# I/O LAB INTRO TO

node JS™

November 2012, @davelester

Monday, November 26, 12

I started writing Node over the summer, and I've been working on a project at Mozilla called Open Badges, that's basically an infrastructure for federating digital badges and recognitions of skills, online. Helping to recognize informal learning, and offering a way to make that visible for others.

Some of this presentation draws from Ryan Dahl's original presentation of Node, and a subsequent talk he gave to the San Francisco PHP meetup. I tried to make this talk more approachable to students in the class; this is focused less on the performance benefits and more about why and how you might use node to build a website. (see: ExpressJS example later)

# OUTLINE

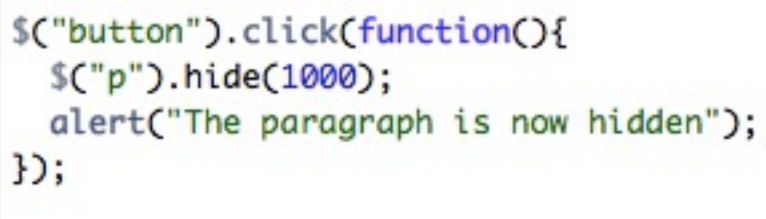
- What is Node.JS and why do I care?
- Introduction and building shit
- Resources (People and projects to know)
- Heroku and App Deployment

Monday, November 26, 12

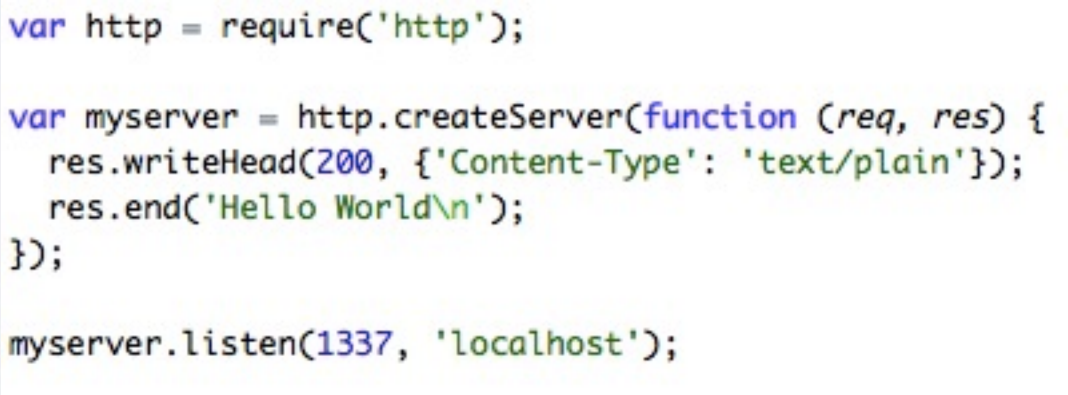
Outline.

Note: I’m not going to cover installing and configuring node, but you can install it easily using homebrew on your mac, OR check out nodejs.org for instructions. there are lots of tutorials

# JAVASCRIPT ON THE SERVER



```
$("button").click(function(){
  $("p").hide(1000);
  alert("The paragraph is now hidden");
});
```



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

var myserver = http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
});

