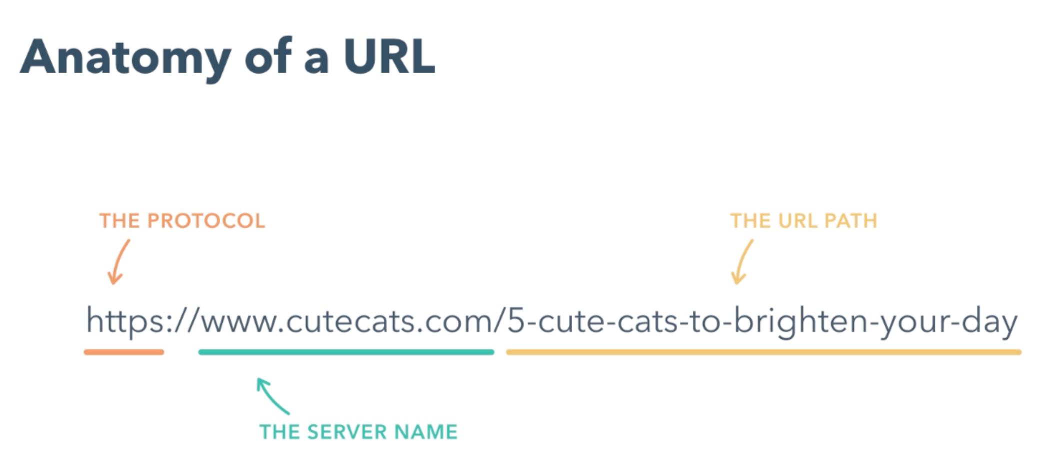
Web Application

A collection of code that's stored on a *remote* serve (not your computer) and delivered over the internet.

Server in laymen terms in just a fancy name for computers. Suppose you click on a link like cutecats.com which show you cute photos of cats, so the web site will send a request to the server for the images and it will get the responses from the server

