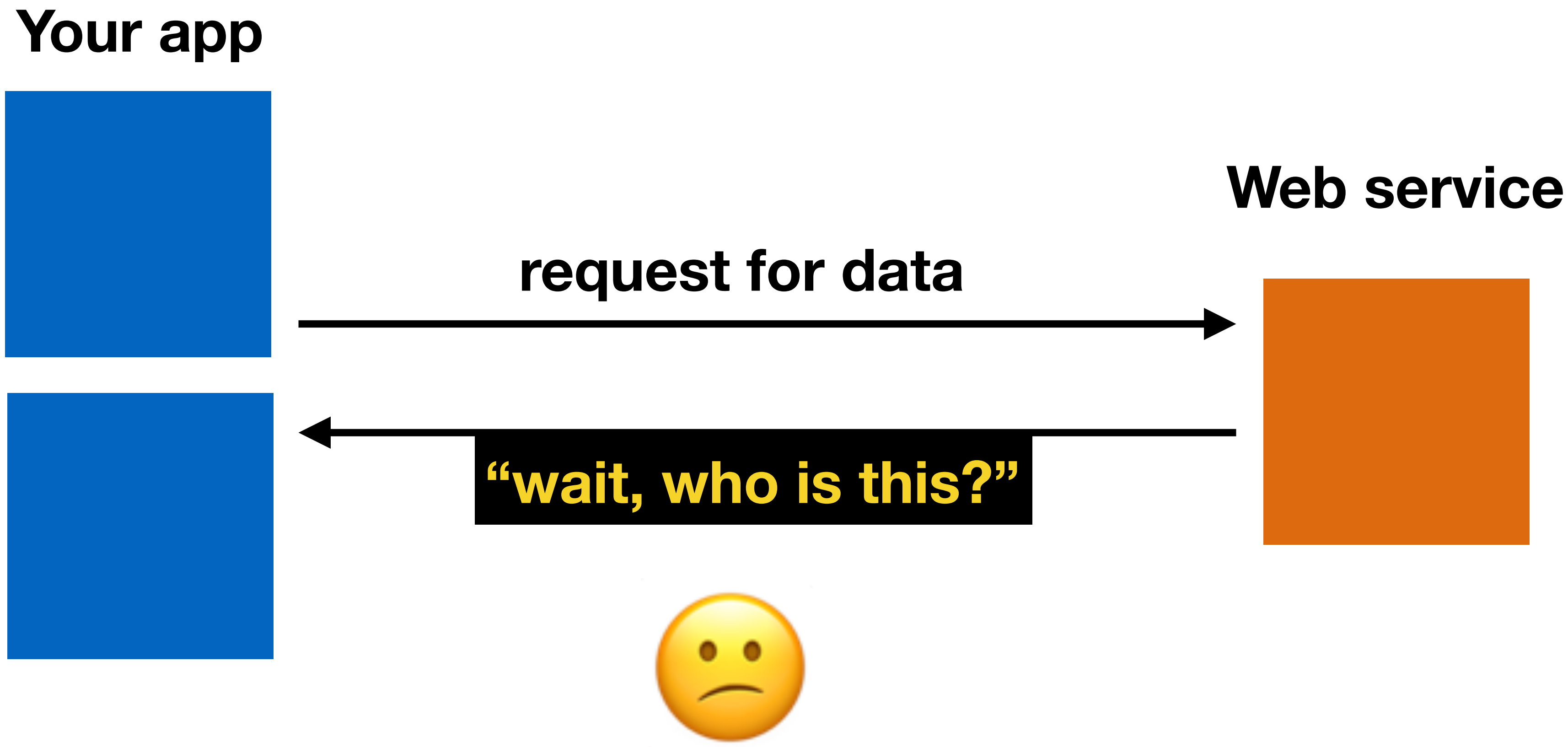
WHAT WE NEED TO KNOW TO USE AN API

TERMS OF SERVICE

HOW TO MAKE A REQUEST

HOW TO UNDERSTAND RESPONSE

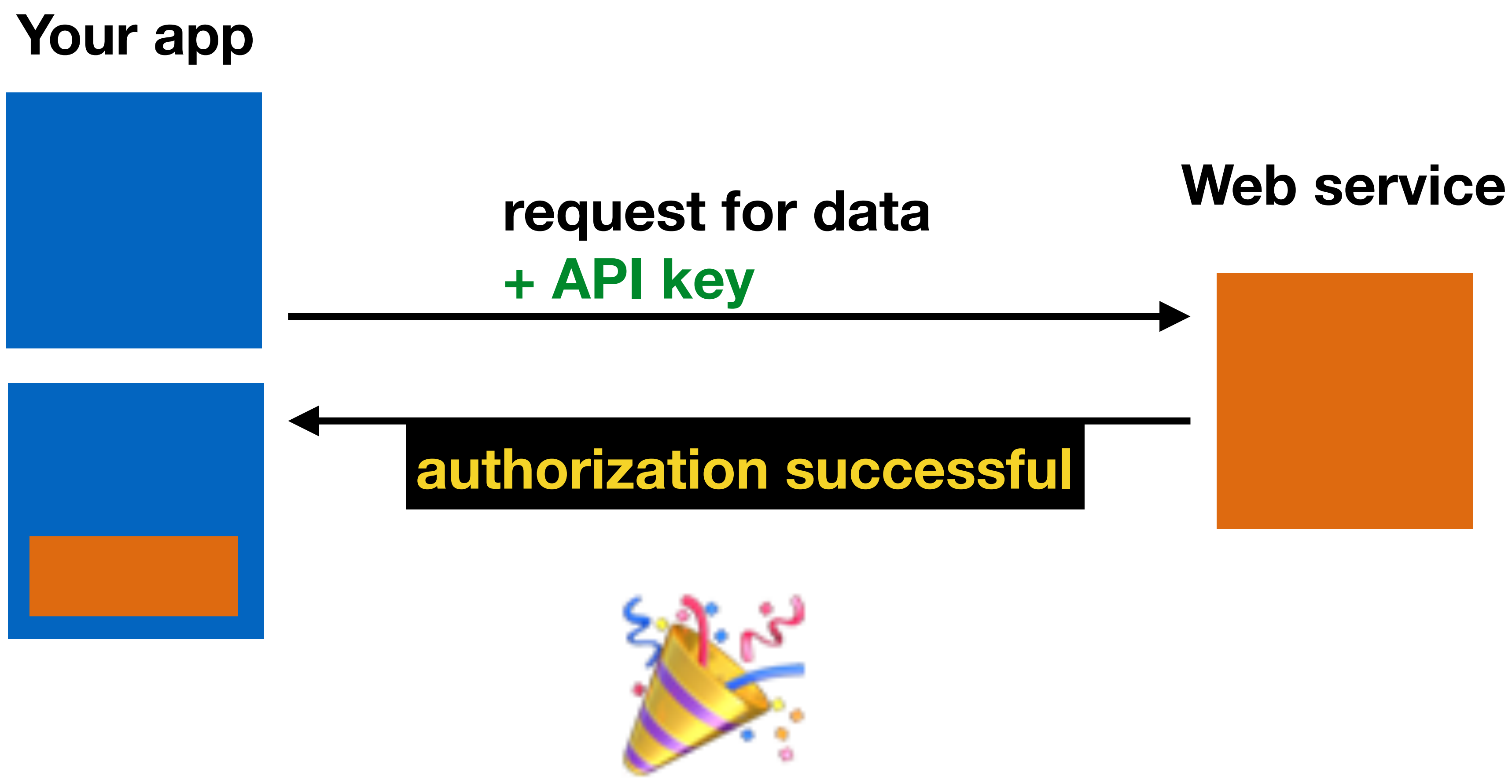
AN API MIGHT REQUIRE AUTHENTICATION



API KEY

- unguessable character string
- connects your requests to your account

API REQUEST WITH AUTHENTICATION



Your app

request for data

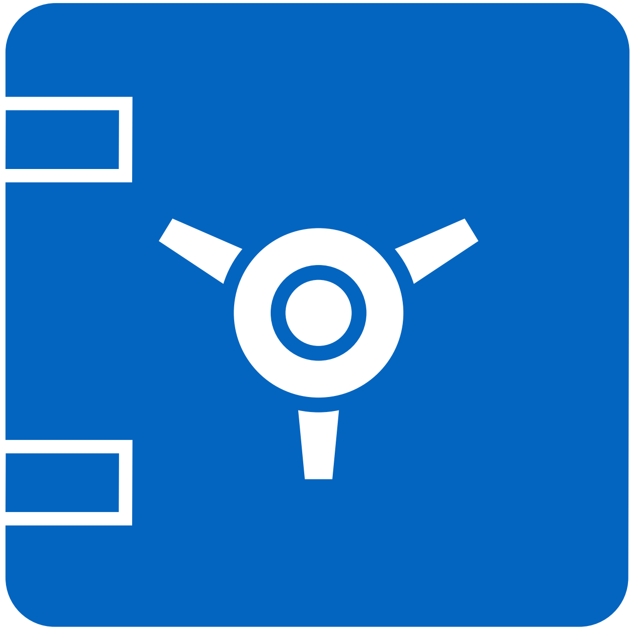
+ API key

Web service

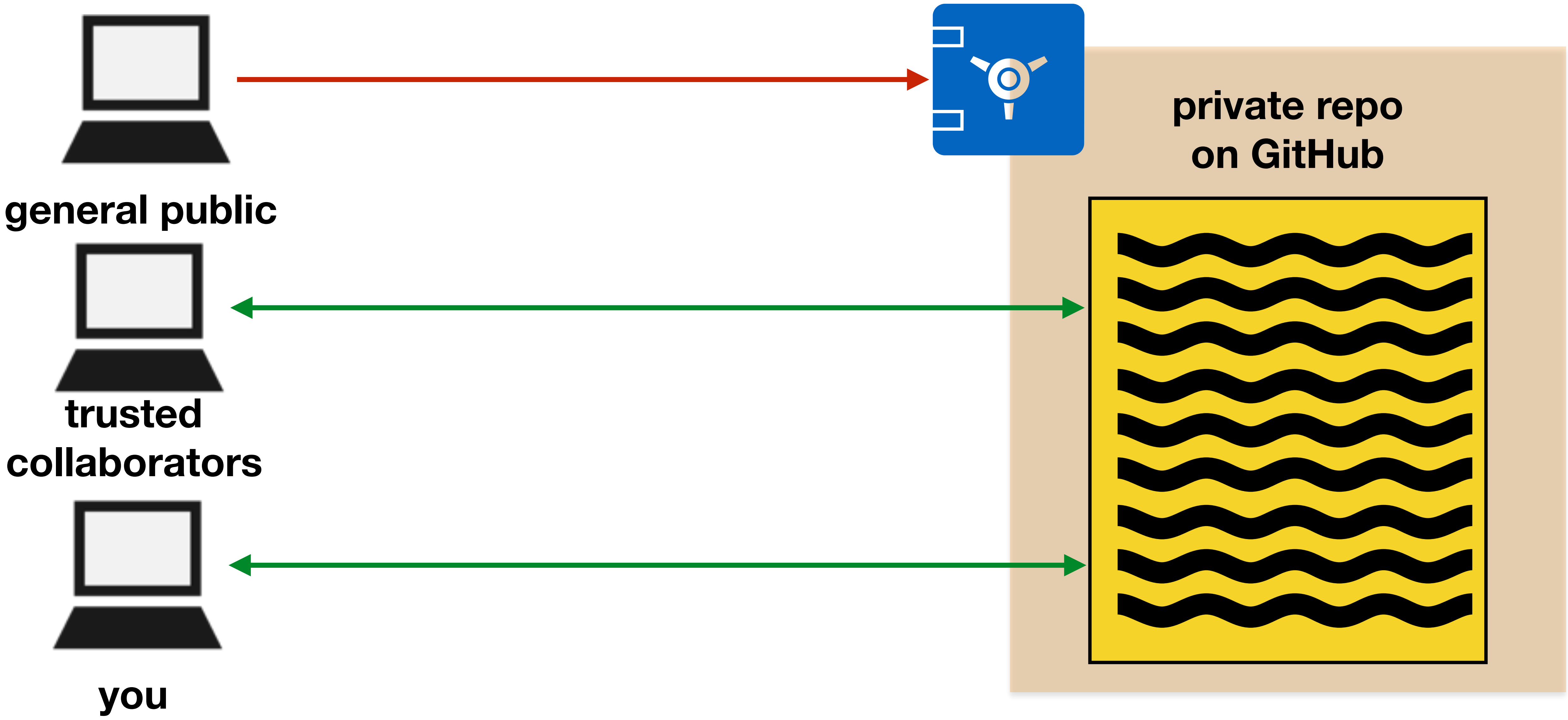
