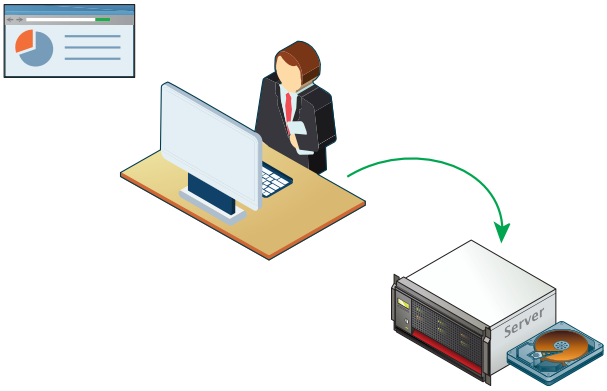
Appendix D - Web Development Basics

Web Dev, How the Web Works

NodeJS

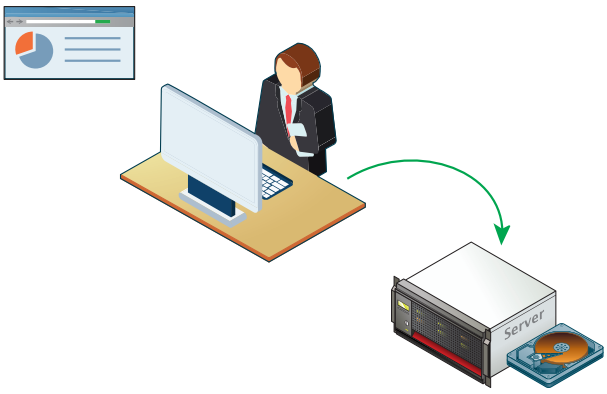
Definitions and History

Static Websites versus Dynamic Websites



Definitions and History

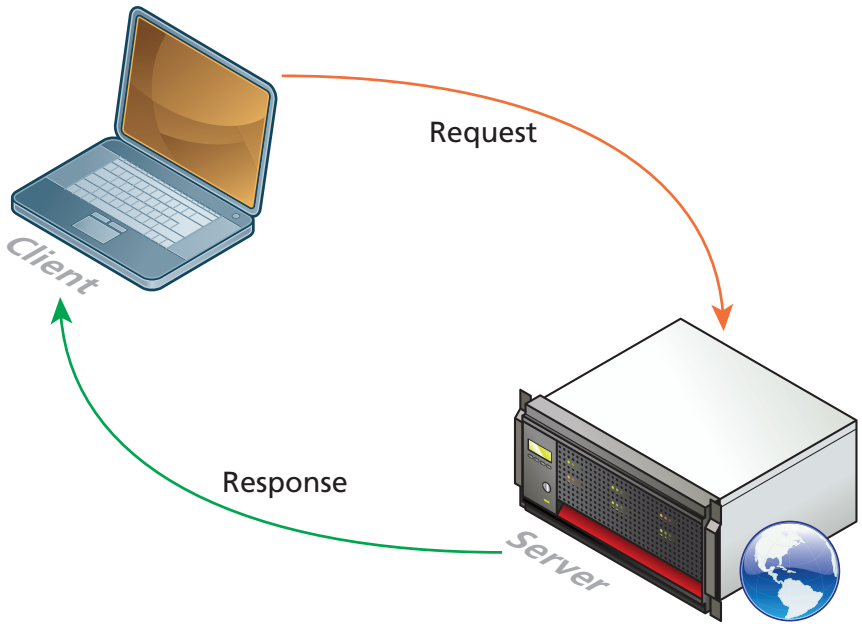
Static Websites versus Dynamic Websites



The Client-Server Model

The Client-Server Model

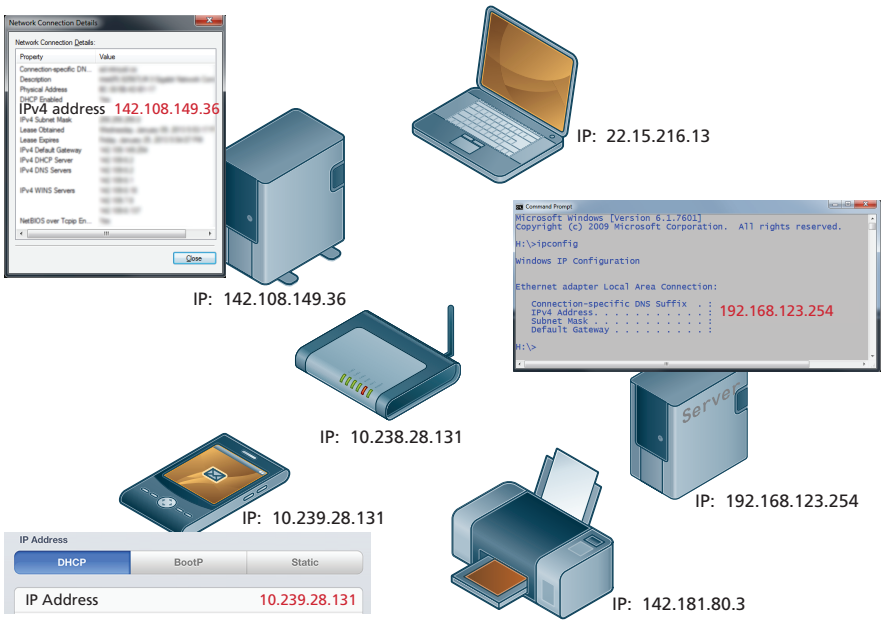
The Request-Response Loop



Internet Protocols

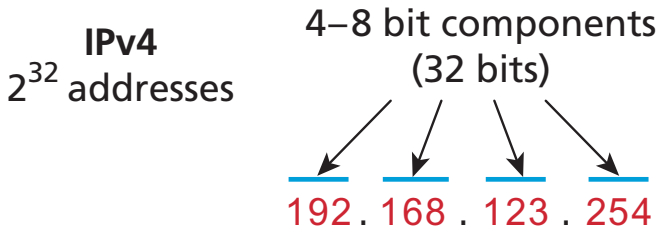
Internet Protocols

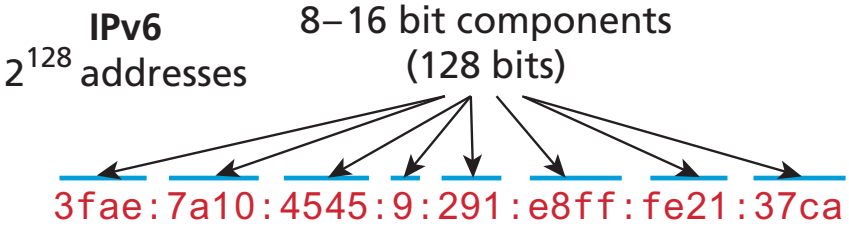
Internet Layer (IP)



Internet Protocols

IP addresses

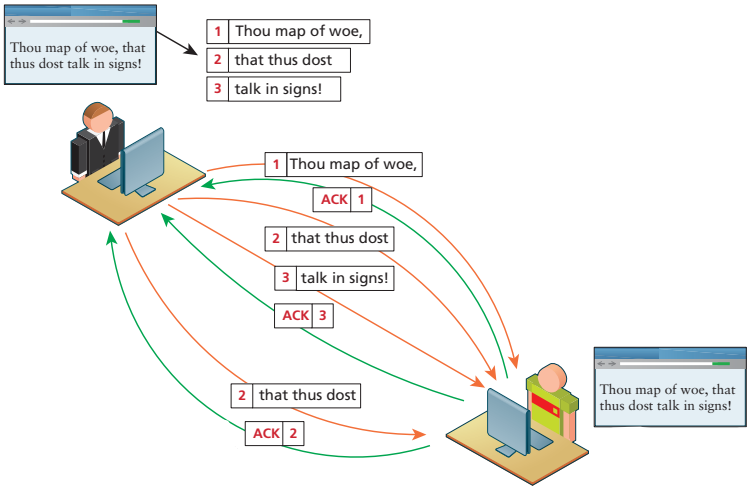




Internet Protocols

Transport Layer (TCP)

- Ensures transmissions arrive in order and without error



Internet Protocols

Application Layer

There are **many** application layer protocols. Web developers should be aware of :

- **HTTP.** The Hypertext Transfer Protocol is used for web communication.
- **SSH.** The Secure Shell Protocol allows remote command-line connections to servers.
- **FTP**. The File Transfer Protocol is used for transferring files between computers.
- **POP/IMAP/SMTP**. Email-related protocols for transferring and storing email.
- **DNS**. The Domain Name System protocol used for resolving domain names to IP addresses.

