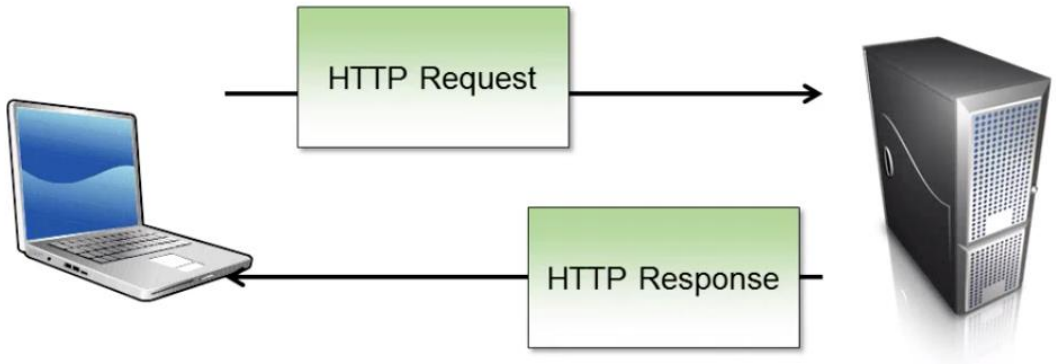
FSW-115: Communication Over the Web – HTTP, AJAX and APIs

Week 1: HTTP & JSON
Week 2: URLs & GET Request with Postman

- Hypertext Transfer Protocol (HTTP):
  - technology on how data is exchanged over the web.
  - a protocol is a predefined set of rules that allows computers to , communicate with each other (tcp, ip, http, ftp, ssh, smtp, etc).
  - clients (browser) transmit message containing http methods (verbs) that tell their **request** to the receiving computer (server).
  - server receives these requests, reacts and then send the **response** back to the client.

A single HTTP Transaction (request – response cycle)
(always initiated by the client):



https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

HTTP Request Methods: (tells the server what you want to do):

- GET – Retrieve a resource (requesting data)
- POST – Create a new resource (sending data)
- PUT – Update a resource (update data)
- DELETE – Remove a resource (remove data)

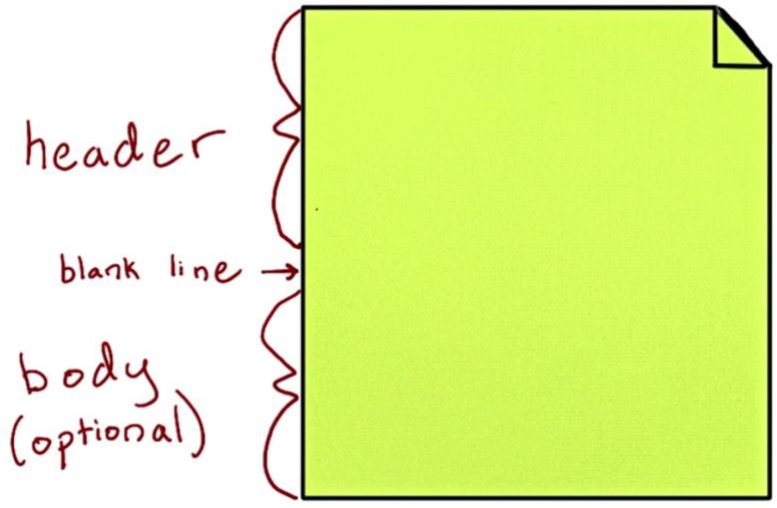
What is an API and a REST API?

- APIs connect one application to the data services of another application. It is mainly just a set of functions.
- REST (Representation State Transfer) is an architectural style for networked applications on the web. It is a guideline on how to build an API.
- REST APIs are web service APIs which uses URIs and HTTP and JSON for its data format

Great video: https://www.youtube.com/watch?v=SLwpqD8n3d0

What is a Request?

- A way to exchange data between a client and server
- Client send the request
- Server responds to the request
- Each **request** is sent to a specific URL
- Message returned by the server is called the **response**



