



Cloud Tech: Your new bag of tricks

Jeffrey Bisti

IBM Poughkeepsie

jbisti@us.ibm.com

#SHAREorg



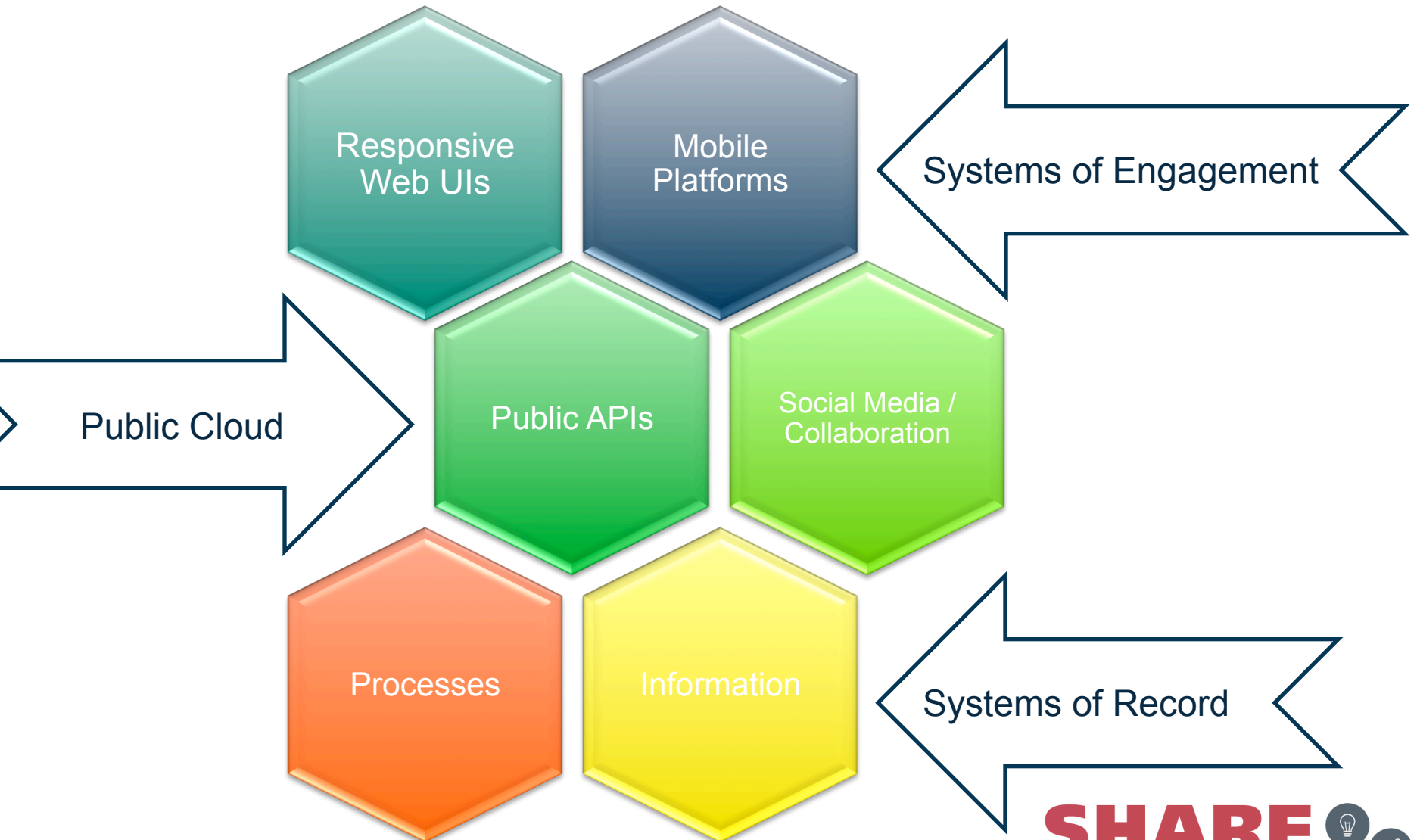
SHARE is an independent volunteer-run information technology association that provides **education, professional networking and industry influence.**



