Fill out the attendance form at acm.mines.edu/attend

Join our Discord

Join our Mailing List

Introduction to the Web

By Ethan Richards

mines acm

HTTP & HTTPS

HTTP (HyperText Transfer Protocol) is the basis of the web

- It defines **methods** (next slide)
- It defines **Requests** and **Responses**
- *..and more!*

In a nutshell, HTTPS is secure HTTP (TLS encryption)

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

HTTP Request Methods

GET - Retrieves data

POST - Sends data to the server

PUT - Creates/replaces a resource

DELETE - Deletes something

HTTP Request Methods follow the **CRUD Operations**: **C**reate (POST), **R**ead (GET), **U**pdate (PUT), **D**elete (DELETE)

For more: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

Application Programming Interface (API)

What does that actually mean?

Application Programming Interface (API)

What does that actually mean?

You wouldn’t want people to have direct access to your application, right?

Application Programming Interface (API)

What does that actually mean?

You wouldn’t want people to have direct access to your application, right?

An API is a tool for developers to safely interact with a program, made by the developers of the original program. It’s an “outward-facing part” of an app.

See this StackOverflow post for differences between libraries, frameworks, and APIs.

REST Principles

REST stands for **RE**presentational **S**tate **T**ransfer is a software architecture that makes things easy to send on the web.

REST architectures are **stateless** and have separate **clients** and **servers**.

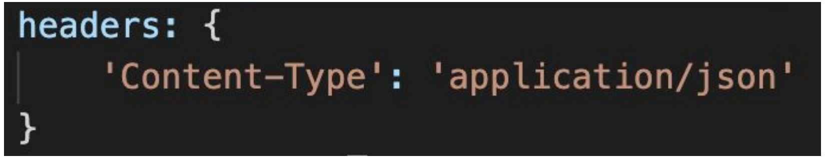
- *Clients send requests, servers send responses!*
- The frontend and backend may be “decoupled” to be “truly RESTful”

You’ll hear “**RESTful APIs**” and similar terms often. It’s nothing to be afraid of!

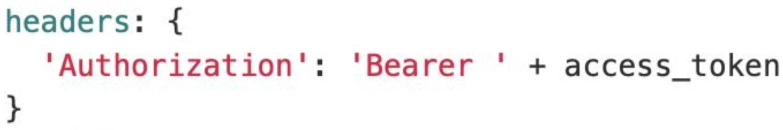
HTTP Headers & Bodies

Headers

Headers to a request indicate what type of data is to be expected and other useful information.



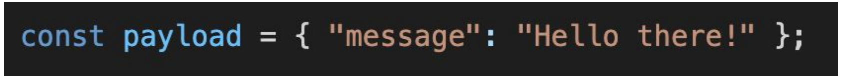
```
headers: {
    'Content-Type': 'application/json'
}
```



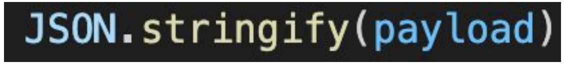
```
headers: {
  'Authorization': 'Bearer ' + access_token
}
```

Content Bodies

HTTP requests can have content bodies, too. This is typically done in JSON format. Sometimes this is called a "payload."



```
const payload = { "message": "Hello there!" };
```



```
JSON.stringify(payload)
```

Read more: HTTP Headers, Request Bodies, JSON.stringify()

HTTP Response Codes

200 - OK!

400 - Bad Request

401 - Unauthorized

403 - Forbidden

404 - Not Found

405 - Method Not Allowed

418 - I'm a teapot

500 - Internal Server Error

504 - Gateway Timeout

Mozilla classifies HTTP response codes in five different number ranges:

Informational (100-199)

Successful (200-299)

Redirection (300-399)

Client error (400-499)

Server error (500-599)

Read more: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Programming time!

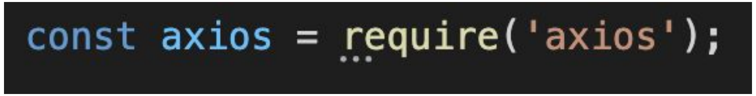
Open up your editor of choice - my examples will use JavaScript or Python.

