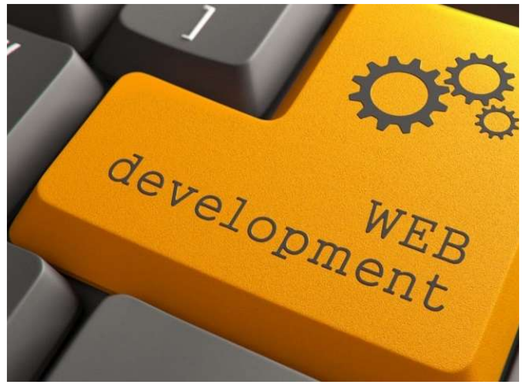
31.03.21 09:48

CSCI 165
Introduction to the Internet and the World Wide Web
Lec 4: Backend Programming



Jetic Gū

Overview

- Focus: Web Development
- Architecture: Internet
- Core Ideas:
    1. Backend
    2. NodeJS

What is Backend?

Summary

P1
Backend

What we've done so far

- Front-end
  - HTML + CSS + Javascript
  - Executed on the browser
  - Front-end controls the UI of the web application, how information is represented to the user, and how the user interacts with information
  - What it doesn't do: accessing information from a secure database, etc.

Review

Front vs Back

- Front-end
  - HTML + CSS + Javascript
  - Executed on the browser
  - UI of the web application
    - Information Representation
    - User Interaction

- **Back-end**
  - PHP, **NodeJS**
  - **Executed on the Server**
  - **Brain** of the web application
    - **Database** Interaction
    - **Complex calculation**, etc.

Concept

Front vs Back

- Front-end developing
  - Works with Clients and Users
  - End users of the product
    Work is mainly dependent on the customer's needs and what they will see.

- **Back-end developing**
  - Works with Front-end engineers
  - Realising functions of the product
    Doesn’t really care about the customer, yells at front-end quite a lot

P1
Backend

Front vs Back

- Common Front-End tools

HTML5

- Markup language to create Web Pages and Web components
- Parsed By the browsers

CSS3

- Cascading style sheet to add Custom themes for your HTML Code.
- Parsed by the browser

Javascript

- Programming language for providing functionally and Interactions in your front end Development
- Interrupted by browsers

Jquery

- JS library that provides a ton of services and tools for most needed functionally in the web
- Interrupted by browsers

Bootstrap[1]

- framework for Html5, css, javascript
- ready made styles and components
- Interrupted by browsers

Others

- There are tons
- More added everyday
- You need to learn as you work and progress

Review

1. I do not plan to cover this

P1
Backend

Backend

- Common Back-End tools

Languages
- PHP, NodeJS, Python, etc.
- Develop backend services, programme that runs on the server

Servers
- Unlike Frontend, Backend needs Servers to understand backend Code and provides compilers for
- Apache, Nginx, Node, etc.

Databases
- Most important part
- Provides data storage
- e.g. SQL, MongoDB, Postgre etc.

Frameworks
- Libraries that provide database access and server request functions
- Django, Spring, etc.