authorization successful

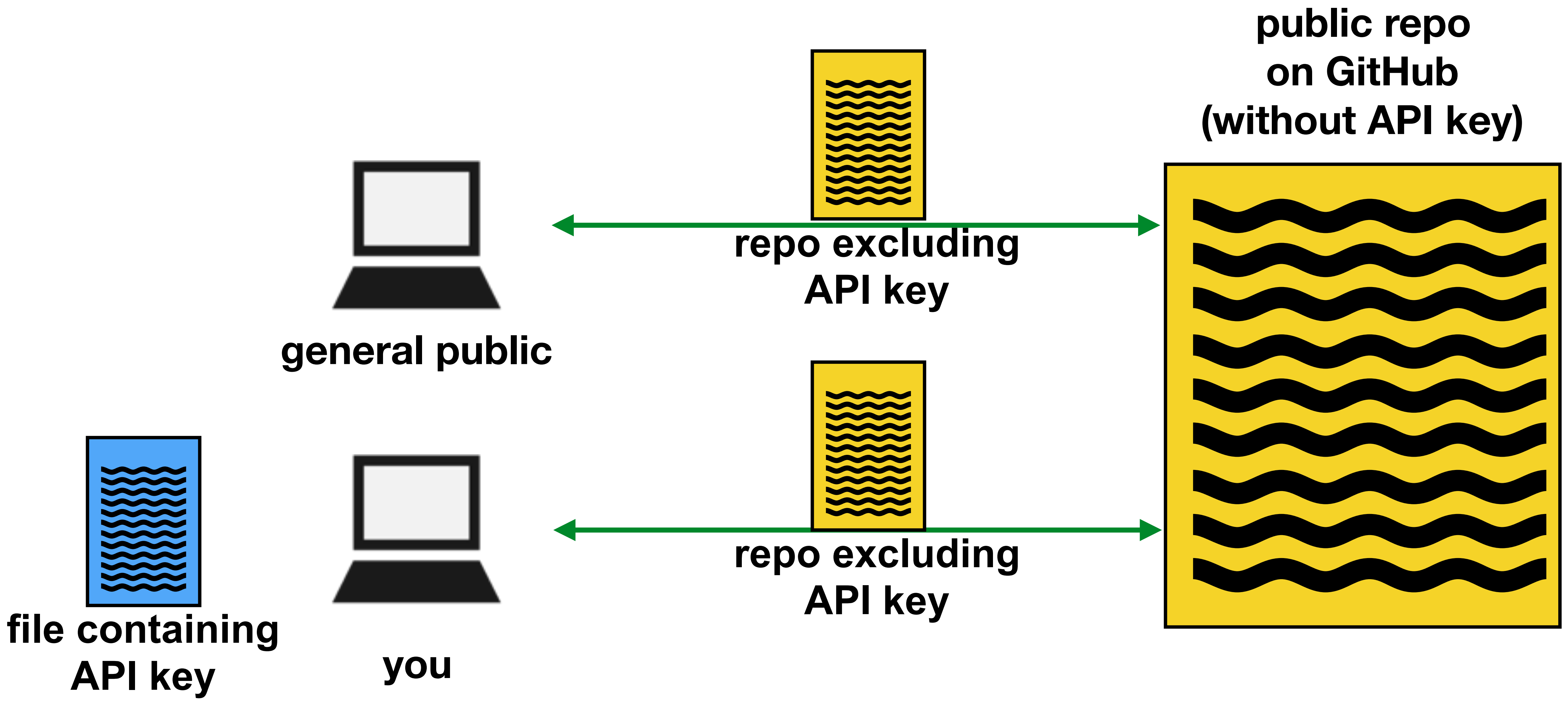
🎉

KEEP YOUR API CREDENTIALS PRIVATE



- Don’t post to a public code repo
- Don’t share with other developers outside of your organization

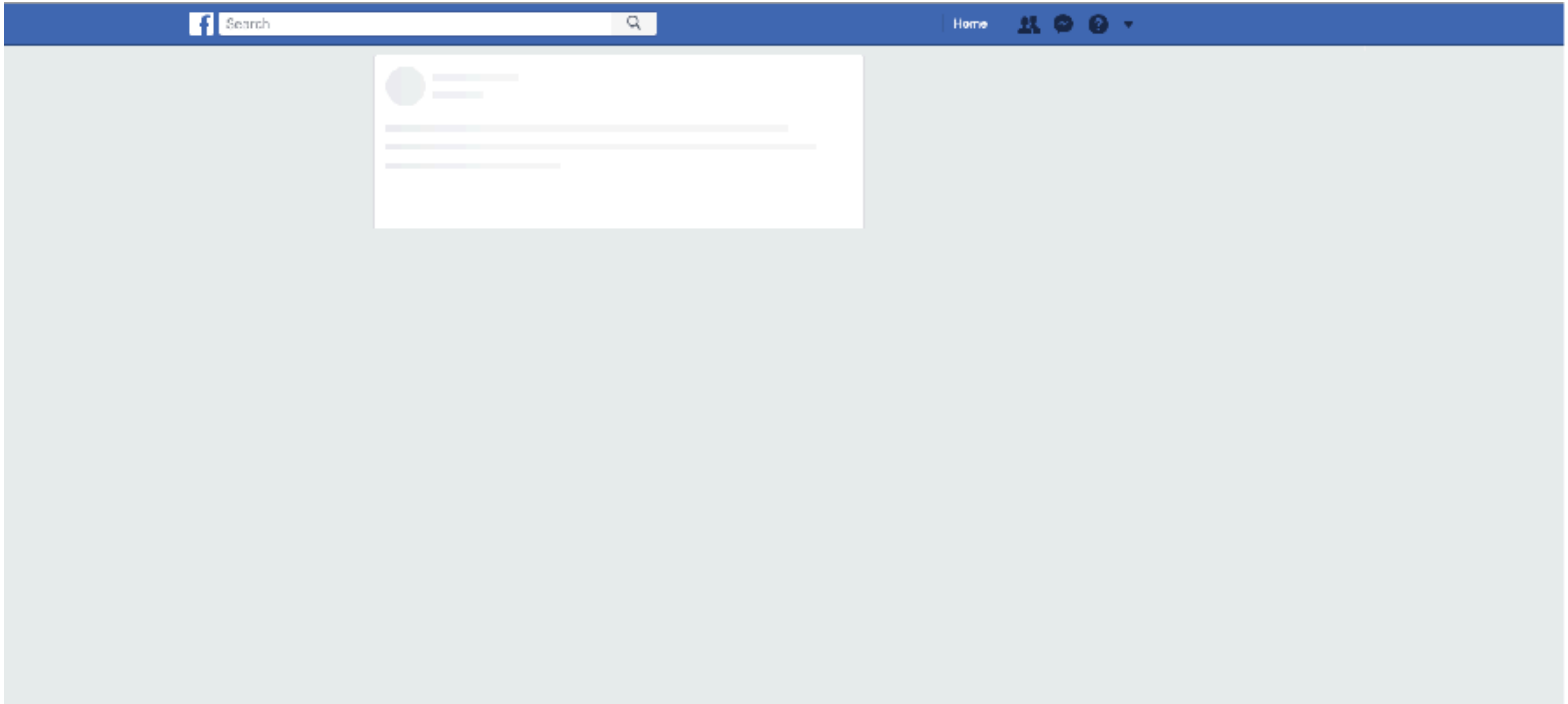

general public
trusted collaborators
you
private repo on GitHub


public repo
on GitHub
(without API key)
general public
repo excluding
API key
repo excluding
API key
file containing
API key
you