Request and Response Data

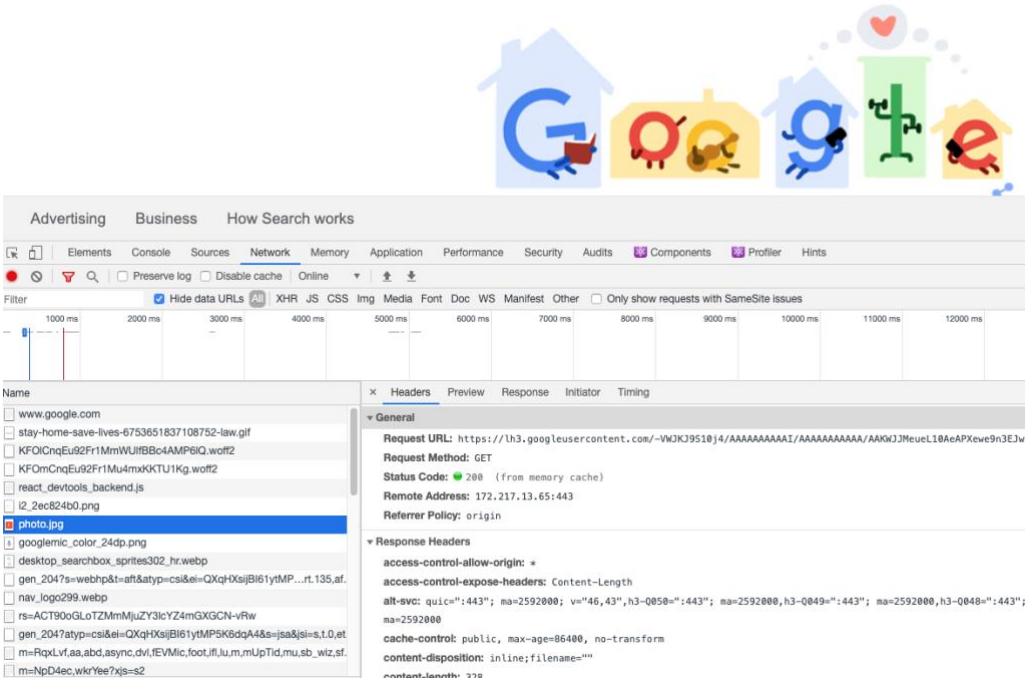
- Form data
- JSON ({“key”:”value”})
- Files (images, .zip .html, .css)
- XML (structured data, but used less now that JSON is more popular)

Request Components

- Method (GET, POST, DELETE, PUT)
- Data (optional)
- Headers (meta information – type of data, etc.)
- Authentication (token – verify who you are; usually stored in header)

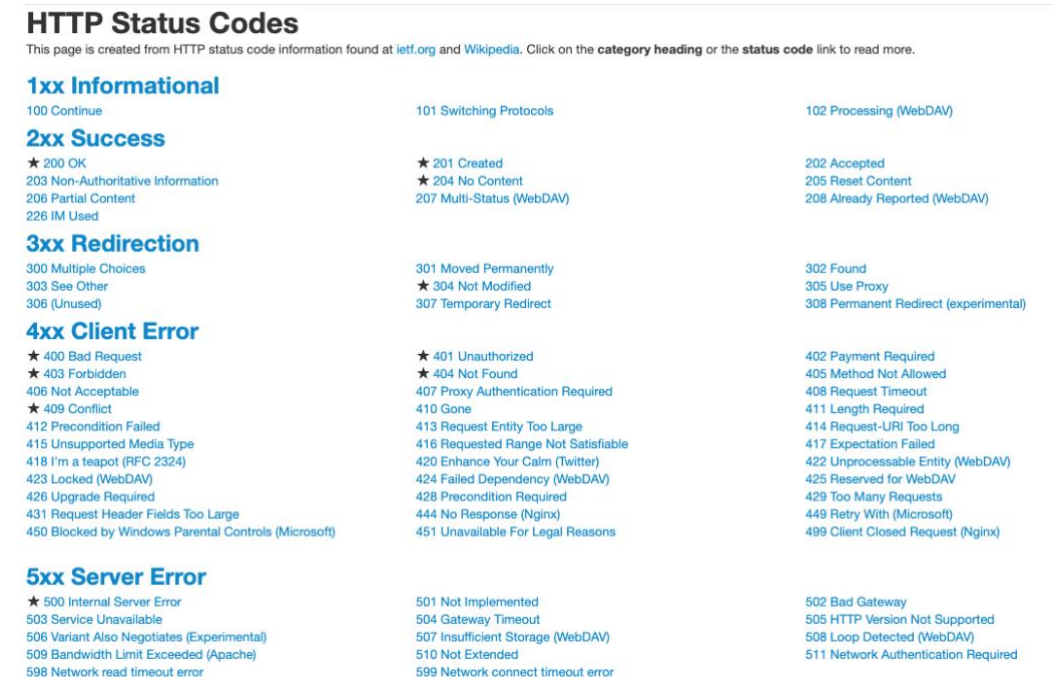
Response Components

- Data (type of data you are requesting - JSON, .html, .css, images, etc.)
- Status Code (2xx, 3xx, 4xx, 5xx); 1XX – Information, 2XX – Success, 3XX – Redirect, 4XX – Client Error, 5XX – Server Error
- Headers (tells you the type of data returned)



A web page can contain many HTTP requests. (network tab).

Status Codes (issued by the server from a request)



HTTP Status Codes

This page is created from HTTP status code information found at ietf.org and Wikipedia. Click on the **category heading** or the **status code** link to read more.