myserver.listen(1337, 'localhost');
```

Node.JS is Javascript that runs on your server. In fact, you can use it to create HTTP servers, as well as other fun things.

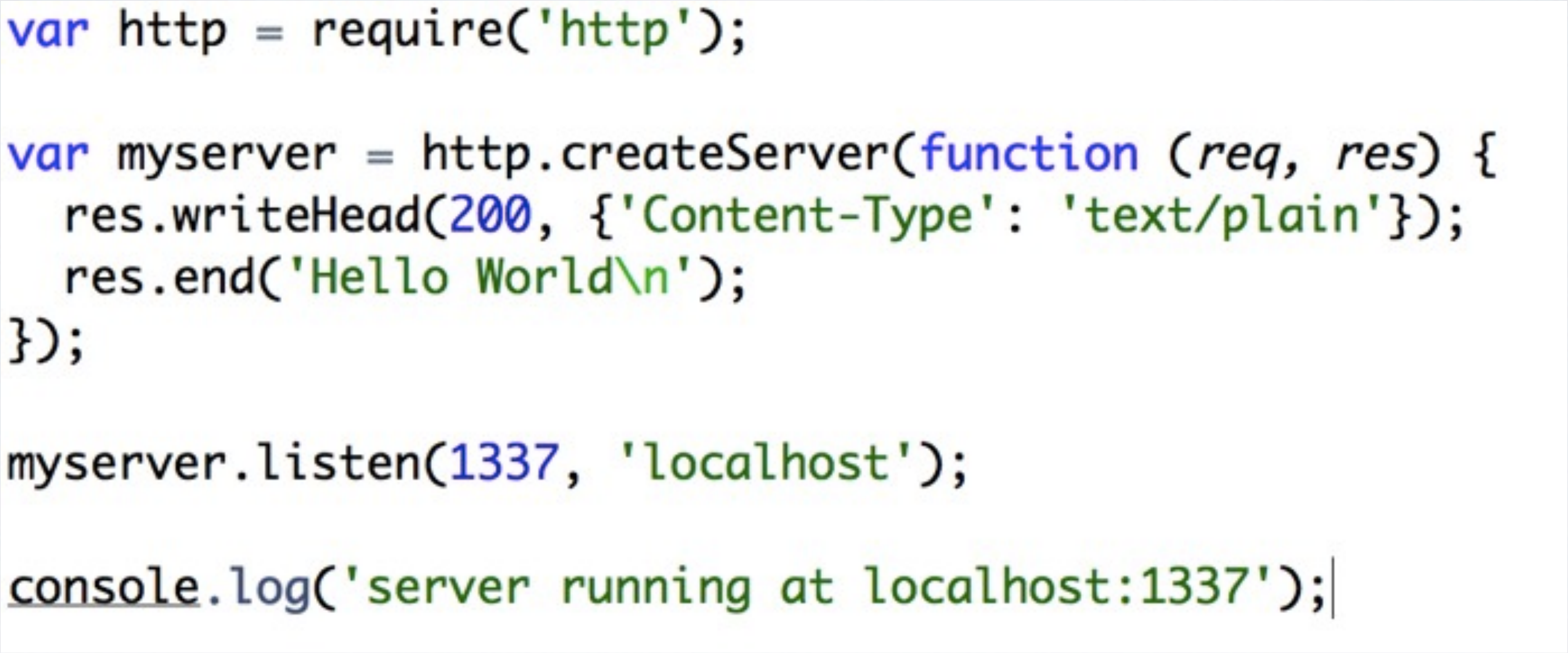
At this point we're all familiar with some basics of javascript, and we've all written some jquery. When you load a webpage, that code is sent to the web browser (your client), and rendered there.

Node.JS is different. Node runs on the server, and while it can be accessed by your web browser in