JavaScript

npm init

npm install axios

touch index.js

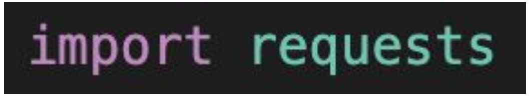


```
const axios = require('axios');
```

Python

pip install requests

touch app.py



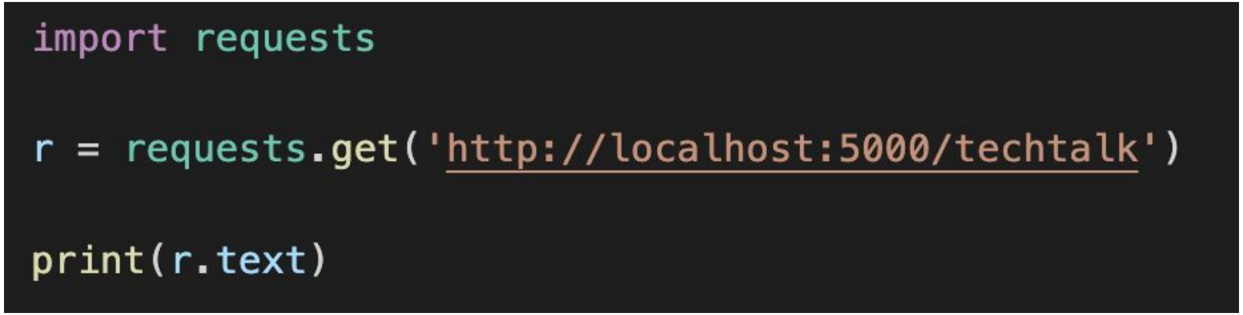
```
import requests
```

https://localhost:8000/techtalk

I have a message for you - come GET it from me!

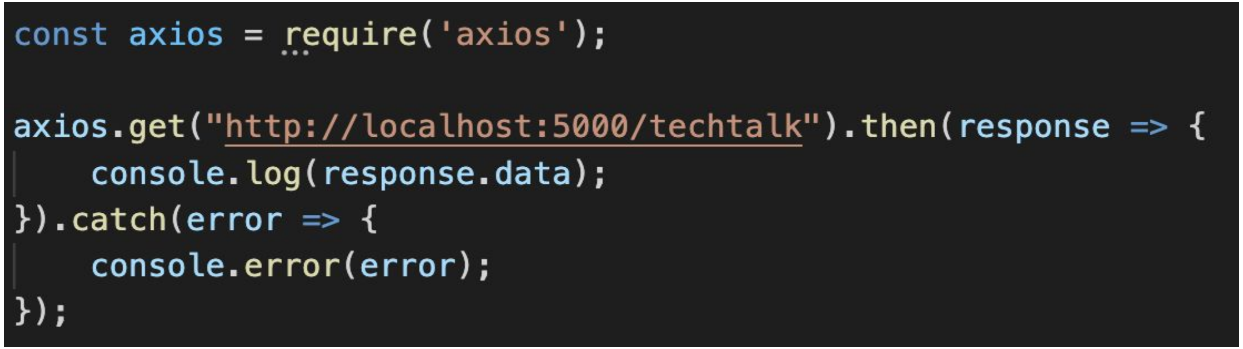
This is called an **endpoint**! Servers host endpoints for data to be accessed that way.

Python GET



```python
import requests

r = requests.get('http://localhost:5000/techtalk')

print(r.text)
```