1xx Informational

100 Continue
101 Switching Protocols
102 Processing (WebDAV)

2xx Success

★ 200 OK
★ 201 Created
202 Accepted
203 Non-Authoritative Information
★ 204 No Content
205 Reset Content
206 Partial Content
207 Multi-Status (WebDAV)
208 Already Reported (WebDAV)
226 IM Used

3xx Redirection

300 Multiple Choices
301 Moved Permanently
302 Found
303 See Other
★ 304 Not Modified
305 Use Proxy
306 (Unused)
307 Temporary Redirect
308 Permanent Redirect (experimental)

4xx Client Error

★ 400 Bad Request
★ 401 Unauthorized
402 Payment Required
★ 403 Forbidden
★ 404 Not Found
405 Method Not Allowed
406 Not Acceptable
407 Proxy Authentication Required
408 Request Timeout
★ 409 Conflict
410 Gone
411 Length Required
412 Precondition Failed
413 Request Entity Too Large
414 Request-URI Too Long
415 Unsupported Media Type
416 Requested Range Not Satisfiable
417 Expectation Failed
418 I'm a teapot (RFC 2324)
420 Enhance Your Calm (Twitter)
422 Unprocessable Entity (WebDAV)
423 Locked (WebDAV)
424 Failed Dependency (WebDAV)
425 Reserved for WebDAV
426 Upgrade Required
428 Precondition Required
429 Too Many Requests
431 Request Header Fields Too Large
444 No Response (Nginx)
449 Retry With (Microsoft)
450 Blocked by Windows Parental Controls (Microsoft)
451 Unavailable For Legal Reasons
499 Client Closed Request (Nginx)

5xx Server Error

★ 500 Internal Server Error
501 Not Implemented
502 Bad Gateway
503 Service Unavailable
504 Gateway Timeout
505 HTTP Version Not Supported
506 Variant Also Negotiates (Experimental)
507 Insufficient Storage (WebDAV)
508 Loop Detected (WebDAV)
509 Bandwidth Limit Exceeded (Apache)
510 Not Extended
511 Network Authentication Required
598 Network read timeout error
599 Network connect timeout error

https://restfulapi.net/http-status-codes/

Data Format: JSON (JavaScript Object Notation)

https://www.w3schools.com/whatis/whatis_json.asp

Example:

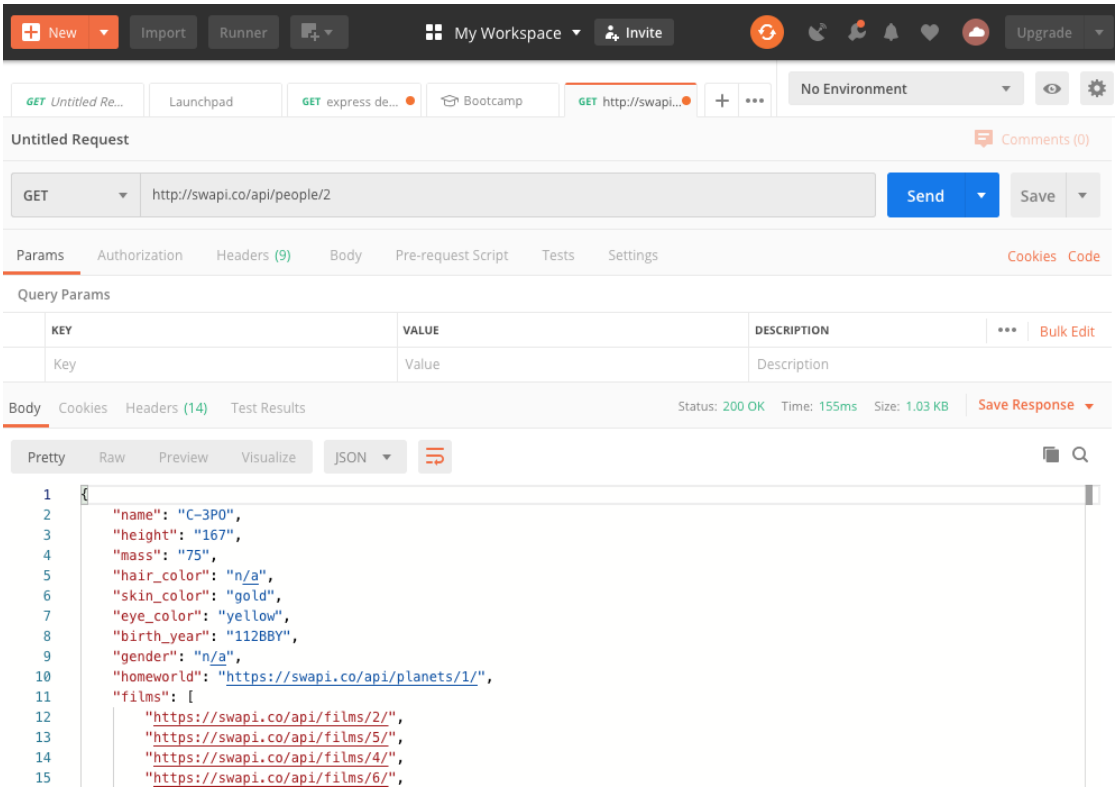
https://codepen.io/frankstepanski/pen/MWwdzqO
https://codepen.io/frankstepanski/pen/JjdQYGp

