CP 104 - Introduction to Node.js

Summary	Since the CloudPortal client code is written in JavaScript, we thought we'd have the server component to CloudPortal written in JavaScript using Node.js. This asynchronous, event-driven framework provides a different perspective on traditional server code. This course will give a very brief overview of the technology, as most interaction will be done with the Express layer that sits on top of Node.
Level	Advanced
Time	1 hour
Prerequisites	CP 102
Author	Howard Abrams
Editor	John Hopprich

Introduction

Node.js (or just *Node*) is a stand-alone engine that executes JavaScript applications on the server. This technology is quite fast (built with Google's V8 Engine), but is quite low-level and somewhat verbose. Typically, you will layer high-level abstractions and frameworks on top of it (like Express).

Advantages of applications written in Node.js include:

- Fast
- Scalable
- Light-weight and efficient

These advantages are due to its event-driven, non-blocking I/O model. This approach may take a while to get used to, however.

Class Outline

- Introduction and Background to Node.js
- Installation
- Example Code
    - Trivial Example
    - Serving static files
    - Answering REST requests
    - Accessing a Database
    - Accessing back-end REST Servers
- Debug Node.js Applications
- References

Conclusion / Expected Outcome

At the end of this course, you should be able to do the following:

- Have Node.js and NPM installed in your Development Environment. You probably should just run the CloudPortal Golden Script to do this.
- Create a simple web server that serves a static message for all HTTP requests.
- Understand how to trouble-shoot typical asynchronous gotchas that often arrive in Node.js programs.
- Show how to debug a Node.js application from the STS IDE.

Note: Most of our interaction with Node, will actually be through Express.