6.170: Software Studio
Web Servers

Arvind Satyanarayan +
Daniel Jackson
Today

Hyptertext Transfer Protocol (HTTP)
  URLs, IPs, and the DNS
  Request & Response

Node.js: Server-Side JavaScript
  Express: Web Server Framework
  Routes
  Templates
  Middleware
  Sessions & Cookies

Wednesday

RESTful API Design

AJAX

Asynchrony
  Continuations & Callbacks
  Promises
  Await / Async
Protocol

Host = Subdomain(s) + Domain + Top-level Domain

Path

http://stellar.mit.edu/S/course/6/fa18/6.170/materials.html?showKind=lectureNotes#content

URL Fragment identifies a section on the page.

Query String = Parameter-value pairs, separated by ampersands (&)
1. Go To URL
2. Lookup Domain Name
3. Return server's IP address
4. HTTP Request
5. Route Request
6. Process Request
7. SQL
8. Build Response
9. HTTP Response
10. Parse Response
11. Render Page

Web Browser

DNS

Server

Database
Client Side

1. Go To URL
2. HTTP Request
3. Route Request
4. Process Request
5. Build Response
6. HTTP Response
7. Parse Response
8. Render Page

Server Side

Database

SQL
Method: What do you want to do? GET, POST, ...

Headers: Additional metadata sent with request and response.

Request: What browser/version, what kinds of content can be rendered, ...

Response: Status (e.g., 404 Not Found), last modified, ...
Why node.js over other technology?

✓ Use a single language (JavaScript) across client and server.

✓ Fill gaps in core JavaScript (modules, input/output, networking).


✓ Very little magic (vs. e.g., Rails).
Our first web server

```javascript
let http = require('http');
let url = require('url');

http.createServer(function (req, res) {
  let path = url.parse(req.url).
    pathname.substring(1);

  res.end("Hello " + path);
}).listen(3000);

console.log("Listening on port 3000...");
```
Sessions and cookies

1. Login
2. HTTP Request with credentials
3. Check credentials
4. Start session
   ```
   {'session_id': 123, 'user': 'arvind'}
   ```
5. HTTP Response with cookie to identify session
6. Store

 cookie = {'session_id': 123}
Sessions and cookies

1. Add to Cart

2. HTTP Request with cookie

3. Lookup session
   
   ```json
   {  
     session_id: 123,
     user: "arvind"
   }
   ```

4. Process request

5. HTTP Response