Anatomy of a URL

The Protocol + The Server Name + The URL Path

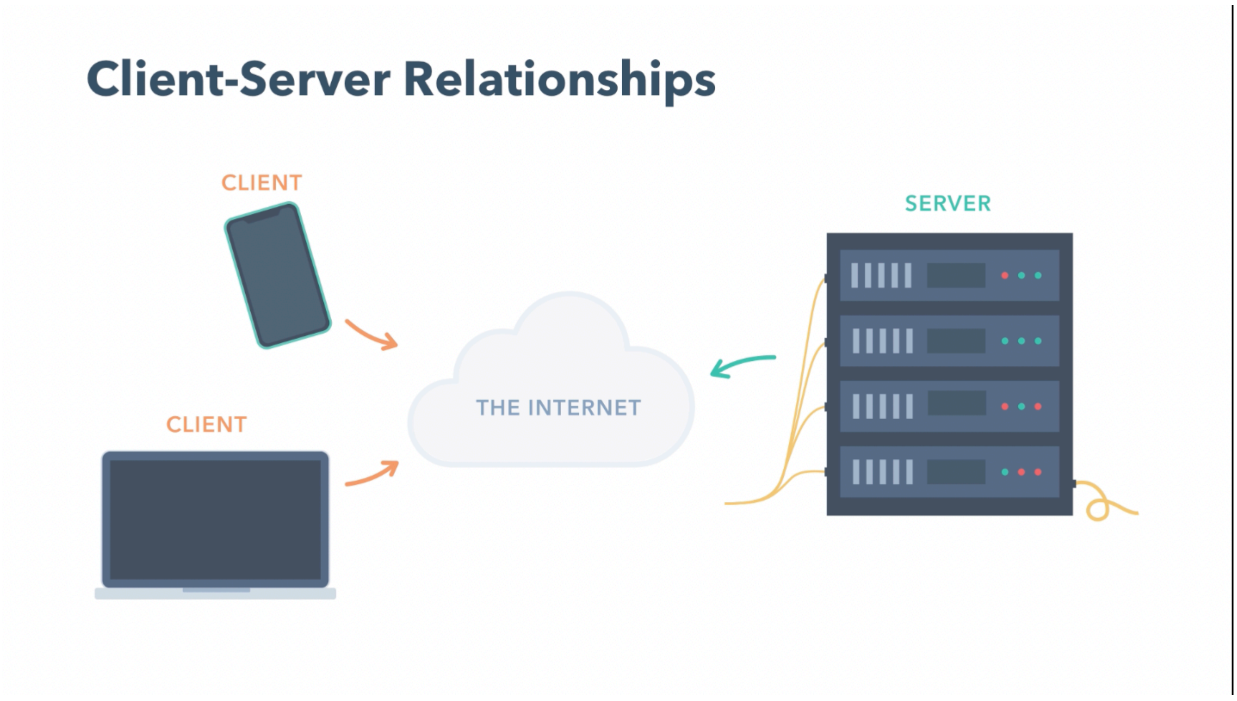


Protocols

- **Protocol:** It declares how your web browser should communicate with a web server when sending or fetching a web page or document. Just like English to communicate
- **HTTP (Hyper Text Transfer Protocol):** The most common protocol.
- **HTTPS:** Security-focused counterpart of HTTP, where the browser encrypts the data it's sending.

Servers and URLs

- **Server:** Stores and serves information (like your application).
- **Clients:** All the devices requesting that information (like your laptop or phone).
- **URL path:** It will tell you the path where the file you are looking for, is stored on the server

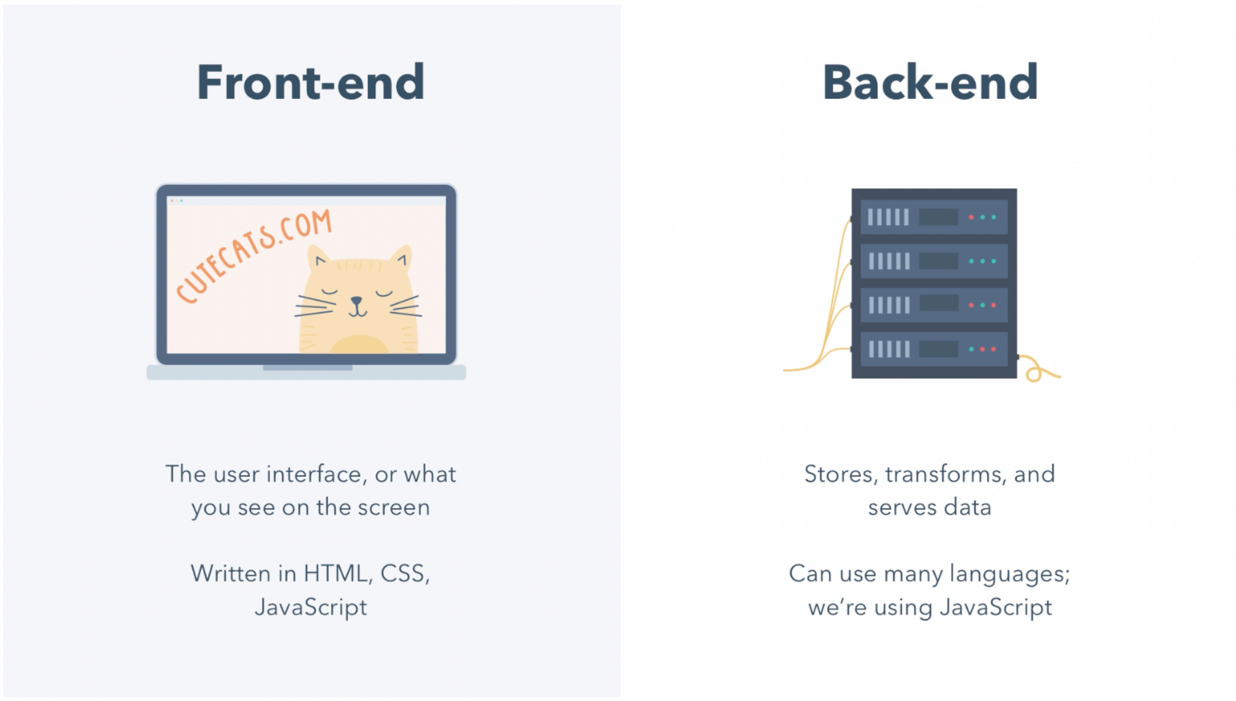


Environments

Applications can be served in different environments which might have different configurirations for different purposes.

- **Production:** Stable, secure and ready for traffic.
  - **Heroku**
    - Production server host
    - Cloud platform
    - Free service
    - Resources available
- **QA (or Quality Assurance):** Environment for testing an application on the server before it goes to users. SE uses most of them QA or Local.
- **Local:** A server on your compiuter, often used when writing code.

