

Testing Your Application on / for Google App Engine

Narinder Kumar

Inphina Technologies

inphina

IndicThreads.com Conference On
Upcoming Technology



2010: Cloud Computing

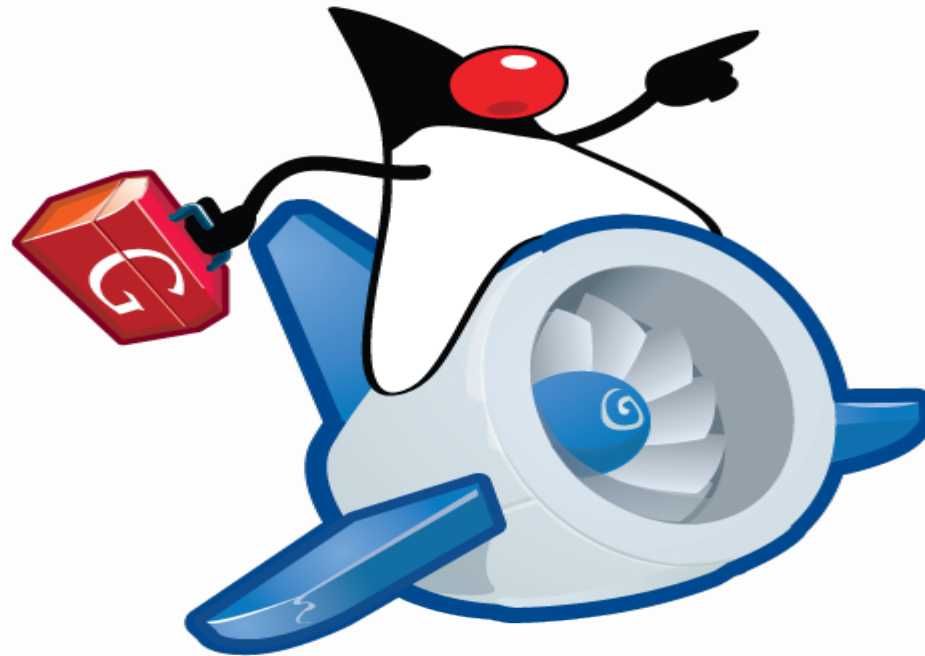
Pune, India

Agenda

- ◆ Problem Context
- ◆ App Engine Testing Framework
 - ◆ Local DataStore Testing
 - ◆ Authentication API Testing
 - ◆ Memcache Testing
- ◆ Google Cloud Cover - An Overview



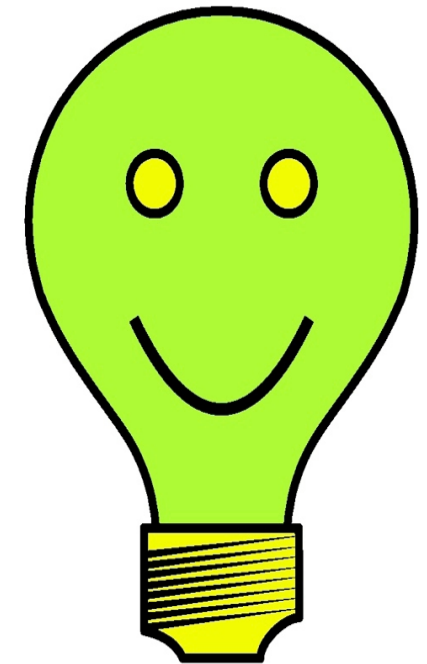
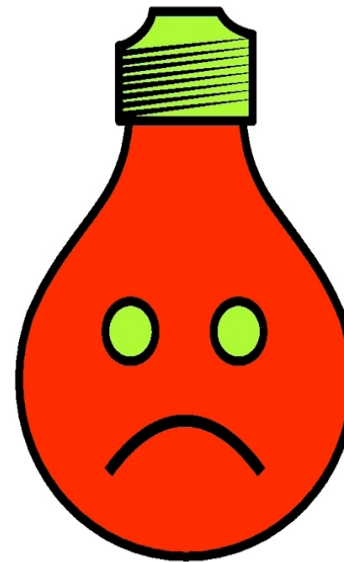
Easy to build, Easy to Maintain, Easy to Scale



Why Testing

- ◆ To verify correctness of the code
- ◆ To be assured of continued correctness of old code
- ◆ To avoid surprises
- ◆ To safely make large refactorings

Debugging
sucks.



Testing rocks.

I feel sad and naked without Good Test Coverage

Max Ross (Member of Google App Engine Team)



Do we need App Engine Specific Testing Strategies ?



Our Goal

To be able to test in our local environment using

- ◆ Spring
- ◆ Maven
- ◆ JPA/JDO
- ◆ JUnit
- ◆ Continuous Integration



Google Infrastructure Services

- ◆ DataStore
- ◆ Memcache
- ◆ TaskQueue
- ◆ Authentication
- ◆ ...



App Engine Testing Framework

- ◆ LocalServiceTestHelper
 - ◆ LocalDataStoreServiceTestConfig
 - ◆ LocalMemCacheServiceTestConfig
 - ◆ LocalTaskQueueTestConfig
 - ◆ LocalUserServiceTestConfig
 - ◆ ...



