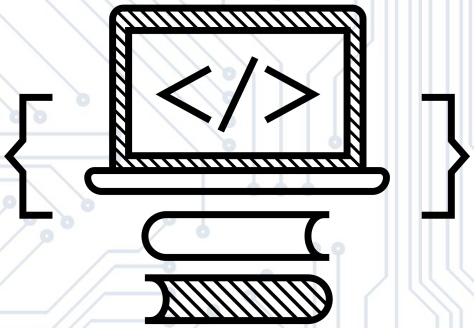
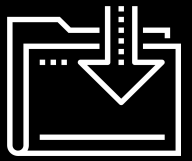


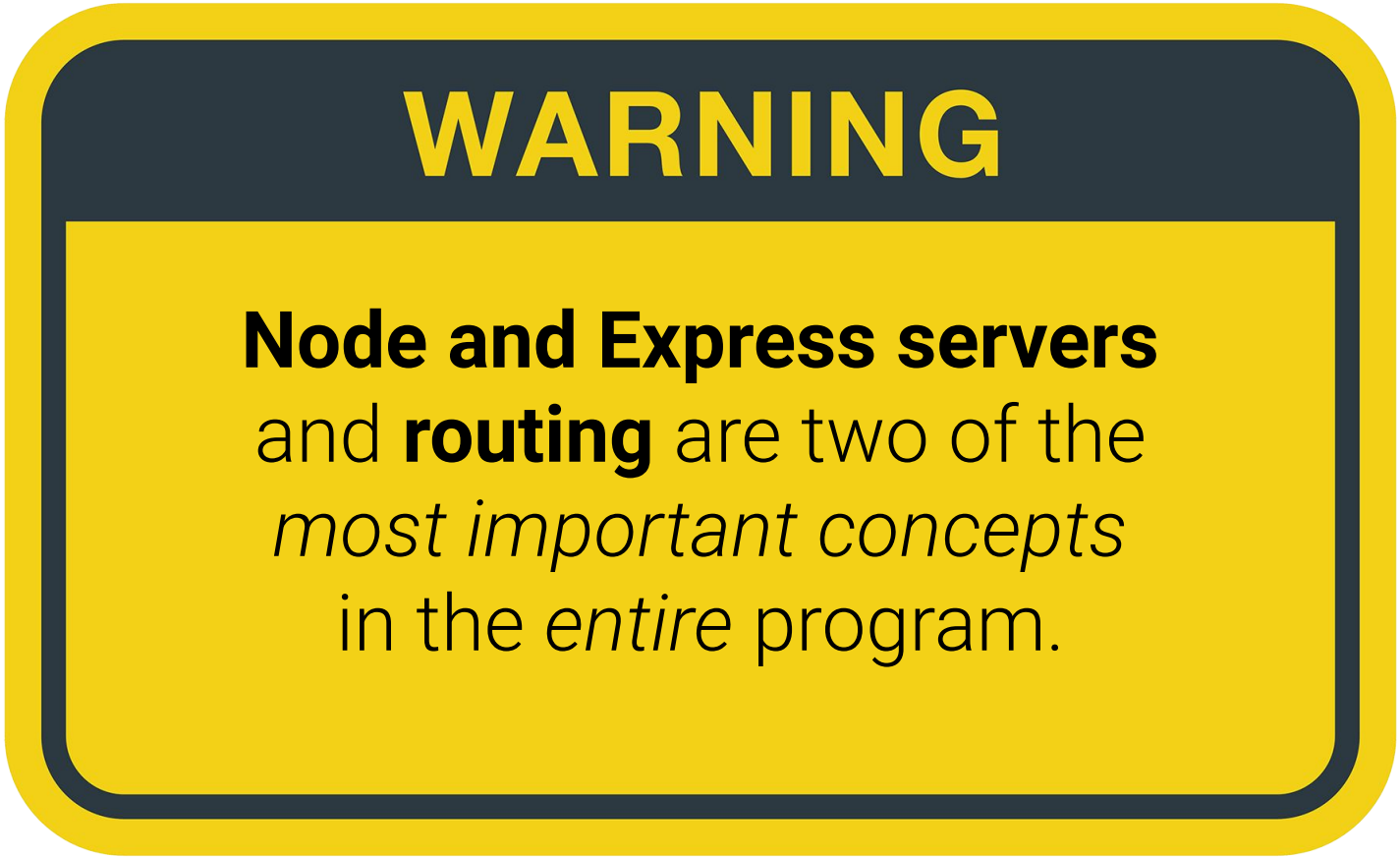
Intro to Node Servers

Web Development Boot Camp

Lesson 11.1



A Moment of Caution


WARNING
Node and Express servers
and routing are two of the
most important concepts
in the entire program.

Forewarning: This Is the Challenging Stuff

The topics we'll cover over the next three weeks will be challenging at times, but it's also some of the most important material. This is when you'll go from humble HTML/CSS hackers to employable full-stack engineers. Bring your A game!

Be Confident!

Remember: If you've made it this far, you've proven that you have what it takes to succeed. Keep going!

Tips for Success

01

Form a study group. Get in the habit of explaining in-class exercises to one another, and work together on homework assignments.

02

Take notes! Jot down key concepts and ideas that come out of the in-class activities to help you keep things straight!

03

Ask "conceptual" questions. It's important to understand the big picture, so if a key concept is confusing, be sure to ask about it!

04

Come to office hours. Seeking help during office hours will be increasingly valuable as the material becomes more complex. Come to these sessions and ask questions! We're here to help you.

Want a Helpful Resource?

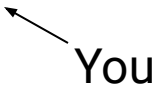
Sign up for a Pluralsight trial account and take their Building Blocks of Express.js course.

pluralsight.com

Remember Our Mantra:

When it comes to web development...

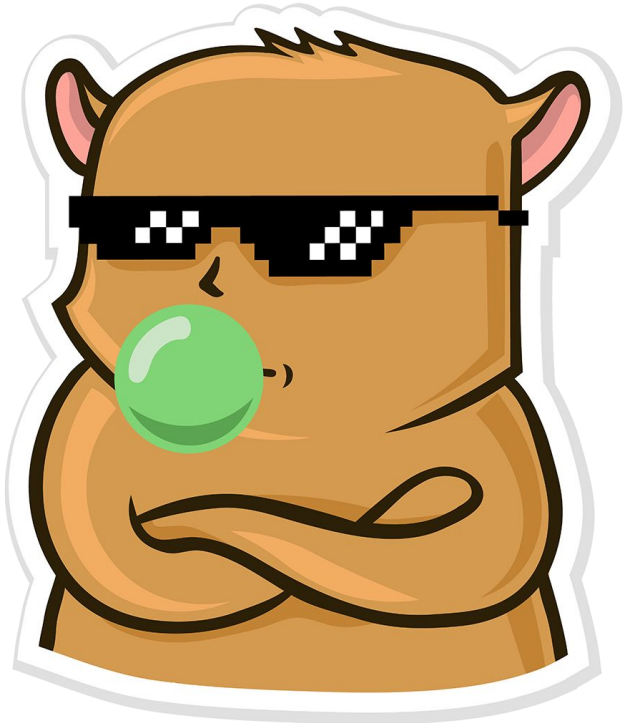
I know nothing.