Agenda

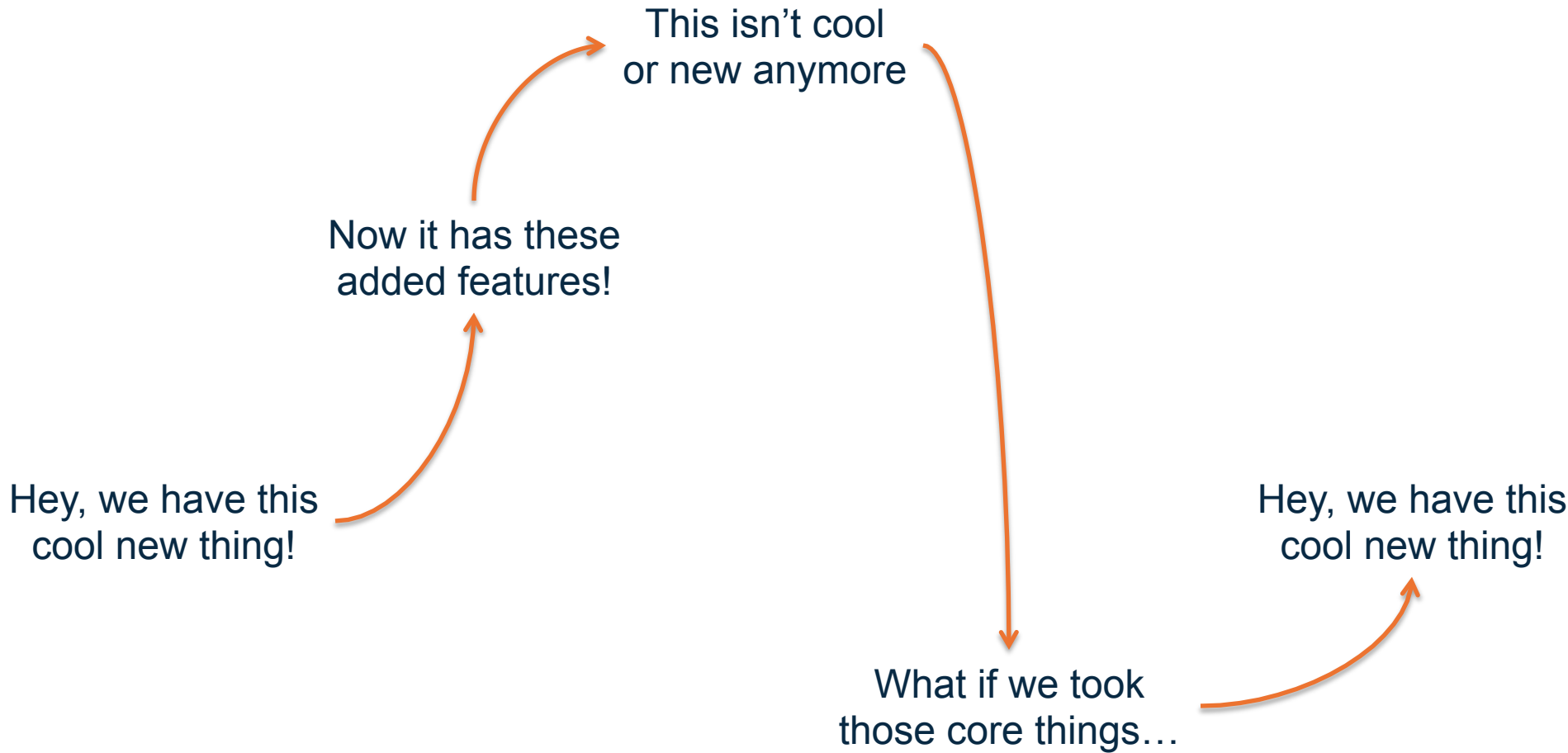
- **What's **really** changed?**
- **Some new Tricks**
- **The Plan of Attack**

