What is Node.js?

Session ID:
Agenda Key:
First, how do IBM i keep up?

LPP OPS – watch this space
... Node.js
... Python
... more coming
... ( http://www.ibm.com/developerworks/ibmi/techupdates/opensource )

<table>
<thead>
<tr>
<th>5733-OPS Option 4</th>
<th>Python 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5733-OPS Option 5</td>
<td>Node.js v4</td>
</tr>
<tr>
<td>5733-OPS Option 6</td>
<td>Git</td>
</tr>
</tbody>
</table>
Open Source Technologies on IBM i

<table>
<thead>
<tr>
<th>Node.js</th>
<th>API Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMBA on IBM i</td>
<td></td>
</tr>
<tr>
<td>Python</td>
<td></td>
</tr>
</tbody>
</table>

- DB2 for i access APIs (with synchronized support)
- DB2 for i access APIs - old
- Toolkit for IBM i APIs
What is Node.js?
Node.js® is a platform built on Chrome's JavaScript runtime

- easily building fast, scalable network applications
- event-driven, non-blocking I/O model
- lightweight and efficient
- perfect for data-intensive real-time applications

Node.js moves to IBM i
Yes, Node.js on IBM i!

Node.js, like all successful Open Source, has many contributors

- IBM i Node.js contribution
  - DB2 access (new)
  - Toolkit XMLService (new)
Node.js® like asynchronous RPG sub-procedures

- Different threads sub-procedure tasks (same time)
- Callback when task completed

**Synchronous**

**Asynchronous**
Node.js example (Bears)
Node.js bears project download.

https://bitbucket.org/litmis/nodejs

In download zip file find:
> node4_tutorial_bears
Step 0 - Node.js bears project requirements
Node.js bears project.

Project requirements.
> JSON API interface
> Browser interface
> Serve 2000/hits second (7.2 million/hour)
> Current LPAR hardware partition (small)
> Isolate Node.js from root of machine
> Integrated into existing Apache site
> Different teams API and GUI (browser)
> Use only JavaScript and html skills

Programmer note:
... partridge in a pear tree, ya crazy zoo keeper ...
Step 1 - Node.js bears isolate by chroot (admin)
When a user profile with a "." (dot) in their home directory path connects using ssh, sftp, or scp path the "/" (root) directory for that connected session is set to be the portion of the directory path prior to the "."

On 400
======
> strTCPSVR SERVER(*SSH)
> mkdir -p /QOpenSys/node4/home/node4
> CRTUSRPRF USRPRF(NODE4) PASSWORD() USRCLS(*PGMR) TEXT('Tony Cairns')
> CHGUSRPRF USRPRF(NODE4) LOCALE(*NONE) HOMEDIR('/QOpenSys/node4/./home/node4')

On laptop
==========
$ ssh node4@myibmi
Step 1 – Isolation node.js from root machine

*** download latest ibmichroot (IBM i use profile-admin) ***
https://bitbucket.org/litmis/ibmichroot/downloads
$ ./chroot_setup.sh chroot_minimal.lst /QOpenSys/node4
$ ./chroot_setup.sh chroot_nls.lst /QOpenSys/node4
$ ./chroot_setup.sh chroot_OPS_SC1.lst /QOpenSys/node
$ ./chroot_setup.sh chroot_gen_OPS_tools.lst /QOpenSys/node4
$ ./chroot_setup.sh chroot_gen_OPS_Node4.lst /QOpenSys/node4

*** make node4 owner of new chroot (IBM i use profile-admin) ***
$ chroot /QOpenSys/node4 /QOpenSys/usr/bin/ksh
$ cd /
$ chown -Rh node4 /
$ exit
Step 2 - Node.js bears json api (node4 user)
Step 2 - sign on to chroot provided by admin.

*** ssh to new chroot (node4 profile) ***
$ ssh -X node4@ut28p63
node4@ut28p63's password:
$ ksh
$ export PATH=.:/QOpenSys/QIBM/ProdData/OPS/Node4/bin:/QOpenSys/QIBM/ProdData/OPS/tools/bin:/QOpenSys/usr/bin
$ bash
$ node --version
v4.4.6
$
Step 2 – npm more node.js toys from the internet.