You

So Let’s Begin

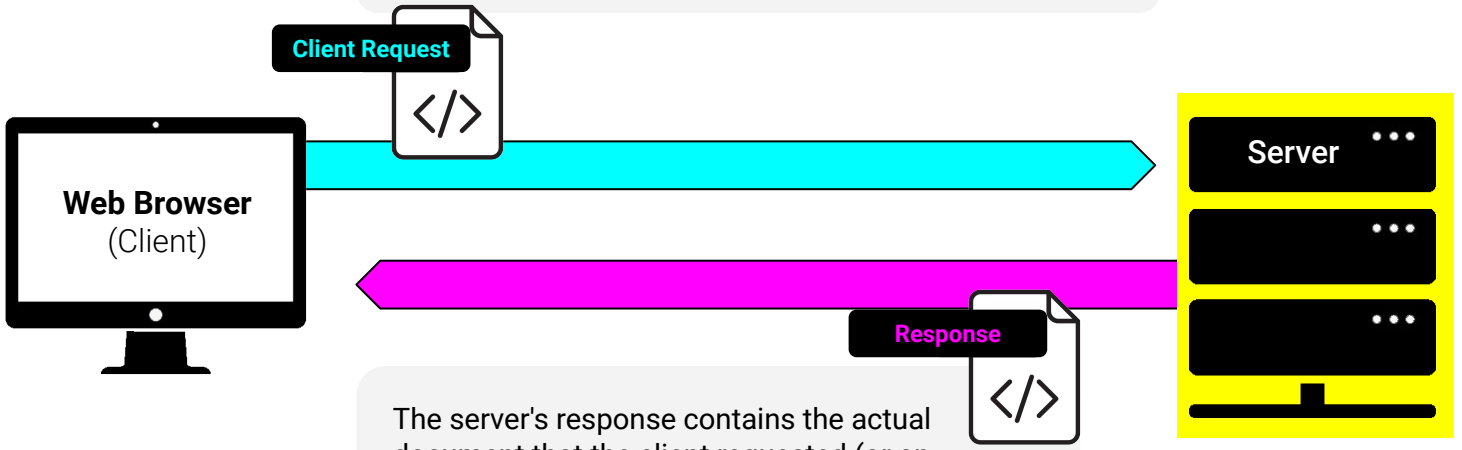
Remind Me Again

What is a **server**?



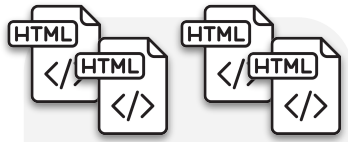
What Is a Server?

A server is the **machine** and **code** that handles client requests and responds to them.



The client's request contains the name and address (the URL) of whatever the client is looking for.

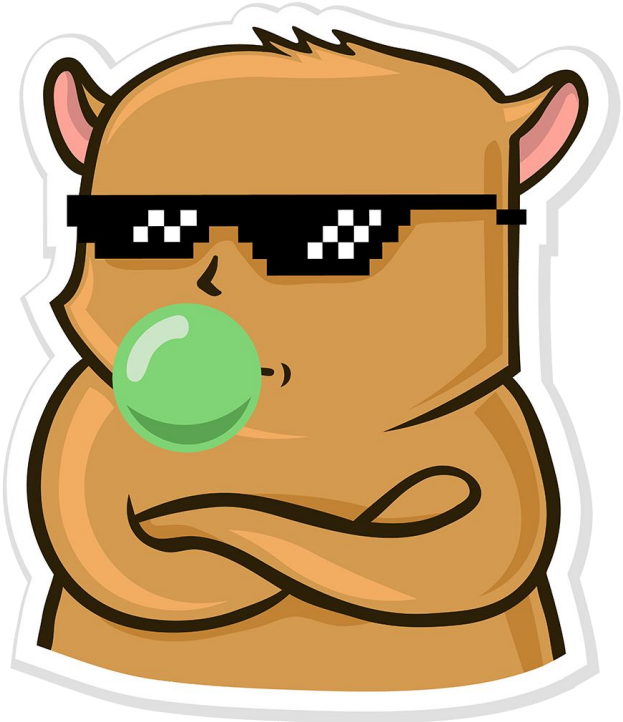
The server's response contains the actual document that the client requested (or an error code if the request cannot be processed).



The server usually has a lot of content that it can send to clients. This content can be webpages, JPEGs, and other resources.

Remind Me Again

What are some examples of **server-side** functions?



Server-Side Code in Action!

Visiting a URL and then being presented with an HTML page

Visiting an API endpoint that parses URL parameters to provide selective JSONs

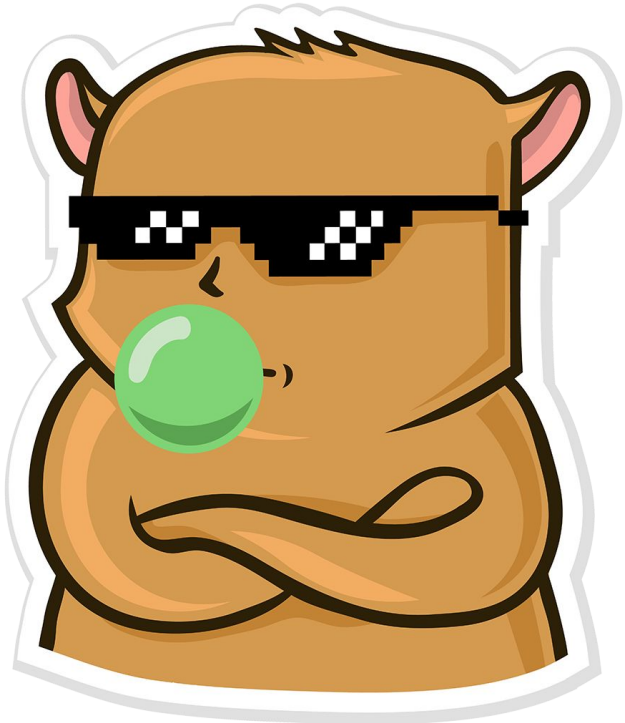
Clicking an invoice that provides a PDF report

Image-processing software that applies a filter to an image and saves the new version