Validate and generators:

https://jslint.com/
https://jsonlint.com/
https://jsonformatter.org/
https://www.json-generator.com/

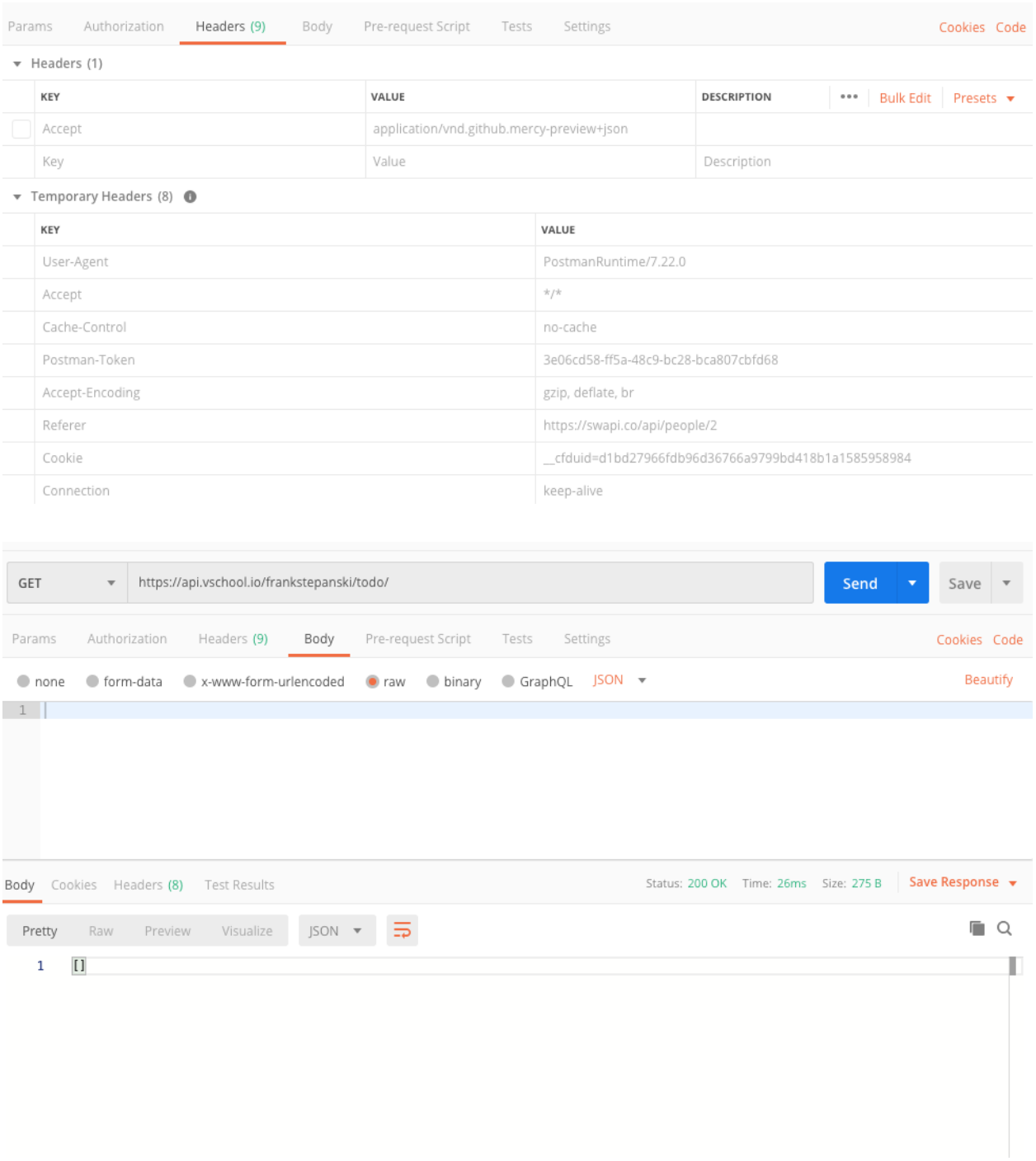
Video: https://www.youtube.com/watch?v=iiADhChRriM