JavaScript GET

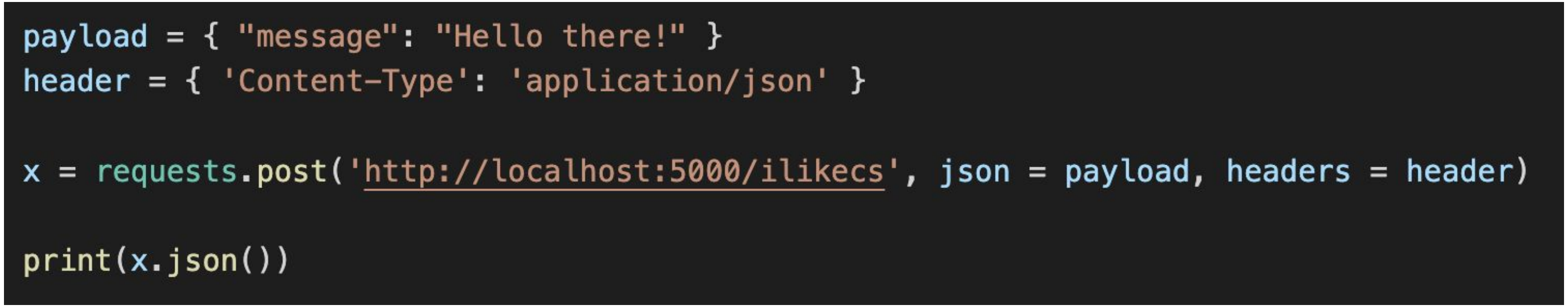


```javascript
const axios = require('axios');

axios.get("http://localhost:5000/techtalk").then(response => {
    console.log(response.data);
}).catch(error => {
    console.error(error);
});
```

https://localhost:8000/ilikecs

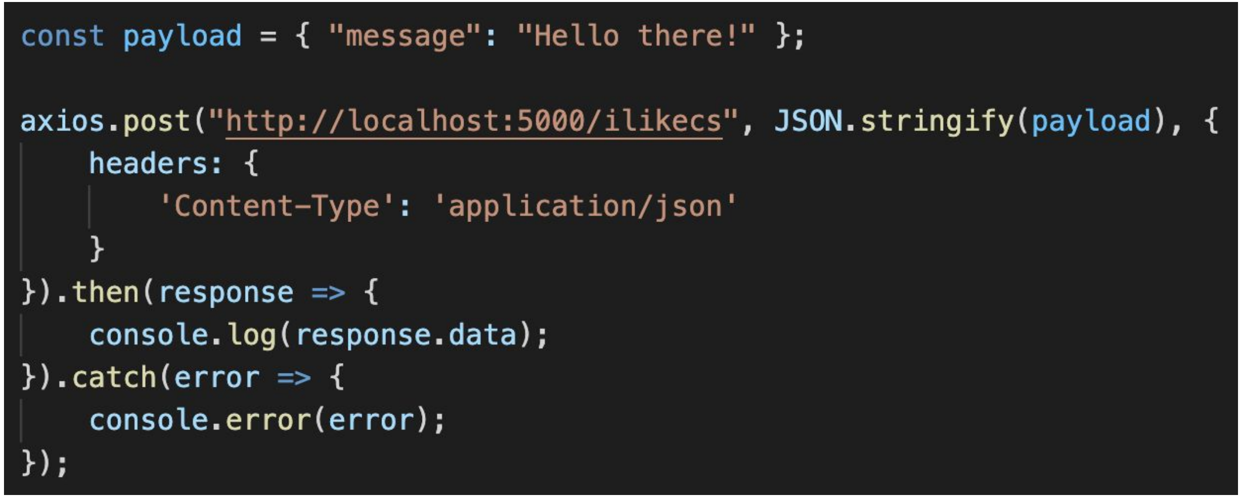
Send me a message via HTTP!