Google providing results relevant to searches on other sites

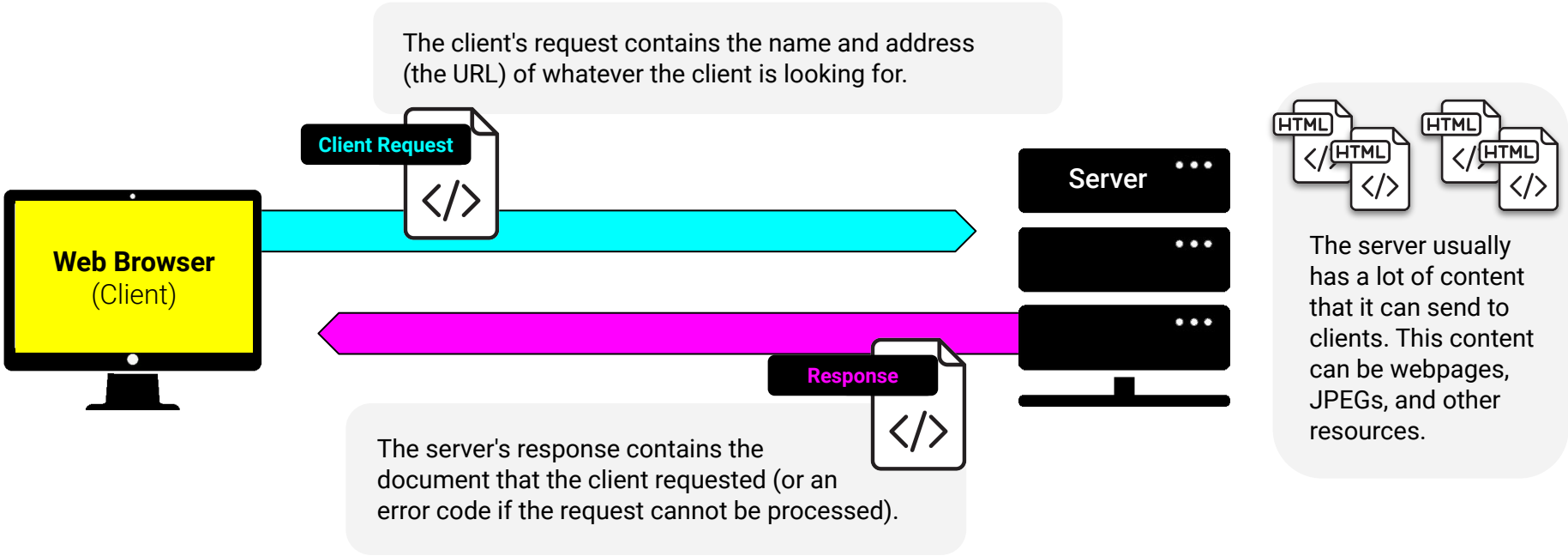
Remind Me Again

What is a **client**?



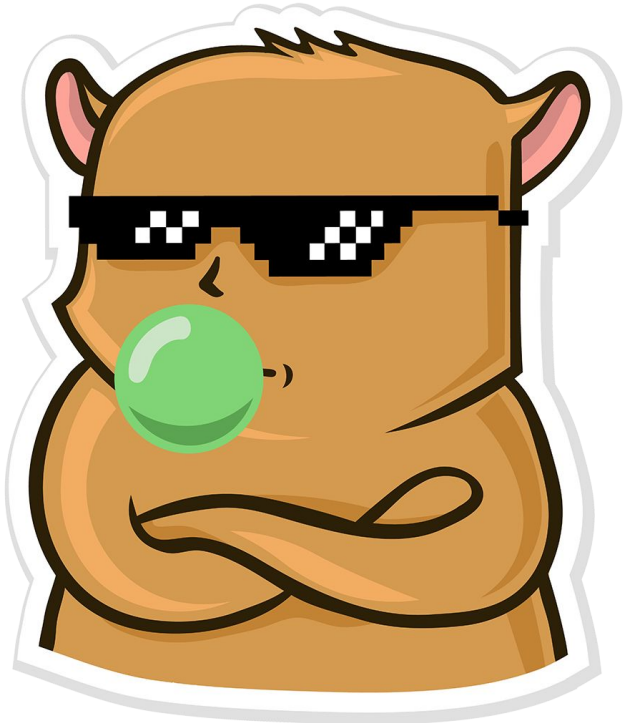
What Is a Client?

A **client** is a user's personal machine which makes "requests" of the server.



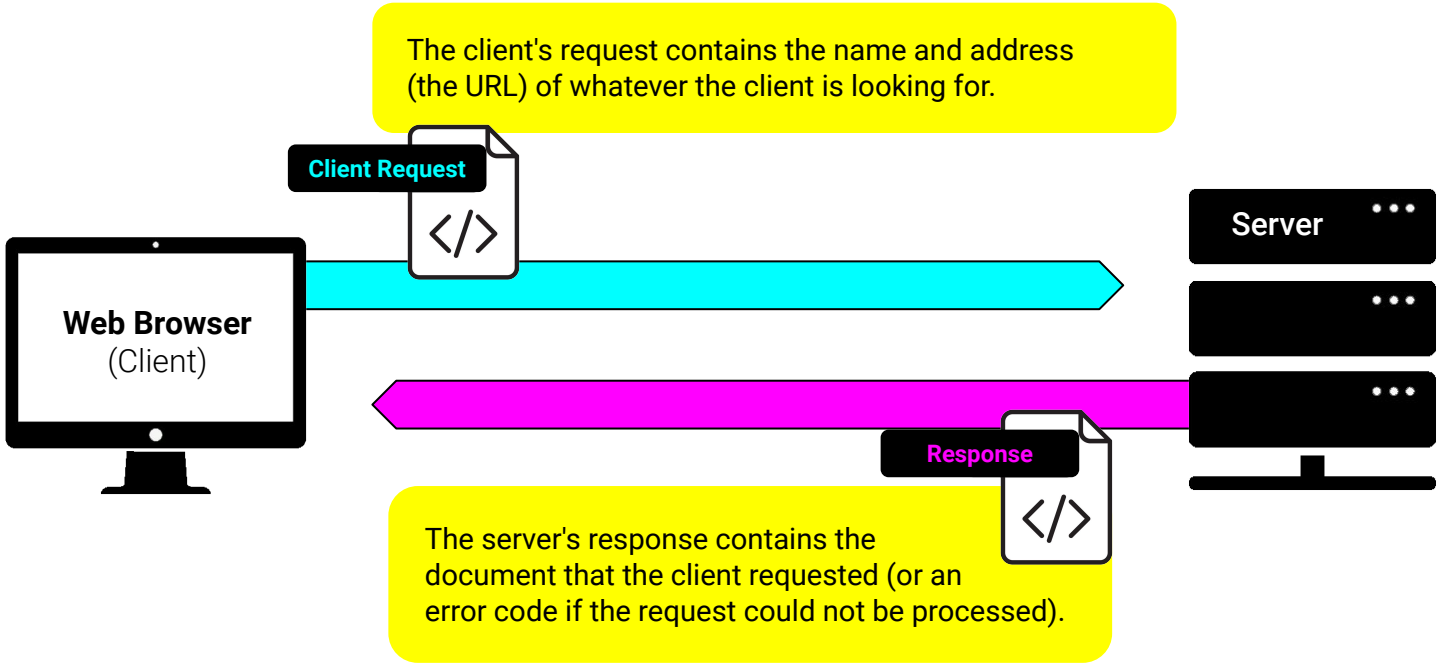
Remind Me Again

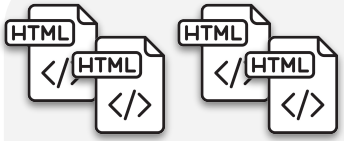
How do the client and server **communicate** with each other?