certain instances the code is run on the server and its JS is not sent to the browser.

In the right example, we're actually creating a HTTP web server that responds 'Hello World', returns a 200 response, with the content-type of text/plain on port 1337.

(note: the examples of JS here obviously do very different things, but they show the similarities in syntax between writing JS for the client vs the server)

# HELLO WORLD IN NODE



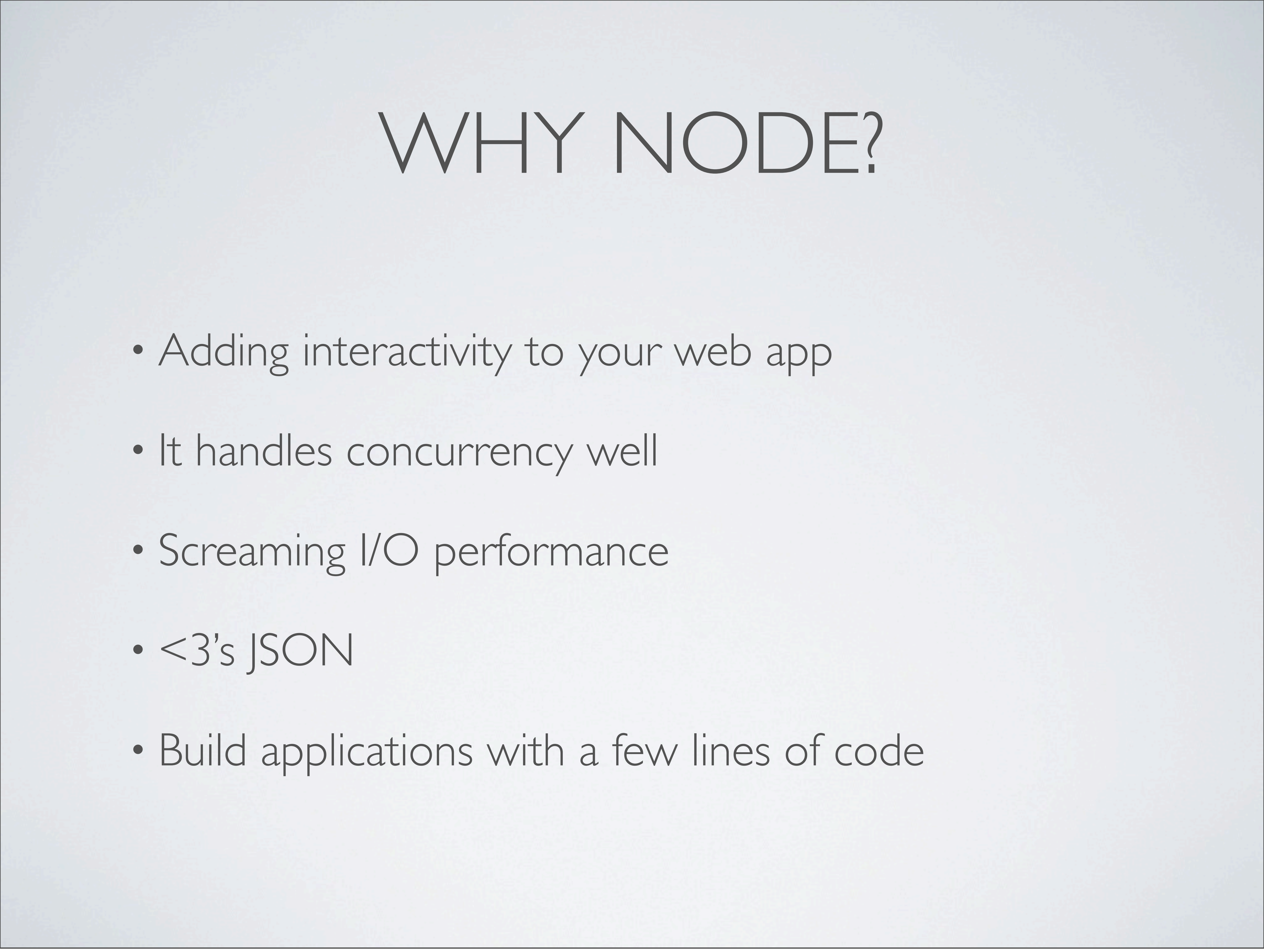
```
var http = require('http');