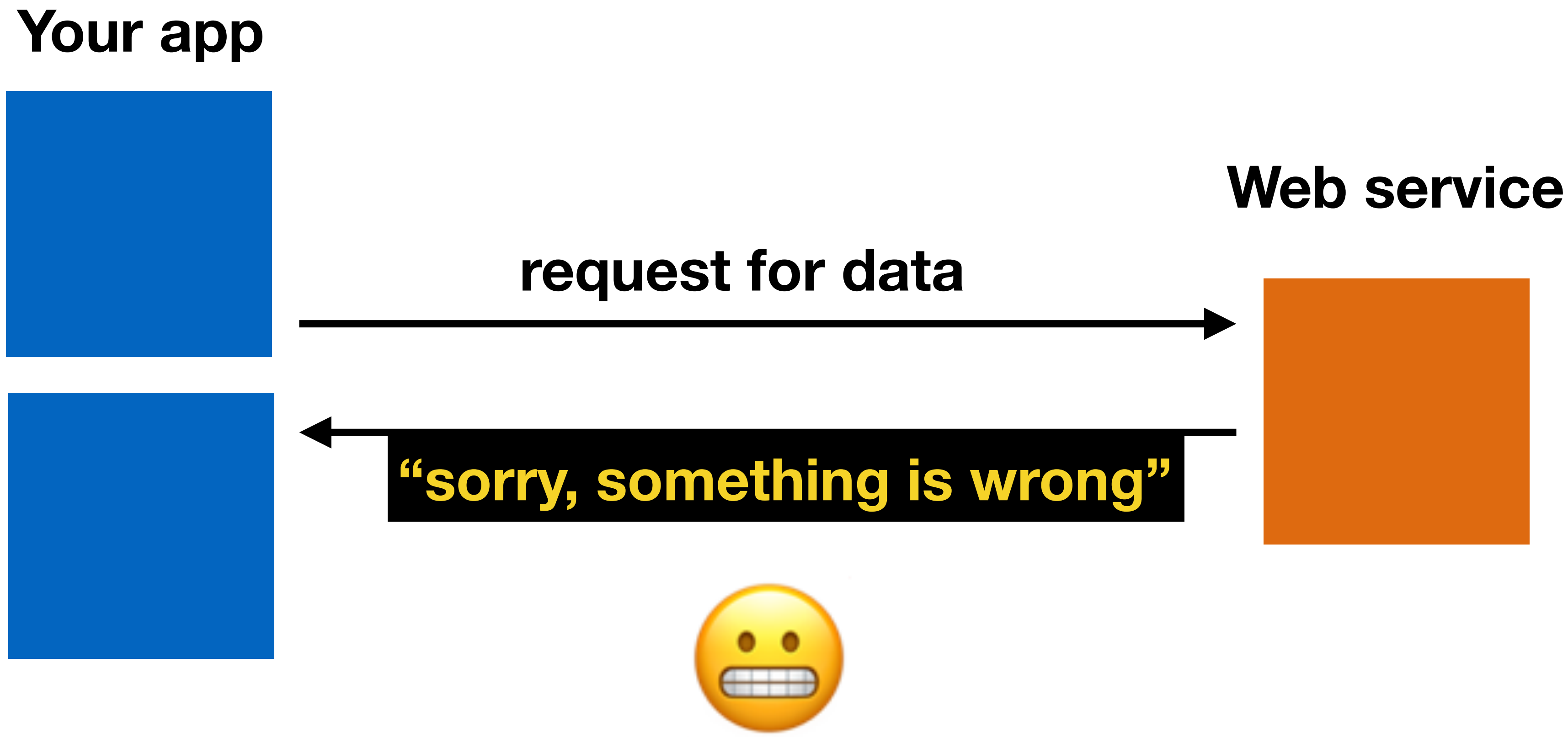
.GITIGNORE

```
js/keys.js
```

YOUR APP MIGHT EXPERIENCE A DELAYED RESPONSE

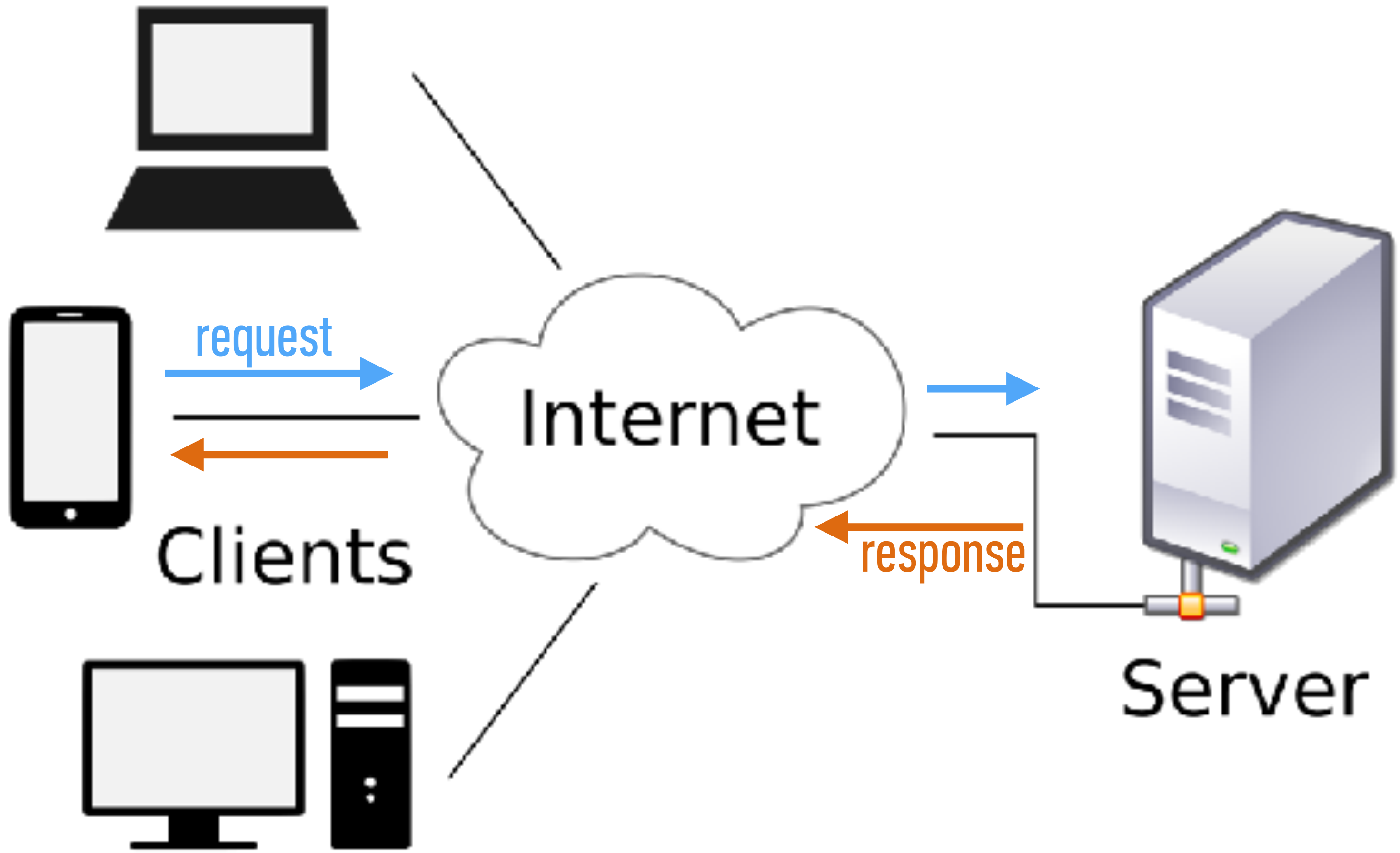


YOUR REQUEST MAY RESULT IN AN ERROR



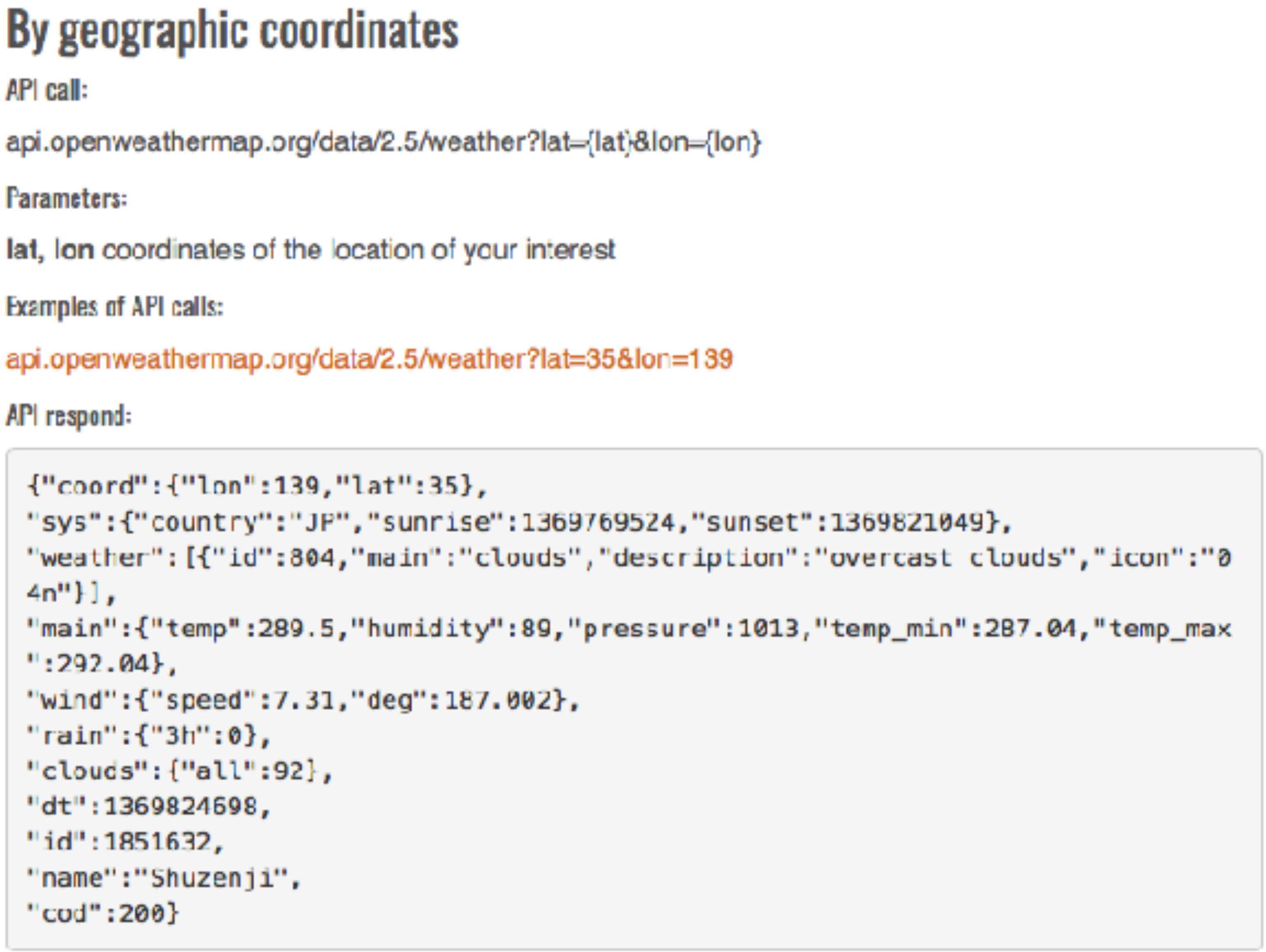
REST (representational state transfer)

- architectural style of web applications
- transfers a representation of the state of a resource between the server and the client



RESTful API

- adheres to REST architecture
- uses
  - a base URL
  - an Internet media type (such as JSON)
  - standard HTTP methods