Evolving to a SoE / SoR Mindset



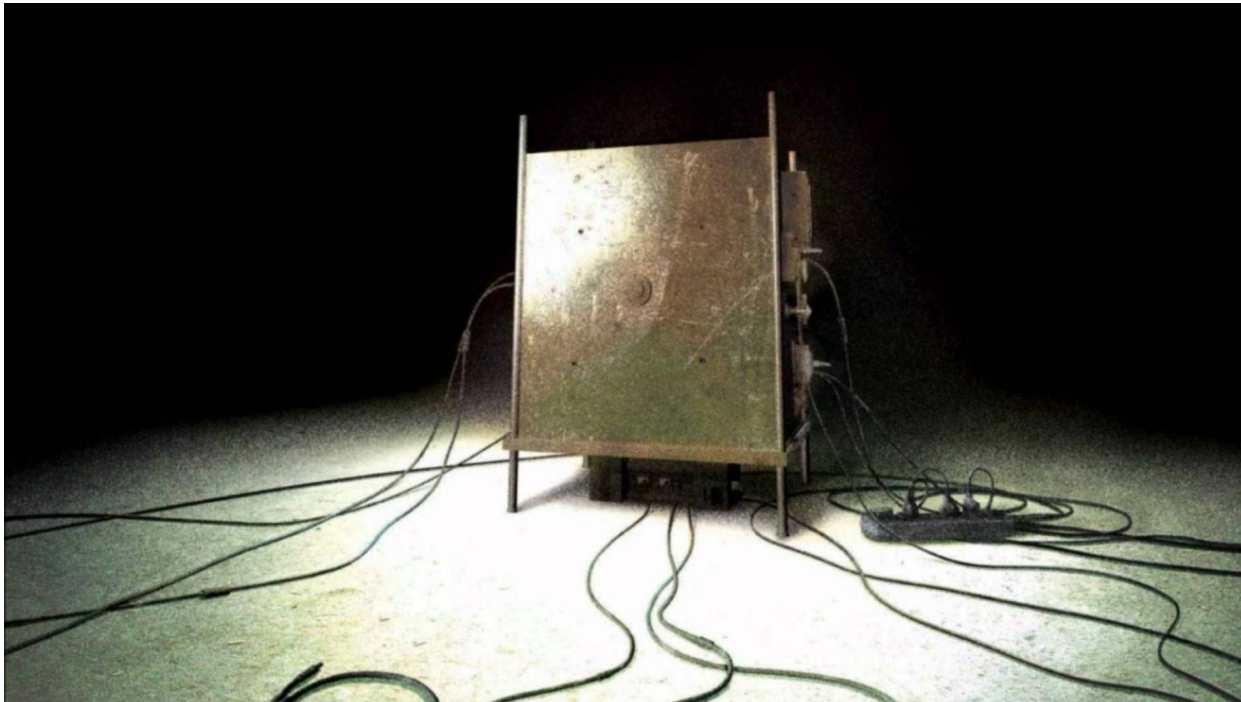
Everything Old is New Again

Except Disco... what were you people thinking?



REST & Statelessness Programming

- REST APIs – The Language of Cloud
 - Representational State Transfer
 - Open (don't need to develop a plugin)
 - Just Google “[thing you want to use] REST API” and GO!



HTTP Verbs

GET

PUT


POST

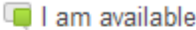
DELETE




Thinking Asynchronously


Find a person

 **Brian**





Brian: Hi
Me: Hi
Brian: How are you?
Me: Fine, thanks
Brian: Can I ask you a question?
Me: Yes
Brian: It's a really quick question, shouldn't take more than five minutes
Me: Ok, what is it
Brian: Ok, here's the question I want to ask you...

b i u 

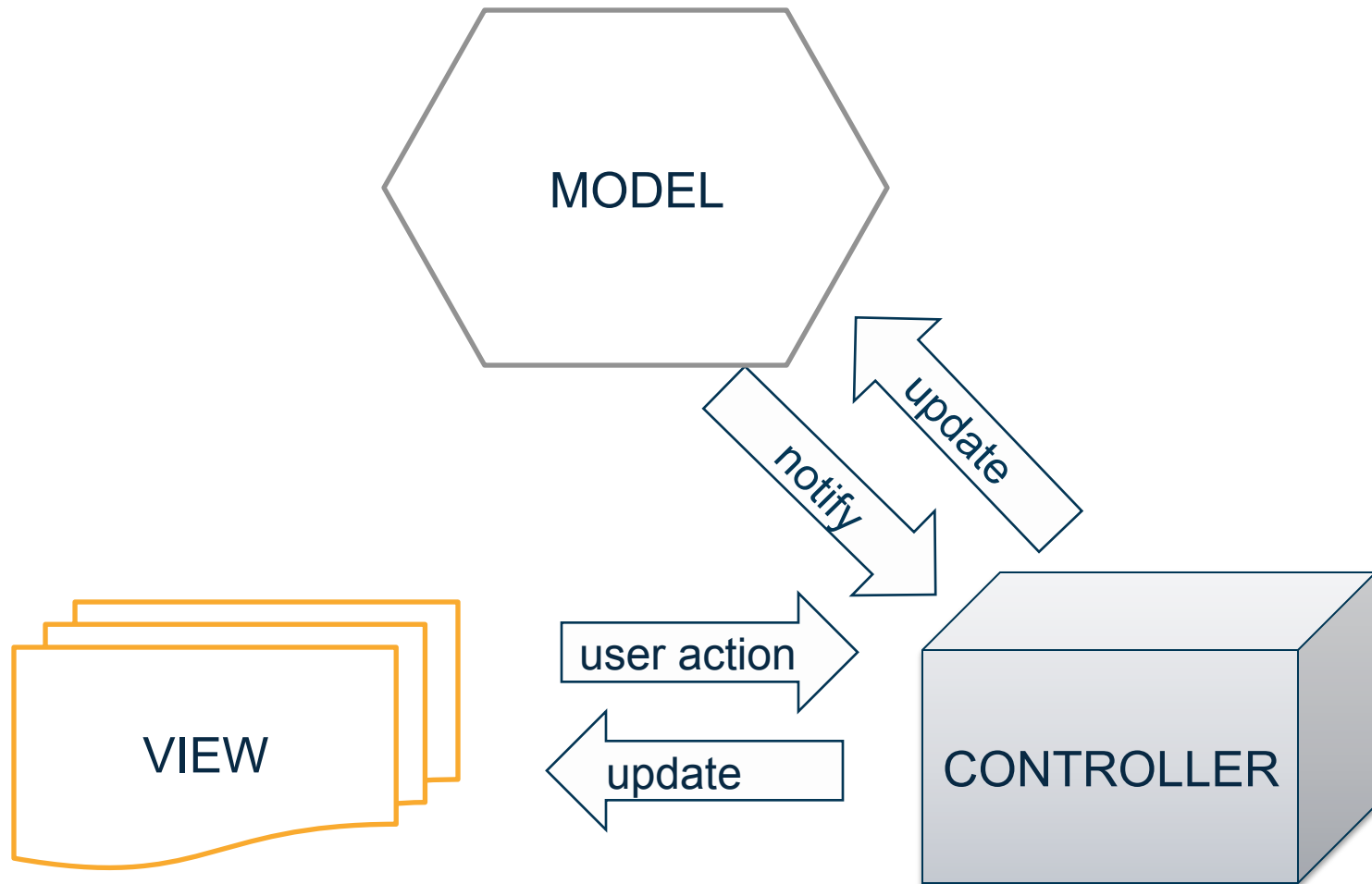
JUST ASK THE QUESTION ALREADY!!!!