Security
- Protect your data
- Protect your server
- Protect your wallet

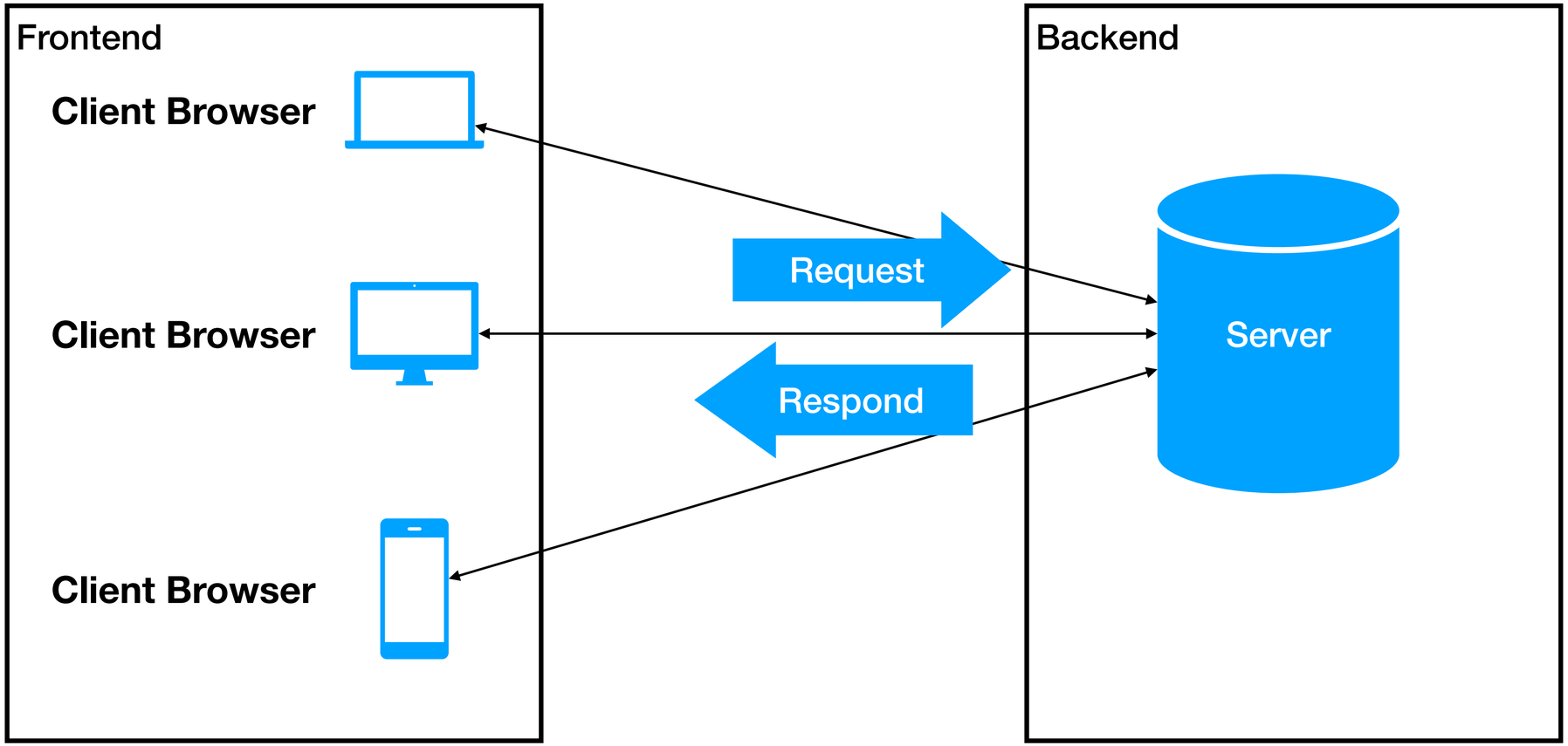
Others
- There are tons
- More added everyday
- You need to learn as you work and progress

Concept

1. We are not going to do too much here, only an intro

P1
Backend

Backend



Concept

P1
Backend

Full-Stack

- Usually, one is good with Front-End or Back-End, but not both
  They all requires tons of experiences and learning

- Full-stack developer is one that does all, this kind of people are like one-man armies.

  - Full-stack knows about front-end development

  - Full-stack knows about back-end development

  - Full-stack has no hair nor romantic relationship with human

Concept

NodeJS

Doing backend without learning new stuff

Summary

P2
NodeJS

Remember the stuff you've learned

- **All kind of languages**
  - **HTML**: for writing text and stuff
  - **CSS**: for formatting and making stuff look pretty
  - **Javascript**: for making HTML and CSS look even fancier and more interactive
  - Traditionally, you'd do the same thing for backend: C# + PHP + #*'*!$§!"

Review

NodeJS

- When it came out, everybody fell in love
- For the first time, you can **do backend with Javascript**!
- No more learning a new language, yeeee!

