

Native Android Development with Spring

Roy Clarkson

Spring Mobile Projects Lead

SpringSource, a division of VMware

@royclarkson

@springandroid

- **What are the purposes of the Spring for Android and Spring Mobile projects?**
 - Spring for Android provides support for building native Android applications utilizing Spring technologies, where applicable.
 - In contrast, Spring Mobile provides support for building mobile web applications.



Spring for Android

- Android Overview
- Define the Problem
- Review of REST
- Basic Rest Template Example
- Rest Template Overview
- Maven Can Help
- Showcase and Demos
- Questions



Android Adoption

- Year-on-year growth rate of more than 250%
- 850,000 new Android devices are activated each day
- Over 300 million devices around the world
- Over 450,000 apps available in Google Play
- Over 1 billion app downloads per month

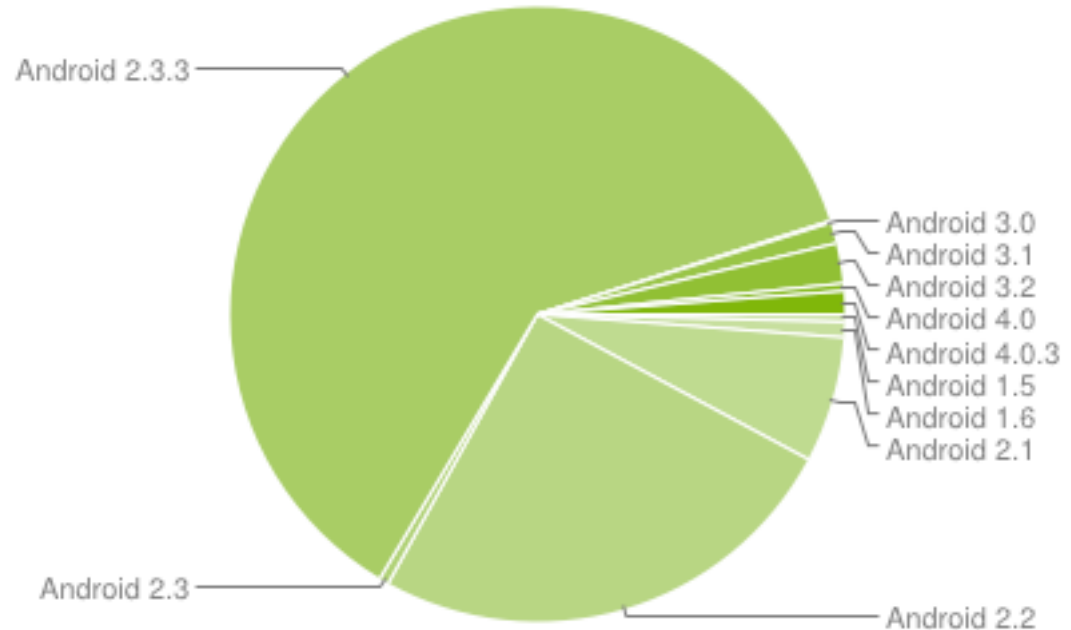


<http://googlemobile.blogspot.com/2012/02/androidmobile-world-congress-its-all.html>

Android Fragmentation

■ Current Distribution

- Gingerbread 62%
- Froyo 25.3%
- Eclair 6.6%
- Honeycomb 3.3%
- Ice Cream Sandwich 1.6%



<http://developer.android.com/resources/dashboard/platform-versions.html>

Applications are written in Java!
(well, sort of)



But not *that* Familiar...

■ Class Files

- Android apps are compiled to class files.

■ DEX Format

- Classes are compiled into the Dalvik Executable (DEX) format.
- DEX format is required to run on the Dalvik Virtual Machine.

■ Dalvik VM

- Not a true Java VM, because it does not operate on Java byte code.
- Based on a subset of the Apache Harmony project.
- Many of the classes from Java SE are available, but not all.

■ Unique Linux User ID

- Android OS assigns each app a unique Linux user ID.

■ Process Isolation

- Within the VM, each app runs in its own Linux process.

■ Managed Lifecycle

- The system manages the starting and stopping.

Components of an Android App

- **Activities** - An activity represents a single screen with a user interface.
- **Services** - A service is a component that runs in the background to perform long-running operations or to perform work for remote processes.
- **Content Providers** - A content provider manages a shared set of application data.
- **Broadcast Receivers** - A broadcast receiver is a component that responds to system-wide broadcast announcements.

<http://developer.android.com/guide/topics/fundamentals.html>

Android Manifest

- contains the permissions requested by the app
- Lists all of the components of an app

```
<uses-sdk android:minSdkVersion="10" />
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
        android:name=".HelloAndroidActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

A simple Activity

```
package org.springframework;

import android.app.Activity;
import android.os.Bundle;

public class HelloAndroidActivity extends Activity {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

HelloAndroid Demo



<http://www.springsource.org/springsource-tool-suite-download>

<http://developer.android.com/sdk/eclipse-adt.html>

How can Maven help?