Front-end vs Back-end



Front-end

The user interface, or what you see on the screen

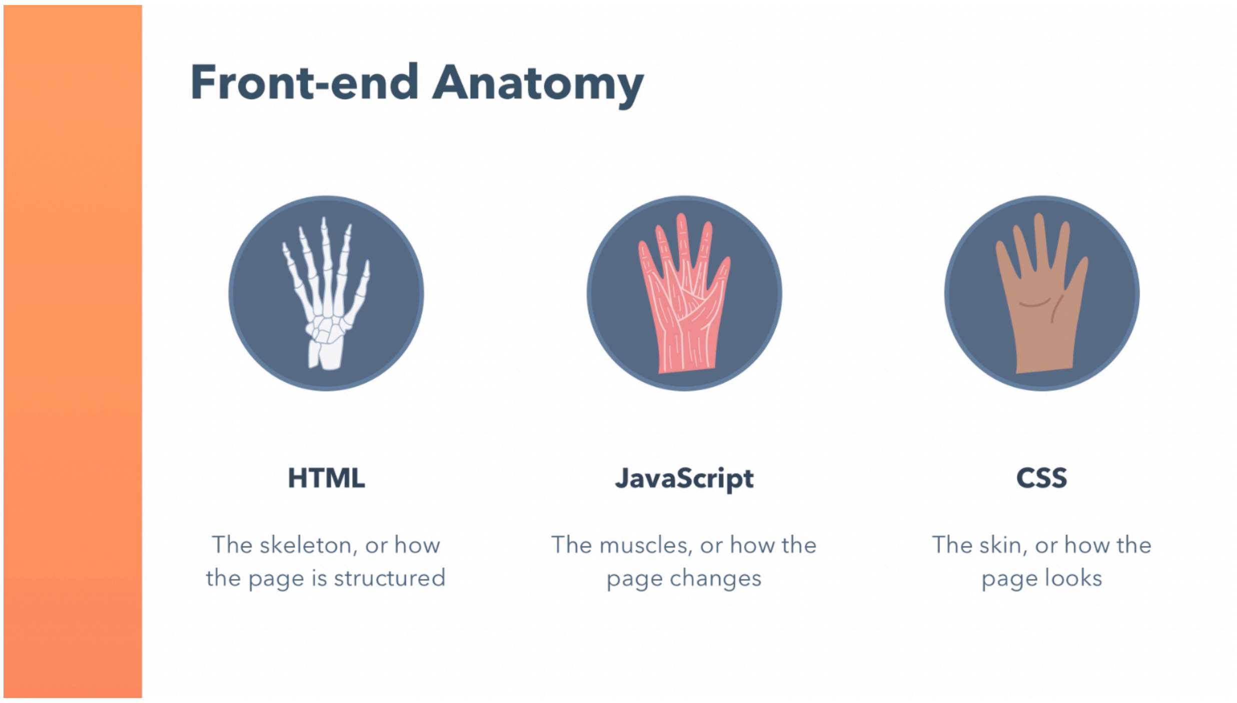
Written in HTML, CSS, JavaScript

Back-end

Stores, transforms, and serves data

Can use many languages; we're using JavaScript

Front-end Anatomy



Front-end Anatomy

HTML

The skeleton, or how the page is structured

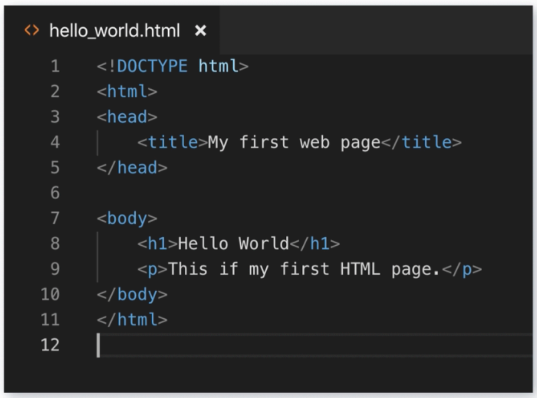
JavaScript

The muscles, or how the page changes

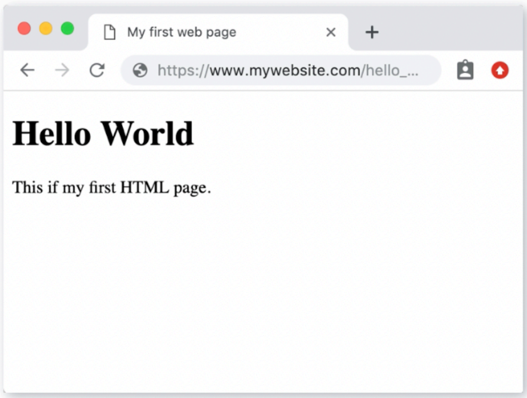
CSS

The skin, or how the page looks

HTML (The Skeleton)


hello_world.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>My first web page</title>
</head>