Thinking Asynchronously with Callbacks

```
fs.readdir(source, function(err, files) {
  if (err) {
    console.log('Error finding files: ' + err)
  } else {
    files.forEach(function(filename, fileIndex) {
      console.log(filename)
      gm(source + filename).size(function(err, values) {
        if (err) {
          console.log('Error identifying file size: ' + err)
        } else {
          console.log(filename + ' : ' + values)
          aspect = (values.width / values.height)
          widths.forEach(function(width, widthIndex) {
            height = Math.round(width / aspect)
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)
            this.resize(width, height).write(destination + 'w' + width + '_' + filename, function(err) {
              if (err) console.log('Error writing file: ' + err)
            })
          }).bind(this)
        }
      })
    })
  }
})
```

Callbackhell.com

MVC – Model / View / Controller



MVC – Model / View / Controller



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

MVC – Model / View / Controller

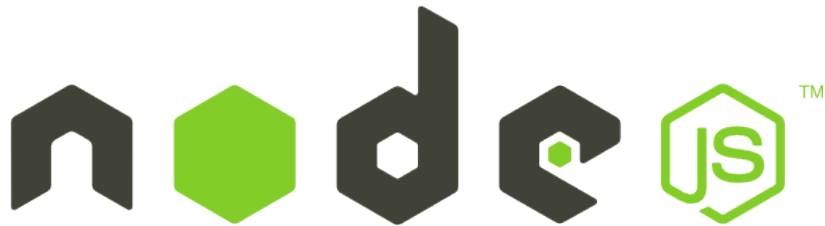


Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Modern Javascript

More than just scrolling text and blinking lights this time

```
var server = net.createServer(function (socket) {  
  socket.end("goodbye\n");  
});  
  
// grab a random port.  
server.listen(function() {  
  address = server.address();  
  console.log("opened server on %j", address);  
});
```



JSON – JavaScript Object Notation

JSON Uses JavaScript Syntax

Because JSON uses JavaScript syntax, no extra software is needed to work with JSON within JavaScript.

With JavaScript you can create an array of objects and assign data to it like this:

Example

```
var employees = [  
  {"firstName":"John", "lastName":"Doe"},  
  {"firstName":"Anna", "lastName":"Smith"},  
  {"firstName":"Peter", "lastName": "Jones"}  
];
```

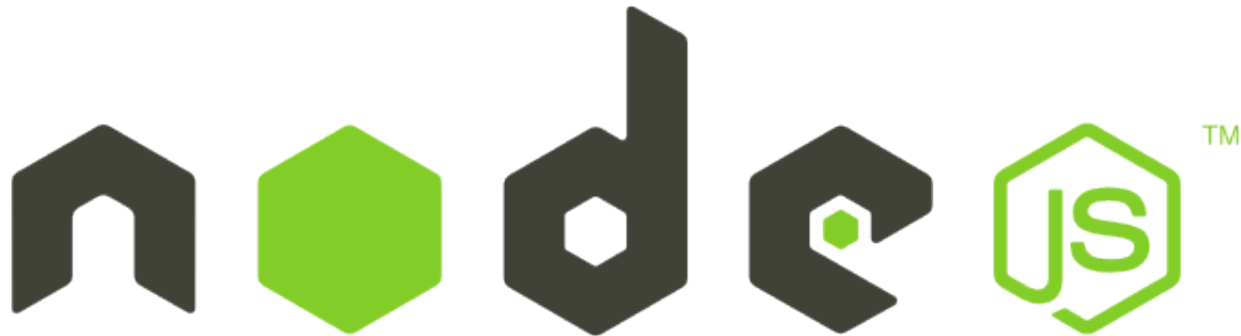
The first entry in the JavaScript object array can be accessed like this:

```
employees[0].firstName + " " + employees[0].lastName;
```