■ Android4Maven

- This project compiles android.jar from source and pulls out source and resource files to replicate android.jar in the SDK
- <http://sourceforge.net/projects/android4maven/>

■ Maven Android SDK Deployer

- If you need to use Google maps, then you have to go this route
- <https://github.com/mosabua/maven-android-sdk-deployer>

■ Android Maven Plugin

- Provides support for Maven dependency management within Android projects
- <http://code.google.com/p/maven-android-plugin/>

Android Maven Plugin Configuration

```
<packaging>apk</packaging>
```

```
<plugin>
  <groupId>com.jayway.maven.plugins.android.generation2</groupId>
  <artifactId>android-maven-plugin</artifactId>
  <version>3.1.1</version>
  <configuration>
    <sdk>
      <platform>${android-platform}</platform>
    </sdk>
    <deleteConflictingFiles>true</deleteConflictingFiles>
    <undeployBeforeDeploy>true</undeployBeforeDeploy>
  </configuration>
  <extensions>true</extensions>
</plugin>
```

<https://repository.sonatype.org/index.html#nexus-search;quick~com.google.android>

Maven Commands

Build your Android App

```
$ mvn clean install
```

Start the Emulator

```
$ mvn android:emulator-start
```

Deploy the app to the emulator

```
$ mvn android:deploy
```

Run the app

```
$ mvn android:run
```

<http://maven-android-plugin-m2site.googlecode.com/svn/plugin-info.html>

Maven Demo



■ **Android Configurator for M2E Maven Integration for Eclipse**

- An Eclipse plugin that adds support for integrating m2eclipse, Android Developer Tools, and the Android Maven Plugin
- <http://rgladwell.github.com/m2e-android/>

■ **Maven Android archetypes**

- This project provides several Maven archetypes for Android. These archetypes allow you to quickly bootstrap a Maven project to develop an android application.
- <https://github.com/akquinet/android-archetypes>

This sounds too good!



Ricardo Gladwell
@rgladwell



looks as though ADT 17 breaks the build for m2e-android so please hold-off upgrading until this is fixed github.com/rgladwell/m2e-...

@rgladwell

Spring for Android



What problem are we trying to solve?

■ Concerns

- REST has become a popular choice for architecting both public and private web services
- The Android runtime provides HTTP clients capable of making HTTP connections and requests, but it does not have a fully featured REST client

■ Spring for Android Solution

- The goal of Spring for Android Rest Template is to provide an easy to use, and functional REST client that supports marshaling objects from XML and JSON.

■ Origin

- The term Representational State Transfer was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation.

■ His paper suggests these four design principles:

- Use HTTP methods explicitly.
 - POST, GET, PUT, DELETE
 - CRUD operations can be mapped to these existing methods
- Be stateless.
 - State dependencies limit or restrict scalability
- Expose directory structure-like URIs.
 - URI's should be easily understood
- Transfer XML, JavaScript Object Notation (JSON), or both.
 - Use XML or JSON to represent data objects or attributes

Basic Rest Template Example

■ Google search example

```
RestTemplate restTemplate = new RestTemplate();

String url =
    "https://ajax.googleapis.com/ajax/services/search/web?v=1.0&q={query}";

String result = restTemplate.getForObject(url, String.class, "SpringSource");
```

■ Multiple parameters example

```
RestTemplate restTemplate = new RestTemplate();

String url = "http://example.com/hotels/{hotel}/bookings/{booking}";

String result = restTemplate.getForObject(url, String.class, "42", "21");
```

Google Search Demo

■ Based on SpringFramework

- Originated as a fork of RestTemplate from SpringFramework.
- Modifications were made to support Android.

■ RestTemplate class is the heart of the library

- Entry points for the six main HTTP methods
 - DELETE - delete(...)
 - GET - getForObject(...)
 - HEAD - headForHeaders(...)
 - OPTIONS - optionsForAllow(...)
 - POST - postForLocation(...)
 - PUT - put(...)
 - any HTTP operation - exchange(...) and execute(...)

- **SimpleClientHttpRequestFactory**

- Delegates to the standard J2SE facilities

- **HttpComponentsClientHttpRequestFactory**

- Delegates to the native HttpComponents HttpClient

- **CommonsClientHttpRequestFactory** (deprecated)

- Delegates to the Commons HttpClient which is not native to Android

■ MappingJacksonHttpMessageConverter

- object to JSON marshaling supported via the Jackson JSON Processor

■ SimpleXmlHttpMessageConverter

- object to XML marshaling supported via the Simple XML Serializer

■ SyndFeedHttpMessageConverter

- RSS and Atom feeds supported via the Android ROME Feed Reader

Spring for Android Showcase

■ Examples

- HTTP GET
 - JSON
 - XML
- HTTP GET with Parameters
 - JSON
 - XML
- HTTP POST
 - String
 - JSON
 - XML
 - MultiValueMap
- RSS
- ATOM

Rest Template Demos

■ What is Spring Social?

- Spring Social is an extension of the Spring Framework that allows you to connect your applications with Software-as-a-Service (SaaS) providers such as Facebook and Twitter.

■ How does it relate to Spring for Android?

- Utilizes RestTemplate to make RESTful requests to providers.

■ How can Spring for Android help?

- Auth library in Spring for Android provides a SQLiteConnectionRepository for use with Spring Social.

Additional Resources

■ Slides:

- <http://portal.sliderocket.com/vmware/Native-Android-Development-with-Spring>

■ Project Home:

- <http://www.springsource.org/spring-android>

■ Sample Code:

- <https://github.com/SpringSource/spring-android-samples>

■ Webinar:

- <http://www.springsource.org/node/3505>

■ Blog Posts:

- <http://blog.springsource.com/author/rclarkson/>