var myserver = http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
});

myserver.listen(1337, 'localhost');

console.log('server running at localhost:1337');
```

Walk line-by-line through code.



# WHY NODE?

- Adding interactivity to your web app
- It handles concurrency well
- Screaming I/O performance
- <3's JSON
- Build applications with a few lines of code

Monday, November 26, 12

Lots of big companies using Node. LinkedIn, Microsoft, Yahoo..

there are tons of interactive websites and features where you need to:
 * game, moving around and need to relay movements
 * progress bar, update the user while a file is being uploaded.
 * chat window, pass around your messages

Handles concurrency well, because it has non-blocking i/o. I'll explain more about what that means in the next slide

Screaming I/O performance. particularly important if you're reading or writing a ton -- think of examples, like you're building an API that others will use, or a mobile application that will need to be reading and writing data to a server. Or you're reading data from someone else's API. or writing to a database.

Since it's in Javascript, it's easy and friendly to interact with Javascript and databases like Mongodb that send JSON back and forth.

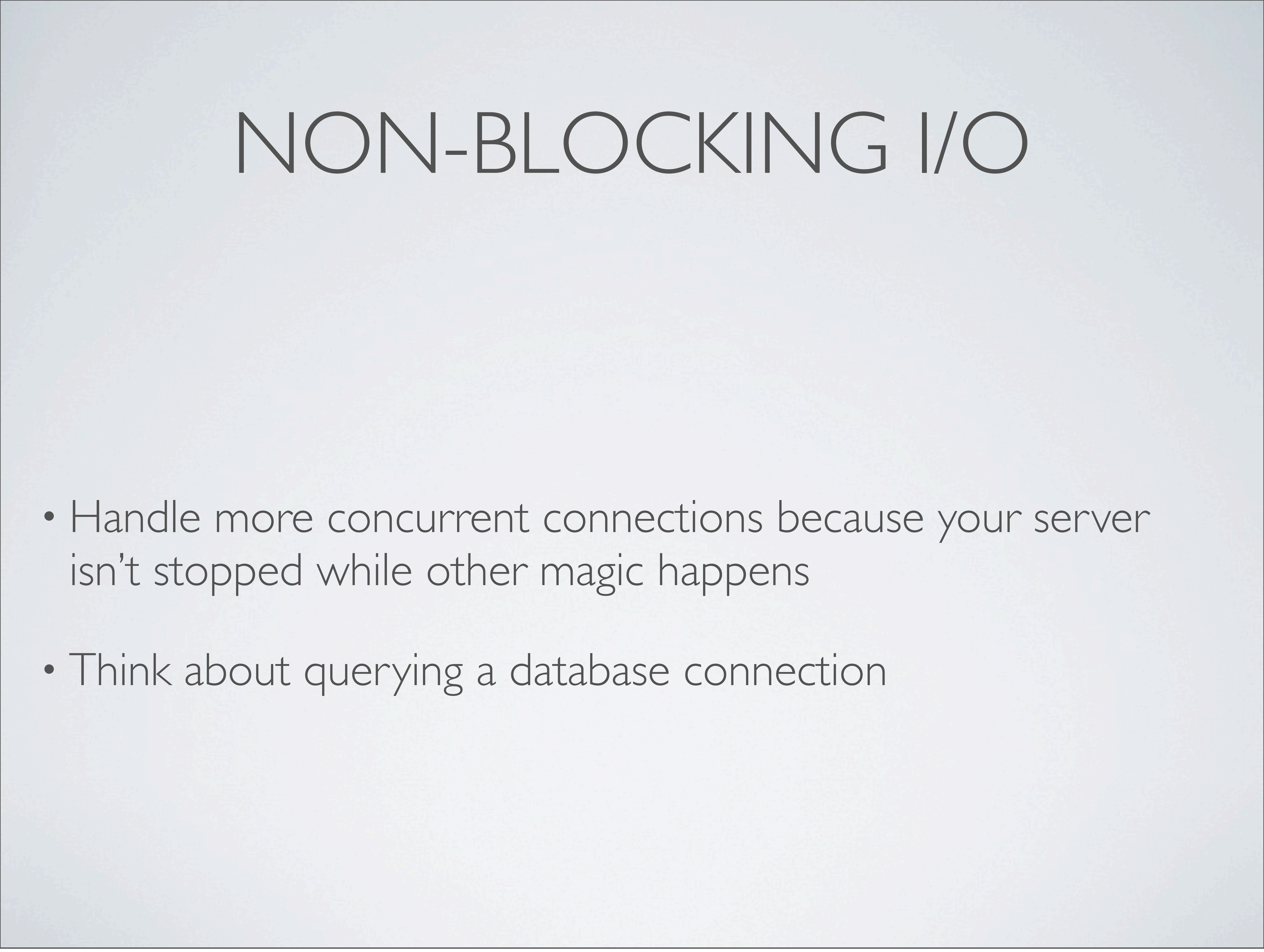
Easy to build apps -- I'll demo and show off NPM (node package manager) in a bit -- there are tons of examples of applications that are small, light, and modular.

# DEMO

- http://nodeknockout.com/

