

Cubstart Web

Lecture 4



[start recording]

Administrivia

- HW3 is due on Friday at midnight!
 - Make an Ed post in the HW3 Megathread for help
 - We encourage students to answer each others questions for more practice!
- If you need to email cubstart@calhacks.io, please enter the subject line as [Cubstart Web] or somehow indicate that you are in the Web Track



Recap



Document Object Model Manipulation

- `const newButton= document.createElement("button")`
- `const box = document.getElementById("box")`
- `function sayHi() {`
`console.log("hi")`
`}`
- `newButton.addEventListener("click", sayHi)`
- `box.appendChild(newButton)`



Introduction to APIs



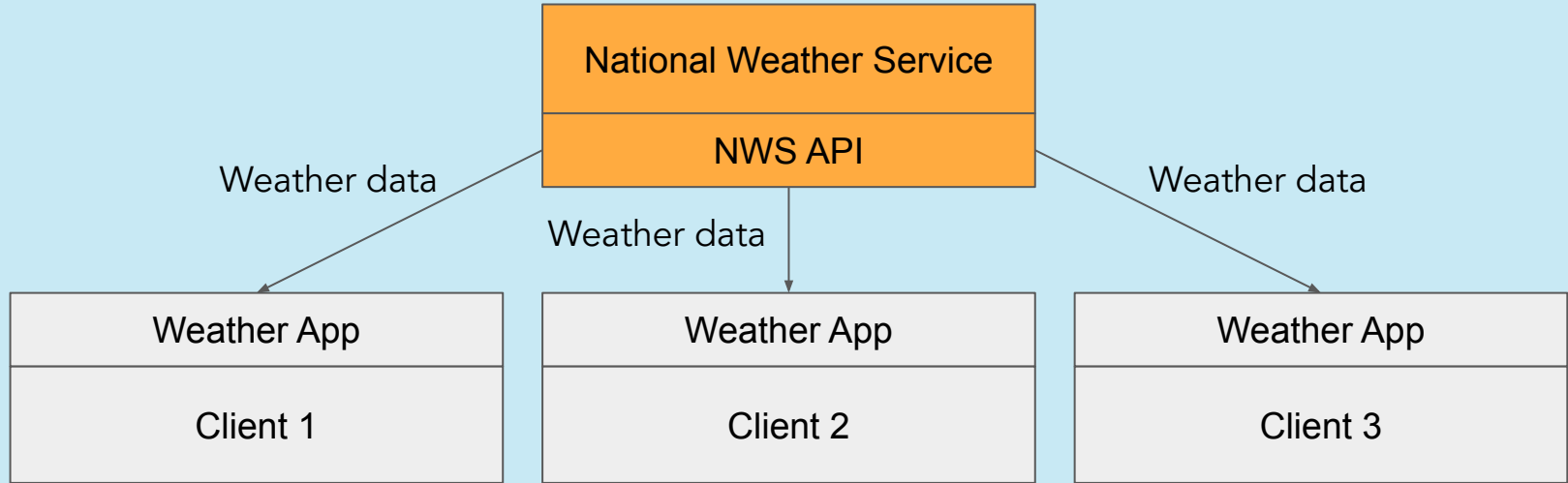


What is an API?

What is an API?


Imagine you work at the National Weather Service, and you have a system that stores weather data. Other software developers are writing weather apps that need access to NWS data. How do you serve these apps the data they need?







What is an API?

- Problem: Software written by different people sometimes need to interact with each other
 - Solution: API (Application Programming Interface), an interface exposed by a service that allows other services to interact with it
- 

What is an API?

- Simple analogy: Vending machine
 - Engineer: Set A3 as chocolate chip cookies, B5 as chips, C2 as gummy bears, etc.
 - You: Press A3, B5, C2, etc. to receive different snacks
 - You don't know how exactly the machine works, how the snacks are made, etc. You just get the snacks you need

What is an API?

