Your Next Week

Tuesday March 31	Wednesday April 01	Thursday April 02	Friday April 03
6:30 PM **— DUE Class 05 Lab** **— DUE Class 06 Reading** — Class 06A	*6:30 PM* — Class 06B *MIDNIGHT* **— DUE Class 06 Learning Journal**	*6:30 PM* — Co-working	
Saturday April 04	**Sunday April 05**	**Monday April 06**	**Tuesday April 07**
9 AM **— DUE Class 06 Code Challenge** **— DUE Class 06 Lab** **— DUE Class 07 Reading** — Class 07 *MIDNIGHT* **— DUE Class 07 Learning Journal**	*MIDNIGHT* **— DUE Career: Networking Gameplan** **— DUE Class 06-07 Feedback**		*6:30 PM* **— DUE Class 07 Code Challenge** **— DUE Class 07 Lab** **— DUE Class 08 Reading** — Class 08A

Lab 05 Review

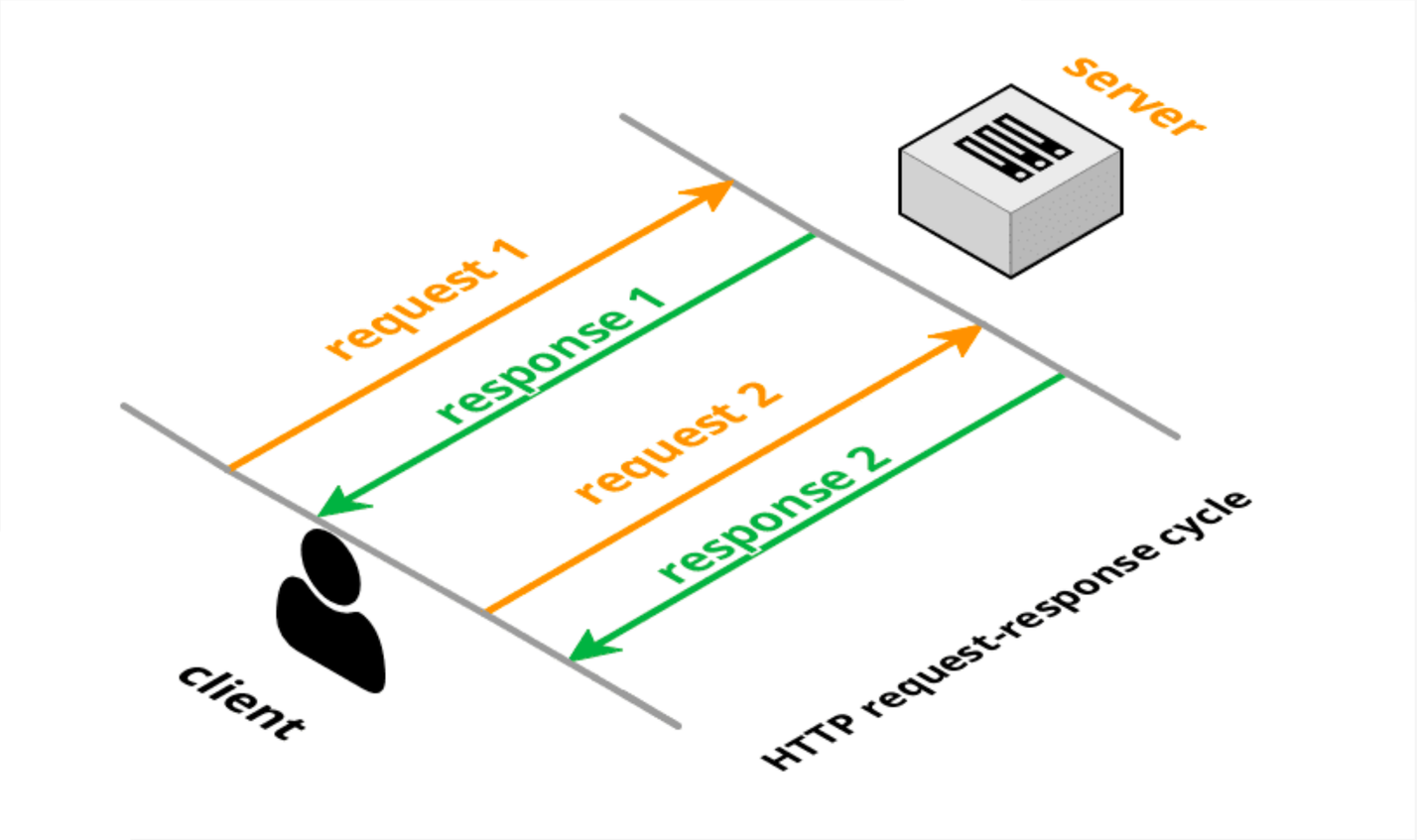
Class 06

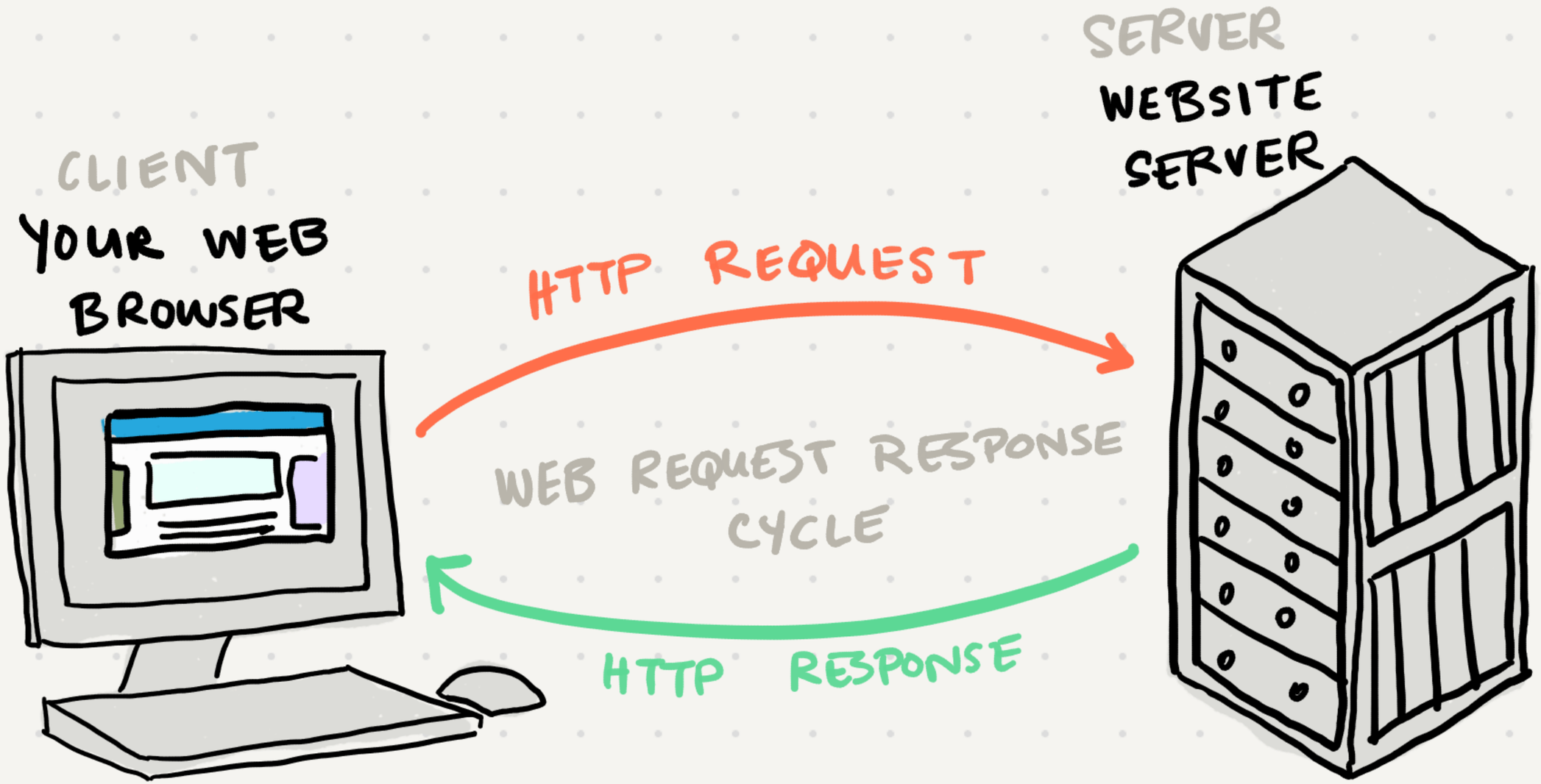
HTTP and REST

seattle-javascript-401n16

Web Request Response Cycle

- A pattern that describes how data is shared on the web
- **Clients** make requests to a server
- **Servers** respond to clients