By geographic coordinates

API call:

api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}

Parameters:

lat, lon coordinates of the location of your interest

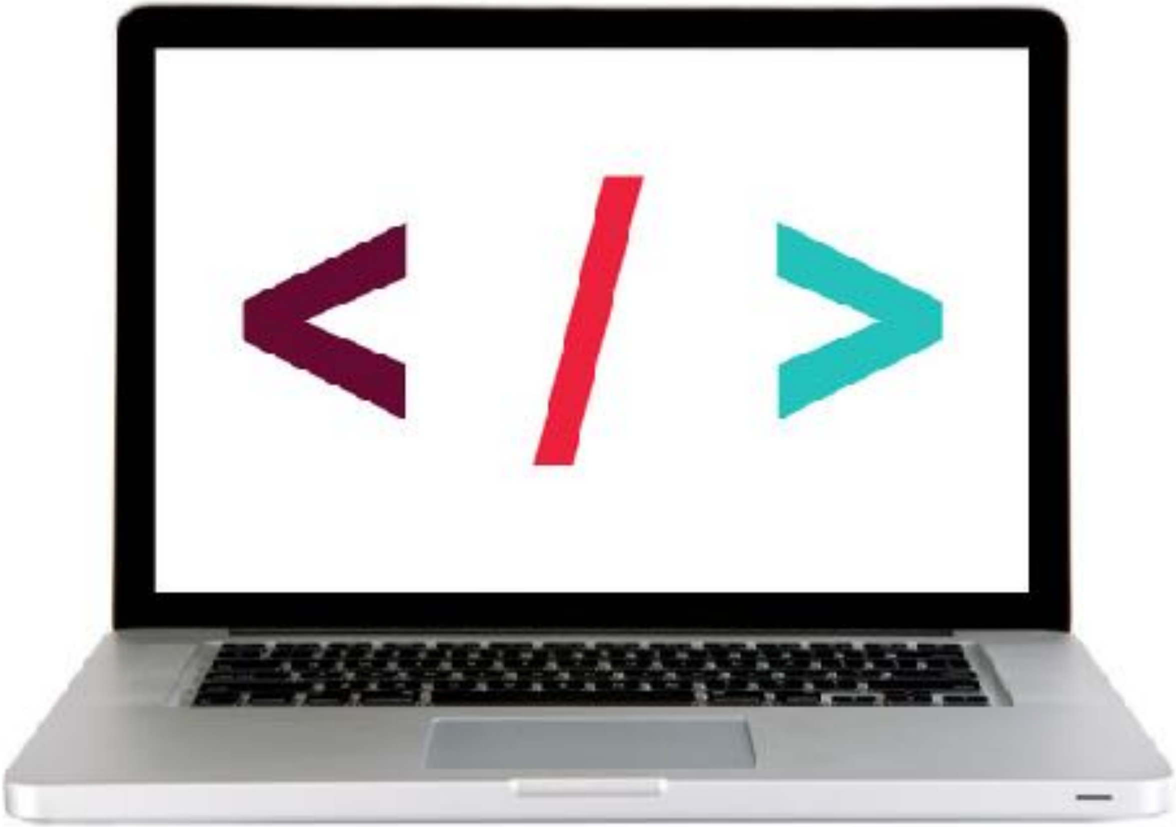
Examples of API calls:

api.openweathermap.org/data/2.5/weather?lat=35&lon=139

API respond:

```
{"coord":{"lon":139,"lat":35},
"sys":{"country":"JP","sunrise":1369769524,"sunset":1369821049},
"weather":[{"id":804,"main":"clouds","description":"overcast clouds","icon":"0
4n"}],
"main":{"temp":289.5,"humidity":89,"pressure":1013,"temp_min":287.04,"temp_max
":292.04},
"wind":{"speed":7.31,"deg":187.002},
"rain":{"3h":0},
"clouds":{"all":92},
"dt":1369824698,
"id":1851632,
"name":"Shuzenji",
"cod":200}
```

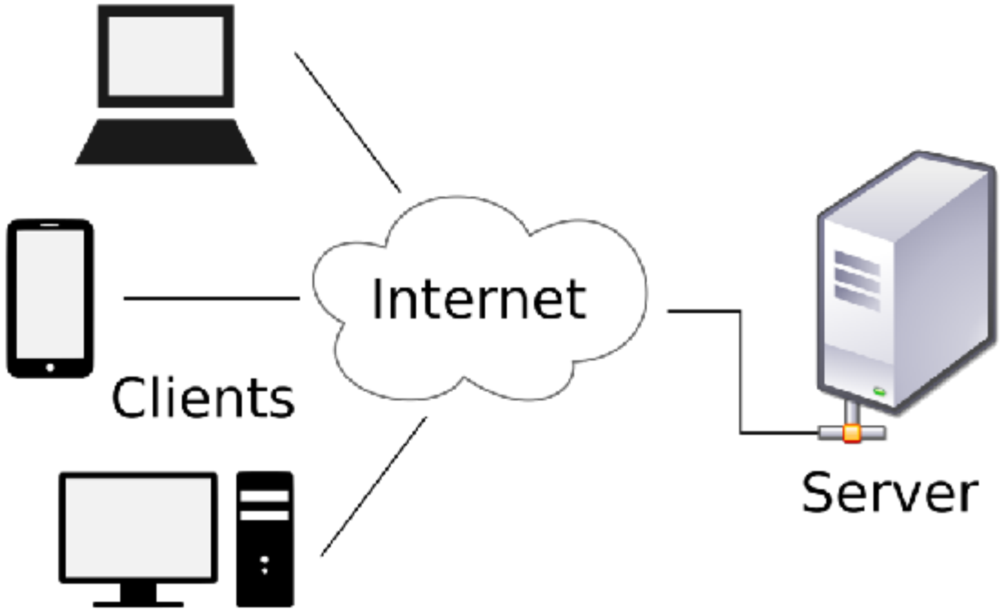
LET'S TAKE A CLOSER LOOK



HTTP

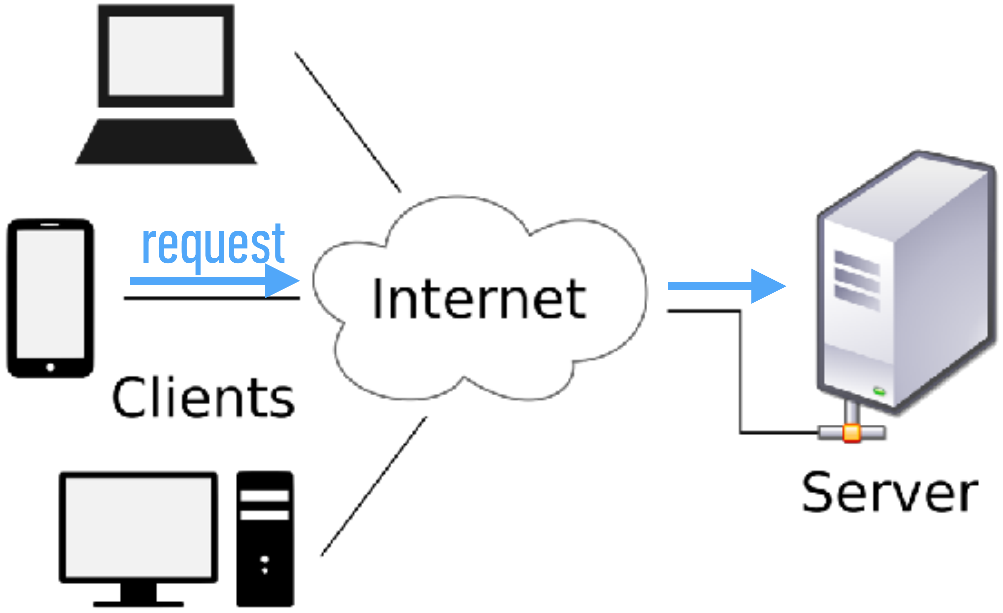
HTTP (hypertext transfer protocol)

- System of rules for how web pages are transmitted between computers
- Defines the format of messages passed between HTTP clients and HTTP servers



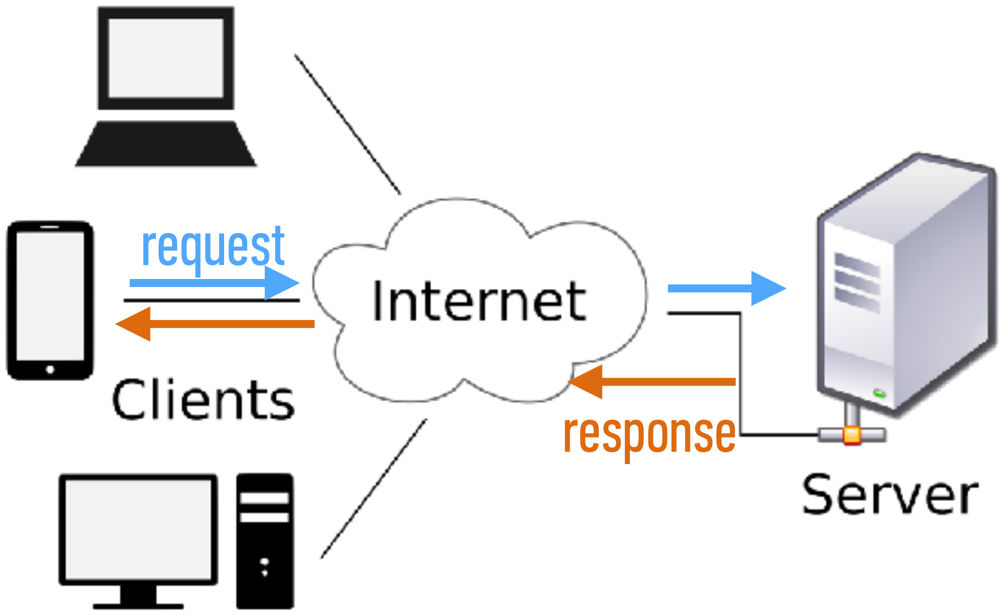
HTTP (hypertext transfer protocol)

- A client sends a **request** to a server.

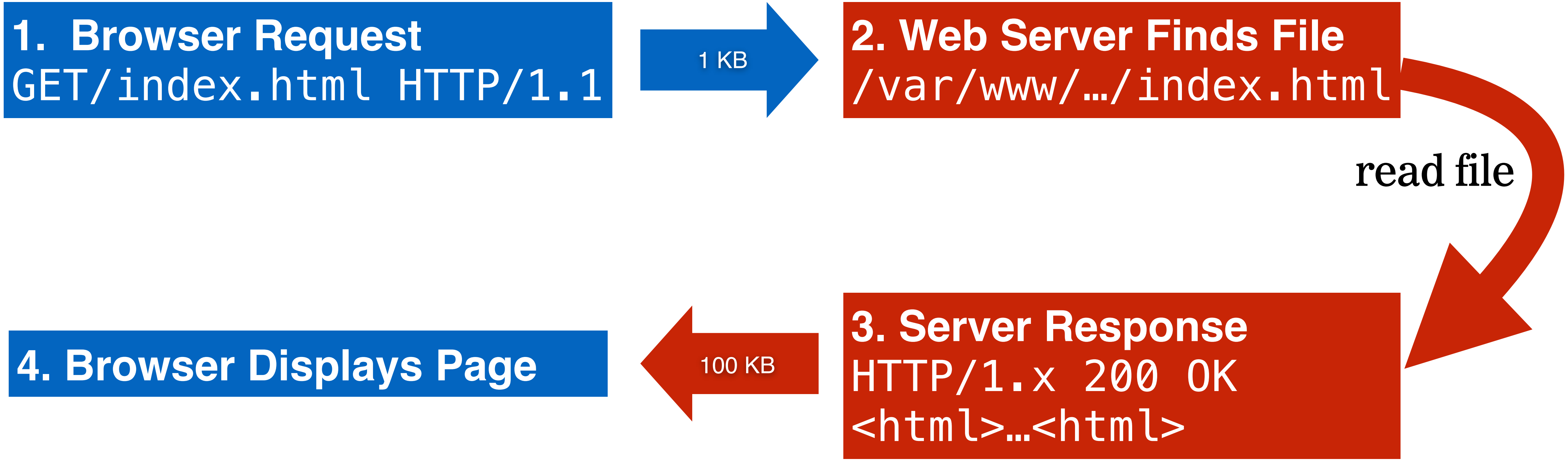


HTTP (hypertext transfer protocol)

- A server sends a **response** back to a client.