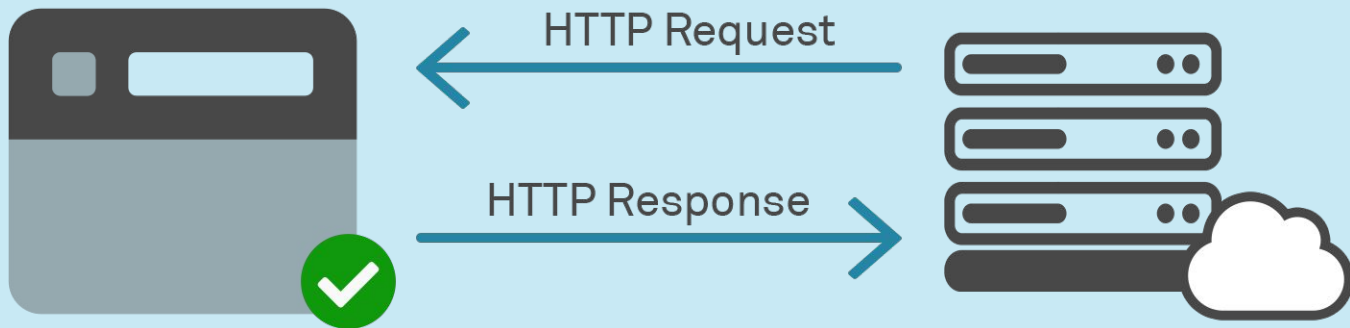
- Analogy: going to a restaurant
 - To order food, you make a “request” with the waiter
 - The waiter executes your request with the kitchen behind the scenes
 - Eventually, the waiter serves you the food
 - You, as the customer, never had to worry about how the food you ordered was made
 - Here, the waiter is the “API”, or the interface between you and the “service”



HTTP APIs

HTTP

- Hypertext Transfer Protocol
- Communications between web servers and clients
- Requests vs. responses
- Text, layout description, images, videos, scripts, and more

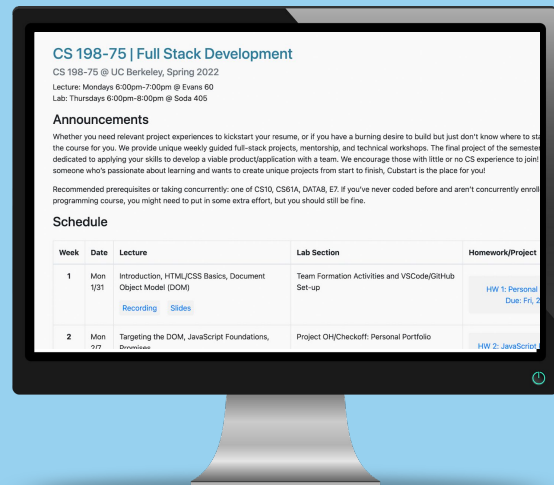


Components of HTTP Systems

- Client makes the request
- Server provides a response
- Application layer: HTTP

Client

- What is initiating the request?
 - user-agent
- The browser is always the entity initiating the request
- Example: visiting a website!



Web Server

- Runs the API
- Serves requested assets
- Load balanced
- Can link to a database





How does a request look like?

• **Analogy: Mail!**

- You want to send mail to a friend
 - Ask how they're doing?
 - Send them a gift?
- To do so, you need to know:
 - Their address
 - What to write in the letter

• **Analogy: Mail!**

- You want to send mail to a friend (**API**)
 - Ask how they're doing? **GET request**
 - **GET information**
 - **Friend will send information back**
 - Send them a gift? **POST request**
 - **POST (send) information**
 - **BOTH are requests!**

Example: GET and POST

- Example request:
 - API: Database for Dad Jokes
 - **I want to send a GET request to the API to receive a Dad Joke**
 - I want to send a POST request to the API to add my own Dad Joke to the database



Check-in Question:

If I wanted to retrieve information about the weather from a Weather API, what type of request would I send?



Parts of an HTTP Request/Response

Endpoints

- To send mail, you need to know:
 - Their address: **Endpoint**
- Fake API: **`https://animal-images.com/`**
 - GET `https://animal-images.com/dog`
 - GET `https://animal-images.com/cat`
 - POST `https://animal-images/add`
 - We'll teach you how to add your own information to the request later on!

