



Welcome!

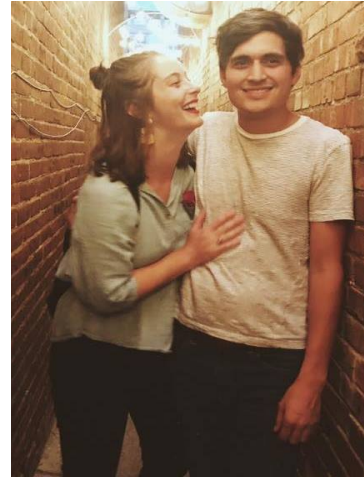
Dynamic Web Applications with JavaScript

1. **WIFI - network: GEvents password: GEvents1644**
2. **Create a Github account**
3. **Download A Text Editor**
4. **Go to <https://github.com/colemanimhoff/i-luv-beer>**

Meet Your Instructors!



- RILEY PRICE
- Security BRO
 - Back End dood
 - Veteran
 - Pilot
 - LOVES PUNS
 - A real life CO native



- COLEMAN IMHOFF
- Music Enthusiast
 - AmeriCorps Allstar
 - React Savant
 - Actually German
 - SASSy Developer



Learning Objectives

-[Git Fork and Clone](#)

-[Fetch Requests](#), [Promises](#) and [.then\(\)](#)

-[console.log\(\)](#) and [Data Structures](#)

-[document.querySelector\(\)](#) and [forEach\(\)](#)

-Style That Bad Boy!



What is an API?

Application Programming Interface

-Essentially an endpoint for you to hit and get data from, or add data to

-Many companies offer APIs for you to access and provide docs to help you

Web API Endpoint Reference

Our Web API endpoints give external applications access to Spotify catalog and user data.

Web API Base URL: <https://api.spotify.com>

[User Guide](#) | [Tutorial](#) | [Code Examples](#)

Search:

METHOD	ENDPOINT	USAGE	RETURNS
GET	/v1/albums/{id}	Get an album	album
GET	/v1/albums?ids={ids}	Get several albums	albums
GET	/v1/albums/{id}/tracks	Get an album's tracks	tracks*
GET	/v1/artists/{id}	Get an artist	artist
GET	/v1/artists?ids={ids}	Get several artists	artists
GET	/v1/artists/{id}/albums	Get an artist's albums	albums*
GET	/v1/artists/{id}/top-tracks	Get an artist's top tracks	tracks

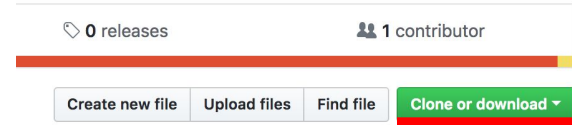
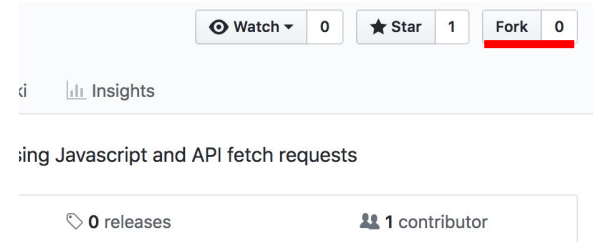
-Head to <https://github.com/colemanimhoff/i-luv-beer>

-Go to the Master branch

-Click the Fork button in the top right

-Once you've forked the repo, you have your own version to work on!

-Now clone it (download it) to your computer



Git Fork and Clone



Making Fetch Happen

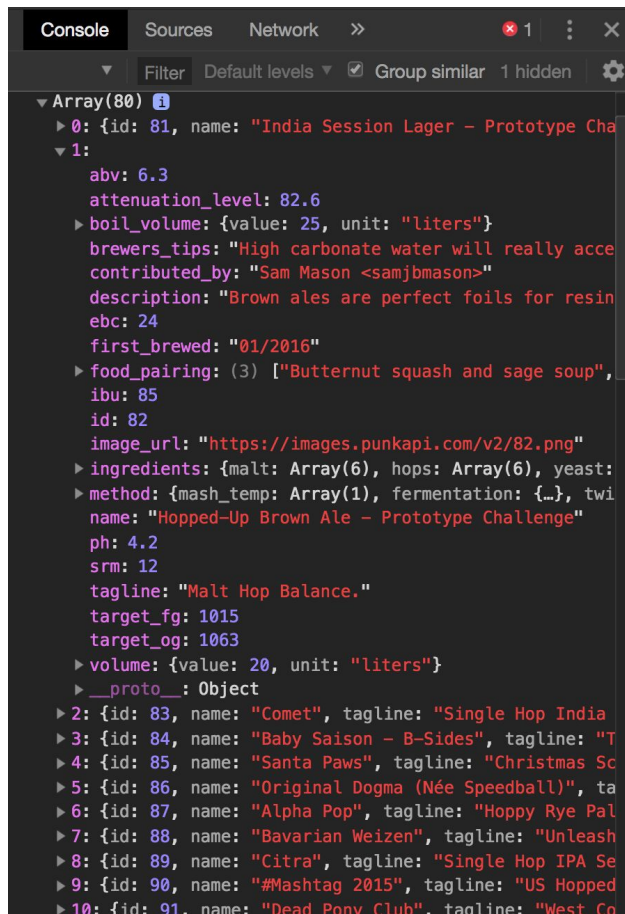
-Tell your app to hit an API endpoint

```
function getData (url) {  
  return fetch(url)  
    .then(response => response.json())  
    .then(beers => {  
      createBeerCard(beers)  
      console.log(beers)  
    })  
    .then(beers => flipCard(beers))  
}
```

Log Your Work!