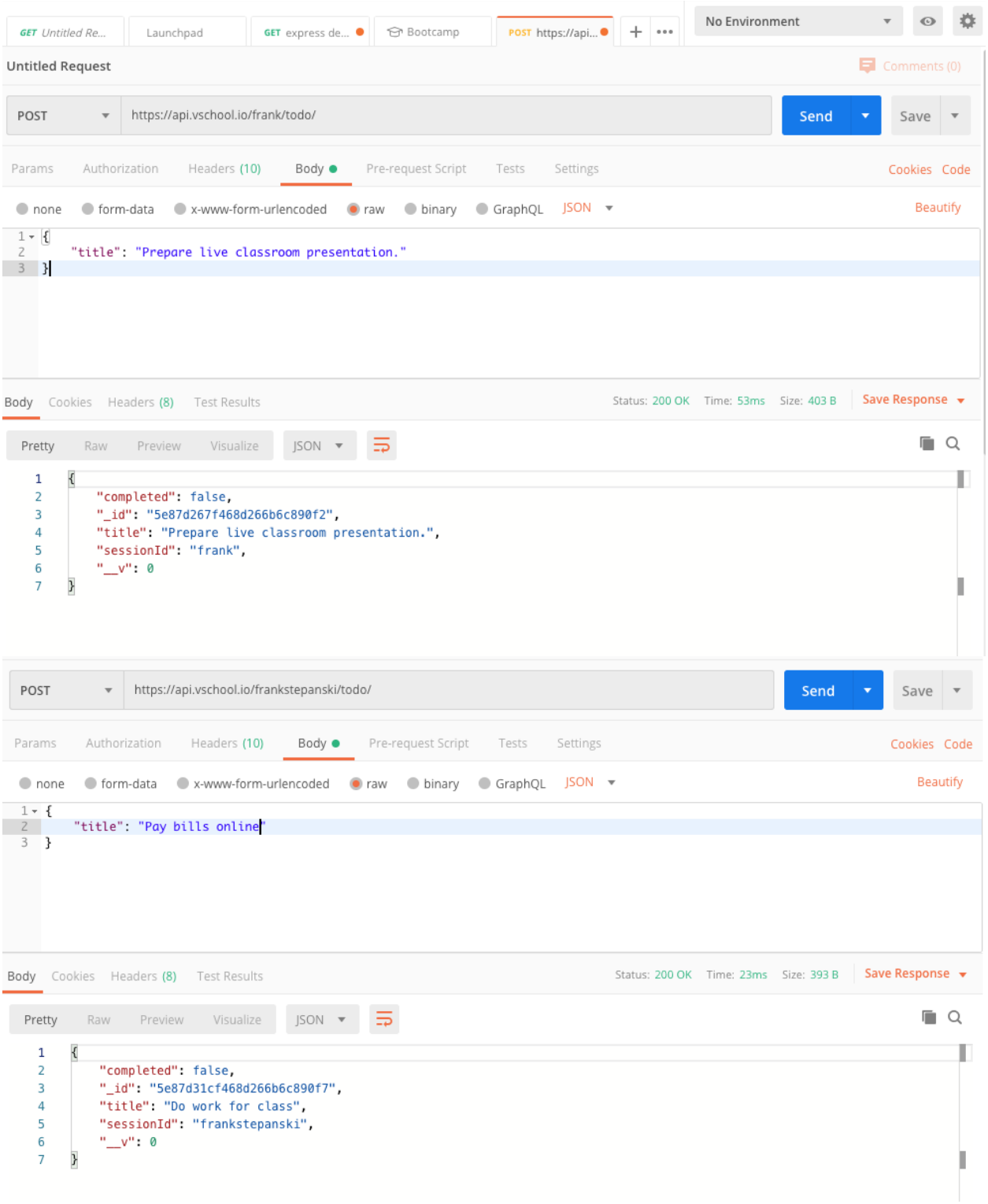
Postman



Learning: https://learning.postman.com/getting-started/

Running through the steps in this article: https://coursework.vschool.io/postman/




GET Untitled Re...
Launchpad
GET express de...
Bootcamp
POST https://api...
No Environment
Untitled Request
Comments (0)
POST
https://api.vschool.io/frank/todo/
Send
Save
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
Code
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"title": "Prepare live classroom presentation."
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 53ms
Size: 403 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"completed": false,
"_id": "5e87d267f468d266b6c890f2",
"title": "Prepare live classroom presentation.",
"sessionId": "frank",
"__v": 0
}
POST
https://api.vschool.io/frankstepanski/todo/
Send
Save
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
Code
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"title": "Pay bills online"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 23ms
Size: 393 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"completed": false,
"_id": "5e87d31cf468d266b6c890f7",
"title": "Do work for class",
"sessionId": "frankstepanski",
"__v": 0
}

GET https://api.vschool.io/frankstepanski/todo/ Send Save

Body Cookies Headers (8) Test Results Status: 200 OK Time: 19ms Size: 736 B Save Response