Concept

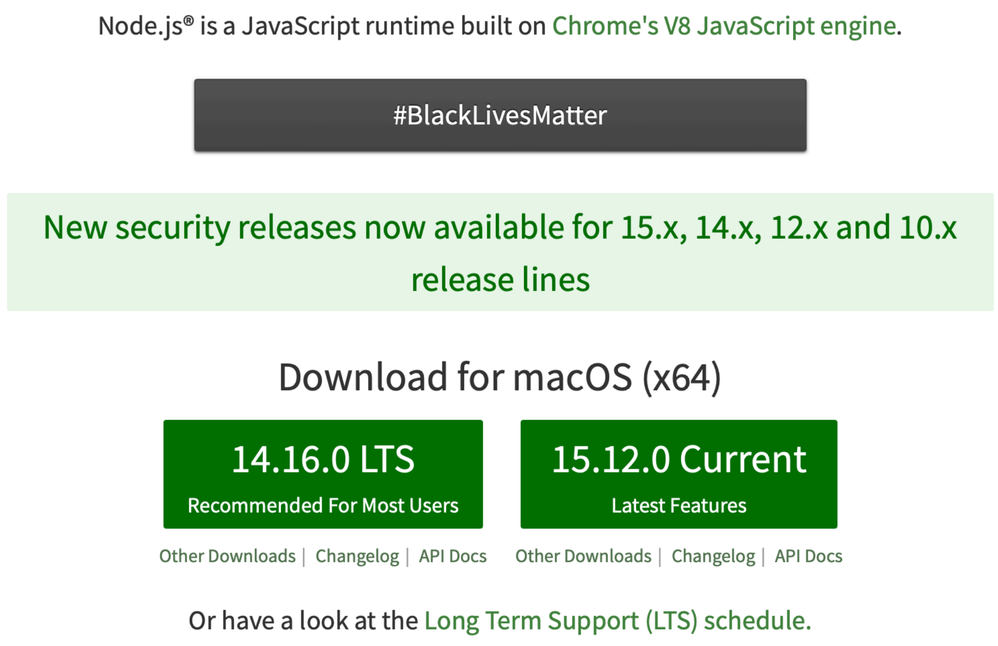
NodeJS

- Open Source Runtime Environment for Server-Side
- Uses Google’s JS V8 Engine
- Cross-Platform Support
  Linux, MacOS, even puny Windows

Concept

Installation

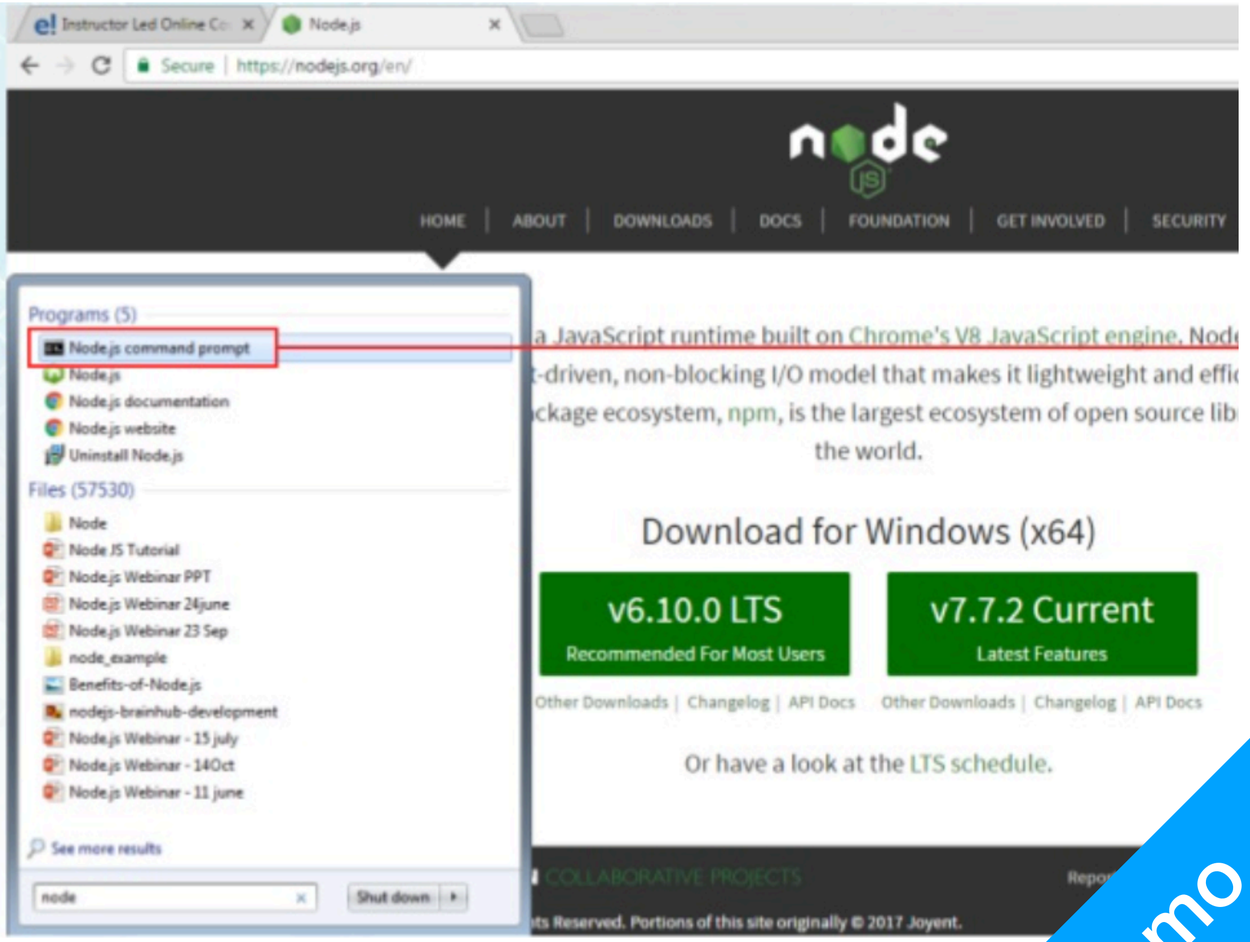
- Step 1: go to https://nodejs.org
- Step 2: download Node.JS