-console.log() is EVERYTHING as a developer

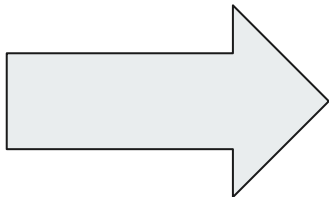
-It lets you see where your code is breaking, see what data your API is returning, and what kind of error you're getting!

A screenshot of a web browser's developer console, specifically the 'Console' tab. The console shows a log of an array of 10 beer objects. The first object is expanded, showing detailed properties like 'id', 'name', 'abv', 'attenuation_level', 'boil_volume', 'brewers_tips', 'description', 'first_brewed', 'food_pairing', 'ibu', 'id', 'image_url', 'ingredients', 'method', 'ph', 'srn', 'tagline', 'target_fg', 'target_og', 'volume', and '__proto__'. The array contains 10 objects, each with a unique ID and name, such as 'India Session Lager - Prototype Challenge', 'Comet', 'Baby Saison - B-Sides', 'Santa Paws', 'Original Dogma (Née Speedball)', 'Alpha Pop', 'Bavarian Weizen', 'Citra', '#Mashtag 2015', and 'Dead Pony Club'.

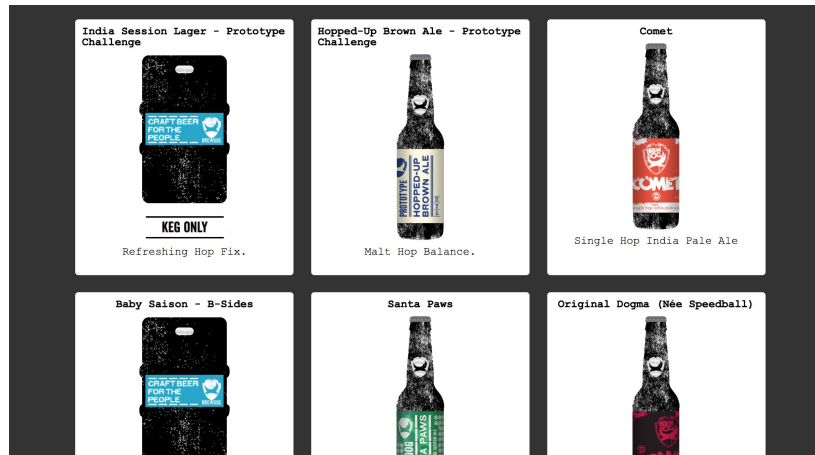
Moving From the Console to the Page

Go from HERE

```
Console Sources Network >> 1
▼ Filter Default levels ▾ Group similar 1 hidden
▼ Array(80)
  ▶ 0: {id: 81, name: "India Session Lager - Prototype Challenge"}
  ▼ 1:
    ▶ abv: 6.3
    ▶ attenuation_level: 82.6
    ▶ boil_volume: {value: 25, unit: "liters"}
    ▶ brewers_tips: "High carbonate water will really accentuate the malt character."
    ▶ contributed_by: "Sam Mason <sam@beers.com>"
    ▶ description: "Brown ales are perfect foils for resinous hops."
    ▶ first_brewed: "01/2010"
    ▶ food_pairing: (3) ["Butternut squash and sage soup", "Braised lamb shanks", "Roasted chicken"]
    ▶ id: 82
    ▶ image_url: "https://images.punkapi.com/v2/82.png"
    ▶ ingredients: {malt: Array(6), hops: Array(6), yeast: Array(1)}
    ▶ methods: {mash_temp: Array(1), fermentation: (1), bottling: (1)}
    ▶ name: "Hopped-Up Brown Ale - Prototype Challenge"
    ▶ phi: 4.2
    ▶ size: 12
    ▶ tagline: "Malt Hop Balance."
    ▶ target_fg: 1015
    ▶ target_og: 1063
    ▶ volume: {value: 20, unit: "liters"}
    ▶ ...proto...: Object
    ▶ 2: {id: 83, name: "Comet", tagline: "Single Hop India"}
    ▶ 3: {id: 84, name: "Baby Saison - B-Sides", tagline: "T"}
    ▶ 4: {id: 85, name: "Santa Paws", tagline: "Christmas Sc"}
    ▶ 5: {id: 86, name: "Original Dogma (Née Speedball)", ta}
    ▶ 6: {id: 87, name: "Alpha Pop", tagline: "Hoppy Bye Pal"}
    ▶ 7: {id: 88, name: "Bavarian Helzen", tagline: "Unleash"}
    ▶ 8: {id: 89, name: "Citrus", tagline: "Single Hop IPA Se"}
    ▶ 9: {id: 90, name: "Hopped-Up", tagline: "TWO Hopped"}
    ▶ 10: {id: 91, name: "Head Don't Ruin", tagline: "Head Don't Ruin"}
```



To here!





Make it Purdy

- CSS is what allows us to style HTML information
- Remember, we used JavaScript to move data to the HTML that the browser renders

```
body {  
  font-family: "Courier New", Courier, monospace;  
  background: □rgb(51, 51, 51);  
}  
  
header {  
  height: 10vh;  
  background: ■rgb(185, 255, 141);  
  width: 100vw;  
  display: flex;  
  align-items: center;  
  box-shadow: ■rgb(255, 105, 180) .5vw .5vw .5vw;  
}  
  
h1 {  
  color: ■rgb(255, 105, 180);  
  font-size: 4vw;  
  text-shadow: □rgb(51, 51, 51) .2vw .2vw .2vw;  
}
```



It's That Simple!

- Let us know what you think!
- What else are you curious about?
- Do you have any questions?