Awesome node app, demonstrating a number of features and possibilities of using Node.

Code on github: https://github.com/nko/website



# NON-BLOCKING I/O

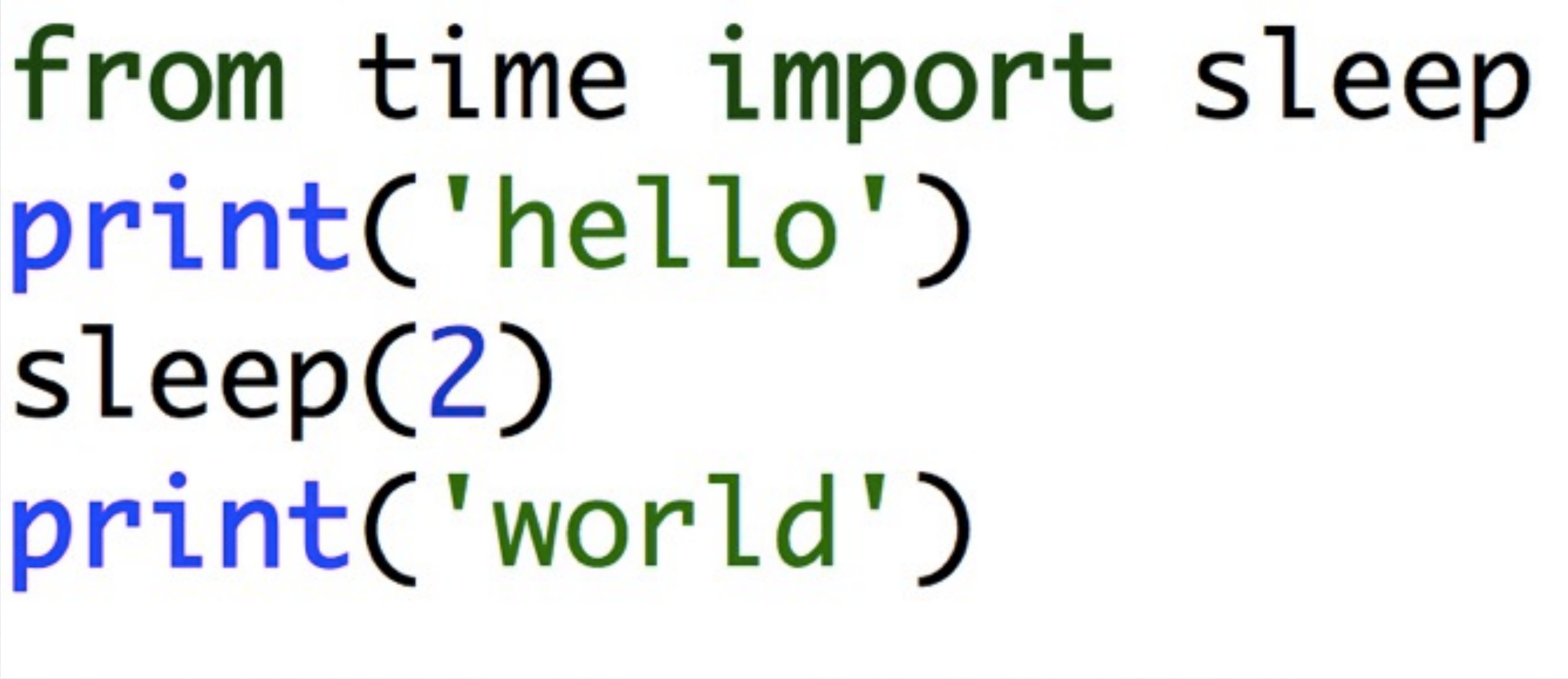
- Handle more concurrent connections because your server isn’t stopped while other magic happens
- Think about querying a database connection

Monday, November 26, 12

Think about this pause in terms of anyy planned (or even uncertain) activity on your server when you wait.

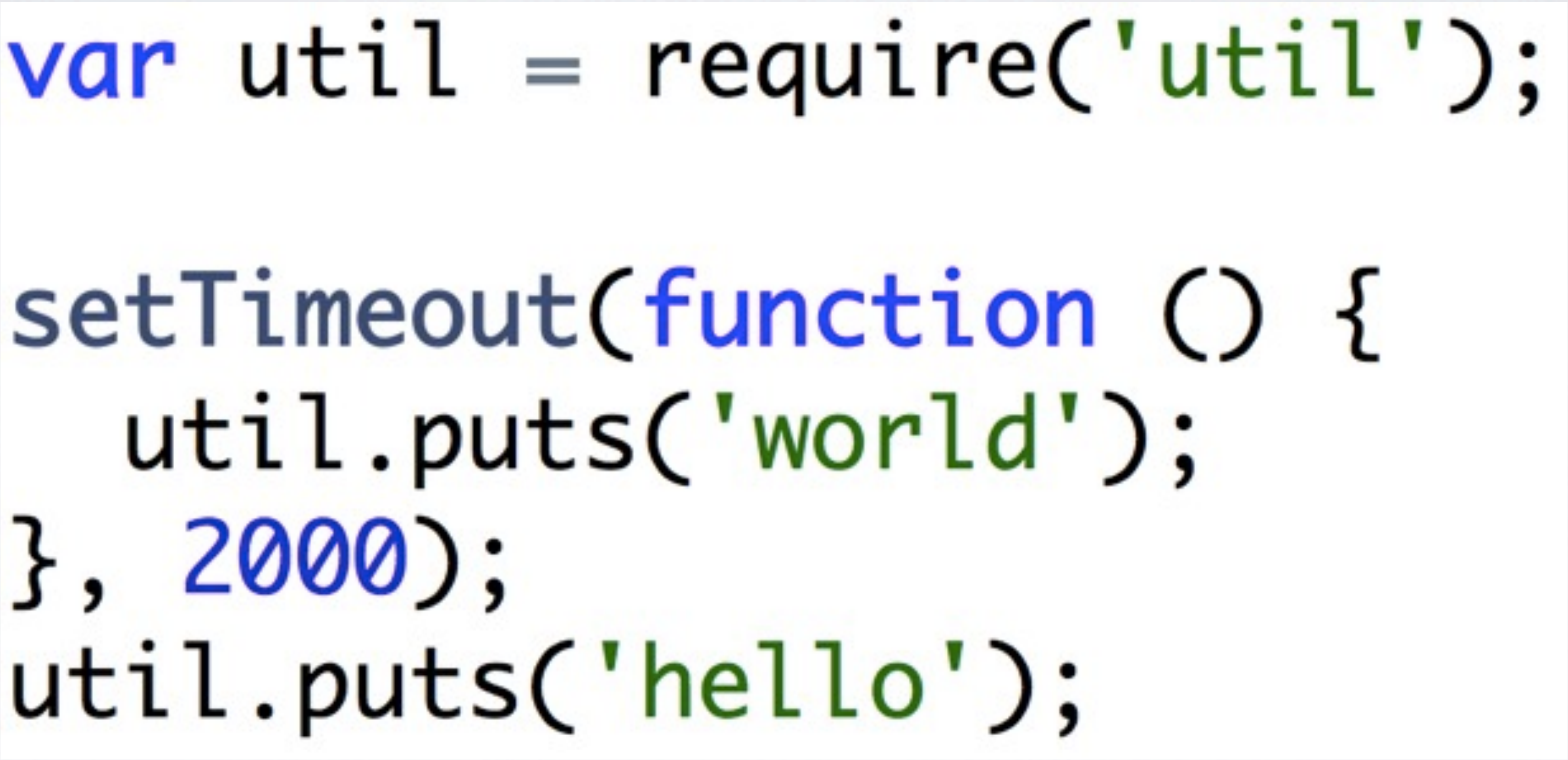
API, database call, interacting with another server of any kind..

# WHAT DOES THIS DO?



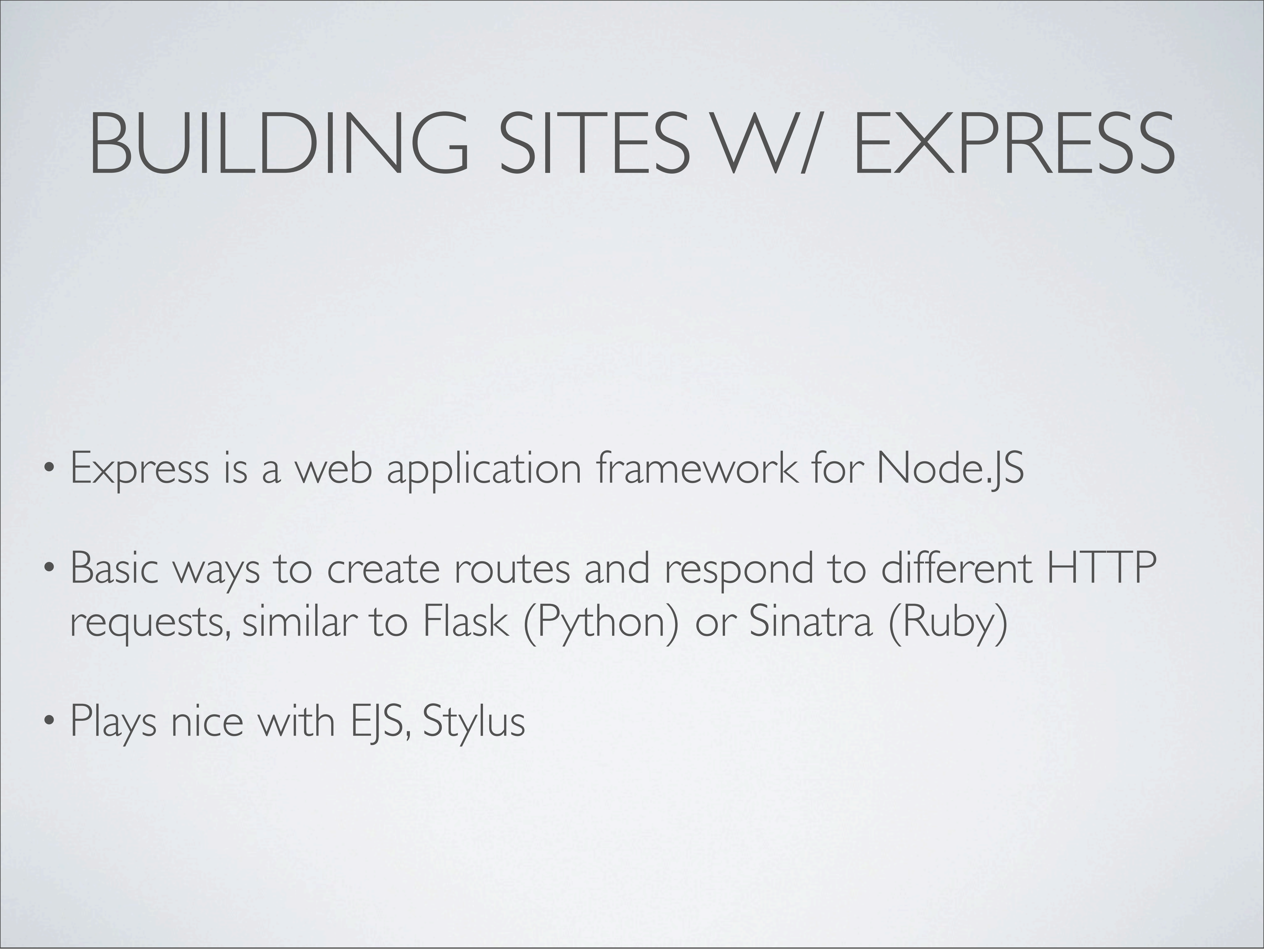
```python
from time import sleep
print('hello')
sleep(2)
print('world')
```

# WHAT DOES **THIS** DO?