<body>
    <h1>Hello World</h1>
    <p>This if my first HTML page.</p>
</body>
</html>
```

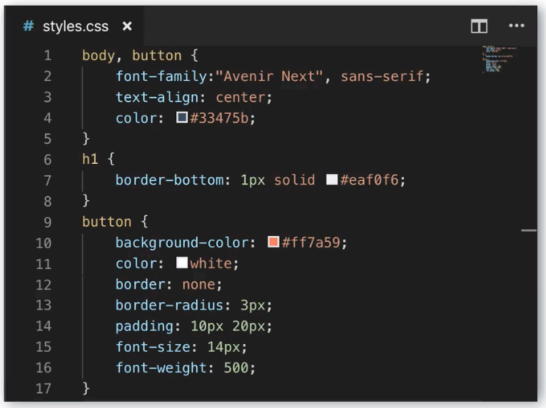

My first web page

https://www.mywebsite.com/hello_...

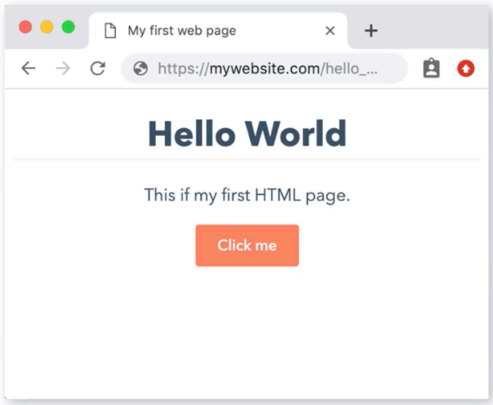
Hello World

This if my first HTML page.

CSS (The Skin)


styles.css

