Kraken Forge at PayPal
Day 2: Userland and Best Practices

Objectives

On completion of this course, students will be able to:

- Use the internal debugging tools to step through code.
- Understand when and how to apply debugging flags.
- Understand the principles of unit testing and be able to apply them using Mocha.
- Demonstrate test coverage.
- Understand the components of an Express.js app.
- Understand when to use nconf and when not to.
- Create Dust.js (LinkedIn fork) templates.
- Manage user sessions in Express.
- Write sophisticated asynchronous control flows with clarity.
- Generate application scaffolding with Yoeman.

Training Units

- Debugging Node.js
- TDD and Testing with Mocha
- Express.js
- Flow Control with Async
- Introducing Yeoman

Detailed Syllabus

- Debugging Node.js
    - Adding breakpoints
    - Using the internal debugger
    - V8 Flags
    - Attaching a remote debugger
    - Connecting to WebStorm
    - Using node-inspector
    - Debugging async code
- TDD and Testing with Mocha
    - Principles of Unit Testing
    - BDD style testing
    - Test Coverage
    - NOTE: XUnit Reporter compatibility a plus
        - Still an open issue for Mocha: https://github.com/visionmedia/mocha/issues/10
- Express.js
    - Express and Connect
    - Configuration
        - nconf
    - Routing and Controllers
    - Error Handling
    - View Rendering
        - Dust.js (LinkedIn fork)
    - Handling Sessions
        - Session lifecycle
        - Anti-patterns: in-memory sessions

    - Middleware
- Flow Control with Async
    - When to use Callbacks, Event Emitters, or Streams
    - Using the Async
    - Understanding when to use Async vs Callbacks
- Introducing Yeoman
    - Generate application scaffolding