HTTP REQUEST AND RESPONSE



1. Browser Request
`GET/index.html HTTP/1.1`

2. Web Server Finds File
`/var/www/…/index.html`

3. Server Response
`HTTP/1.x 200 OK`
`<html>…<html>`

4. Browser Displays Page

HTTP (hypertext transfer protocol)

HTTP client

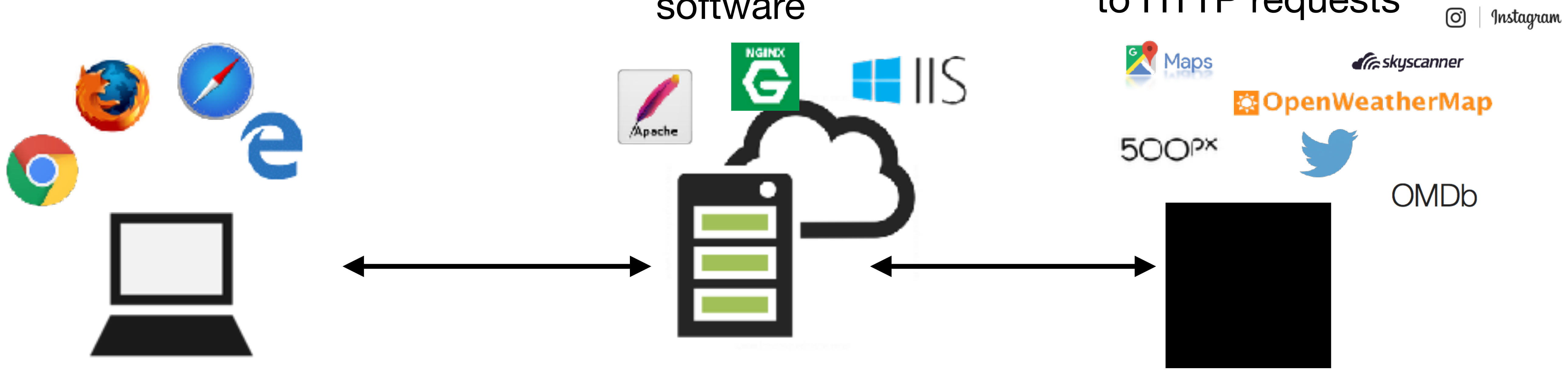
web browser

HTTP server

web server software

Web service

app that responds to HTTP requests

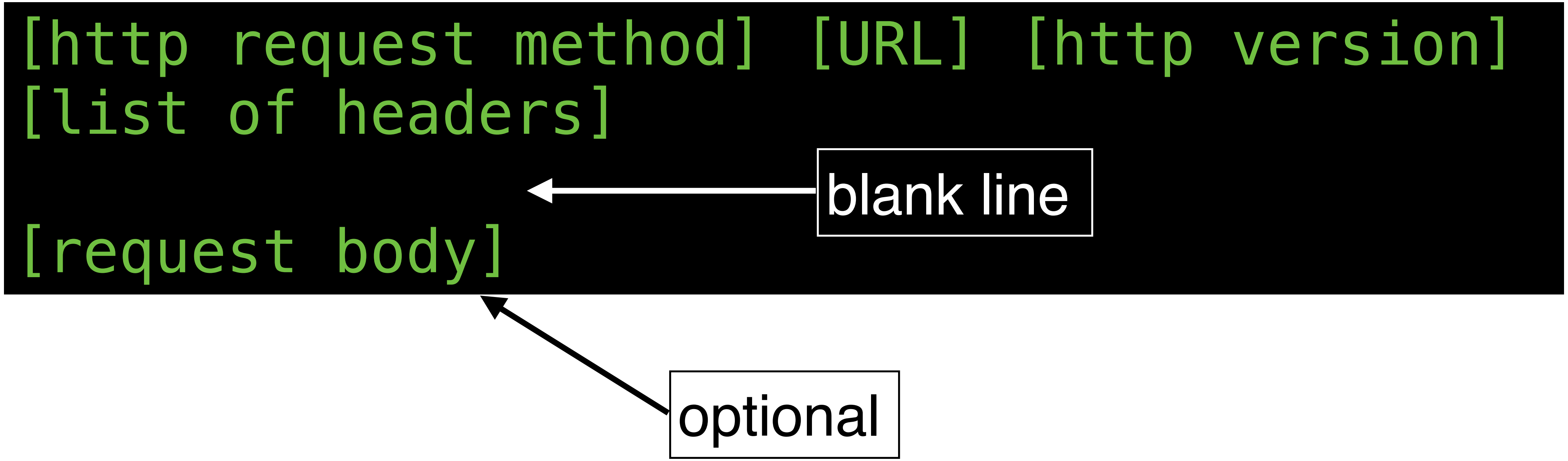


HTTP REQUESTS IN EVERYDAY LIFE

protocol host resource path query

```
https://www.domain.com/path/to/resource?a=b&x=y
```

HTTP REQUEST STRUCTURE



```
[http request method] [URL] [http version]
[list of headers]

[request body]
```

blank line

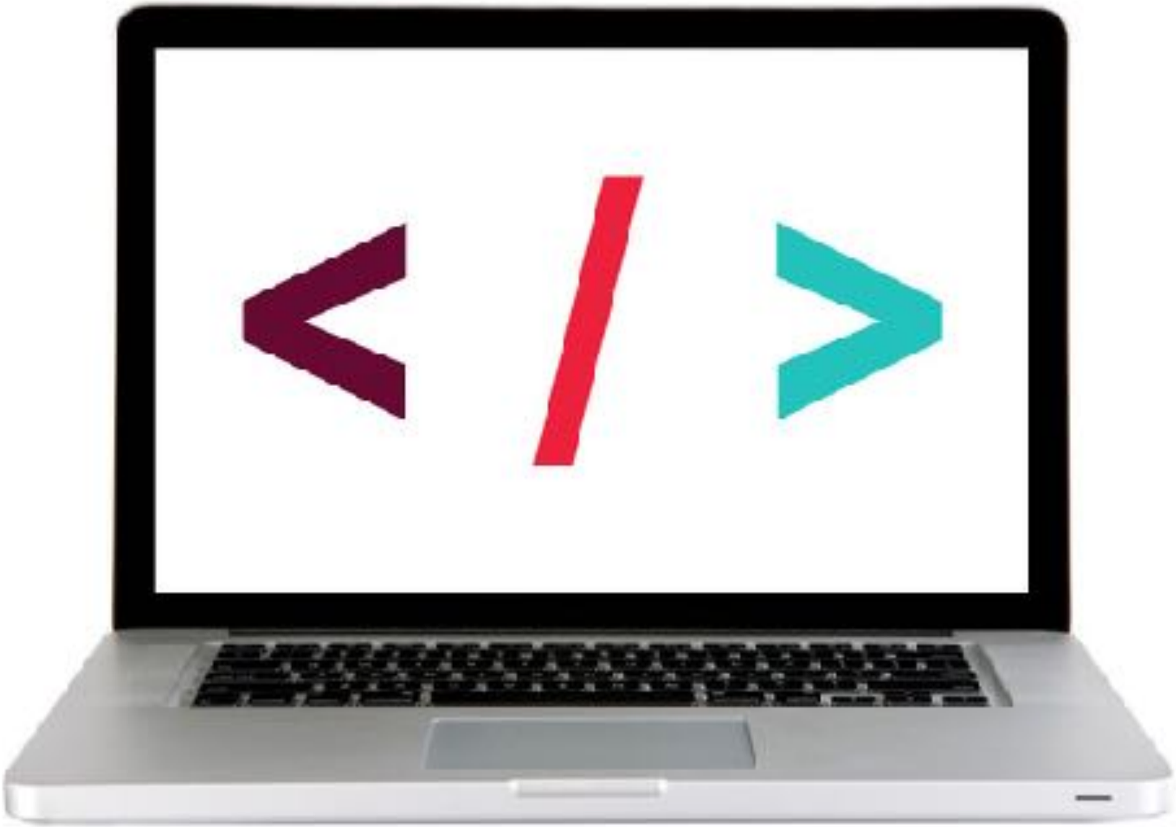
optional

HTTP REQUEST METHODS (“HTTP VERBS”)

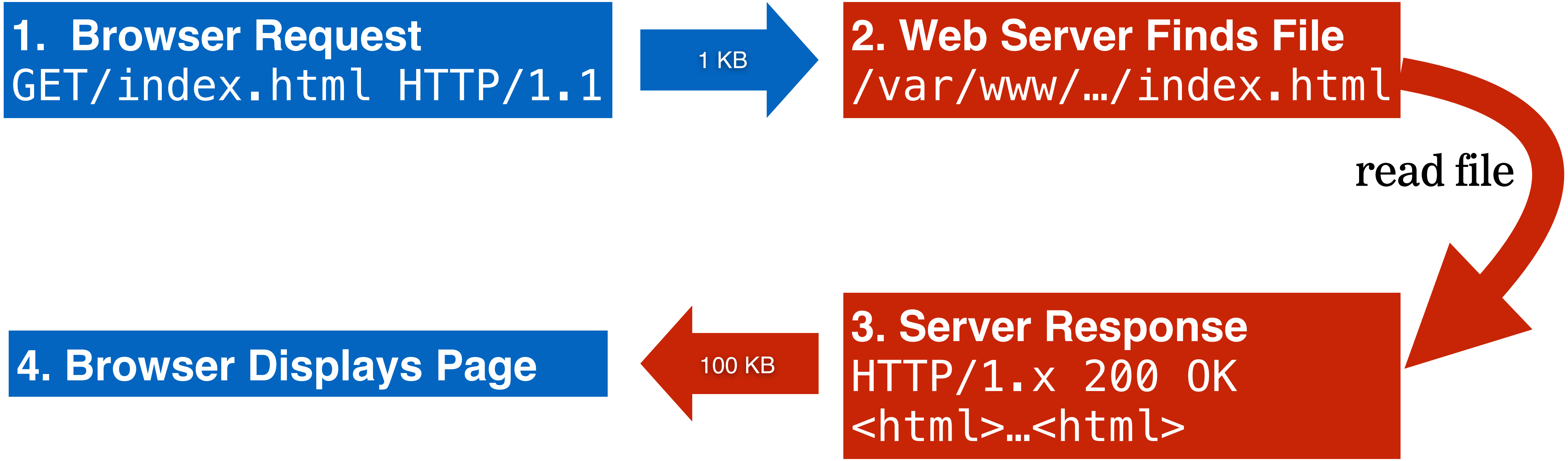
GET	Retrieve a resource
POST	Create a resource
PATCH	Update an existing resource
PUT	Replace an existing resource
DELETE	Delete a resource