```css
body, button {
    font-family:"Avenir Next", sans-serif;
    text-align: center;
    color: #33475b;
}
h1 {
    border-bottom: 1px solid #eaf0f6;
}
button {
    background-color: #ff7a59;
    color: white;
    border: none;
    border-radius: 3px;
    padding: 10px 20px;
    font-size: 14px;
    font-weight: 500;
}
```

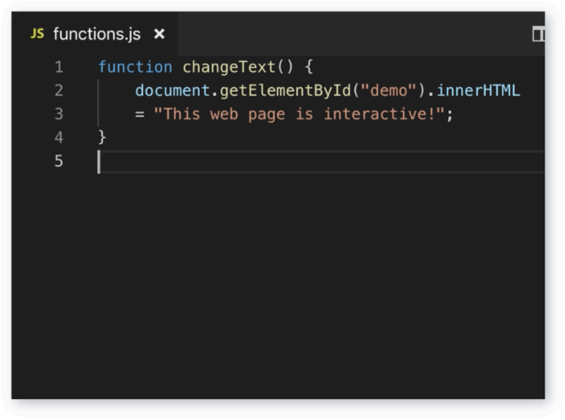

My first web page

https://mywebsite.com/hello_...

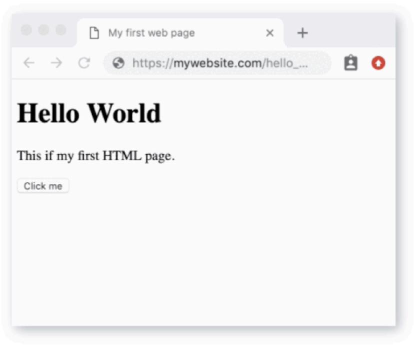
Hello World

This if my first HTML page.

JavaScript (The Muscles)



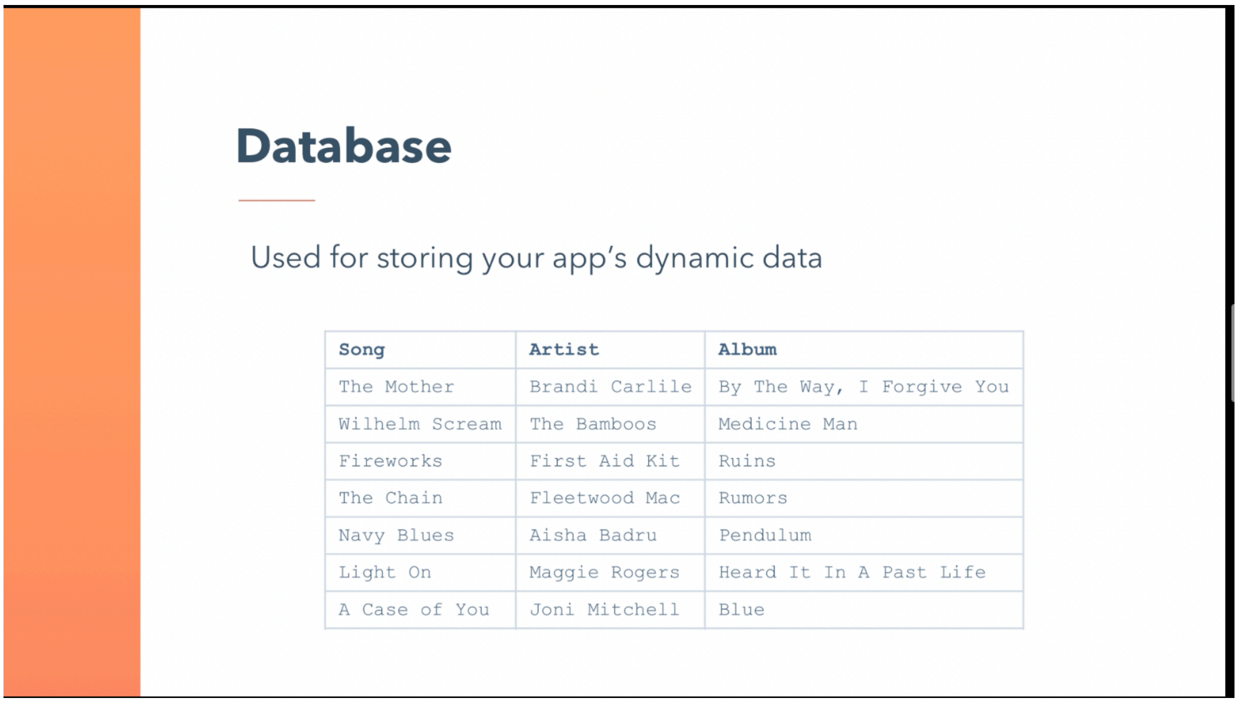
```
function changeText() {
    document.getElementById("demo").innerHTML
    = "This web page is interactive!";
}

```



Hello World

This if my first HTML page.

Back-end Anatomy



Database

Used for storing your app's dynamic data