- Clients receive response, and then can make more requests




CLIENT
YOUR WEB BROWSER
SERVER
WEBSITE SERVER
HTTP REQUEST
WEB REQUEST RESPONSE CYCLE
HTTP RESPONSE

HTTP

- What does it mean to start urls with **http**? (**HyperText Transfer Protocol**)
- Telling the browser to open up a connection
- HTTP is the ruleset that connection must follow
- Connect to some Uniform Resource Locator (URL)
- HTTPS is more secure than HTTP

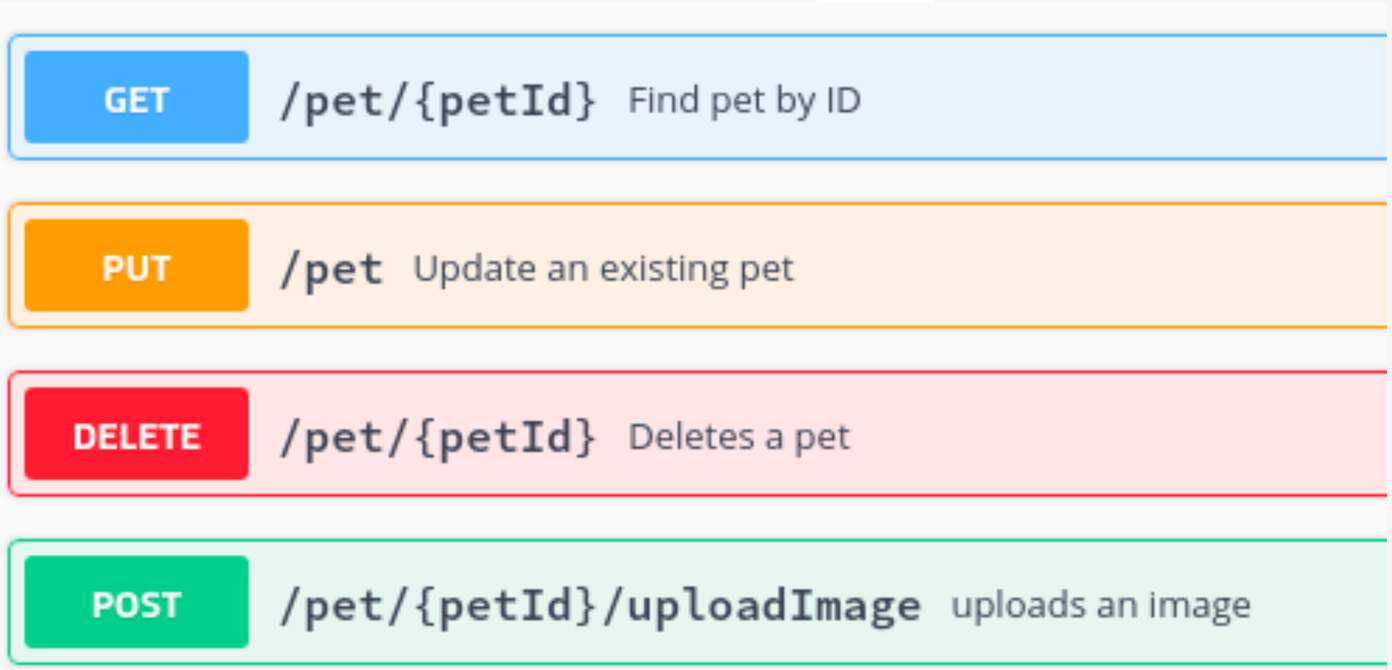
http://localhost:3000/path/to/something?name='Sonia'

