**Topic:**

Node.js Cookbook – Patterns and Good Practices when writing a Node.js REST Service

**Contact Information:**

**Name: Guillaume Brossard and David Richer**

**Location: Montreal**

**Division & Team:** **M&E – Cloud Services – Core Team**

**E-mail:**  **guillaume.brossard@autodesk.com; david.richer@autodesk.com**

**Speaker/Exhibitor Biography:**

|  |
| --- |
| Guillaume Brossard previously worked at GRIP Entertainment, where he designed and built an artificial intelligence behavior tree system for the video game industry, and at Bluestreak Technology, developing on a real-time Adobe Flash compatible rendering engine for embedded devices.  David Richer also previously worked at GRIP Entertainment where he designed and built a real-time character system for video game digital extras. He has over 10 years’ experience in C++.  Both speakers are experienced C++ programmers that became JavaScript enthusiasts. They are working together in the M&E Cloud Service Core Team, building Web Services and Tools that are the building blocks of M&E Cloud Service Web Application. |

**Proposal Abstract:**

The M&E Cloud Service team has been using Node.js for a year, and a lot patterns have emerged from our REST services.   
  
The goal of this presentation is to share a set of patterns and good practices when writing a REST service with Node.js. This presentation is for developers by developers, and will focus at helping developers writing robust and reusable REST services. Some key topics includes:  
- JavaScript and Node.js Patterns  
- Express Middlewares  
- Exceptions Handling  
- Minimize Callback Hell  
- Useful libraries  
  
This will be done using real-life examples from Project Scotty sources, and demonstrated using Project Scotty development tools.

**Intended Audience & Prerequisites:**

* Node.js developers, from beginners to experienced
* Back-end developers that are curious about Node.js
* Node.js knowledge preferred, but not required

**Presentation outline:**

* Service Project Hierarchy
* Express.js
* Good Patterns
  1. Controllers
  2. Classical Inheritance
  3. Middleware
* Exception Handling
  1. Node.js Error Convention
  2. Exception Throwing
  3. Domains
* Avoiding Callback Hell
  1. async
  2. Promises
* Good Software Development Practices
  1. JSLint
  2. Testing
* Service configuration
  1. nconf
* Useful Libraries
  1. superagent
  2. underscore / lodash
* Testing
  1. Jasmine
* Automation
  1. Grunt
* Debugging
  1. node-inspector
  2. WebStorm
* Production Foresight
  1. node-inspector
  2. logging (winston)
  3. tracing (tracing middleware + winston)
  4. longjohn

**Additional materials:**

Project Scotty Source Code

Project Scotty Development Tools

NPM Registry Shopping List