Song	Artist	Album
The Mother	Brandi Carlile	By The Way, I Forgive You
Wilhelm Scream	The Bamboos	Medicine Man
Fireworks	First Aid Kit	Ruins
The Chain	Fleetwood Mac	Rumors
Navy Blues	Aisha Badru	Pendulum
Light On	Maggie Rogers	Heard It In A Past Life
A Case of You	Joni Mitchell	Blue

To access this data we use APIs

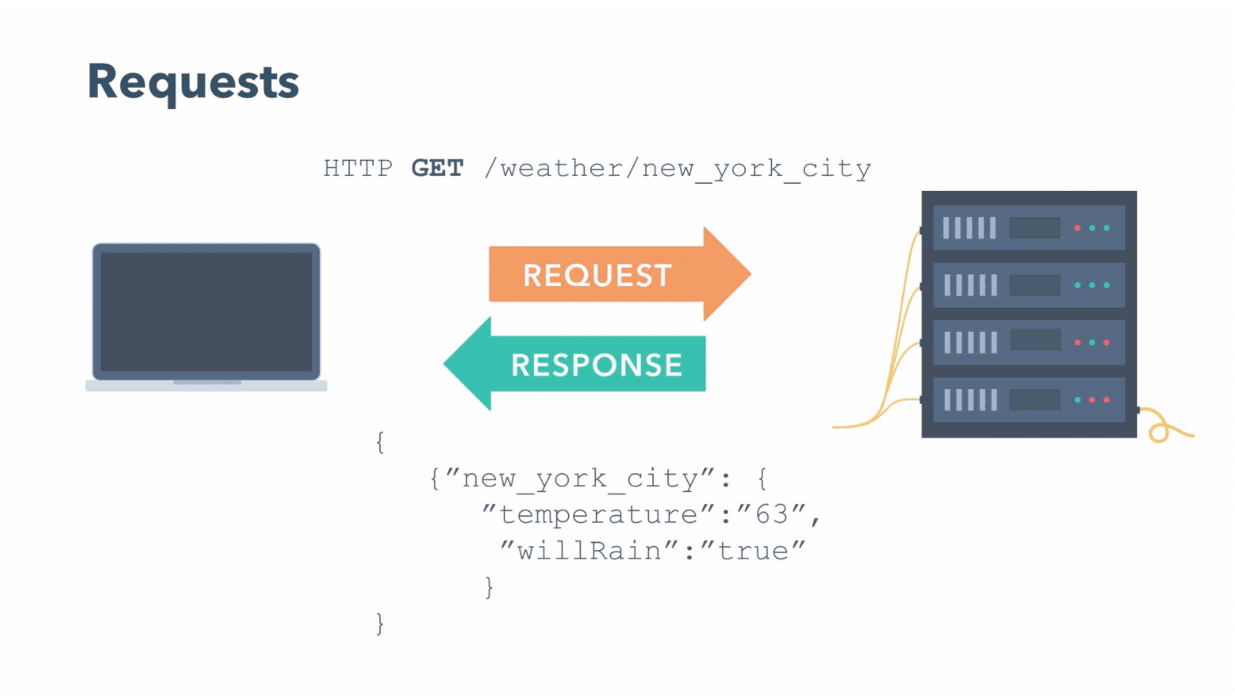
APIs

- Stands for Application Programming Interface.
- How the front-end of your app communicates with the back end.
- Lets you retrive and modify data in your database.
- It is a set of operations - with specific inputs and outputs - used for interacting with

something, like a database, web service, or hardware.

We can make these APIs requests from javascript.

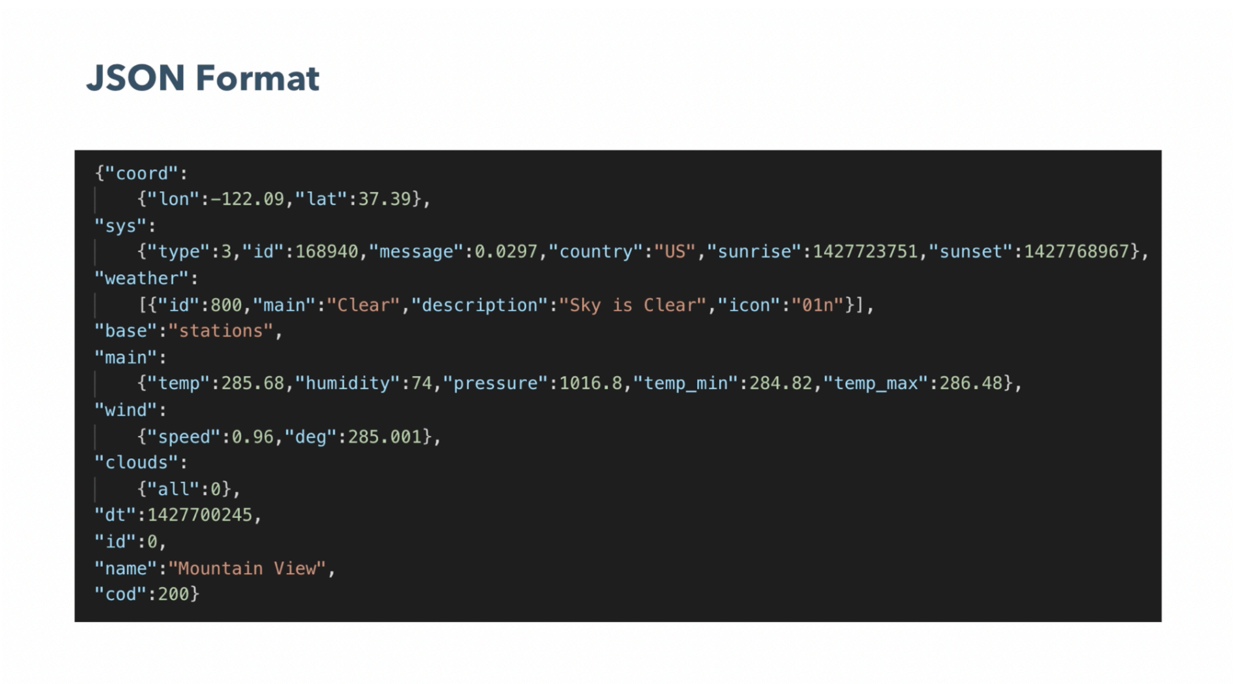
Example:-