Uniform Resource Locators

Uniform Resource Locators

Overview

Uniform Resource Locators

Protocol

Recall that we listed several application layer protocols on the TCP/IP stack. FTP, SSH, HTTP, POP, IMAP, DNS, …

Requesting

- **ftp**://example.com/abc.txt → sends out an FTP request on port 21, while
- **http**://example.com/abc.txt → transmits an HTTP request on port 80.

Uniform Resource Locators

Domain

- The domain identifies the server from which we are requesting resources.
- Since the DNS system is case insensitive, this part of the URL is case insensitive.
- Alternatively, an IP address can be used for the domain

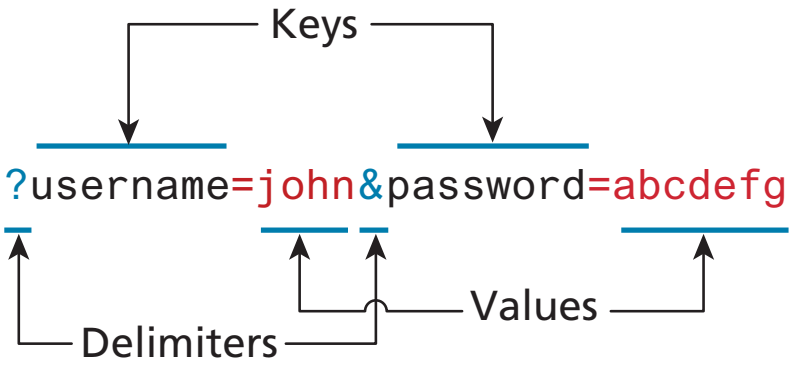
Uniform Resource Locators

Port

- The optional port attribute allows us to specify connections to ports other than the defaults
- Add a colon after the domain, then specify an integer port number.

Uniform Resource Locators

Query String



Uniform Resource Locators

Fragment

A way of requesting a portion of a page.

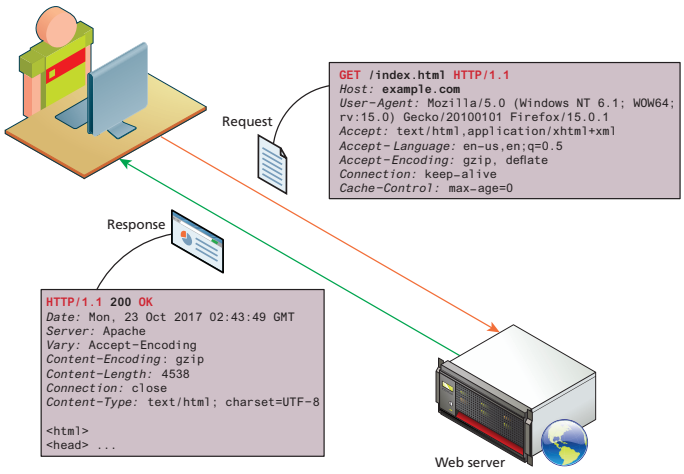
- Browsers will see the fragment in the URL, seek out the tag anchor in the HTML, and scroll the website to it.

Hypertext Transfer Protocol

Hypertext Transfer Protocol

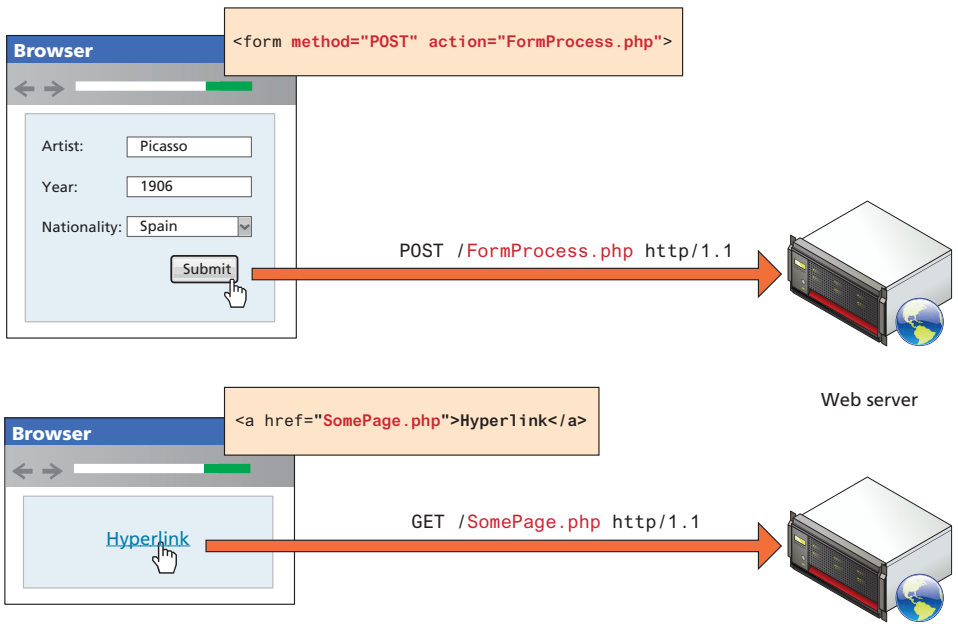
Headers

- **Request headers** include data about the client machine.
- **Response headers** have information about the server answering the request and the data being sent



