Node

Notes from Lecture

Content

Getting Started

A simple project structure

/

Your project directory, typically referred to as *root*.

package.json

app.js

app.js - our entry point

```
var http = require('http'); //import the http module

// create a simple HTTP server running on port 3000
http.createServer(function(request, response){
    response.writeHead(200);
    response.write('Hello world!');
    response.end();
}).listen(3000);
```

Start the app

1. Open your terminal
2. Go to, or make sure you are in, the project directory
3. Type `node app.js`
4. Press `Enter` key

Modules

What is a module?

- An encapsulation of related code into a single unit
- For example, http is a module we referenced previously
  - it encapsulates http-related functionality
  - it gave us the ability to create a server via the `createServer` method

Example: A simple module

Directory and file setup

1. Create a folder called `modules` under your root directory
2. Create a file called `greetings.js`

greetings.js

```
// greetings.js
var sayHello = function() {
  return "Hello.";
};

var sayGoodbye = function() {
  return "Goodbye.";
};
```

Module Exports

Why export a module?

Code can be used by other files, increasing its utility (via the magic of encapsulation).

Example: Export

greetings.js

```
// greetings.js
var exports = {}; // new object literal

exports.sayHello = function() { // a function of exports object
    return “Hello”;
};

exports.sayGoodbye = function() { // a function of exports object
    return “Goodbye”;
};

module.exports = exports; // make this module available when
                          // require invoked
```

greetings.js (an alternate version)

```
// greetings.js
module.exports = {
    sayHello: function() {
        return “Hello”;
    },

    sayGoodbye: function() {
        return “Goodbye”;
    }
};
```

Module Imports

How to import a module

- Use keyword `require` to import module
  - Returns an object that references the value of `module.exports` for a given file

Example: Import

app.js

```
var greetings = require('./modules/greetings.js'); //import

// we can access the methods in greetings.js
console.log(greetings.sayHello());
console.log(greetings.sayGoodbye());
```

The Request Object

Try it: console.log(request)

```
http.createServer(function(request, response){
    console.log(request);
     response.writeHead(200);
     response.write('Hello world!');
     response.end();
}).listen(3000);
```

The request object is very useful. For example, request.url shows our url relative to the server's root. Knowing this, we could send different content based on that url.

Resources

Initializing the node package manager (npm)

1. Navigate to your terminal
2. Type `npm init`
3. Follow the prompts (see potential defaults on next slide)

npm init defaults

field	value	explanation
name	(default)	Leave the NAME and VERSION alone. We will talk more about app naming and semantic versioning as we continue at Prime, for now the defaults are fine.
version	(default)	
description	`My first Node application`	
entry point	(default)	app.js
test command	(default)	Also, go ahead and leave test command, git repository, and keywords alone (note that we can change all of this later).
git repository	(default)	
keywords	(default)	
author	<your name>	Make sure to enter your name (gotta give yourself credit!)
license	(default)	License is fine at ISC for now.

curl

- Basic GET request via curl (plus other examples)