Most widely used

LET'S TAKE A CLOSER LOOK



HTTP REQUEST AND RESPONSE



1. Browser Request
`GET/index.html HTTP/1.1`

1 KB

2. Web Server Finds File
`/var/www/…/index.html`

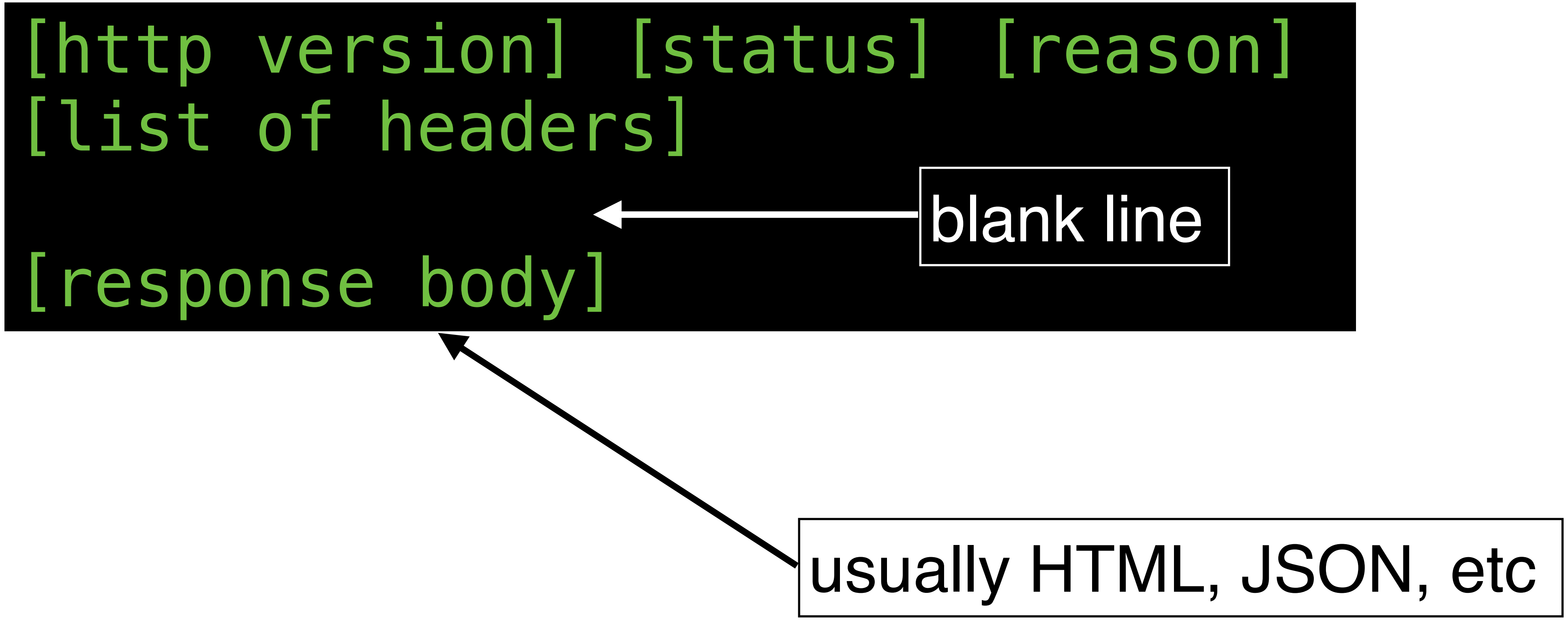
read file

3. Server Response
`HTTP/1.x 200 OK`
`<html>…<html>`

100 KB

4. Browser Displays Page

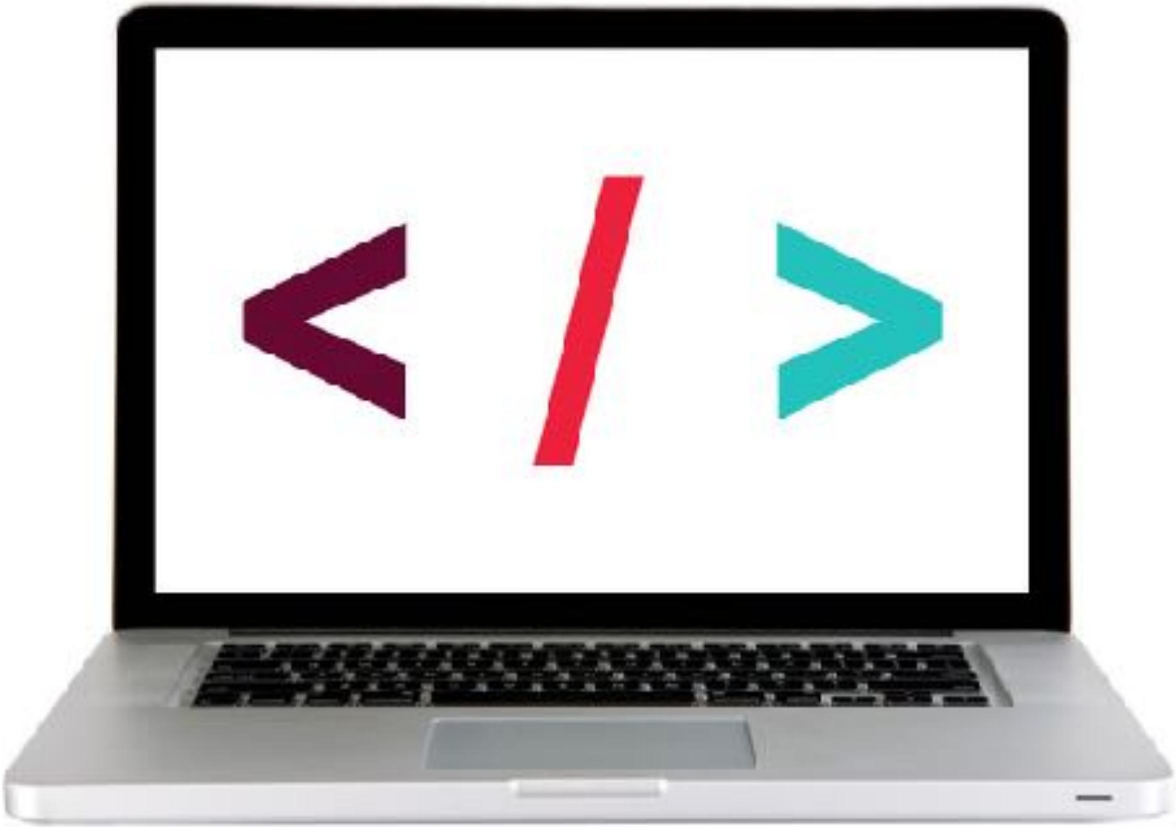
HTTP RESPONSE STRUCTURE



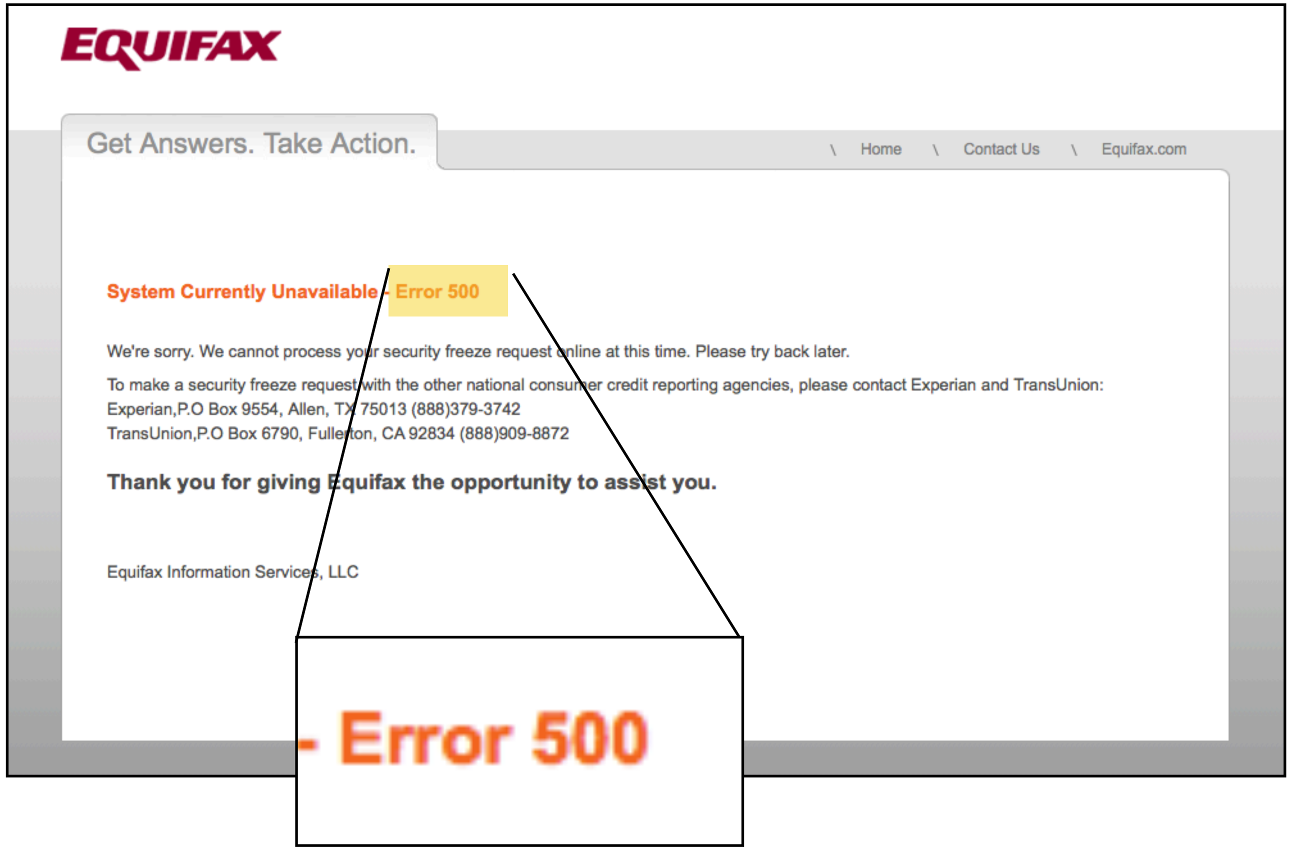
```
[http version] [status] [reason]
[list of headers]

[response body]
```

LET'S TAKE A CLOSER LOOK



HTTP STATUS CODES



EQUIFAX

Get Answers. Take Action.