HTTP/HTTPS

Clients and servers communicate with each other using a series of requests and responses defined by **HTTP / HTTPS (Hypertext Transfer Protocol Secure).**

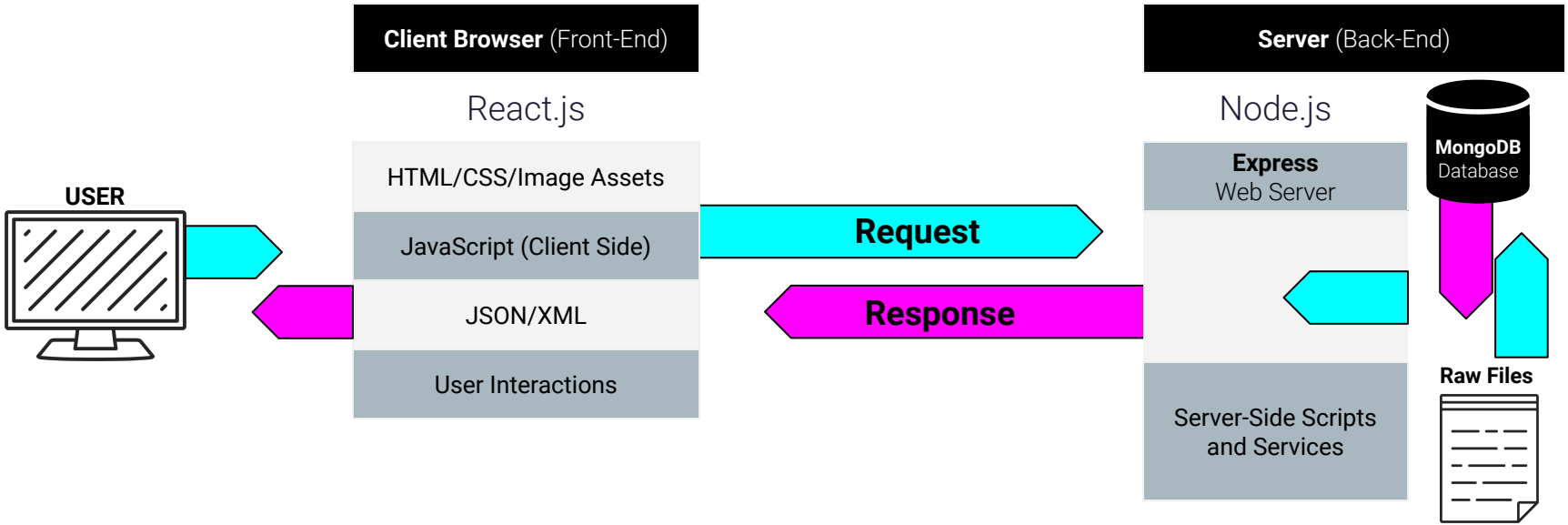




The server usually has a lot of content that it can send to clients. This content can be webpages, JPEGs, and other resources.

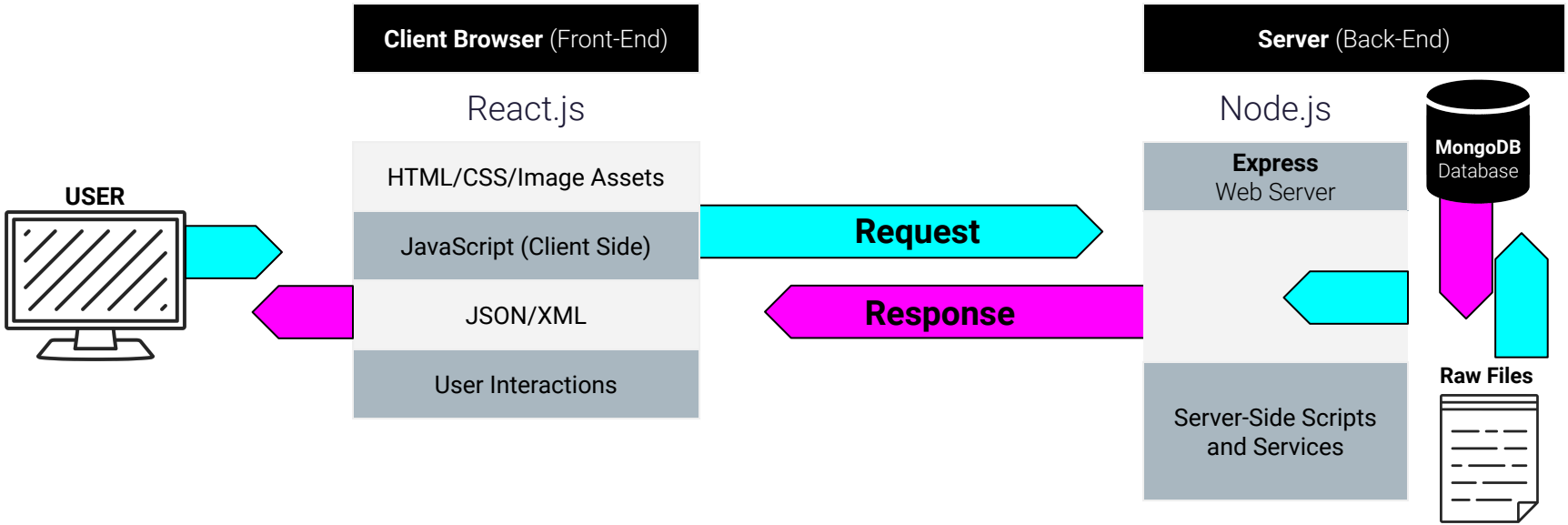
Full-Stack Development

In modern web applications, there is constant back-and-forth communication between the visuals displayed on the user's browser **(front-end)** and the data and logic stored on the server **(back-end)**.