Pretty Raw Preview Visualize JSON

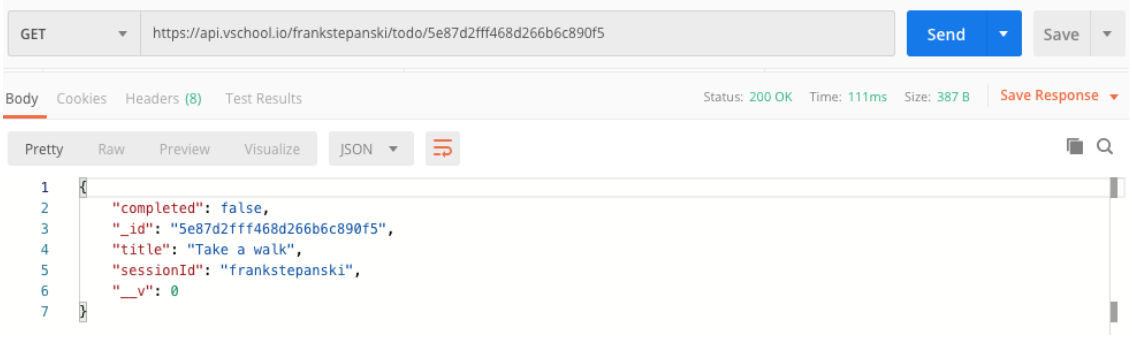

```
[
    {
        "completed": false,
        "_id": "5e87d2fff468d266b6c890f5",
        "title": "Take a walk",
        "sessionId": "frankstepanski",
        "__v": 0
    },
    {
        "completed": false,
        "_id": "5e87d30df468d266b6c890f6",
        "title": "Take a nap",
        "sessionId": "frankstepanski",
        "__v": 0
    },
    {
        "completed": false,
        "_id": "5e87d31cf468d266b6c890f7",
        "title": "Do work for class",
        "sessionId": "frankstepanski",
        "__v": 0
    },
    {
        "completed": false,
        "_id": "5e87d333f468d266b6c890f8",
        "title": "Pay bills online",
        "sessionId": "frankstepanski",
        "__v": 0
    }
]
```