\ Home \ Contact Us \ Equifax.com

System Currently Unavailable - Error 500

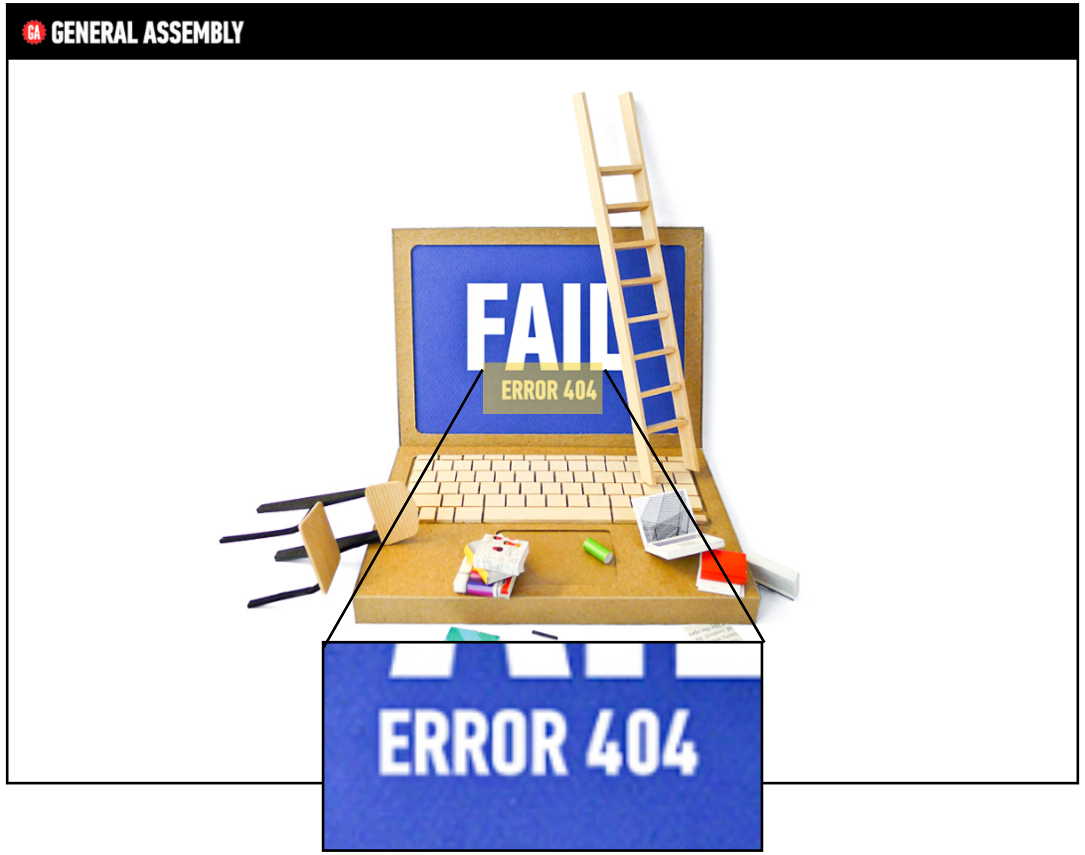
We're sorry. We cannot process your security freeze request online at this time. Please try back later.

To make a security freeze request with the other national consumer credit reporting agencies, please contact Experian and TransUnion:
Experian,P.O Box 9554, Allen, TX 75013 (888)379-3742
TransUnion,P.O Box 6790, Fullerton, CA 92834 (888)909-8872

Thank you for giving Equifax the opportunity to assist you.

Equifax Information Services, LLC

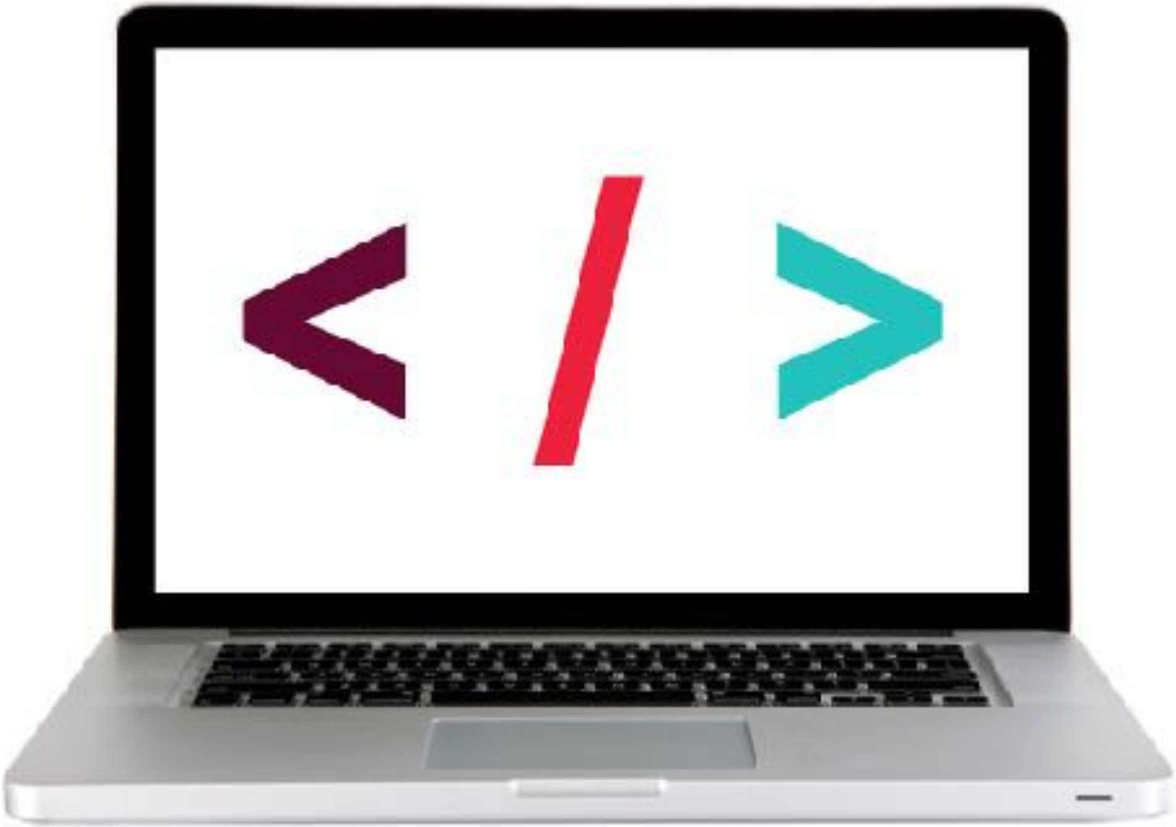
- Error 500



HTTP STATUS CODES

`200`	Okay
`301`	Moved permanently
`302`	Moved temporarily
`400`	Bad request
`403`	Forbidden
`404`	Not found
`418`	I’m a teapot
`500`	Internal server error

LET'S TAKE A CLOSER LOOK



Ajax

Ajax

Asynchronous

JavaScript

And

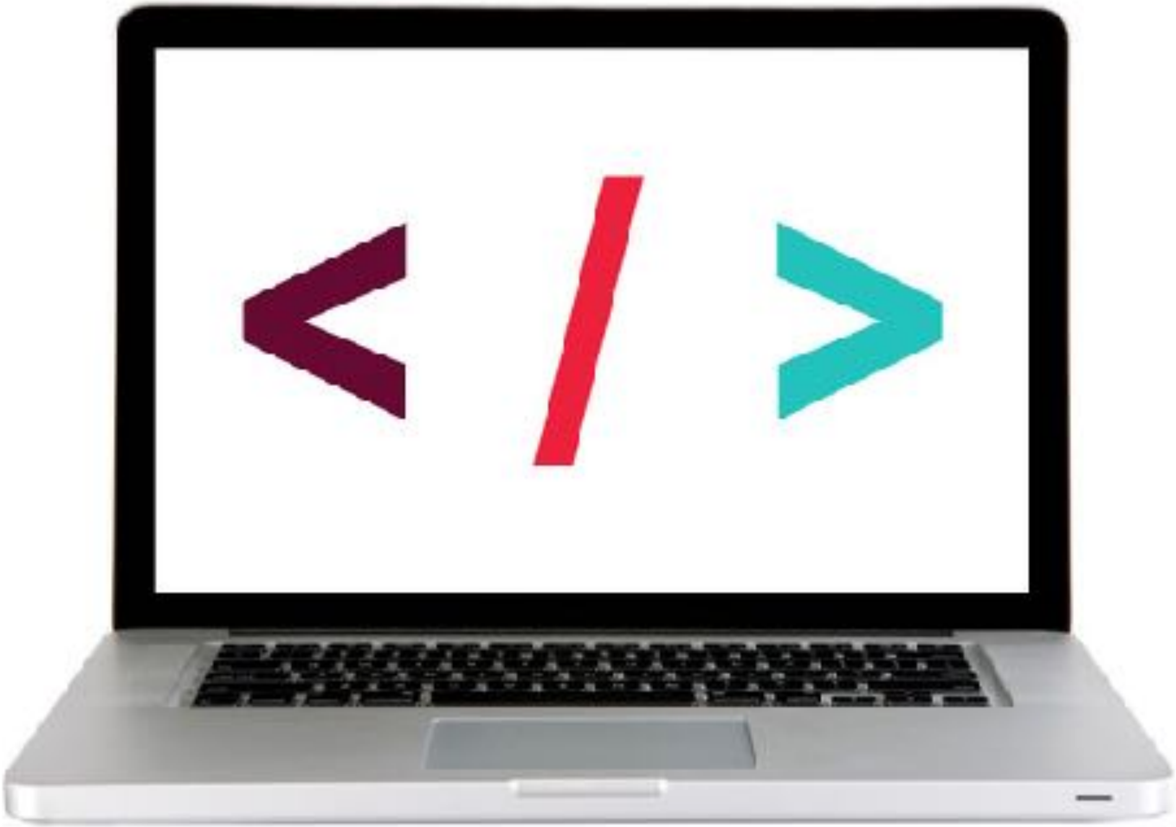
XML **or JSON!**

Ajax in vanilla JS

Fetch = Ajax requests in vanilla JavaScript

```
fetch(url)
  .then((response) => {
    // check if request was successful
  })
  .then((data) => {
    // do something with the data
  });
```