Full-Stack Development

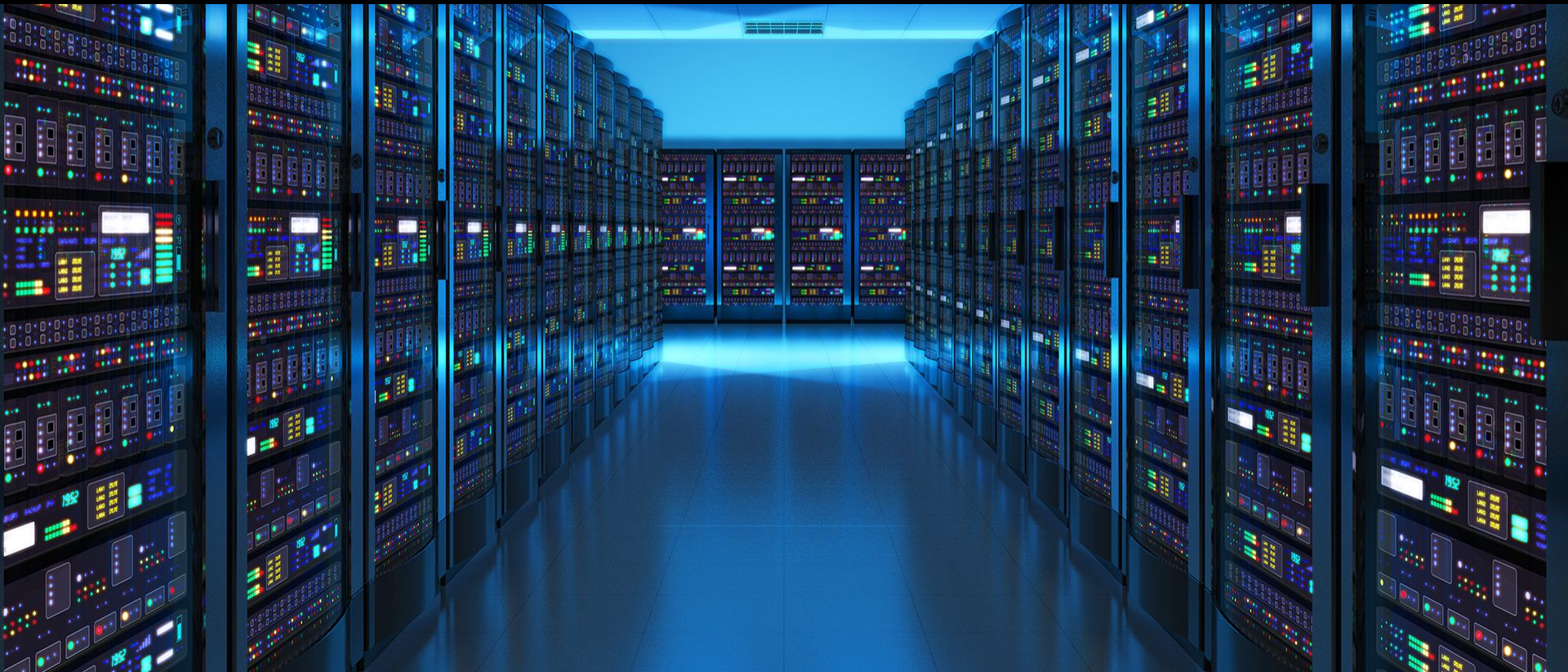
You can think of the server and the client as two distinct “machines.”



Deep Dive into Servers

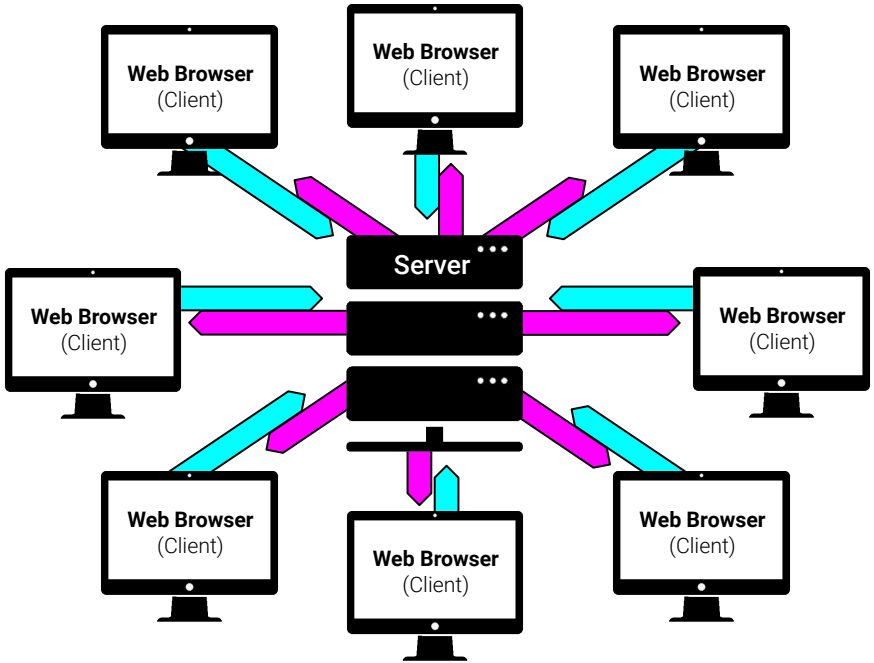
Visualizing Servers

Server hardware looks like large, ruggedized versions of desktop computers.



Visualizing Servers

These machines and their code handle all requests coming in from the browsers accessing a website.



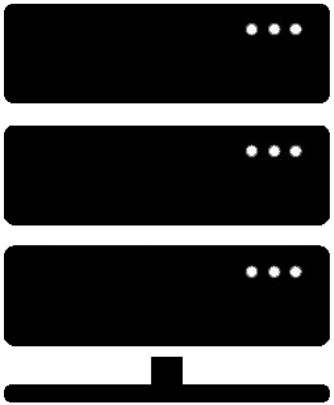
Visualizing Servers

Where does the server live?

Where Do Servers Live?

Servers live in dedicated hardware intended to handle the requests and responses from many clients.

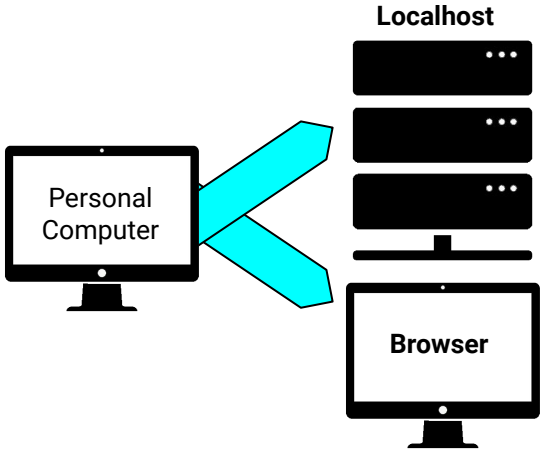
Servers most often live on cloud platforms like AWS, Heroku, and Google Cloud.



Servers During Development

During development, our personal computers will simulate both the client and the server.

We will create a local server (Localhost) on our computer and then use our browser to interact with it.



Building a Server

Building a Server

For our purposes, “building a server” means writing code that handles what the server will do.