Local Data Store Testing



Step 1 : Make RunTime Libraries Available Locally

```
<dependency>
  <groupid>com.google.appengine</groupid>
  <artifactid>appengine-testing</artifactid>
  <version>1.3.4</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupid>com.google.appengine</groupid>
  <artifactid>appengine-api-labs</artifactid>
  <version>1.3.4</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupid>com.google.appengine</groupid>
  <artifactid>appengine-api-stubs</artifactid>
  <version>1.3.4</version>
  <scope>test</scope>
</dependency>
```

Some of these libraries are not available in Central Maven Repositories



Step 2 : Enhance Domain Classes

<plugin>

```
<groupid>org.datanucleus</groupid>
<artifactid>maven-datanucleus-plugin</artifactid>
<version>1.1.4</version>
<configuration>
<mappingincludes>**/domain/*.class</mappingincludes>
  <verbose>>true</verbose>
  <enhancername>ASM</enhancername>
  <api>JPA</api>
</configuration>
<executions>
  <execution>
    <phase>compile</phase>
    <goals>
      <goal>enhance</goal>
    </goals>
  </execution>
</executions>
```

.....

<plugin>



Prepare Run-Time Environment and Dependencies

```
@RunWith(SpringJUnit4ClassRunner.class)
@TestExecutionListeners( { DependencyInjectionTestExecutionListener.class } )
@ContextConfiguration(locations = { "classpath:test-applicationContext.xml",
"classpath:test-applicationContext-dao.xml"})
public class LocalDatastoreSpringTestCase extends TestCase {

    private final LocalServiceTestHelper helper =
        new LocalServiceTestHelper(new LocalDatastoreServiceTestConfig());

    @Before
    public void setUp() {
        helper.setUp();
    }

    @After
    public void tearDown() {
        helper.tearDown();
    }
}
```



Write Specific Tests

```
public class EmployeeDaoSpringTest extends LocalDatastoreSpringTestCase {  
  
    @Autowired  
    private EmployeeDao employeeDao;  
  
    @Test  
    public void testShouldPersistEmployee() {  
        Employee employee = new Employee();  
        employee.setFirstName("Scott");  
        employee.setLastName("Adams");  
        employee.setHireDate(new Date());  
  
        employeeDao.createEmployee(employee);  
  
        Collection<employee> list = employeeDao.list();  
  
        Assert.assertEquals(1, list.size());  
    }  
}
```



Authentication API Testing

```
public class AuthenticationTest {  
  
    private final LocalServiceTestHelper helper =  
        new LocalServiceTestHelper(new LocalUserServiceTestConfig())  
            .setEnvIsAdmin(true).setEnvIsLoggedIn(true);  
  
    @Test  
    public void testIsAdmin() {  
        UserService userService = UserServiceFactory.getUserService();  
        assertTrue(userService.isUserAdmin());  
    }  
  
    @Before  
    public void setUp() {  
        helper.setUp();  
    }  
  
    @After  
    public void tearDown() {  
        helper.tearDown();  
    }  
}
```



Memcache Testing

```
public class LocalMemcacheTest {  
  
    private final LocalServiceTestHelper helper =  
        new LocalServiceTestHelper(new LocalMemcacheServiceTestConfig());  
  
    @Test  
    private void testInsert() {  
        MemcacheService ms = MemcacheServiceFactory.getMemcacheService();  
        assertFalse(ms.contains("yar"));  
        ms.put("yar", "foo");  
        assertTrue(ms.contains("yar"));  
    }  
  
    ....  
    // SetUp and Tear Down
```



Google Cloud Cover – An Overview



Key Features

- ◆ Provides ability to run your tests in Cloud
- ◆ Designed to run Existing Test Suites
- ◆ Tests Execute in Parallel
 - ◆ Creates one Task Queue per Test
 - ◆ Number of workers determined by Queue Config
- ◆ Allows to Run Large Test Suites faster : Acts as a Test Grid



How to Set it up

- ◆ Create a Standard GAE/J web application with all Production Code & Test Code and dependencies
- ◆ Add Cloud Cover Dependencies to WAR
- ◆ Create a TestRunner Config around your Test Suite
- ◆ Add Cloud Cover Servlet

http://<your_app_id>/cloudcover.html



Google Cloud Cover in Action



Different from Normal Testing

- ◆ Each Test must complete in 30 seconds
- ◆ Application Code and Test Code must obey
Sandbox restrictions
- ◆ Need to invoke Tests via HTTP



Conclusions

- ◆ Local RunTime Environment very helpful during Development Phase
- ◆ Google Cloud Cover can be a good aid in certain areas but need more refinement



inphina

nkumar@inphina.com

www.inphina.com

<http://thoughts.inphina.com>



References

- ◆ <http://code.google.com/appengine/docs/java/tools/localunittesting.html>
- ◆ <http://code.google.com/p/cloudcover/>
- ◆ <http://thoughts.inphina.com/2010/06/28/unit-testing-maven-based-jpa-application-on-gae/>
- ◆ <http://objectuser.wordpress.com/category/software-development/google-app-engine/>
- ◆ <http://code.google.com/events/io/2010/sessions.html#App%20Engine>

