Groovy Automation with Spok Framework Gradle project

Day1

Groovy Fundamentals

- Differences between Groovy and Java
- Compiling and executing Groovy programs
- The basic Groovy data types and optional typing
- Writing Groovy scripts
- Declaring classes
- Overriding operators and type coercion
- GDK: the Groovy library
- Groovy strings
- Regular expressions in Groovy

Groovy Collections

- Ranges
- Lists
- Maps
- Iterators and polymorphic algorithms

Closures in Groovy

- Declaring closures
- Available options for calling closures

Groovy Control Structures

- The "Groovy truth"
- Conditional execution
- Looping constructs

Using Classes and Scripts

- Groovy fields and local variables
- Methods and operations
- Safe dereferencing with the ?. operator
- Organizing classes in packages
- Using inheritance
- POGO's vs. POJO's

Regular Expressions

- Regular Expression syntax
- The =~ operator
- The ==~ operator
- Common methods that use Regular Expressions

Builders and Slurpers

- What are Builders and Slurpers?
- NodeBuilder
- MarkupBuilder
- AntBuilder
- Using the ConfigSlurper

Unit Testing in Groovy

- GroovyTestCase
- Unit-testing Groovy code
- Unit-testing Java code
- Code coverage and automation

Groovy Builders

- NodeBuilder
- MarkupBuilder
- SwingBuilder

Survey of the GDK

- Simplified File access
- Templates
- Groovlets

Day 2

Database Access with Groovy

- Basic database operations
- DataSet operations
- Groovy and ORM solutions

Working with XML

- Reading and parsing XML documents
- Using XPath expressions
- Distributed processing with XML

Working With JSON,CSV and Text Files

- Simple Metaprogramming
- Metaclasses
- Injecting methods with Categories
- The Expandmetaclass

Web Development with Grails

- The Grails framework
- Creating web applications using Grails

The Gradle Project

- About the project
- Documentation / Resources / Support

Gradle Introduction

- The Gradle philosophy
- Installing Gradle
- Gradle build scripts
- The build lifecycle
- The Gradle build daemon

Tasks

- Declaring tasks
- Executing tasks
- Configuring tasks
- Implementing custom tasks
- Extending tasks
- Task dependencies
- Task exclusion
- Conditionally skipping tasks
- The Task Graph API
- Task rules

Day 3

Spock, JUnit & TestNG

- Configuration
- Reporting
- Cookie management
- Jar and class names
- Example Projects

Build System & Framework Integrations

- Grails
- Gradle
- Maven

Grails in a hurry...

- Why Grails?
- Getting set up
- Our sample program: a Web 2.0 QOTD
- Creating the domain model
- Adding UI actions
- Improving the architecture
- Summary and best practices

Modeling the domain

- Hubbub: starting our example application
- Your first domain class object
- Validation: stopping garbage in and out
- Defining the data model—1:1, 1:m, m:n
- Summary and best practices

Putting the model to work

- Creating instant UIs with scaffolding
- Groovy querying with dynamic finders and Query by Example
- More sophisticated query options
- Bootstrapping reference data
- Summary and best practices

Controlling application flow

- Controller essentials
- Services: making things robust and maintainable
- Data binding
- Command objects
- Working with images
- Intercepting requests with filters
- Creating custom URL mappings
- Summary and best practices

Day 4

Developing tasty views, forms, and layouts

- Understanding the core form tags
- Extending views with your own tags
- Adding delicious layouts
- Applying Ajax tags
- Summary and best practices

Building reliable applications

- Why should we test software?
- Unit testing
- Integration testing
- Functional testing
- Summary and best practice

Using plugins: adding Web 2.0 in 60 minutes

- Taking advantage of others' hard work
- Adding charts and graphs
- Adding mail support
- Full-text search: rolling your own search
- GrailsUI makeover
- Summary and best practices

Using JSON Plugin

- Introduction to JSON
- Provides JSON and XML Conversion for common Objects (Domain Classes, Lists, Maps, POJO)

Advanced GORM

- Domain model kung fu
- Caching : moving from 2 users to 2^10
- Legacy integration kung fu: dealing with multiple data sources
- Dealing with difficult legacy databases
- Summary and best practices

Spring and transactions

- Spring and Grails
- Using transactions with GORM
- Summary and best practices

Beyond compile, test, and run

- The Grails build system
- Build integration—not for the hobbyist
- Coping with a changing data model
- Summary and best practices