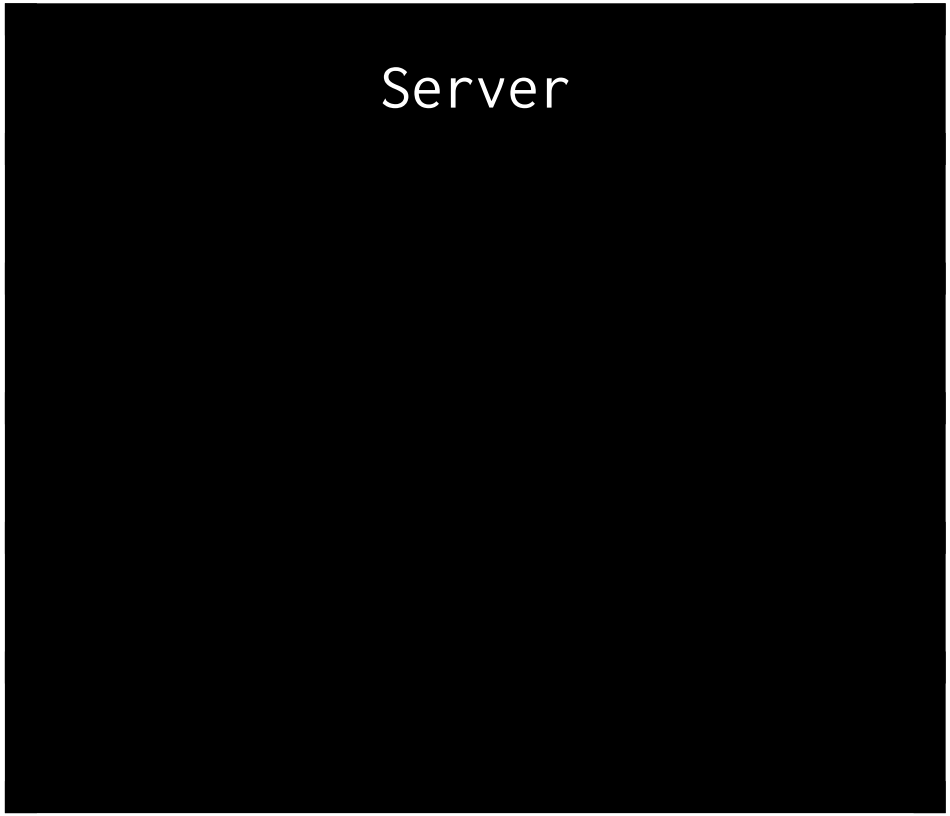
It’s important to note that even though you pay for server-side hardware, you still need to write the code that goes inside.

This code enables functions such as:

- Establishing connections to the database
- Handling client-side URL requests
- Performing server-side processes
- Authenticating user requests
- Logging client requests

Your Server: A Big Box

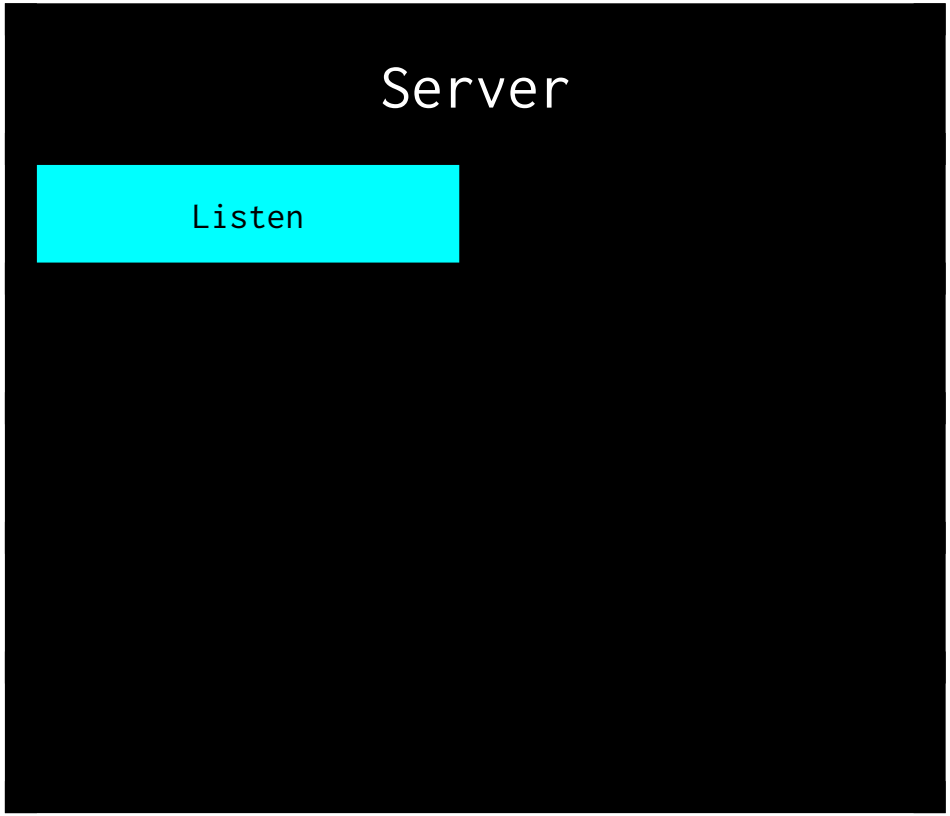
- Throughout this week, imagine your server as a **big, empty box**.
- We will add **code** snippets and **modules** to give our big, empty box the power to do stuff in response to all the requests that come in.



Inside the Box: Connections

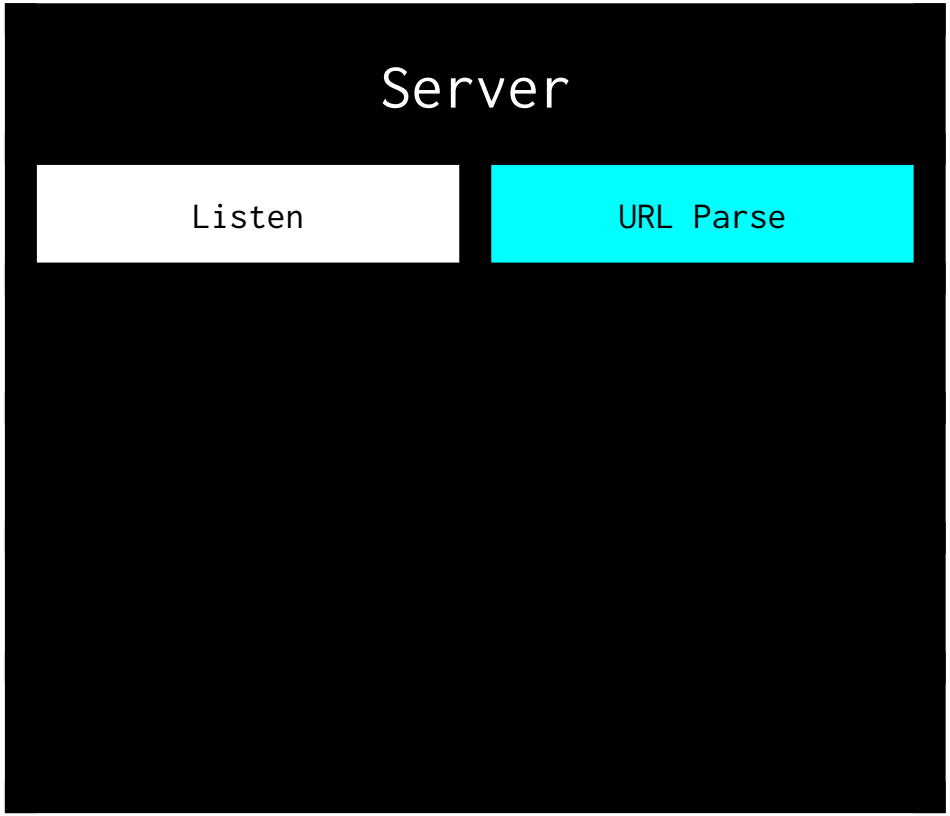
How do we allow the client to connect to the server?