Python POST



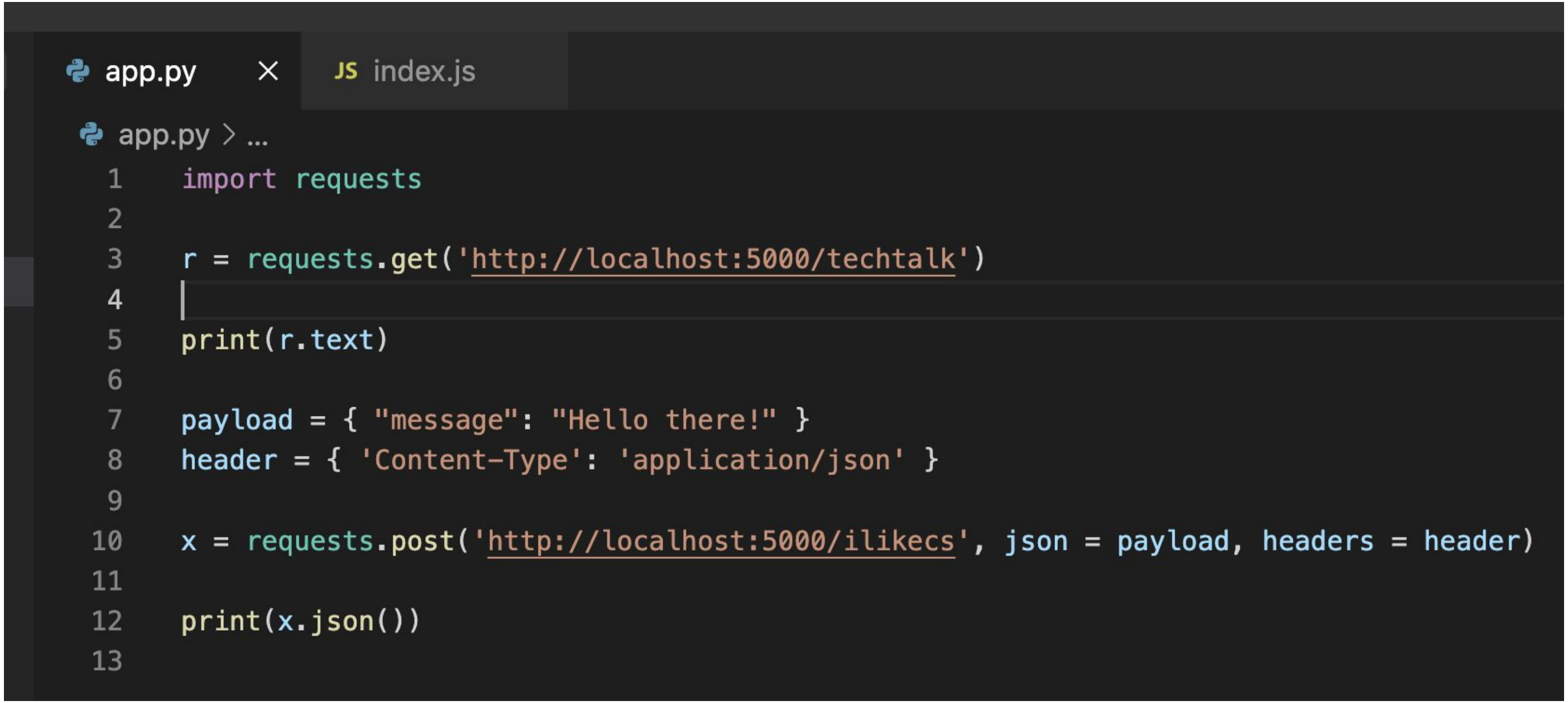
```python
payload = { "message": "Hello there!" }
header = { 'Content-Type': 'application/json' }

x = requests.post('http://localhost:5000/ilikecs', json = payload, headers = header)

print(x.json())
```