The returned content will be:

```
John Doe
```

Node.JS



Node.js® is a platform built on **Chrome's JavaScript runtime** for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Current Version: v0.10.35

[INSTALL](#)

[DOWNLOADS](#)

[API DOCS](#)

Frameworks and Toolkits



The stuff you're always going to use
should always be there

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Express

Create an Express Project

Create a new Node.js Express project (using `express` CLI underneath, that should be installed before with `npm install -g express`)

Project name:

Use default location

Location:

Template Engine: Jade ejs jshtml hogan.js

Stylesheet Engine: CSS LESS Stylus





Working sets

Add project to working sets

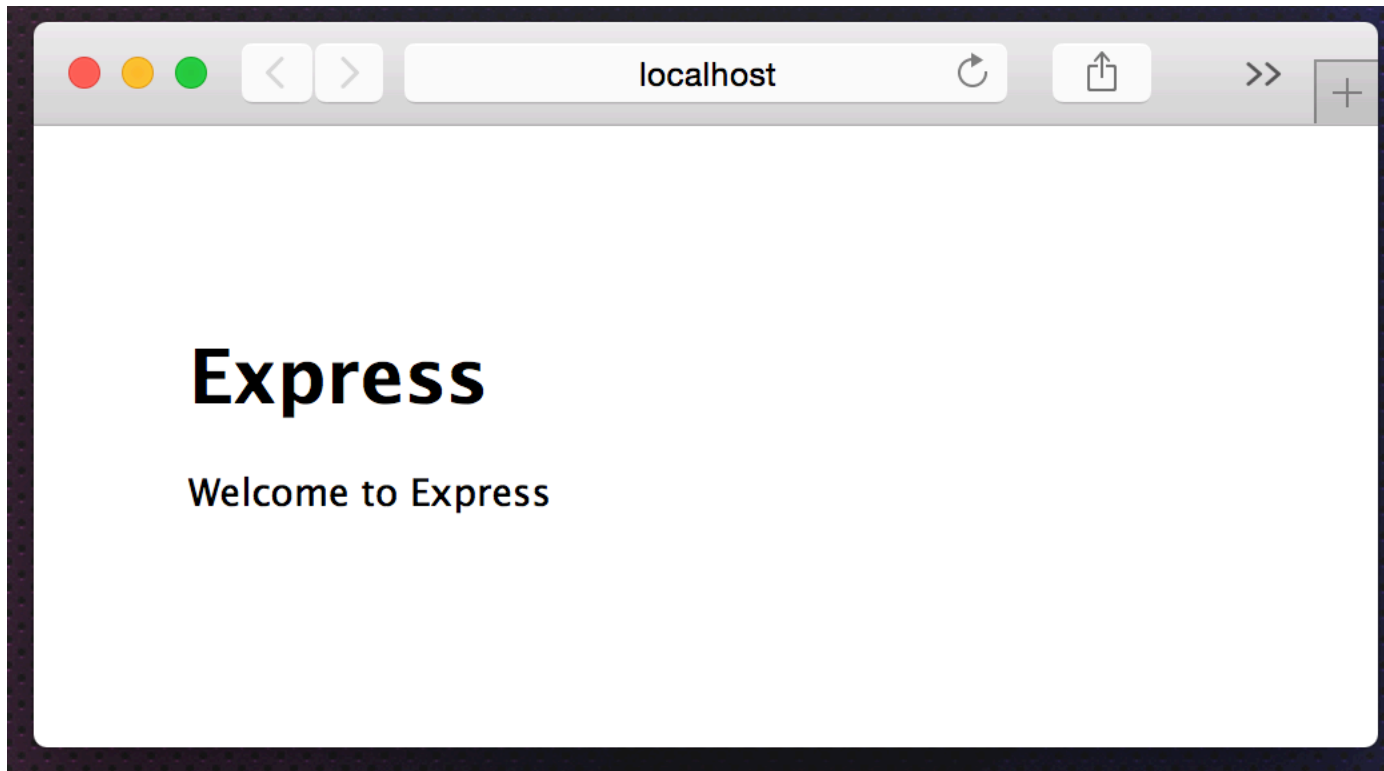
Working sets:



Express

Console  Debug  Markdown HTML Preview  Problems  Error Log

MyAwesomeNewApp-app.js [Node Application] Node.js Process
Express server listening on port 3000