We'll add a listener to the server that allows it to **listen** for client requests.



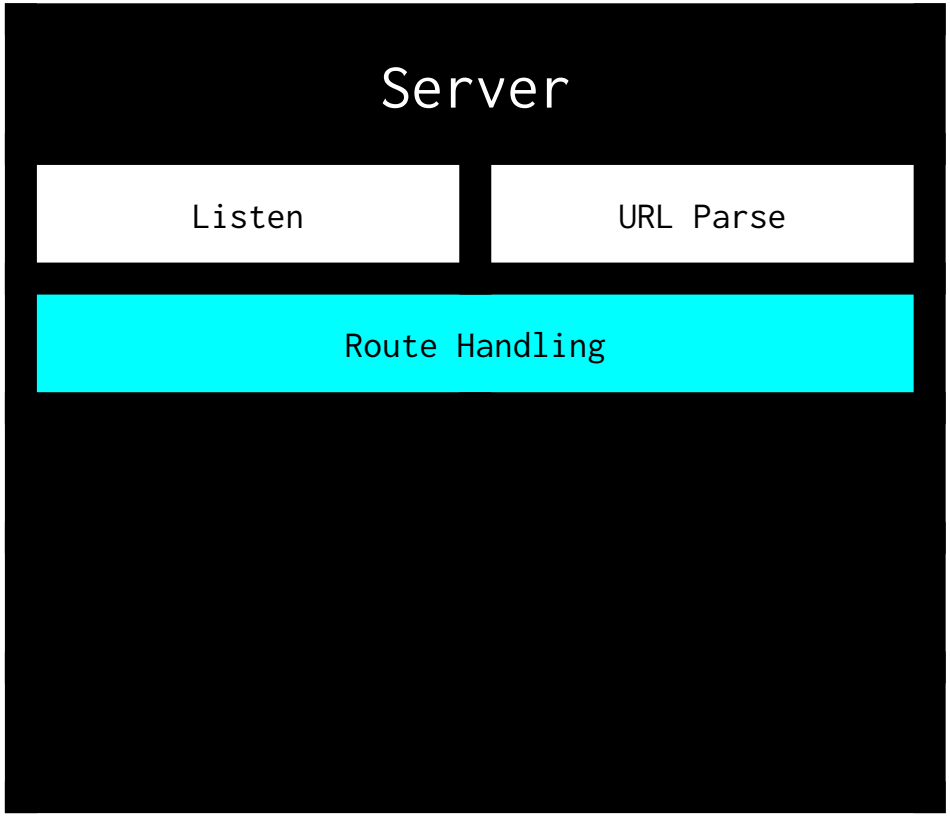
Inside the Box: Parsing

We give the server the ability to **parse**, or interpret, URLs that the user requests.



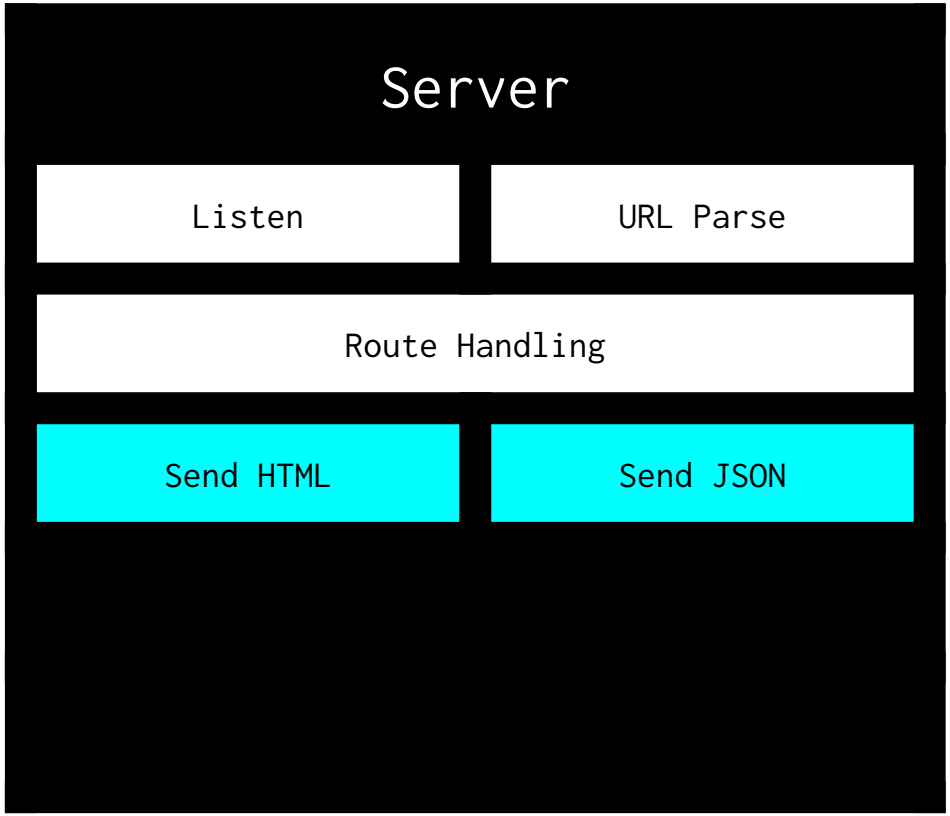
Inside the Box: Routing

Then, based on a URL's keywords, the server will be able to **route**, or direct, the flow of logic to initiate other processes.