*** Defining our Node Packages package.json ***

```
$ cat package.json
{
    "name": "node-api",
    "main": "server.js",
    "dependencies": {
        "express": "~4.0.0",
        "body-parser": "~1.0.1"
    }
}
```

*** Installing Our Node Packages ***

```
$ npm install
npm WARN package.json node-api@ No repository field.
npm WARN package.json node-api@ No license field.
body-parser@1.0.2 node_modules/body-parser
  : express@4.0.0 node_modules/express
```
Step 2 – try initial api server

server_step_1.js - Initial server

*** Setting Up Our Server server.js ***
$ cp server_step_1.js server.js

*** Starting Our Server and Testing ***
$ node server.js
Magic happens on port 8080

http://myibmi:8080/api
{"message":"hooray! welcome to our api!"}
Step 2 – set up bear model

*** bear_step_2.js - Initial REST model ***
$ cp app/models/bear_step_2.js app/models/bear.js

*** Testing ***
$ node test/test_bear_ctor.js
$ node test/test_bear_save.js
$ node test/test_bear_find.js
$ node test/test_bear_findById.js nbr
$ node test/test_bear_removeById.js nbr
**server_step_3.js - REST route and view**
$ cp server_step_3.js server.js
$ node server.js
Magic happens on port 8080

*** Testing ***
GET list
$ curl http://ut28p63:8080/api/bears

GET by id
$ curl http://ut28p63:8080/api/bears/3

POST save name
$ curl -d "name=Sally" http://ut28p63:8080/api/bears

DELETE by id
$ curl -X DELETE http://ut28p63:8080/api/bears/3
Step 2 – check the speed (Uh Oh)

***before cache (ab tool test) ***

$ ab -t 15 -c 60 http://ut28p63:8080/api/bears/2
Concurrency Level: 60
Time taken for tests: 15.072 seconds
Complete requests: 1351
Failed requests: 0
Write errors: 0
Total transferred: 276606 bytes
HTML transferred: 37719 bytes
Requests per second: 89.64 [#/sec] (mean)
Time per request: 669.361 [ms] (mean)
Time per request: 11.156 [ms] (mean, across all concurrent requests)
Transfer rate: 17.92 [Kbytes/sec] received
Step 2 – check speed with cache (Yahoo)

*** copy model cache code ***
$ cp app/models/bear_step_4.js app/models/bear.js

*** after cache (ab tool test) ***
$ ab -t 15 -c 60 http://ut28p63:8080/api/bears/2
Concurrency Level:      60
Time taken for tests:   15.003 seconds
Complete requests:      38197
Failed requests:        0
Write errors:           0
Total transferred:      7563006 bytes
HTML transferred:       1031319 bytes
Requests per second:    2545.91 [#/sec] (mean)
Time per request:       23.567 [ms] (mean)
Time per request:       0.393 [ms] (mean, across all concurrent requests)
Transfer rate:          492.28 [Kbytes/sec] received
Step 3 - Node.js bears gui (browser)
Step 3 – GUI team set-up view to use our JSON API

*** app/view ***
index.html -- list bears
bear_find.js -- list bears create
bear_add.js -- add a bear name
bear_findById.js -- single bear
bear_id.html -- view single bear action
bear_gone.html -- remove bear
bear_remove.js -- remove bear delete
helper.js -- helper js functions
bear.css -- view layout
Step 3 – configure gui (admin)

*** view (admin -- outside chroot) ***

*** config ***
> cp /QOpenSys/node4/home/node4/node4_tutorial_bears/conf/httpd.conf /www/apachedft/conf/.
> cd /www/apachedft/htdocs
> ln -sf /QOpenSys/node4/home/node4/node4_tutorial_bears/app/view bears
bash-4.3$ ls -l
lrwxrwxrwx    1 adc      0               112 Oct 20 10:06 bears -> /QOpenSys/node4/home/node4/node4_tutorial_bears/app/view
-rwx---r-x    1 qsys     0              1008 Oct  7 2009  index.html
```

*** run (browser) ***
http://ut28p63.rch.stglabs.ibm.com/bears/
Step 3 – Open for business test

**Bear List**

- Add

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>78</td>
<td>Bob</td>
</tr>
<tr>
<td>82</td>
<td>Gina</td>
</tr>
<tr>
<td>72</td>
<td>Jimmy</td>
</tr>
<tr>
<td>92</td>
<td>Lee</td>
</tr>
<tr>
<td>89</td>
<td>Liv</td>
</tr>
<tr>
<td>90</td>
<td>Mildred</td>
</tr>
<tr>
<td>91</td>
<td>Paul</td>
</tr>
<tr>
<td>85</td>
<td>Sally</td>
</tr>
</tbody>
</table>

**Bear Find By Id**

- 89 - Liv (remove)