Type of HTTP Request

- GET
- POST
- PUT
- DELETE

/books

GET	/books	Lists all the books in the database
DELETE	/books/{bookId}	Deletes a book based on their id
POST	/books	Creates a Book
PUT	/books/{bookId}	Method to update a book
GET	/books/{bookId}	Retrieves a book based on their id



Check-in Question:

If I wanted to add a book to the database, what would I use?



Check-in Question:

If I wanted to update a book to the database,
what would I use?

HTTP Headers

- Request headers
 - Provide information about the request context
 - e.g. Accept, Auth, User-Agent
- Response headers
 - A more detailed context of the response
 - e.g. Age, Location or Server

Example HTTP Request

POST [/api/message](#) HTTP/1.1

User-agent: Chrome/17.2

Accept: text/xml,text/html,text/plain

Accept-Language: en-us, en

Accept-Encoding: gzip

Authorization: Token 20193481039

Hello world! This is my message

HTTP Status Codes

- 1xx (CONTINUE)
 - Request received by server, resolving
- 2xx (SUCCESS)
 - Request received, processed, successfully resolved
- 3xx (REDIRECT)
 - Redirected to another endpoint
- 4xx (BAD REQUEST)
 - No clearance, paywall, not found
- 5xx (SERVER ERROR)

HTTP Status Codes

Level 200 (Success)

200 : OK

201 : Created

203 : Non-Authoritative
Information

204 : No Content

Level 400

400 : Bad Request

401 : Unauthorized

403 : Forbidden

404 : Not Found

409 : Conflict

Level 500

500 : Internal Server Error

503 : Service Unavailable

501 : Not Implemented

504 : Gateway Timeout

599 : Network timeout

502 : Bad Gateway

Example: A Real API

- Dad Joke API: “<https://icanhazdadjoke.com/>”
 - GET a joke: <https://icanhazdadjoke.com/>
 - GET a specific joke by ID:
“https://icanhazdadjoke.com/j/<joke_id>”

There are so many APIs!

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

Dog.ceo
Catfacts.ninja
Agify.io



What are some tools we can use
to make API requests?



Postman



- Postman is an API development tool which helps to build, test and modify APIs
- Lots of features!
 - API Client
 - Automated Testing
 - Documentation
 - Monitoring
- In this course, we're going to use Postman as a tool for making API requests



Try it out!

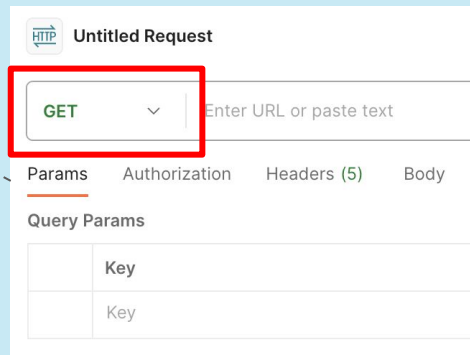
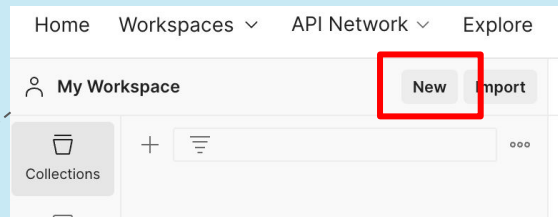
Task #0: Make a Postman account

<https://www.postman.com>



Task #1: Set up your GET request

1. Open Postman and sign in
2. You may have to navigate to “My Workspace”
3. Click New in the top left corner
4. Select “HTTP”
5. Select whatever type of request you want (in this case, its GET)



Task #2: Read documentation

API documentation = technical content that describes how to use and integrate the API in detail

Like an instruction manual!

www.boredapi.com (This API has short documentation because it's fairly simple. Other more complex APIs will have much longer documentation that you should get used to reading through!)

Task #3: Make a GET request

1. Enter a URL (<http://www.boredapi.com/api/activity/>)
2. Click send
3. The response will appear at the bottom in “Body”

JSON

JSON = JavaScript **O**bject **N**otation

- Format to store data in key/value pairs

Syntax:

{ key0: value0, key1: value1, ... }

Example:

```
{  
  "title": "1984",  
  "author": "George Orwell",  
  "details": {  
    "publisher": "Secker & Warburg",  
    "Date": 1949,  
    "Pages": 328  
  }  
}
```



What if I want my GET request to only return responses with certain constraints/parameters?