LET'S TAKE A CLOSER LOOK



EXERCISE - CREATING AN AJAX REQUEST

EXERCISE

LOCATION

▸ `starter-code > 1-fetch-ajax-exercise`

TIMING

5 min

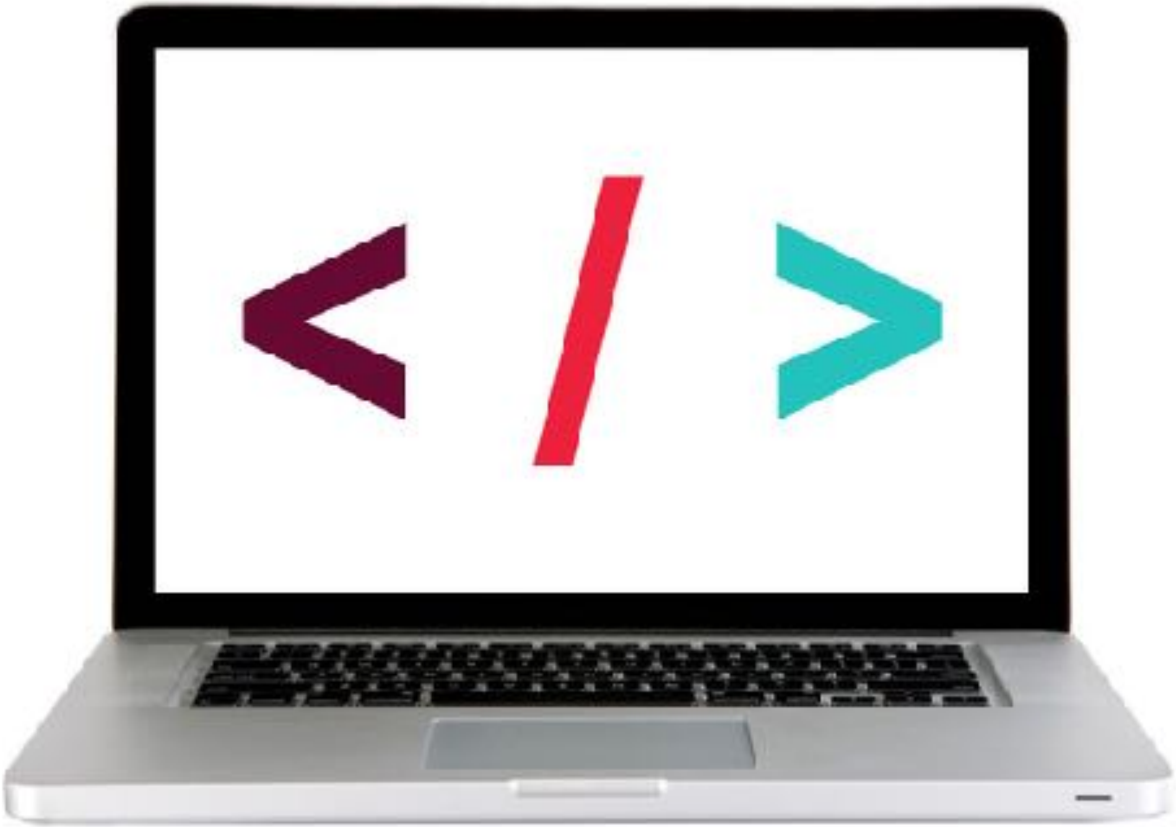
1. Copy the code from the codealong to the main.js file.
2. Change the URL to the one shown in the instructions.
3. Verify that a new set of results is shown in the console.
4. BONUS: Customize the error message to display the text of the HTTP status message.
   (Hint: look at https://developer.mozilla.org/en-US/docs/Web/API/Response/statusText)
5. BONUS: Refactor the code to work with user interaction. In the index.html file there is a "Get Health Data" button. Modify your code so data is only requested and logged to the console after a user clicks the button.

jQuery Ajax

Using Ajax with jQuery

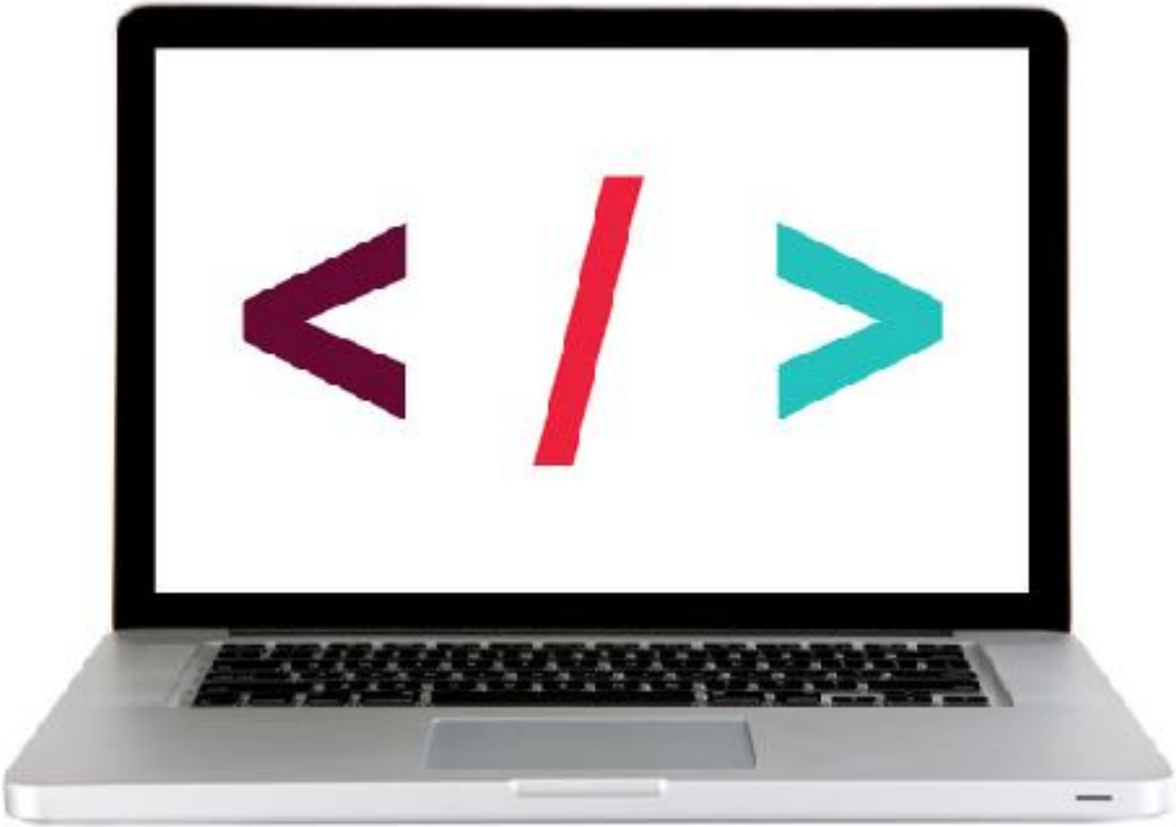
method	description
`$.get()`	loads data from a server using an HTTP GET request
`$.ajax()`	performs an Ajax request based on parameters you specify

LET'S TAKE A CLOSER LOOK



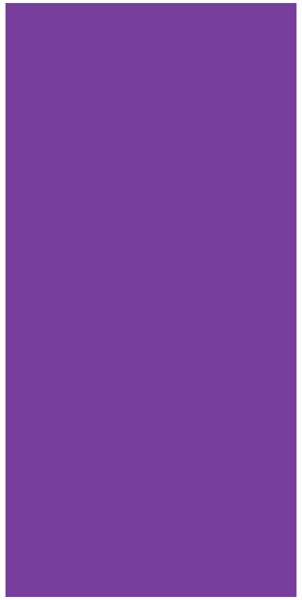
Code organization

LET'S TAKE A CLOSER LOOK



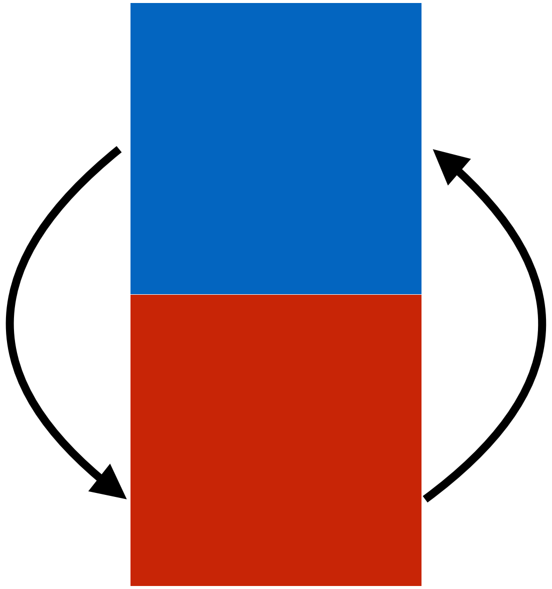
SEPARATION OF CONCERNS

code for data and view intermingled



✗

code for data



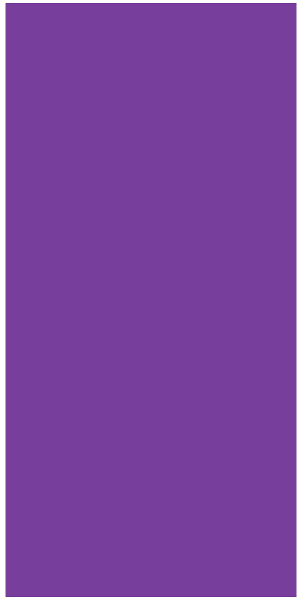
code for view

parts of code call each other, but are maintained separately

✓

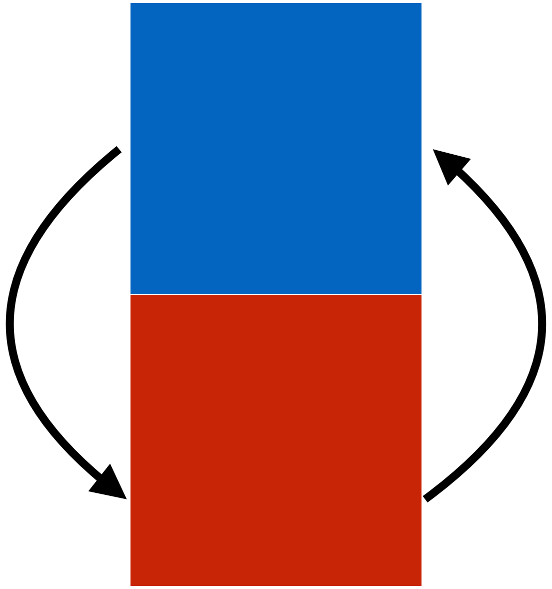
SEPARATION OF CONCERNS – HTTP

code for client
and for HTTP
requests
intermingled



✗

code for client



code for HTTP

parts of code
call each
other, but are
maintained
separately

✓

LAB — JQUERY AJAX

OBJECTIVE

- Create an Ajax request using jQuery or Fetch.

LOCATION

- `starter-code > 4-ajax-lab`

EXECUTION

until 9:20

1. Open index.html in your editor and familiarize yourself with the structure and contents of the file.
2. Open main.js in your editor and follow the instructions.

Exit Tickets!

(Class #9)

LEARNING OBJECTIVES - REVIEW

- Use event delegation to manage dynamic content.
- Use implicit iteration to update elements of a jQuery selection
- Identify all the HTTP verbs & their uses.
- Describe APIs and how to make calls and consume API data.
- Access public APIs and get information back.
- Implement an Ajax request with Fetch.
- Create an Ajax request using jQuery.

NEXT CLASS PREVIEW

Asynchronous JavaScript and Callbacks

- Describe what asynchronous means in relation to JavaScript
- Pass functions as arguments to functions that expect them.
- Write functions that take other functions as arguments.
- Build asynchronous program flow using Fetch
- Integrate string and variable values using template literals

Q&A