JavaScript POST



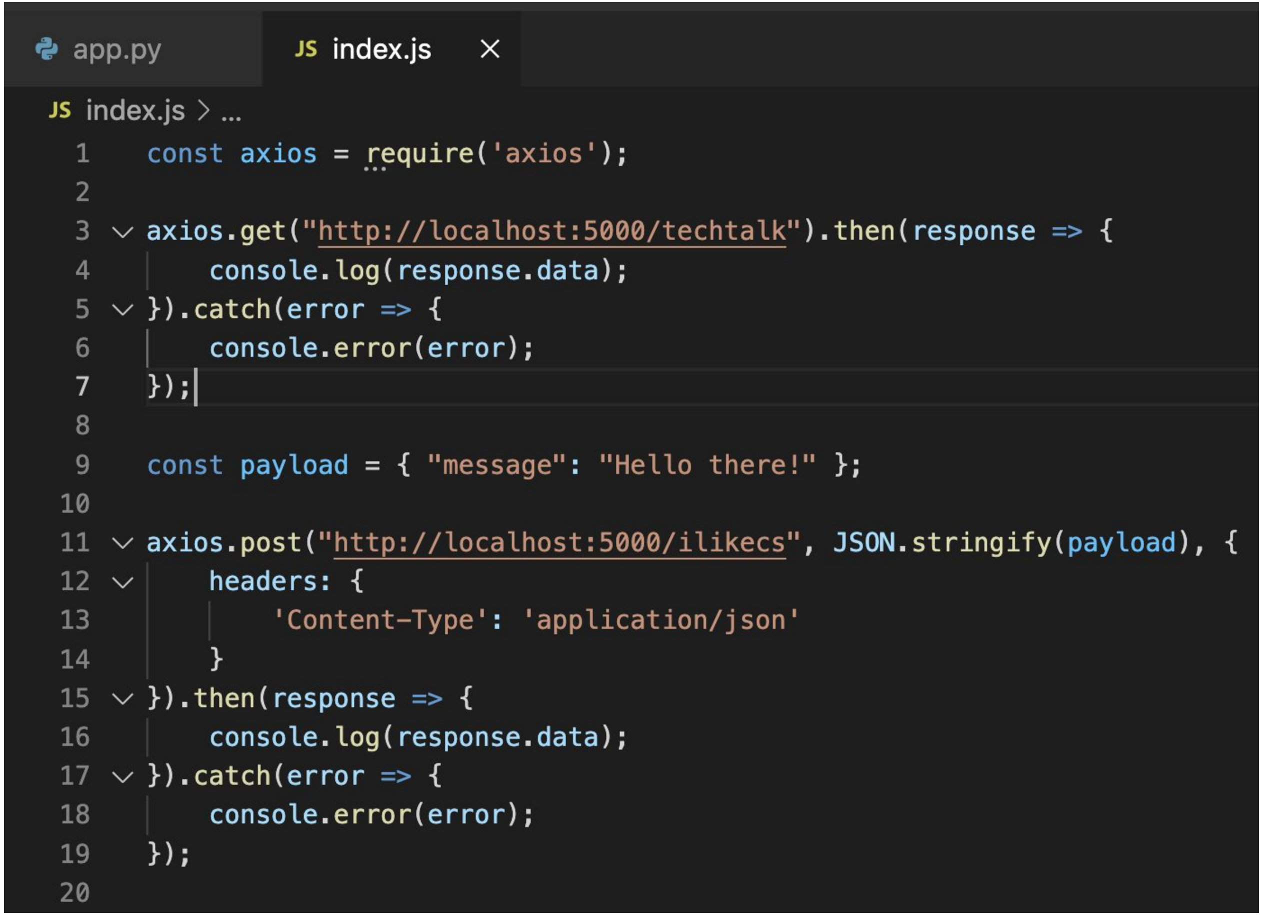
```javascript
const payload = { "message": "Hello there!" };

axios.post("http://localhost:5000/ilikecs", JSON.stringify(payload), {
    headers: {
        'Content-Type': 'application/json'
    }
}).then(response => {
    console.log(response.data);
}).catch(error => {
    console.error(error);
});
```

Full Python Code



```python
import requests

r = requests.get('http://localhost:5000/techtalk')

print(r.text)

payload = { "message": "Hello there!" }
header = { 'Content-Type': 'application/json' }

x = requests.post('http://localhost:5000/ilikecs', json = payload, headers = header)

print(x.json())

```

Full JavaScript Code



```javascript
const axios = require('axios');

axios.get("http://localhost:5000/techtalk").then(response => {
    console.log(response.data);
}).catch(error => {
    console.error(error);
});

const payload = { "message": "Hello there!" };

axios.post("http://localhost:5000/ilikecs", JSON.stringify(payload), {
    headers: {
        'Content-Type': 'application/json'
    }
}).then(response => {
    console.log(response.data);
}).catch(error => {
    console.error(error);
});

```