# Publishing

Sang Shin
Michèle Garoche
www.javapassion.com
“Learn with Passion!”

# Disclaimer

- Portions of this presentation are modifications based on work created and shared by the Android Open Source Project
  - http://code.google.com/policies.html
- They are used according to terms described in the Creative Commons 2.5 Attribution License
  - http://creativecommons.org/licenses/by/2.5/

# Topics

- Signing your application
- Setting up publisher account
- Preparing your application
- Exporting and Signing application
- Publishing the application

# Signing Your Application

# Why Signing Android Applications?

- Because the android platform will not install an application that is not signed
  - On both emulator and real device
- You can use self-signed certificate to sign your applications.
  - No certificate authority signed certificate is needed.
- During debug mode (development time), the SDK tools automatically signs the app with a debug key
  - This is why you did not have to deal with the signing during development
- When you are ready to release your application for end-users, you must sign it with your own private key.
  - You can not publish an application that is signed with a debug key

# Certificate Expiration Date Checking

- The system tests a signer certificate's expiration date only at install time.
  - If an application's signer certificate expires after the application is installed, the application will continue to function normally.

# Debug-mode Signing during Development & Testing

- While developing and testing, you work in debug mode.
- In debug mode, the SDK build tools use the Keytool utility, included in the JDK, to create a keystore and key with a known alias and password.
- At each compilation, the tools then use the debug key to sign the application .apk file.
- Because the password is known, the tools don't need to prompt you for the keystore/key password each time you compile

# Release Mode Signing

- When your application is ready for release, you must build the application in release mode and then sign the .apk with your own private key.
- Two ways of doing release-mode signing
  - Option #1: Using command line tools
  - Option #2: Using Eclipse ADT

# Option #1: Steps of Signing using Command-line tools

1. Build the application into unsigned .apk file
2. Create private key using “keytool” (from JDK)
3. Sign the .apk file using “jarsigner” (from JDK)
4. Perform additional optimization using “zipalign” (from Android SDK)

# Option #2: Signing Using Eclipse ADT

- Use Export Wizard for performing all the steps
- Or you might want to create private key using “keytool” command line tool then use Export Wizard to perform the rest of the steps

# Setting up a Publisher Account

# What is Android Market?

- Android Market is an online software store developed by Google for Android devices.
- An application program ("app") called "Market" is pre-installed on most Android devices and allows users to browse and download apps published by third-party developers, hosted on Android Market.

# What is Publisher Account?

- You need a publisher account before uploading your application to Android market
- You also manage/control licensing (we will cover this in the “licensing” topic)

# Preparing Your Application Before Publication

# Preparation Tasks

- Set application verison
  - Make sure “versionCode” and “versionName” are set in the manifest file
- Make sure icon and label are set
- Test your application on actual devices

# Where Version Info. is used

- Users need to have specific information about the application version that is installed on their devices and the upgrade versions available for installation.
- Other applications — including other applications that you publish as a suite — need to query the system for your application's version, to determine compatibility and identify dependencies.
- Services through which you will publish your application(s) may also need to query your application for its version, so that they can display the version to users.

# versionCode & versionName

- android:versionCode
  - An integer value that represents the version of the application code, relative to other versions.
  - The value is an integer so that other applications can programmatically evaluate it, for example to check an upgrade or downgrade relationship.
- android:versionName
  - A string value that represents the release version of the application code, as it should be shown to users.
  - The value is a string so that you can describe the application version as a <major>.<minor>.<point> string, or as any other type of absolute or relative version identifier.