Task #4: Read more documentation

Let's say I'm bored and I want to use the Bored API to figure out an activity to do with my friend Ddoski.



- Conveniently, the Bored API has a parameter called “participants” that you can pass into an endpoint!

GET /api/activity?participants=:participants

Find a random activity with a given number of participants

Sample Query:

<http://www.boredapi.com/api/activity?participants=1>

Response:

```
{
  "activity": "Learn how to fold a paper crane",
  "accessibility": 0.05,
  "type": "education",
  "participants": 1,
  "price": 0.1,
  "key": "3136036"
}
```

Task #4: Make a GET request w/ parameters

1. Click the “Params” tab, and make a Query entry with key: “participants” and value: “2”. Notice what this does to the URL
2. Click send. Now, everytime you make a request from this endpoint, you will only get activities that involve 2 people!
3. Try changing the query param and value. What do you think this does?

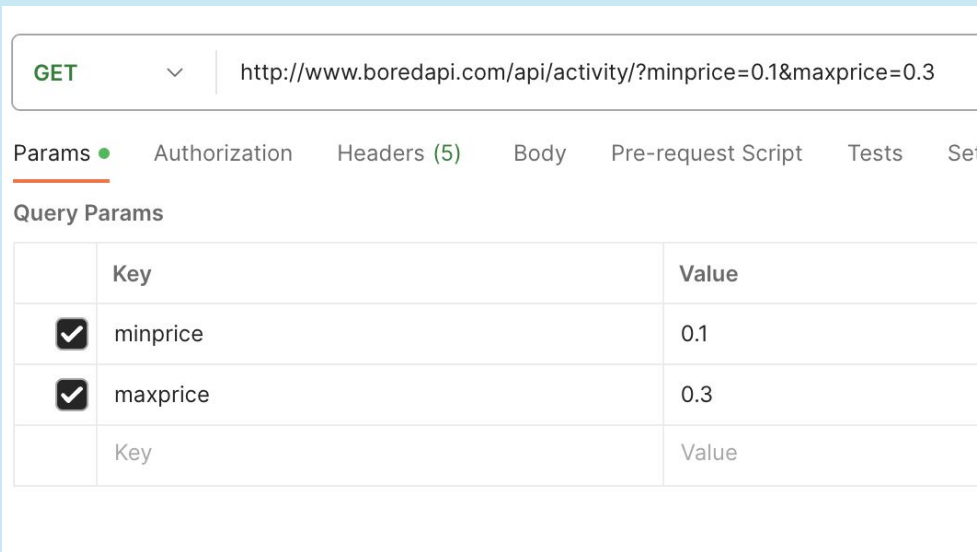
The screenshot shows a REST client interface with a GET request to `http://www.boredapi.com/api/activity/?participants=2`. The 'Params' tab is selected, and a table shows a query parameter 'participants' with value '2'.

Key	Value
<input checked="" type="checkbox"/> participants	2
Key	Value

Check-in Question

Find an activity using the Bored API that is in between the price range 0.1 to 0.3 (Hint: read the documentation!)

Answer:



The screenshot shows a REST client interface with a GET request to `http://www.boredapi.com/api/activity/?minprice=0.1&maxprice=0.3`. The 'Params' tab is selected, displaying a table of query parameters. The table has two columns: 'Key' and 'Value'. Two parameters are listed: 'minprice' with a value of '0.1' and 'maxprice' with a value of '0.3'. Both parameters have a checked checkbox in the first column. Below the table, there is a header 'Query Params' and a table with two columns: 'Key' and 'Value'.

	Key	Value
<input checked="" type="checkbox"/>	minprice	0.1
<input checked="" type="checkbox"/>	maxprice	0.3
	Key	Value

Next Week

- How do we use the response we get from the API in a web application?
- What are API keys and how do we use them?



[stop recording]

Attendance: Lecture 4

<https://forms.gle/LAAZ28LAEzEcpfP59>

Secret word:

