

# Serving the Web:

## Building a Web Server



By: Arianna Lee, Nigel Heeralall, Marcus Robinson, Sean Dewey

# Agenda:

- Project Overview
- HTML Design
- Head Request
- Post Request
- Multi-threading
- Logging
- Authentication / Authorization
- Cache Requests
- Demo

# Project Overview

- Implemented Option 1:
  - GET, HEAD, POST
  - Authorization / Authentication
  - Logging
  - Multi-threading
  - Cache Request
- Used a simple login page as our index.html so we could do GET and POST
- To-Do: cache request

# HTML Website

## Login

## Welcome to Student Database!

Company	Contact	Country	Action	
Alfreds Futterkiste	Maria Anders	Germany	<a href="#">Edit</a>	<a href="#">Delete</a>
Centro comercial Moctezuma	Francisco Chang	Mexico	<a href="#">Edit</a>	<a href="#">Delete</a>
Ernst Handel	Roland Mendel	Austria	<a href="#">Edit</a>	<a href="#">Delete</a>
Island Trading	Helen Bennett	UK	<a href="#">Edit</a>	<a href="#">Delete</a>
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada	<a href="#">Edit</a>	<a href="#">Delete</a>
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy	<a href="#">Edit</a>	<a href="#">Delete</a>

# HEAD

```
> testHEAD = function(url) {  
  var xmlhttp = new XMLHttpRequest();  
  xmlhttp.open("HEAD", url, true);  
  xmlhttp.onreadystatechange=function() {  
    if(xmlhttp.readyState==4) {  
      console.log(xmlhttp.getAllResponseHeaders());  
    }  
  }  
  xmlhttp.send(null);  
}
```

```
< f (url) {  
  var xmlhttp = new XMLHttpRequest();  
  xmlhttp.open("HEAD", url, true);  
  xmlhttp.onreadystatechange=function() {  
    if(xmlhttp.readyState==4) {  
      console.log(xmlhttp.getAllResponse...  
    }  
  }  
}
```

```
> testHEAD('')
```

```
< undefined
```

```
date: Thu Nov 30 19:33:07 EST 2017
```

```
server: JHTTP 2.0
```

```
content-length: 130
```

```
content-type: text/html
```

```
> |
```

```
if (version.startsWith("HTTP/")) { // send a MIME header  
  sendHeader(out, responseCode: "HTTP/1.0 200 Good",  
             contentType: "text/html; charset=utf-8", body.length());  
}  
out.write(body);  
out.flush();
```

VM87:6

Nov 30, 2017 7:33:07 PM RequestProcessor run  
INFO: /0:0:0:0:0:0:0:1:59708 HEAD / HTTP/1.1

# POST

## Login

sean

•••••

Let me in.

POST http://localhost/

Status: HTTP/1.0 200 Good

### Request Headers

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding	gzip, deflate, br
Accept-Language	en-US,en;q=0.9
Content-Type	application/x-www-form-urlencoded
Origin	http://localhost
Referer	http://localhost/

Upgrade-Insecure-Requests

1

User-Agent Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.94 Saf

### Response Headers

Content-length	110
Content-type	text/html; charset=utf-8
Date	Sun Dec 03 14:57:39 EST 2017
Server	JHTTP 2.0

Dec 03, 2017 2:52:28 PM RequestProcessor run  
INFO: /0:0:0:0:0:0:0:1:56108 POST / HTTP/1.1

# Multithreading (Thread Pool)

Multithreading was already implemented in JHTTP and RequestProcessor

- Each request is its own thread
- Handled by an Executor
- Fixed number of threads

```
ExecutorService pool = Executors.newFixedThreadPool(NUM_THREADS);
```

```
Socket request = server.accept();  
Runnable r = new RequestProcessor(  
    rootDirectory, INDEX_FILE, request);  
pool.submit(r);
```

# Logging

Logging allowed us to monitor the server activity and log the errors that occurred. We advised the errors for authentication, authorization, files not found, etc.

- Here, we showed that the server has started successfully
- Recognized the severe error of the server not starting

```
SEVERE: Server could not start
java.net.BindException: Permission denied (Bind failed)
  at java.net.PlainSocketImpl.socketBind(Native Method)
  at java.net.AbstractPlainSocketImpl.bind(AbstractPlainSocketImpl.java:116)
  at java.net.ServerSocket.bind(ServerSocket.java:375)
  at java.net.ServerSocket.<init>(ServerSocket.java:237)
  at java.net.ServerSocket.<init>(ServerSocket.java:128)
  at JHTTP.start(JHTTP.java:27)
  at JHTTP.main(JHTTP.java:83)
```

```
Dec 03, 2017 4:21:02 PM JHTTP start
INFO: Accepting connections on port 1024
Dec 03, 2017 4:21:02 PM JHTTP start
INFO: Document Root: ./www
```



# Authentication

## Login

Let me in.

## Welcome to Student Database!

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

# Authorization

## Login

Let me in.

## Welcome to Student Database!

Company	Contact	Country	Action	
Alfreds Futterkiste	Maria Anders	Germany	<a href="#">Edit</a>	<a href="#">Delete</a>
Centro comercial Moctezuma	Francisco Chang	Mexico	<a href="#">Edit</a>	<a href="#">Delete</a>
Ernst Handel	Roland Mendel	Austria	<a href="#">Edit</a>	<a href="#">Delete</a>
Island Trading	Helen Bennett	UK	<a href="#">Edit</a>	<a href="#">Delete</a>
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada	<a href="#">Edit</a>	<a href="#">Delete</a>
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy	<a href="#">Edit</a>	<a href="#">Delete</a>

# Cache Requests

- Cache Object contains:
  - File Name and File Body
  - Time stamp to see which cache to replace
- Will store the **five** most recently accessed web pages in a map so it is in memory
  - Need to create a few more web pages to test
- If a user requests a file, we will:
  - Search for the file name
    - Hit: use our map instead of using Scanner to read in the file, update time stamp on object
    - Miss: load file with Scanner, check map for page that oldest time stamp and replace it

**DEMO!**

**Questions, Comments, Concerns?**