```html
home
row found id (89)
```

**Bear Remove**

```html
(89)
home
row found id (89)
```
Step 4 – Show the boss
Happy project manager.

Project requirements.
yes> JSON API interface
yes> Browser interface
yes> Serve 2000/hits second (7.2 million/hour)
yes> Current LPAR hardware partition (small)
yes> Isolate Node.js from root of machine
yes> Integrated into existing Apache site
yes> Different teams API and GUI (browser)
yes> Use only JavaScript and html skills
Bluemix Node.js remote IBM i?
http://myibminodejs.mybluemix.net/

BLUEMIX cloud node.js to IBM i...

Q: New technology?
A: IBM i Node.js toolkit (available IBM i)

(application not always running)
Web Application: http://myibminodejs.mybluemix.net/
Git repository: https://hub.jazz.net/project/rangercairns/MyIBMNodeJs/overview

(application not always running, but source available)
Bluemix app move to IBM i?
Bobby download Git ...

Git repository: https://hub.jazz.net/project/rangercairns/MyIBMiNodeJs/overview

Download and unzip on your pc. Transfer all files up to IBM i, into your chroot location, same directories as in zip (/home/bobby/MyIBMiNodeJs).
Bobby chroot application ...

```bash
$ cd /home/bobby/MyIBMNodeJs
$ ls
License.txt            instructions.md        public
README.md              lib                    views
README.txt             manifest.yml           xmlserviceprovider.js
app.js                 package.json

$ ksh
$ export PATH=/QOpenSys/QIBM/ProdData/Node/bin:/usr/bin
$ export LIBPATH=/QOpenSys/QIBM/ProdData/Node/bin:/usr/lib
$ node --version
v0.10.29
```
Bobby chroot application ...

$ node app.js
module.js:340
  throw err;
Error: Cannot find module 'express'

$ npm install express
module.js:340
  throw err;
Error: Cannot find module 'npmlog'

Never goes perfectly, most presentations lie about 'ops', this one shows it all.
Bobby chroot application ...

$ cd /QOpenSys/QIBM/ProdData/Node/bin
$ ls ../lib/node_modules/npm/bin/npm-cli.js
../lib/node_modules/npm/bin/npm-cli.js
$ ln -sf ../lib/node_modules/npm/bin/npm-cli.js npm
$ ls -l npm
lrwxrwxrwx  1 bobby  0                76 Nov  9 07:48 npm -> ../lib/node_modules/npm/bin/npm-cli.js
$ cd
$ npm --version
1.4.14

$ npm install express
npm ERR! fetch failed https://registry.npmjs.org/buffer-crc32/-/buffer-crc32-0.2.1.tgz
npm ERR! fetch failed https://registry.npmjs.org/mkdirp/-/mkdirp-0.3.5.tgz
npm ERR! fetch failed https://registry.npmjs.org/fresh/-/fresh-0.2.0.tgz
npm ERR! fetch failed https://registry.npmjs.org/batch/-/batch-0.5.0.tgz
express@3.4.7 node_modules/express

Never goes perfectly, most presentations lie about 'ops', this one shows it all.
$ node app.js
App started on port 3000

events.js:72
    throw er; // Unhandled 'error' event
Error: listen EADDRINUSE

$ export VCAP_APP_PORT=8081
$ export VCAP_APP_HOST=ut30p30
$ node app.js
App started on port 8081
module.js:340
    throw err;
^{
Error: Cannot find module 'jade'

Never goes perfectly, most presentations lie about 'ops', this one shows it all.
Bobby chroot application ...

$ npm install jade
npm ERR! fetch failed https://registry.npmjs.org/character-parser/-/character-parser-1.2.0.tgz
jade@1.1.4 node_modules/jade
$ node app.js
App started on port 8081

This IBM i application moved from Bluemix is remotely chatting with Yips IBM i machine. It works my_IBMi<2>yips_IBMi, but, demo assuming your IBM i can talk REST web services outside firewall to HTTP (of course).
Also, Ruby Rails use Node.js?
PowerRuby Rails Asset Pipeline requires JavaScript (Node.js)
- framework concatenate JavaScript and CSS assets
- hybrid lang assets CoffeeScript, Sass and ERB.
- CoffeScript compiles into JavaScript (via Node.js)

> rails new appname

Rails Asset Pipeline
# Assignment:
number = 42
opposite = true
# Conditions:
number = -42 if opposite
# Functions:
square = (x) -> x * x
# Arrays:
list = [1, 2, 3, 4, 5]
# Objects:
math =
  root: Math.sqrt
  square: square
  cube: (x) -> x * square x

var cubes, list, math, number, opposite, square;
number = 42;
opposite = true;
if (opposite) {
  number = -42;
}
square = function(x) {
  return x * x;
};
list = [1, 2, 3, 4, 5];
math = {
  root: Math.sqrt,
  square: square,
  cube: function(x) {
    return x * square(x);
  }
};
Questions?