https://www.facebook.com/soniaKandah?ref=bookmarks

Protocol URL/Host [Port] [Path] [Query]

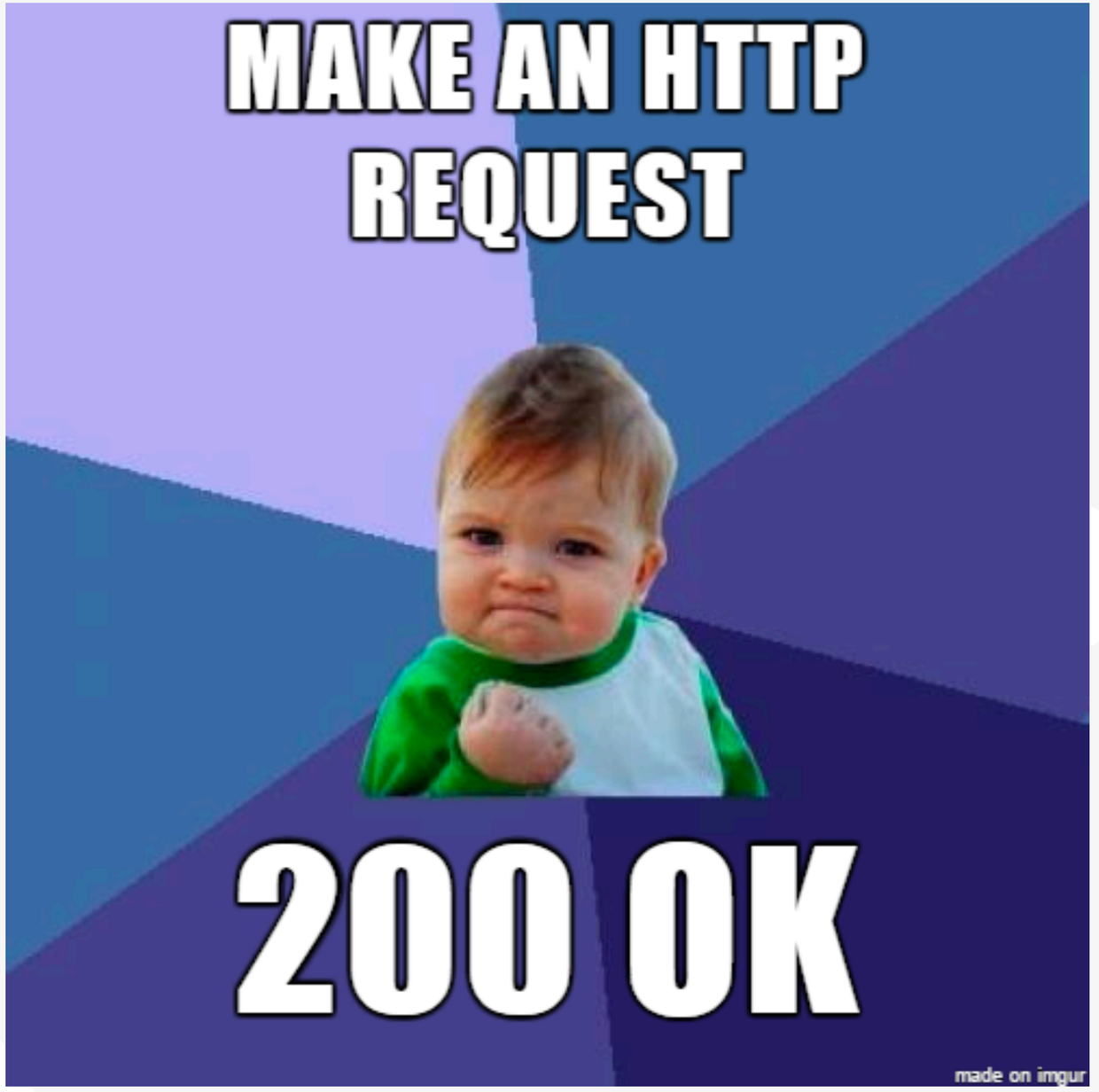
HTTP to CRUD

- CRUD specifies our standard desired operations on data (create, read, update, delete)
- HTTP has specific terms for these operations
  - CREATE == POST
  - READ == GET
  - UPDATE == PUT
  - DELETE == DELETE



Request

- A request usually has the following:
  - Method - GET, POST, PUT, DELETE
  - URL - Where the request should go
  - Headers - administrative information about the client
  - Body (optional) - any data the client needs to send



Response

- A response usually has the following:
  - Body/JSON - any returned data the client asked for
  - Status - a standardized numerical code that tells the client how the request completed (success or error!)

Endpoints

- Clients request to a **URL**
- When you load a website, that's a GET request (`GET https://www.google.com`)
- Servers can have lots of endpoints! Each can do something different
  - `POST http://mysite.com/food`
  - `DELETE http://mysite.com/food/apple`
  - `GET http://mysite.com/drinks`

I Want [to] REST

- Representational State Transfer (REST) describes proper API endpoint structure and documentation
- It's a guideline, not a protocol
- `http://` - the protocol