```
HTTP GET /weather/new_york_city
```

```
{
   {"new_york_city": {
      "temperature":"63",
       "willRain":"true"
      }
}
```

We can make a **GET** request using **HTTP**, (the sam eprotocol which allows us to retrive a webpage), the serve will then send the response in the form of **JSON** (JavaScript Object Notation) in organised format like in ***key : values*** format.



```
{"coord":
    {"lon":-122.09,"lat":37.39},
"sys":
    {"type":3,"id":168940,"message":0.0297,"country":"US","sunrise":1427723751,"sunset":1427768967},
"weather":
    [{"id":800,"main":"Clear","description":"Sky is Clear","icon":"01n"}],
"base":"stations",
"main":
    {"temp":285.68,"humidity":74,"pressure":1016.8,"temp_min":284.82,"temp_max":286.48},
"wind":
    {"speed":0.96,"deg":285.001},
"clouds":
    {"all":0},
"dt":1427700245,
"id":0,
"name":"Mountain View",
"cod":200}
```

How do APIs work?

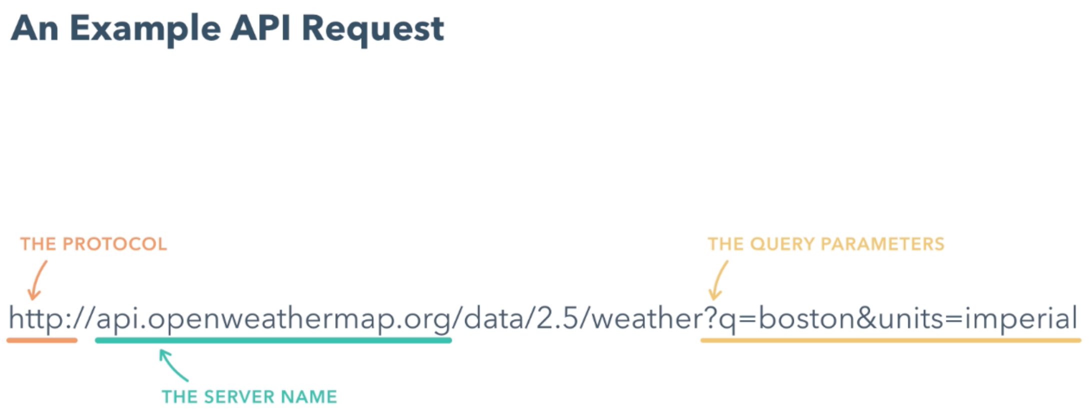
- Use a protocol, like HTTP
- Send an HTTP request to a URL
- Common HTTP requests/methods: **GET**, **POST**, **DELETE**, and **PUT**.

GET Request

An HTTP request type used for retrieving data from a service.

POST Request

An HTTP request type used for sending data to a service.


An Example API Request
THE PROTOCOL
THE QUERY PARAMETERS
http://api.openweathermap.org/data/2.5/weather?q=boston&units=imperial
THE SERVER NAME

To prevent abuse of the API, the user is provided with the **API Key(Access Token)**

API Key(Access Token):

- A string sent along with an API request to signify an authorized user.
- It is usually sent as a query parameter like **API_KEY="sdvwvdiwv217t23"**, which lets the API know that you are authorized user.
- API Keys are **sensitive information** - *do not expose them* in your code because there are bots that are continuously scraping **GitHub** looking for API Keys to abuse.
- You can hide your API Key by setting it as an **environment variable** on **Heroku**

AJAX

It is a technique for sending and receiving data from a server.

AJAX success and error handlers:

- Functions that get called based on the result of your API request.
- Success handler function called when request comes back with desried data.
- Error handler function called when something goes wrong with the request.

Database

An orderly way of storing your app's data.

Firebase

- Fast.
- Free for simple apps.
- Handles authentication easily.