Angular



index.html

```
1. <!doctype html>
2. <html ng-app>
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.10/
angular.min.js"></script>
5.   </head>
6.   <body>
7.     <div>
8.       <label>Name:</label>
9.       <input type="text" ng-model="yourName" placeholder="Enter a name here">
10.     <hr>
11.     <h1>Hello {{yourName}}!</h1>
12.   </div>
13. </body>
14. </html>
```

Angular

Framework for client-side JavaScript, useful for building powerful single-page documents. Directives and Controllers

Rapid development of modern interfaces using cloud resources

Users: MSNBC, NASA, Vevo, Google, Weather.com, Kijiji

Baked-in testability

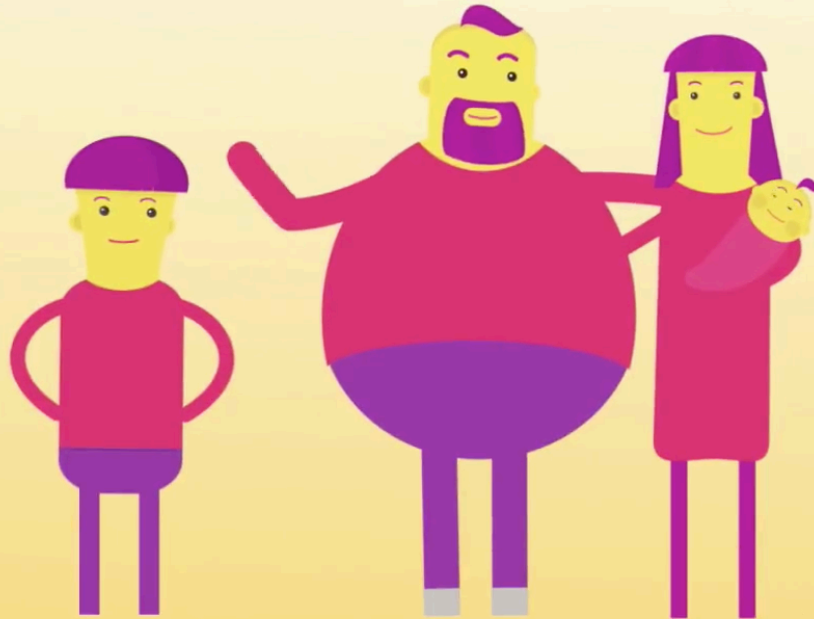
What it's good for: Putting a pretty front end to web resources. Models for Web Services. Last-mile filtering/messaging of data.

What it's NOT good for: Control freaks.

Also: the most adorable online tutorial ever. Ever.

(shaping up with Angular.js - <https://www.codeschool.com/courses/shaping-up-with-angular-js>)

THE FLATLANDERS



NoSQL Databases



redis



Cloudbant



mongoDB



Remember those JSON objects?
Here's a good place to keep them.

[InfoQ](#) has released a video of [Twitter's](#) Kevin Weil speaking at [Strange Loop](#) earlier this year on how the company uses NoSQL. Weil is quick to point out that Twitter is heavily dependent on MySQL. However, Twitter does employ NoSQL solutions for many purposes for which MySQL isn't ideal. According to Weil, Twitter users generate 12 terrabytes of data a day - about four petabytes per year. And that amount is multiplying every year. Read on for our notes on Weil's talk.