```
var util = require('util');

setTimeout(function () {
  util.puts('world');
}, 2000);
util.puts('hello');
```



# BUILDING SITES W/ EXPRESS

- Express is a web application framework for Node.JS
- Basic ways to create routes and respond to different HTTP requests, similar to Flask (Python) or Sinatra (Ruby)
- Plays nice with EJS, Stylus

Monday, November 26, 12

http://expressjs.com/guide.html

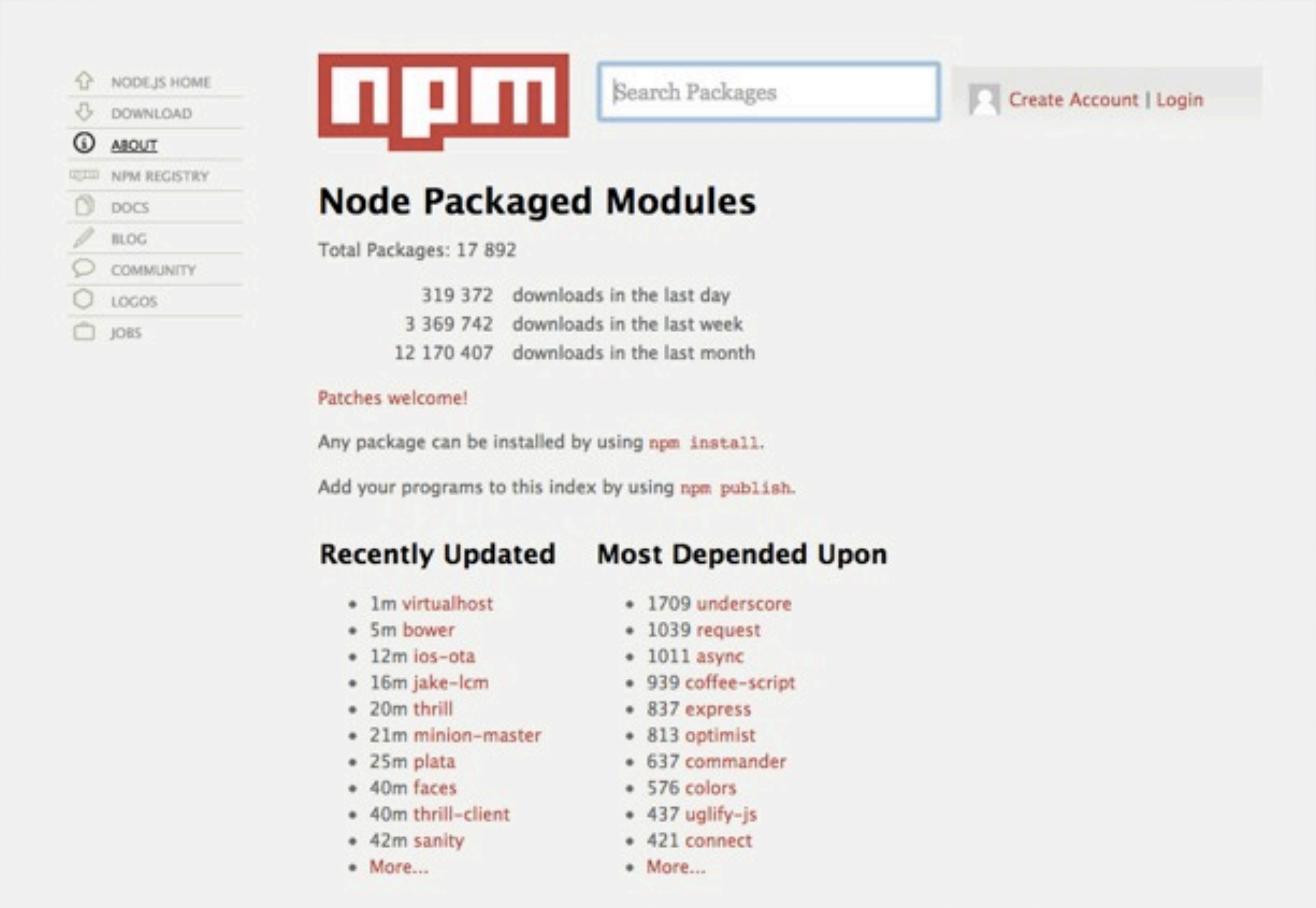
```
express --sessions --css stylus --ejs myapp
```

