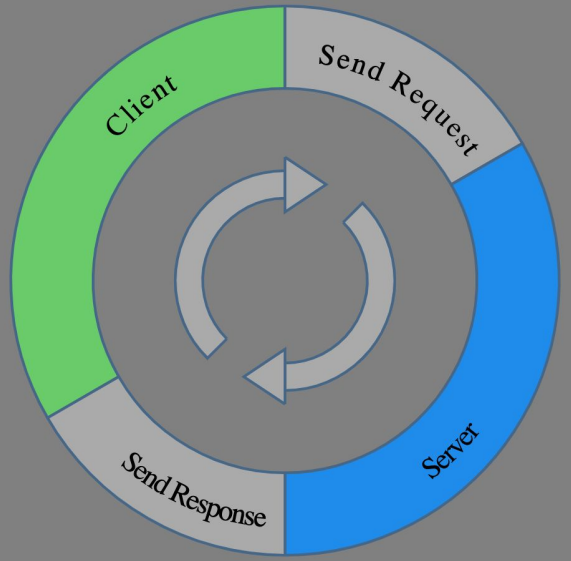
Introduction to Web Servers

From One Click to the Next

Type a URL into a Tab

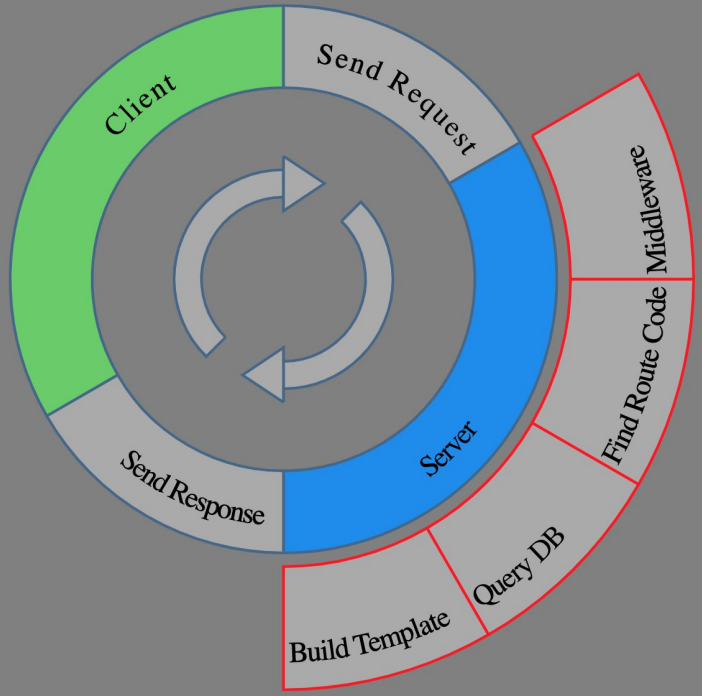
1. Sends a Request to Web Server
2. Server makes HTML Response
3. Response is sent to the Browser
4. Browser Renders the Response

If a User Clicks another Link go to Step 1.



Introduction to Web Servers

- web server inputs
  (a request = an HTTP verb, a path and sometimes data)
- how it receives those inputs
- how it processes those inputs to build customized output (including the role of middleware)
- how it sends that output back to the browser side



Web Server from Scratch

Twitter: @lighthouse_labs | Instagram: @lighthouselabs

Not Really From Scratch (from Request to Response)

The web “Request” includes a Path and Verb

Match the incoming request to custom code

Put the data into HTML

Send HTML Back to Client as a "Response"

What is a “Route”?

The Request includes both an HTTP Verb and a Path in the Headers

We combine these into a single concept so we can match the incoming request to an "Route" (a.k.a. "End Point")

Examples: GET / or POST /widget or …..

Express

(an npm package to help build Web Servers)

How Express Works (from Request to Response)

- Run some “Middleware”
- The Request includes Headers (HTTP Verb + Path) and Body (Form Data)
- Express matches the incoming request to an "End Point" (a.k.a. "Route")
- Express interleaves data into HTML via a "Template"
- The HTML is Sent Back as a "Response"