Complete your session ev

Life With Bluemix

Services // The building blocks of any great app

Watson

Build cognitive apps that help enhance, scale, and accelerate human expertise



Concept Expansion
IBM BETA



IMARS-image-tagger
IBM BETA



Language Identification
IBM BETA



Machine Translation
IBM BETA



Message Resonance
IBM BETA



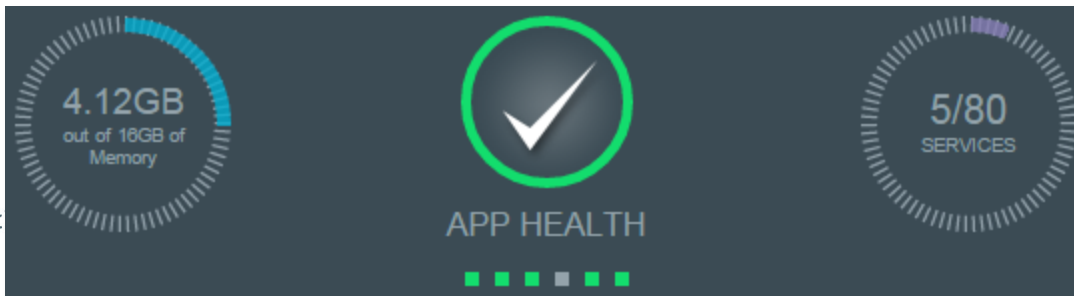
Question Answer
IBM BETA



Relationship Extraction
IBM BETA



Speech Recognition
IBM BETA



4.12GB
out of 18GB of
Memory

APP HEALTH

5/80
SERVICES

Complete your session evaluation

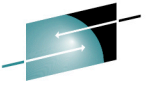
SHARE

San Diego 2015

Lightbulb icon, Share icon, Globe icon

Node-RED

Even *I* can do this

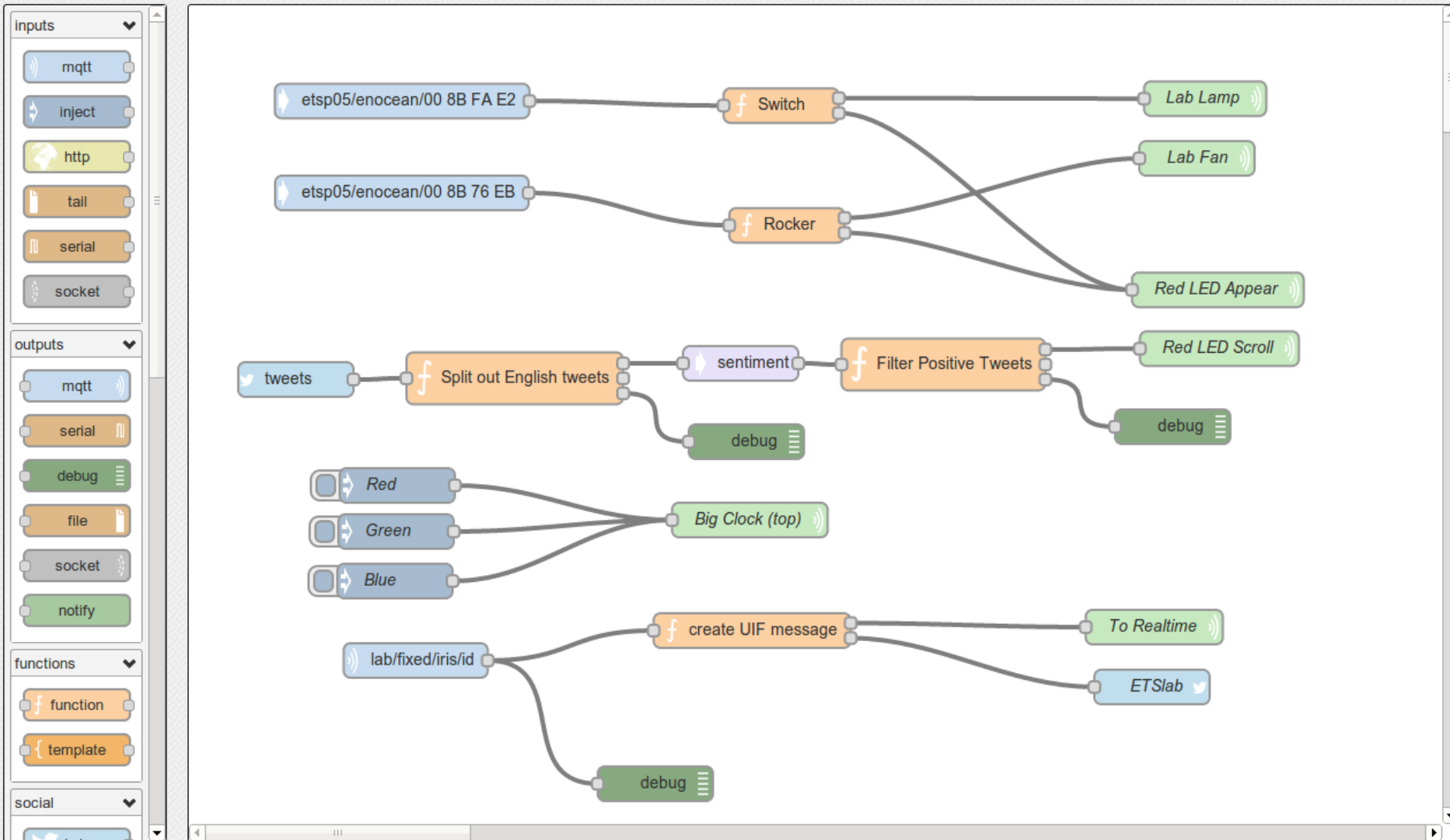


SHARE
Educate · Network · Influence

Node-RED

Deploy

More



Node-RED

Even *I* can do this

A selection of the nodes available



Transform

JS Functions
Mustache Templates



Network

HTTP • TCP
UDP • MQTT



Social

Twitter • Twilio
Email • IRC • XMPP



Hardware

Raspberry Pi
BeagleBone Black



Storage

Filesystem • MongoDB
MySQL • PostgreSQL



Analysis

Sentiment
Statistics

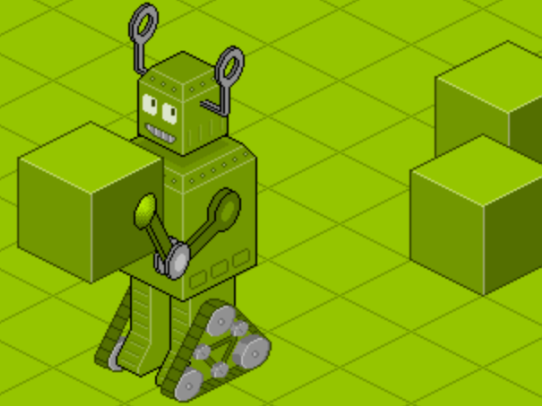
SWAGGER

Admit it, you kind of want to use it, just on name alone



SWAGGER

The World's Most Popular Framework for APIs.



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

SWAGGER

Good documentation isn't difficult. It's ~fancy~

```
File ▾ Preferences ▾ Generate Server ▾ Generate Client ▾ Help ▾
1 # this is an example of the Uber API
2 # as a demonstration of an API spec in YAML
3 swagger: '2.0'
4 info:
5   title: Pictures of Cats
6   description: Allows for the dynamic generation of pictures of cats
7   version: "1.0.0"
8 # the domain of the service
9 host: picturesofcats.mybluemix.net
10 # array of all schemes that your API supports
11 schemes:
12   - https
13 # will be prefixed to all paths
14 basePath: /v1
15 produces:
16   - application/json
17 paths:
18   /products:
19     get:
20       summary: Request a picture of a cat
21       description: |
22
23       parameters:
24         - name: Color
25           in: query
26           description: Preferred color of kitteh
27           required: false
28           type: string
29           format: string
30         - name: Attitude
31           in: query
32           description: Specify the mood of kitteh
33           required: false
34           type: string
35           format: string
36         - name: Quantity
37           in: query
38           description: The number of cat pictures requested
39           required: true
40           type: number
41           format: int
