Effective Unit Testing in Java EE

Rob Di Marco
VP, Technology
Health Market Science
rdimarco@hmsonline.com

3/28/2007

What Will We Accomplish

- Set up our Maven 2.0 project
- Compile the Java code
- Create a JUnit test validating the session and entity beans
- Create a Project Website with diagnostic information
- Integrate with Struts 2 (Time Permitting)

What Technologies Will Be Involved

- Java APIS
  - JDK 5.0
    - Annotations
    - Enhanced For Loop
    - Static Import
  - EJB 3.0
    - Entity Beans
    - Session Beans
- Tools
  - Maven 2.0
  - MavenProxy
  - Continuum
  - Eclipse
- Libraries
  - JUnit 4.0
  - JBoss Embedded EJB Container
  - Hypersonic DB
  - Cobertura
  - PMD
  - FindBugs
  - Struts 2

Our Example Code: Java PetStore 2.0

- Description from Sun

  The Java Pet Store 2.0 is the reference application for building AJAX web applications on Java Enterprise Edition 5 platform.

- Use Entity Beans for demonstrating persistence

- Use Session Beans for business logic
  - Logic moved from the PetStore *CatalogFacade* class

Maven 2.0

- What is Maven?
  - *Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.*
- Why use Maven?
  - Allows for easy establishment of development best practices
  - Creates a consistent cross-project build process
  - Third party tool integration makes for better builds and documentation
- Maven 1.0 vs. Maven 2.0

Starting with Maven 2.0

- Install Maven 2.0
- Create a Project Object Model
  - Using Maven archetypes to setup a base project
  - Create a POM manually
- Optionally can use $HOME/.m2/settings.xml for configuration
  - Allows for creation of internal MavenProxy repository
- Understanding Key Phases of the Maven Lifecycle
- Local dependency repository created at $HOME/.m2/repository/

Reviewing the PetStore Project Setup

- POM
  - Project Information
  - Build Information
    - Setups dependencies to download from the repository
  - Reporting Information
- Directory structure
  - src/main/java
  - src/main/resources
  - src/test/java
  - src/test/resources
- Setting up Eclipse with Maven

Reviewing Pet Store Code

- Entity Beans
  - Address
  - Category
  - Item
  - Product
  - SellerContactInfo
  - Tag
  - ZipLocation
- Session Bean
  - PetStoreBean
  - Adapted from PetStore's CatalogFacade class
- Compile using Maven
  - mvn compile

Using an Embedded JBoss Server

- Embeddable JBoss version that allows for embedding JEE functionality
  - Local JNDI
  - Transaction Manager
  - JMS
  - Local TX datasource/connection pool
  - Stateful, Stateless, Service, Consumer, Producer, and MDBs
  - EJB 3 Persistence
  - EJB Security
- Based off JBoss 5.0 tree

Setting up a EJB Server in the Unit Test

- Use @BeforeClass annotation to set up the server
  - Start JBoss kernel
  - Create in-memory Hypersonic data source
  - Tell JBoss to EJBs in the classpath
    - Looks for classes marked with EJB and JPA annotations
- Use @Before to initialize a context and seed the database
- Use @AfterClass for shutting down the server
- Use @Test to test the code

Building Project Documentation

- Use the site phase of the Maven lifecycle
- Creates a website that documents project information
- Project reports point to potential problems with the application
  - PMD for coding rule enforcement
  - CPD for code duplication
  - FindBugs for bug detection
  - Cobertura for code coverage

Results of Maven install

- Unit tests executed and results logged
- Jar File for EJBs created
- Jar File for EJB client created
- Jar and POM files published to local repository

Integrating our EJBs Using Struts 2

- Create POM for Web project
  - Dependency on EJB client
- Create a Struts 2 Actions
  - No ActionForms, parameters set using reflection
- Use a ServiceLocator
  - Façade pattern abstracts implementation of EJB calls
  - Factory pattern for creation of ServiceLocator allows for better Unit Testing

Comments on the Solution

- Every tool demonstrated has an Open Source
  - Mostly Apache, GPL, or LGPL licenses
  - No initial implementation cost
  - Varying levels of user community and support
- Valid technique even if not using JBoss in production
- Many of the tools are still in beta
  - Be prepared to handle implementation bugs and patches

Relevant Articles

- Maven Site Generation
  - http://www.javaworld.com/javaworld/jw-02-2006/jw-0227-maven_p.html?page=1
- Embedded JBoss
  - http://www.devx.com/Java/Article/30496

References

- Java Pet Store
  - http://java.sun.com/developer/releases/petstore/
- Maven
  - http://maven.apache.org
  - http://maven-proxy.codehaus.org/
  - http://m2eclipse.codehaus.org/
- Cobertura
  - http://cobertura.sourceforge.net/
- Embedded EJB 3.0
  - http://docs.jboss.org/ejb3/embedded/embedded.html

References Continued

- PMD
  - http://pmd.sourceforge.net/
- FindBugs
  - http://findbugs.sourceforge.net
- Struts 2
  - http://struts.apache.org