Demo

Installation

- Step 3: open command-line prompt
  - Windows: Node.js command prompt
  - macOS + Linux: open Terminal, type `node -v` and hit enter



Installation

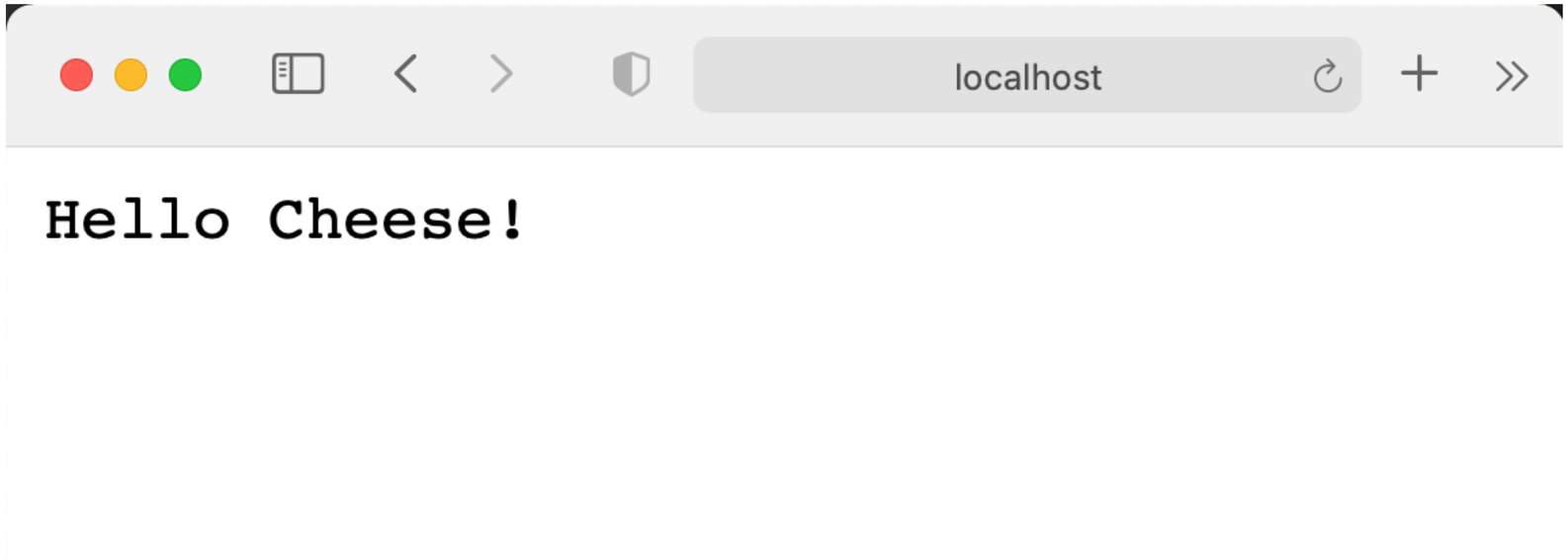
- Step 4: create `helloworld.js`

```
var http = require("http");

var port = 8080;

var server = http.createServer(function (request, reponse) {
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.end('Hello Cheese!\n');
})

server.listen(port, function() {
    console.log('Server running at http://localhost:8080')
})
```

Demo

Installation

- Step 5: Run, and open http://localhost:8080 in your browser
  - Command: `node helloworld.js`



Demo