42       tags:
43         - Products
44       responses:
45         200:
46           description: An array of URLs with cat pictures
```

All changes saved.

Pictures of Cats

Allows for the dynamic generation of pictures of cats

Version 1.0.0

Paths

Filter operations by a tag:

Products Estimates User

/products

GET /products

Products

Summary

Request a picture of a cat

Parameters

Name	Located in	Description	Required	Schema
Color	query	Preferred color of kitteh	No	⇔ string (string)
Attitude	query	Specify the mood of kitteh	No	⇔ string (string)
Quantity	query	The number of cat pictures requested	Yes	⇔ number (int)

Responses

Code	Description	Schema
200	An array of URLs with cat	⇔ [Product {}]

SWAGGER UI : Colorful, yet refined

pet : Everything about your Pets

Show/Hide | List Operations | Expand Operations

POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet
GET	/pet/findByStatus	Finds Pets by status
GET	/pet/findByTags	Finds Pets by tags
DELETE	/pet/{petId}	Deletes a pet
GET	/pet/{petId}	Find pet by ID
POST	/pet/{petId}	Updates a pet in the store with form data
POST	/pet/{petId}/uploadImage	uploads an image

SWAGGER : Swings both ways

PUT /pet

Update an existing pet

Parameters

Parameter	Value	Description	Parameter Type	Data Type				
body	<div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div> <p>Parameter content type: <input type="text" value="application/json"/></p>	Pet object that needs to be added to the store	body	<table border="1"><tr><td>Model</td><td>Model Schema</td></tr><tr><td colspan="2"><pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "doggie", "photoUrls": ["string"], "tags": [{ </pre></td></tr></table> <p>Click to set as parameter value</p>	Model	Model Schema	<pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "doggie", "photoUrls": ["string"], "tags": [{ </pre>	
Model	Model Schema							
<pre>{ "id": 0, "category": { "id": 0, "name": "string" }, "name": "doggie", "photoUrls": ["string"], "tags": [{ </pre>								

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	Invalid ID supplied		
404	Pet not found		
405	Validation exception		

Try it out!

Your Next Steps

- Find a problem
- Sketch
- Research
- Build!

Complete your session evaluations online at www.SHARE.org/Orlando-Eval Jeffrey Bisti – jbisti@us.ibm.com

Design and Priority Considerations

- Keep tech out of the first few rounds of whiteboarding.
- If you can't find a good tutorial, your internet is probably broken.
- Things change. Keep everyone in the loop.



Jeffrey Bisti – jbisti@us.ibm.com

Complete your session evaluations online at www.SHARE.org/Orlando-Eval