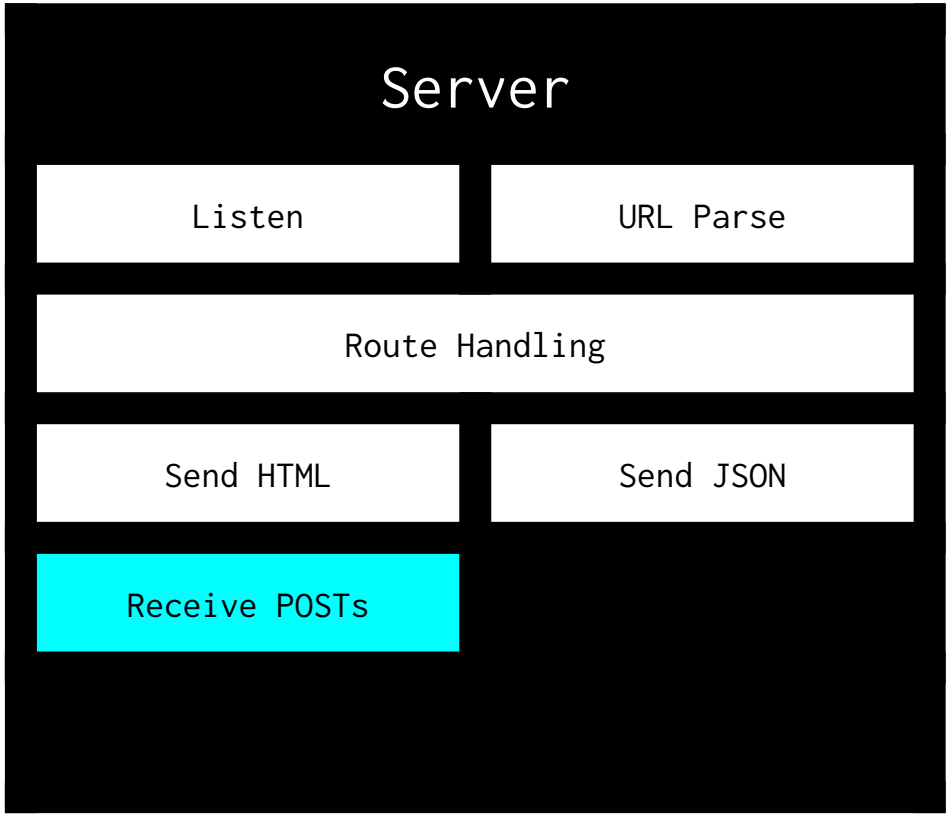
Inside the Box: Sending Files

These “other processes” might be sending an HTML file to be rendered or sending a JSON file to be rendered.



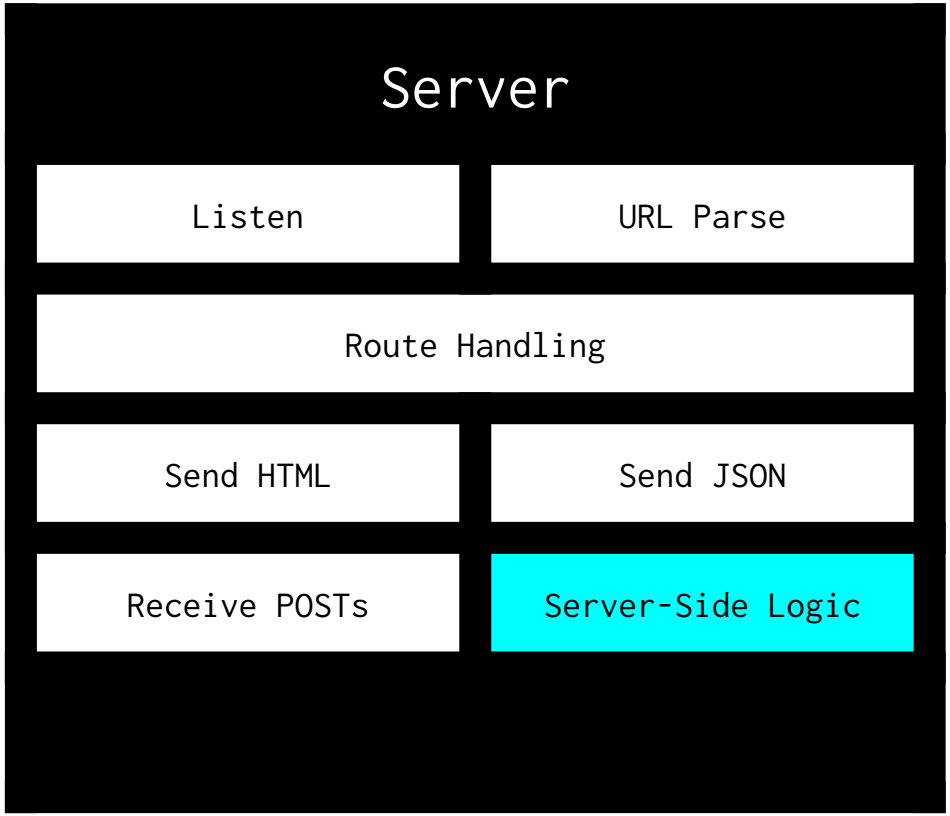
Inside the Box: Receiving Posts

We can also include a module to handle receiving a user's POST requests—the data they send the server.



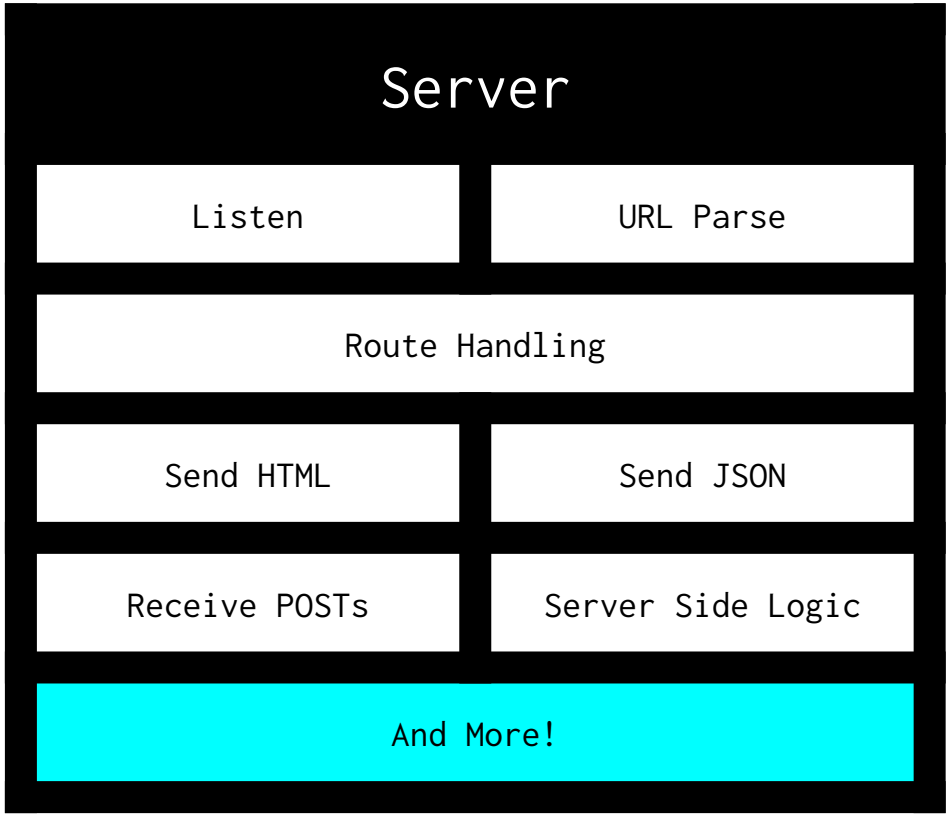
Inside the Box: Performing Logic

We can also have complex server-side logic that will initiate in response to users visiting a route endpoint or sending us data.



Inside the Box: And More!

- But it doesn't stop there!
- We can add functionality for authentication, logging requests, connecting to databases, and so much more.
- Always remember that we're coding out these functionalities into our box!



Questions?

<Time to Code>