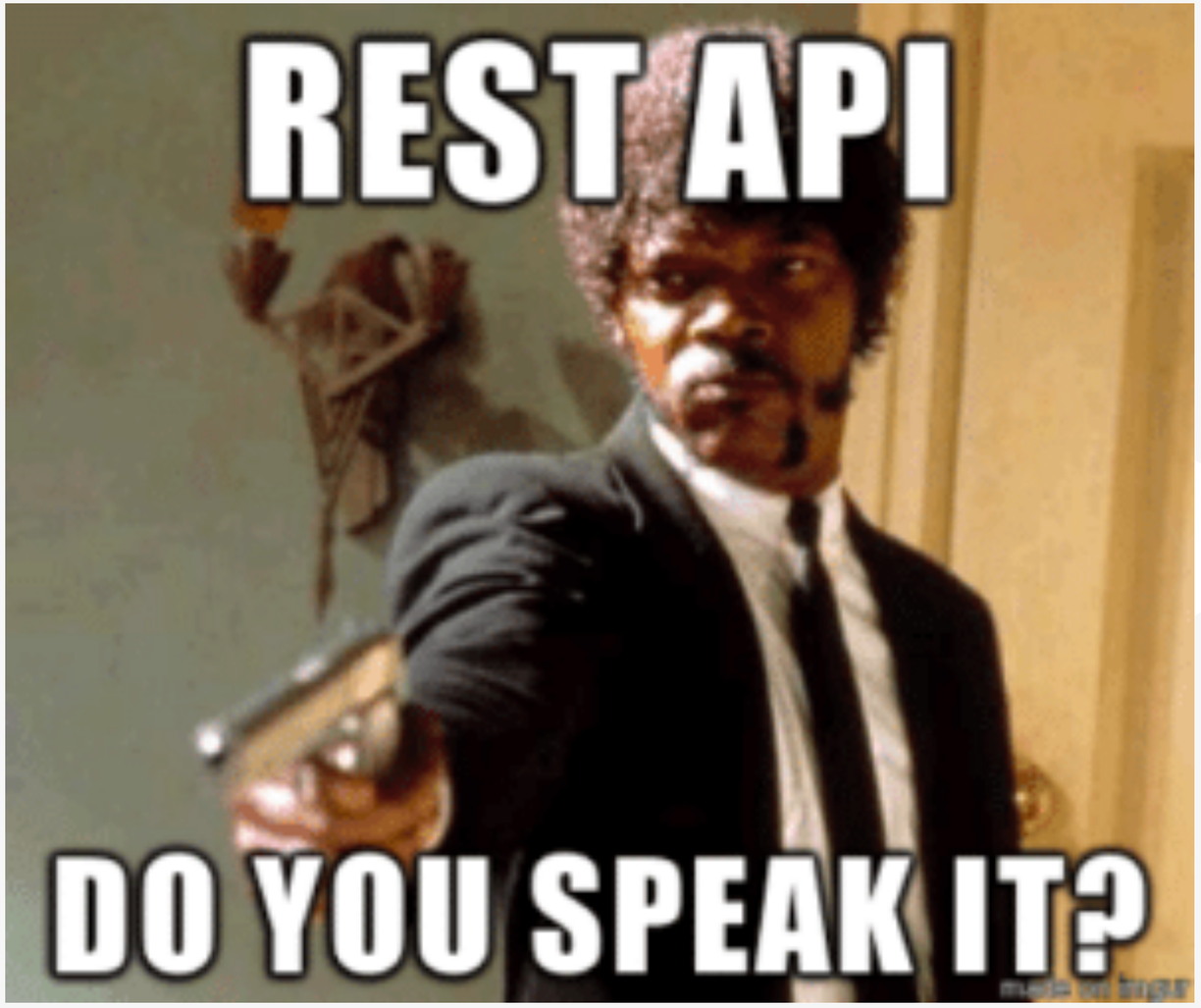
  `www.note-taker.com` - the URL

  `/api` - the API endpoints

  `/v1` - the version of the API

  `/notes` - the collection you want

  `/12345` - the record id



Swagger

- A useful tool that makes it easy to create API documentation
- You can host this documentation on another URL
- All you have to do is pretend to be a Client making requests
    - The documentation will be built with each request!

Vocab Review

Application Programming Interface (API)

Application Programming Interface (API)

A definition of how other applications can interact with your application. APIs define what kind of requests can be made to it from clients.

server

server

A program who’s goal is to provide data or functionality to other programs. Servers are meant to continually run and be ready to “serve” a response to anyone needing it.

client

client

A client is a piece of computer hardware or software that accesses a service made available by a server. Typically, a client access the server over the internet / network.

Web Request Response Cycle (WRRC)

Web Request Response Cycle (WRRC)

Applications over the internet communicate with one another through a series of client requests and server responses. WRRC is just a term for this common interaction.

HTTP

HTTP

Hypertext Transfer Protocol is a set of rules that form the foundation of how data is shared over the web. HTTP defines the structure of a client request and a server response, and it allows for a set of methods (GET, POST, PUT, etc) to be commonly understood by the web.

request

request

A command a client builds and sends to a server, asking for some data to be returned or some data operation to execute. Requests typically have a method, URL, body and headers.

response

response

An object a server builds and returns to a requesting client. Responses typically contain any data the client was looking for, and a status code describing how the request fared (successfully or with an error).

HTTP Status Codes

HTTP Status Codes

A collection of standard numerical codes

that all HTTP compliant servers use.

These codes range from 100-500, and

each has a specific meaning.

endpoint

endpoint

A specific URL that a client can request to. An endpoint is a combination of a URL path an a method, and the server defines specific code to build each endpoint’s response.

REST

REST

A set of guidelines that all web APIs should follow in order to stay consistent. Though these guidelines are not enforced, they cover everything from how to structure endpoint paths, to how to document your API.

Swagger

Swagger

A tool that allows you to quickly and easily generate documentation for your API, so that anyone who wants to be a client can understand how to contact your server.

Lab 06 Overview

Code Challenge 06 Overview