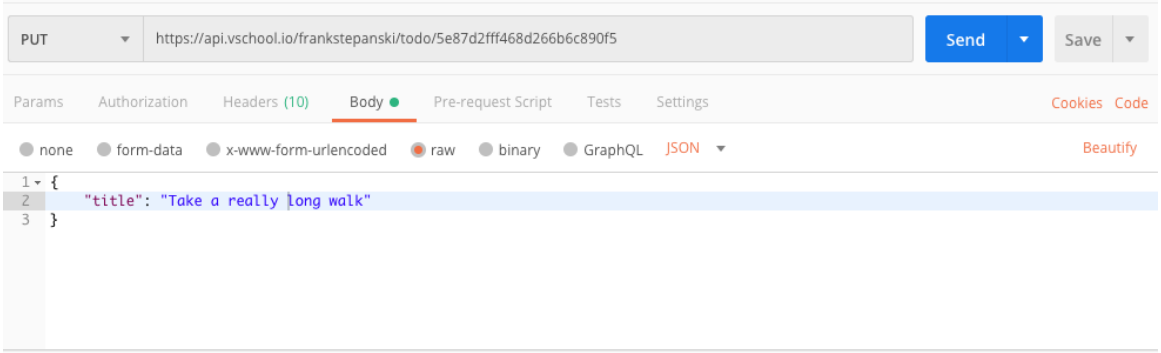

GET https://api.vschool.io/frankstepanski/todo/

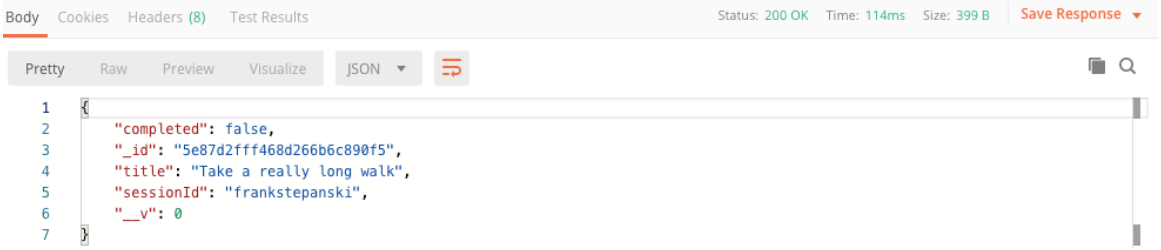
Body Cookies Headers (8) Test Results

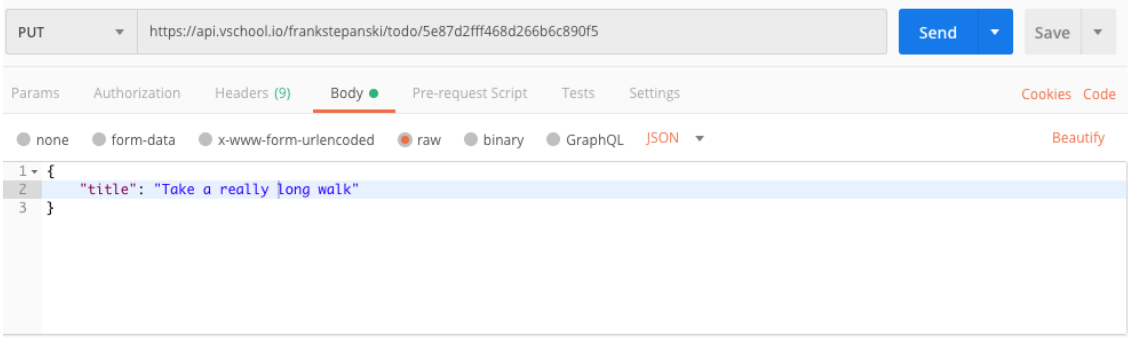
Pretty Raw Preview Visualize JSON

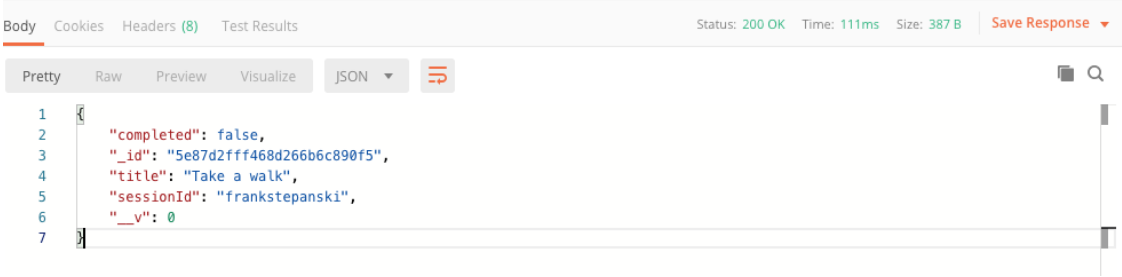

```
[
    {
        "completed": false,
        "_id": "5e87d2fff468d266b6c890f5",
        "title": "Take a walk",
        "sessionId": "frankstepanski",
        "__v": 0
```


GET
https://api.vschool.io/frankstepanski/todo/5e87d2fff468d266b6c890f5
Send
Save
Body Cookies Headers (8) Test Results
Status: 200 OK Time: 111ms Size: 387 B
Save Response
Pretty Raw Preview Visualize JSON
{
"completed": false,
"_id": "5e87d2fff468d266b6c890f5",
"title": "Take a walk",
"sessionId": "frankstepanski",
"__v": 0
}


PUT
https://api.vschool.io/frankstepanski/todo/5e87d2fff468d266b6c890f5
Send
Save
Params Authorization Headers (10) Body Pre-request Script Tests Settings
Cookies Code
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"title": "Take a really long walk"
}


Body Cookies Headers (8) Test Results
Status: 200 OK Time: 114ms Size: 399 B
Save Response
Pretty Raw Preview Visualize JSON
{
"completed": false,
"_id": "5e87d2fff468d266b6c890f5",
"title": "Take a really long walk",
"sessionId": "frankstepanski",
"__v": 0
}


PUT
https://api.vschool.io/frankstepanski/todo/5e87d2fff468d266b6c890f5
Send
Save
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies Code
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"title": "Take a really long walk"
}


Body Cookies Headers (8) Test Results
Status: 200 OK Time: 111ms Size: 387 B
Save Response
Pretty Raw Preview Visualize JSON
{
"completed": false,
"_id": "5e87d2fff468d266b6c890f5",
"title": "Take a walk",
"sessionId": "frankstepanski",
"__v": 0
}