```
$ cd myapp
$ npm install
```

# NODE RESOURCES & PEOPLE

- Great screencasts at Nodetuts.com
- Nodeup.com Node.JS podcast
- Follow these people on Twitter:
  - @izs, @maxogden, @substack, @mikael
- Hang out on IRC
  - #node.js on irc.freenode.net

# NODE PACKAGE MANAGER



## Node Packaged Modules

Total Packages: 17 892

```
   319 372   downloads in the last day
 3 369 742   downloads in the last week
12 170 407   downloads in the last month
```

Patches welcome!

Any package can be installed by using `npm install`.

Add your programs to this index by using `npm publish`.

| Recently Updated | Most Depended Upon |
|---|---|
| 1m virtualhost | 1709 underscore |
| 5m bower | 1039 request |
| 12m ios-ota | 1011 async |
| 16m jake-lcm | 939 coffee-script |
| 20m thrill | 837 express |
| 21m minion-master | 813 optimist |
| 25m plata | 637 commander |
| 40m faces | 576 colors |
| 40m thrill-client | 437 uglify-js |
| 42m sanity | 421 connect |
| More... | More... |

- https://npmjs.org/

REQUEST module: https://github.com/mikeal/request

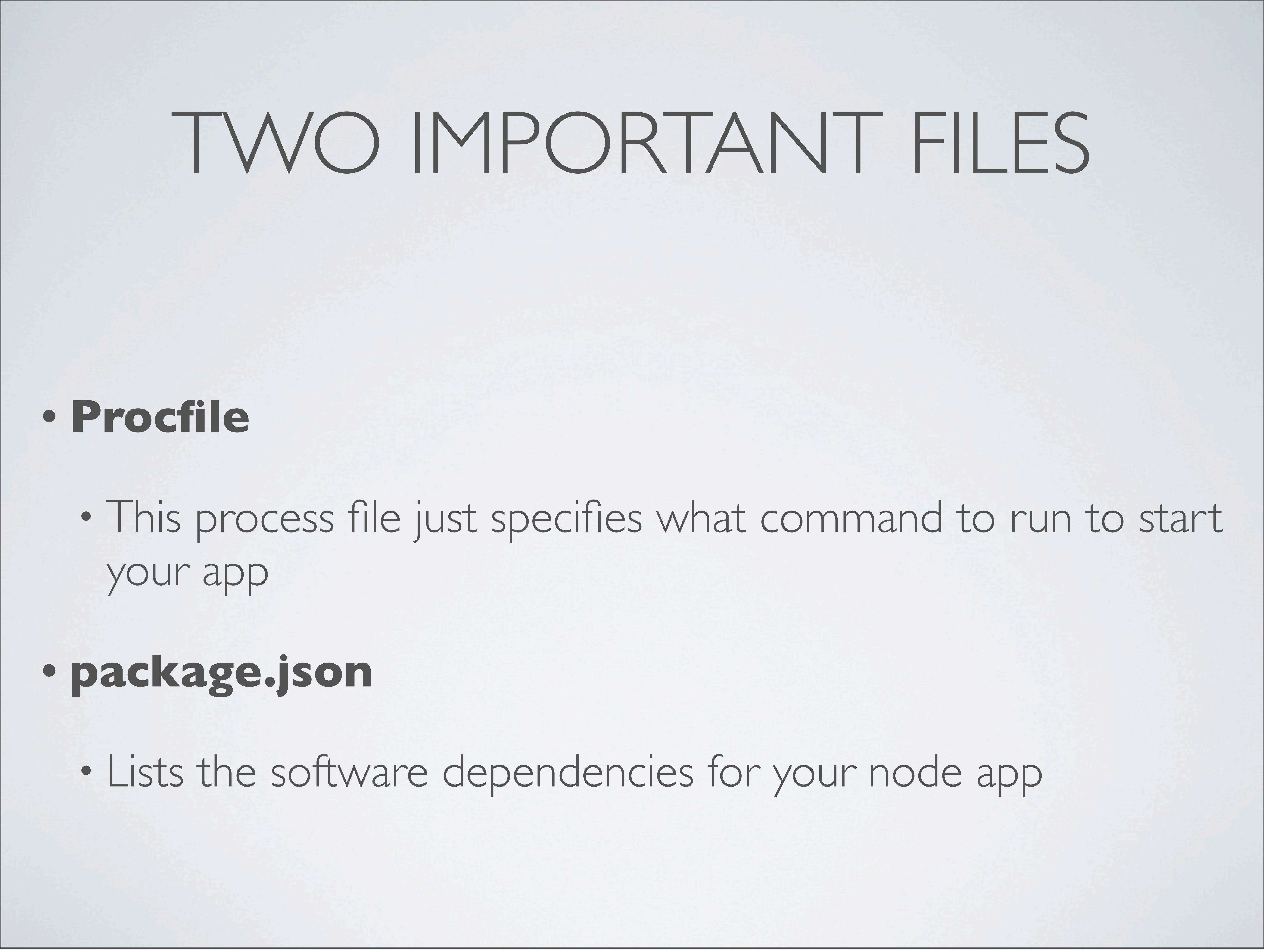
NUNJUCKS (making it possible to use jinja2 templating in node!)

# DEPLOY YOUR APP TO heroku

- Heroku allows you to deploy web applications to their servers, making them accessible via URL
- Not just for Node.JS! Also Python, Java, Scala..
- Also has free add-ons for features like logging, MySQL,

our app should have a package.json w/ necessary magic
Add a procfile with the following contents: 'web: node web.js'

http://www.heroku.com/



# TWO IMPORTANT FILES

- **Procfile**
  - This process file just specifies what command to run to start your app
- **package.json**
  - Lists the software dependencies for your node app

Monday, November 26, 12

Add a procfile with the following contents: ‘web: node web.js’

A few things to add to our package.json (need to specify node and NPM version), left out by default but necessary for Heroku:

```
  "engines": {
    "node": "0.8.x",
    "npm": "1.1.x"
  }
```

# STEPS TO DEPLOYING

- Add it to git
- Create a project on Heroku
- Push your code
- Scale your Server

Monday, November 26, 12

https://devcenter.heroku.com/articles/nodejs

```
foreman start
git init
git add .
git commit -m “init”
heroku create

heroku apps:rename newappname

git push heroku master

heroku ps:scale web=1
heroku ps
```