COMP 20
WEB PROGRAMMING

WEEK 14: CONNECTING NODE.JS

65

ANNOUNCEMENTS

TA Office Hours

Mon 7pm
Tues 3:30pm
Thurs 8pm
Sunday 11am

Additional times available by request.

April 20

Lightning talks (project updates) on Wednesday

See canvas assignment for details

Assignments

Node.js/MongoDb assignment due 4/23

Last Quiz

April 24-26

Topics: MongoDB, NoSQL, NodeJS, Heroku

Extension for Final Project to April 30

Review sessions Apr 30, May 2 and May 4
Watch Piazza to sign up.

66

FINAL PROJECT TIMELINE

- ~~Week 12 (4/5): finalize concept, set up dev environment, establish roles, project architected~~
- ~~Week 13 (4/12): coding, "trial and error" testing~~
- Week 14 (4/19): MVP working / Project update/ lightning talk
- Week 15 (4/26): testing and enhancements

67

LOCAL MODULES: GROW YOUR OWN USING *EXPORTS*

- Example: create a module to get the date/time

```
exports.dtModule = function () {
  return Date();
};
```

68

ADD THE MODULE TO YOUR SCRIPT

- *require* the module in the script
  - `var dt = require('./datemodule.js');`
- Use the module in the event function
  - `res.write("Current date/time: " + dt.dtModule());`

See: mydate.js, mymodule.js

69

69

A MODULE CAN BE AN OBJECT

```
exports.myobj = {
  a: 1,
  b : function() { return 10;}
};
```

See: myobj.js, mymodule2.js

70

70

```
var http = require('http');
var obj = require('./myobj.js');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write("a is " + obj.myobj.a + "<br>");
  res.write("b is " + obj.myobj.b() + "<br>");

  res.end();
}).listen(8080);
```

71

71

TRY IT

- Create a module to serve the price of one of several items. Possible items are hotdogs ($3.50), fries ($2.00) and soda ($1.50)
- You will need to create an object in the module to store the names/prices of the items.
- Create a .js file to load the module and read the value of a query string parameter called "item" from the query string and then display the price of that item

72

72

INSTALLING A 3RD PARTY MODULE USING NPM

- NPM (Node Package Manager) is a command line tool that installs, updates or uninstalls Node.js packages to give you access to additional modules
- The basic format is:
  - npm install <module name>
- Use -g to install the module/package globally on that computer
- Use --save to add the module to the dependency list for this "package"
- Also available is *npm update* and *npm uninstall*

73

73

EXAMPLE: CALENDAR MODULE

- The calendar module has functionality to display a calendar

```
> npm install calendar

c = require("calendar");
cal = new c.Calendar();
m = cal.monthDates(2020,3, function(d) {return (' '+d.getDate()).slice(-2)},
                                                    function(w) {return w.join(' | ')});
console.log ("Calendar for April, 2020");
for (i=0; i<m.length; i++)
    console.log(m[i]);
```

74

74

USE THE CALENDAR MODULE TO WRITE TO THE WEB PAGE

```
c = require("calendar");
var http = require('http');
cal = new c.Calendar();
m = cal.monthDates(2020,3, function(d) {return (' '+d.getDate()).slice(-2)},
      function(w) {
            s = "";
            for (j=0; j<w.length; j++)
                  s += "<div  class='col'>" + w[j] + "</div>";
            return s;
      }
);
```

See: cal_demo1.js, cal_demo2.js

75

75

EXAMPLE CONTINUED

```
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write ("<html><head><style type='text/css'>.col {display: inline-block; width:40px;
border: 1px solid #333;} </style></head>");
  res.write("<body>Calendar for April 2020<br />");
  for (i=0; i<m.length; i++)
     res.write(m[i] + "<br />")
 res.end("</body></html>");
}).listen(8080);
```

76

76

CONNECTING NODE.JS TO A DATABASE

```
MongoClient.connect(url, { useUnifiedTopology: true }, function(err, db) {
  if(err) { console.log("Connection err: " + err); return; }
  var dbo = db.db("Textbooks");
  var coll = dbo.collection('books');
  …
});
```

See: node_mongo.js, node_mongo_insert.js, node_mongo_delete.js, node_mongo_find.js

77

77

HEROKU

- Platform as a Service
- Heroku is a cloud platform that facilitates deployment of apps
- Can manage and scale apps
- Can deploy from Git
- Good for teams
- Free level available at heroku.com

78

78

HEROKU DYNOS

- Whats a dyno: *Dynos are isolated, virtualized Linux containers that are designed to execute code based on a user-specified command.*
- Think of them as virtual computers that can be powered up or down as needed based on the size of the application
- Apps are run in "dynos"
- Price structure reflects flexibility, availability and performance of dynos
- Add more dynos for more storage or for more processing – can also increase the size of the dynos

79

79

HEROKU OR AWS?

- Heroku runs on top of AWS
- Provides a more user friendly interface and management tool

Heroku provides

- Predictability and insight into the cost of application maintenance and support elements
- Clear accountability and SLA's
- Decreased risk
- Ability to focus on innovation, not operations

You don't need to manage the AWS environment – it's all done transparently as part of the Heroku experience

80

80

HEROKU AND GIT

- Add Git to Heroku in the Deploy tab of apps in the Heroku Dashboard.
- You must set up a repo for which you have access

- Updates to the app will automatically push to the git repo
- Test apps can be deployed for git branches that have not yet been merged into the master branch.
- Heroku CI (continuous integration) – an additional Heroku tool that can run your test code in a "disposable environment" to allow testing before committing updates to GitHub

81

81

SETTING UP HEROKU-GIT- NODE

1. Create an account at heroku.com
2. Create a folder with two files
   1. your app (choose a simple hello world app – be sure to write to the web page)
   2. package.json (provided in canvas)
3. In your app, set the port variable as:
   var port = process.env.PORT || 3000;
4. Test the app locally
5. Create a new git repo on github or in an existing git repo, create a folder
6. Using the command line, add the app and the package.json file to the repo.

82

82

ADDING THE FILES TO THE GIT REPO:

- git init
- git add .
- git commit -m "commit message"
- git remote add origin https://github.com/<user id>/<repo name>.git
- git push origin master

- Chek that the files are in the repo

83

83

SETTING UP HEROKU

- Log in to your account and choose "Create new App" (upper right)
- This should bring you to the dashboard for the app – choose Deploy
- Select GitHub as the method and below that connect to the repo
  - You may want to choose automatic depoloyment
- Go to "settings" and select "Add buildpack" – choose node.js
- Go back to the deploy tab and choose "Deploy Branch" – this should start the app
- Lastly – go back to the "settings" tab and scroll down to "Domains" to find the domain to view your app

84

84