Hypertext Transfer Protocol

Request Methods



Hypertext Transfer Protocol

Response Codes

- 2## codes are for successful responses,
- 3## are for redirection-related responses,
- 4## codes are **client** errors,
- 5## codes are **server** errors.

Hypertext Transfer Protocol

(Some) Response Codes

200: OK

301: Moved Permanently

304: Not Modified

307: Temporary redirect

400: Bad Request

401: Unauthorized

404: Not found

414: Request URI too long

500: Internal server error

REST

Repr**E**sentational **S**tate **T**ransfer

Routes describe a resource (type of media) and what actions can be taken

- Actions:
  - GET
  - POST
  - PUT
  - DELETE
- CRUD
  - Create
  - Read
  - Update
  - Delete

NodeJS

Servers

- Node.js (www.nodejs.org)
  - a development framework based on Google's V8 JavaScript engine.

- Most servers are multi-processed or multi-threaded.
  - This is sometimes referred to as **blocking architecture**
- Advantage – independence
- Disadvantage – fixed amount of processes, context switching

Servers

In a blocking (threading) architecture:

- Each thread executes the entirety of the web application
- Threads are blocked while performing lengthy tasks (i.e. database call)
- Response are generated only after all tasks are completed.

JavaScript as a Server

- JavaScript uses a non-blocking single-thread architecture
- **Node.js** is an event-driven execution environment for server-side web applications
  - Built on Chrome's open source JavaScript V8 Engine

In Node.js

- A single thread runs an event loop
- Any blocking tasks are handled asynchronously and will signify the event loop when it is complete

Node Package Manager

NPM

- The **Node Package Manager** is a set of command line tools, or CLI's, that keep track of small software applications called packages
  - packages are folders reusable code (much like what we've done so far) that can add extra functionality or to modularize your application
  - NPM is a quick way for developers to share code with other developers
  - Packages can installed locally to your application or globally with your installation of node.js
  - Essentially these packages make our lives easier

NPM

Global installation

- NPM packages are for many different purposes. They can be even in the form of command line tool (like NPM itself)
- When you install a package locally, NPM creates a node_modules folder if not exist.

  npm install <package>
- NPM looks for a default file in the root of the application called "package.json". To create one, use the command

  npm init
- To save a package to your project use the --save option

  npm install --save <package>
- To port your application, delete the node_modules folder and run

  npm install

  in the new environment

NPM

- Packages can come in many flavours:
  - For development purposes
  - App dependence
- An application dependence - the application will not run without these packages.
- A development dependency - that it is some utility that is required only during the development phase.
  - For example, tests, auto-compiling, transpilers, workflows, etc.
  - npm install --save-dev <package>

NPM

- To list the current packages locally and globally, use the list command
  - `npm list`
  - `npm list –g`
- To check for any updates to any of your packages
  - `npm outdated`
- To uninstall a package use
  - `npm uninstall --save <package>`
  - `npm uninstall --save-dev <package>`
- To make your application consistent with your package.json file use
  - `npm prune`

Express

Webserver

- Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- It has become so popular that many frameworks that are built on the Express package
- Install Express

```
npm install express
```


Express

- Express Reference:
  - https://expressjs.com/en/4x/api.html
- Creates an Express application. The express() function is a top-level function exported by the express module.

```
var express=require('express');
var app=express();
```

- Here the app variable is an HTTP `requestListener` function.
- Express express() functions:
  - `express.static()`
  - `express.json()`
  - `express.urlencoded()`

Express

- Express app() methods:
  - `app.all()`
  - `app.delete()`
  - `app.disable()`
  - `app.disabled()`
  - `app.enable()`
  - `app.enabled()`
  - `app.engine()`
  - `app.listen()`
  - `app.METHOD()`
  - `app.param()`
  - `app.render()`
  - `app.route()`