GET https://api.vschool.io/frankstepanski/todo/ Send Save

Body Cookies Headers (8) Test Results Status: 200 OK Time: 24ms Size: 748 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "completed": false,
        "_id": "5e87d2fff468d266b6c890f5",
        "title": "Take a really long walk",
        "sessionId": "frankstepanski",
        "__v": 0
    },
    {
        "completed": false,
        "_id": "5e87d30df468d266b6c890f6",
        "title": "Take a nap",
        "sessionId": "frankstepanski",
        "__v": 0
    },
    {
        "completed": false,
        "_id": "5e87d31cf468d266b6c890f7",
        "title": "Do work for class",
        "sessionId": "frankstepanski",
        "__v": 0
    },
    {
        "completed": false,
        "_id": "5e87d333f468d266b6c890f8",
        "title": "Pay bills online",
        "sessionId": "frankstepanski",
        "__v": 0
    }
]
```


PUT https://api.vschool.io/frankstepanski/todo/5e87d2fff468d266b6c890f5 Send Save

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Code

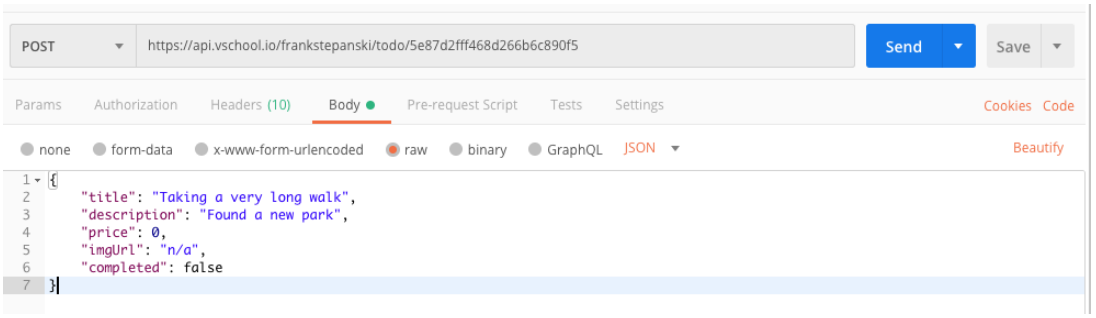
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

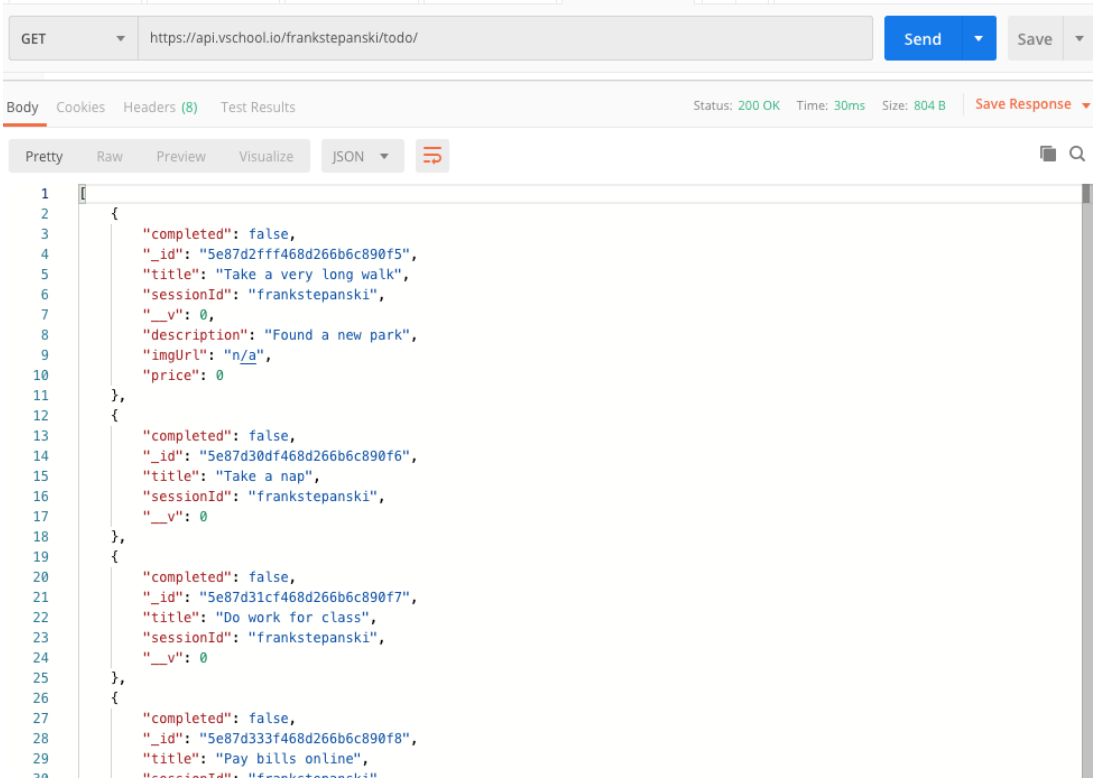

```
{
    "title": "Take a very long walk",
    "description": "Found a new park",
    "price": "No price",
    "imgUrl": "n/a",
    "completed": false
}
```


Body Cookies Headers (8) Test Results Status: 500 Internal Server Error Time: 24ms Size: 368 B Save Response

Pretty Raw Preview Visualize JSON


```
{
    "error": "Cast to number failed for value \"No price\" at path \"price\""
}
```





JSON Parsing Errors

Normally you will not be manually creating a JSON file from scratch. JSON data will normally be created by converting a JS Object or from a database. But whether you create it from hand or programmatically, you may run into parsing errors.

Here is a great article on the most common error and how to try and fix them:

https://filosophy.org/code/fixing-syntaxerror-unexpected-string-token-in-json-at-position/