# versionCode and versionName

- In AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.package.name"
    android:versionCode="2"
    android:versionName="1.1">
  <application android:icon="@drawable/icon"
                 android:label="@string/app_name">
     ...
  </application>
</manifest>
```

# Export and Sign the Application

# Tasks to be performed

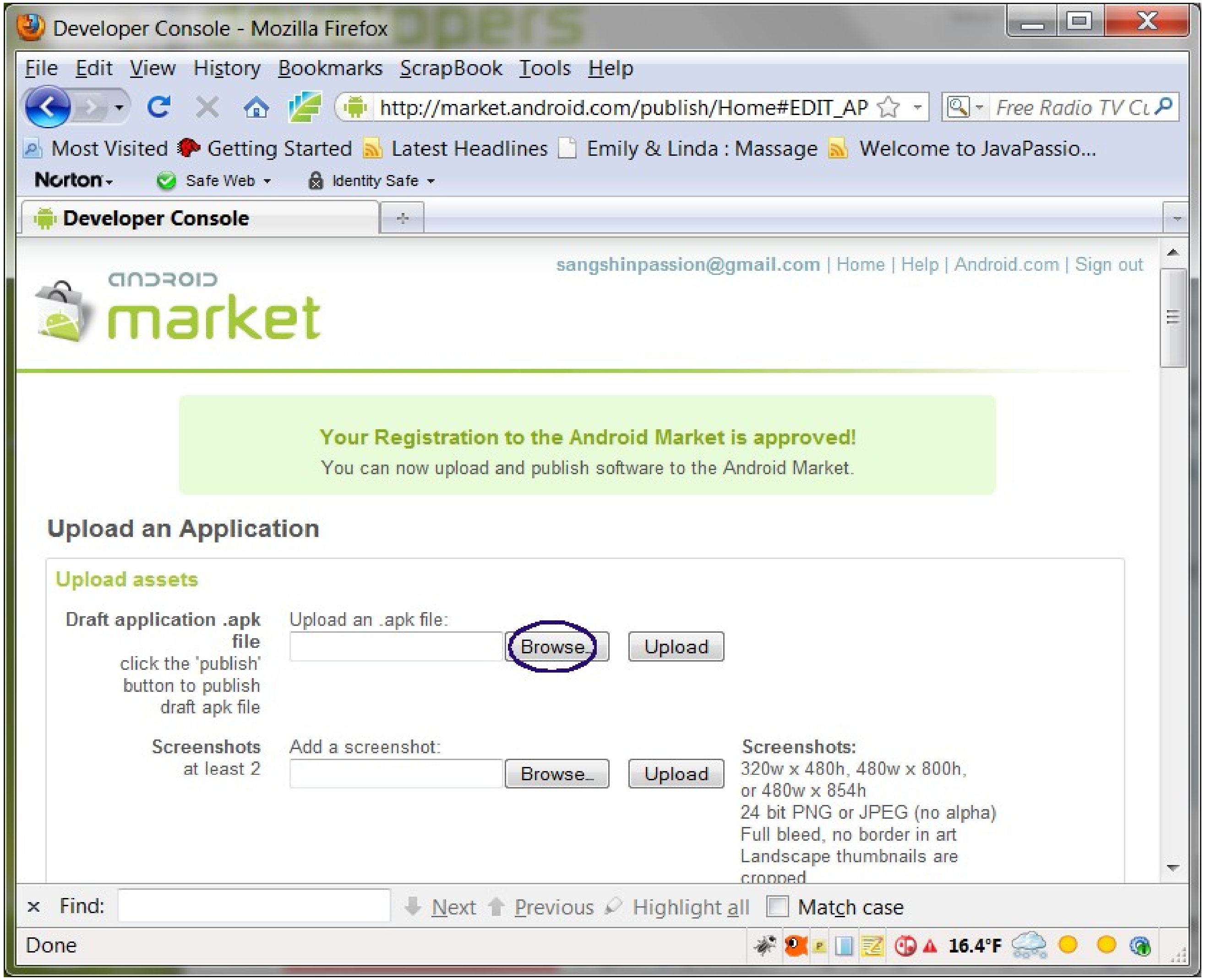
- Signing the application (which is APK file) using private key
  - Use “jarsigner” command line tool (from JDK) or use Eclipse ADT export wizard
- Aligning the APK file
  - Ensures that all uncompressed data starts with a particular byte alignment, relative to the start of the file. Ensuring alignment at 4-byte boundaries provides a performance optimization when installed on a device.
  - “zipalign” command line tool (from Android SDK) or use Eclipse ADT export wizard

# Eclipse ADT Export Wizard

- Performs “signing” and “aligning” in graphic way
  - Could also create keystore (if needed)

# Publishing Your Application

# Publishing Your Application



Developer Console - Mozilla Firefox
File Edit View History Bookmarks ScrapBook Tools Help
http://market.android.com/publish/Home#EDIT_AP
Free Radio TV Cu
Most Visited Getting Started Latest Headlines Emily & Linda : Massage Welcome to JavaPassio...
Norton Safe Web Identity Safe
Developer Console
sangshinpassion@gmail.com | Home | Help | Android.com | Sign out
android market
Find: Next Previous Highlight all Match case
Done
16.4°F

**Your Registration to the Android Market is approved!**

You can now upload and publish software to the Android Market.

Upload an Application

Upload assets

Draft application .apk file
click the 'publish' button to publish draft apk file

Upload an .apk file: Browse... Upload

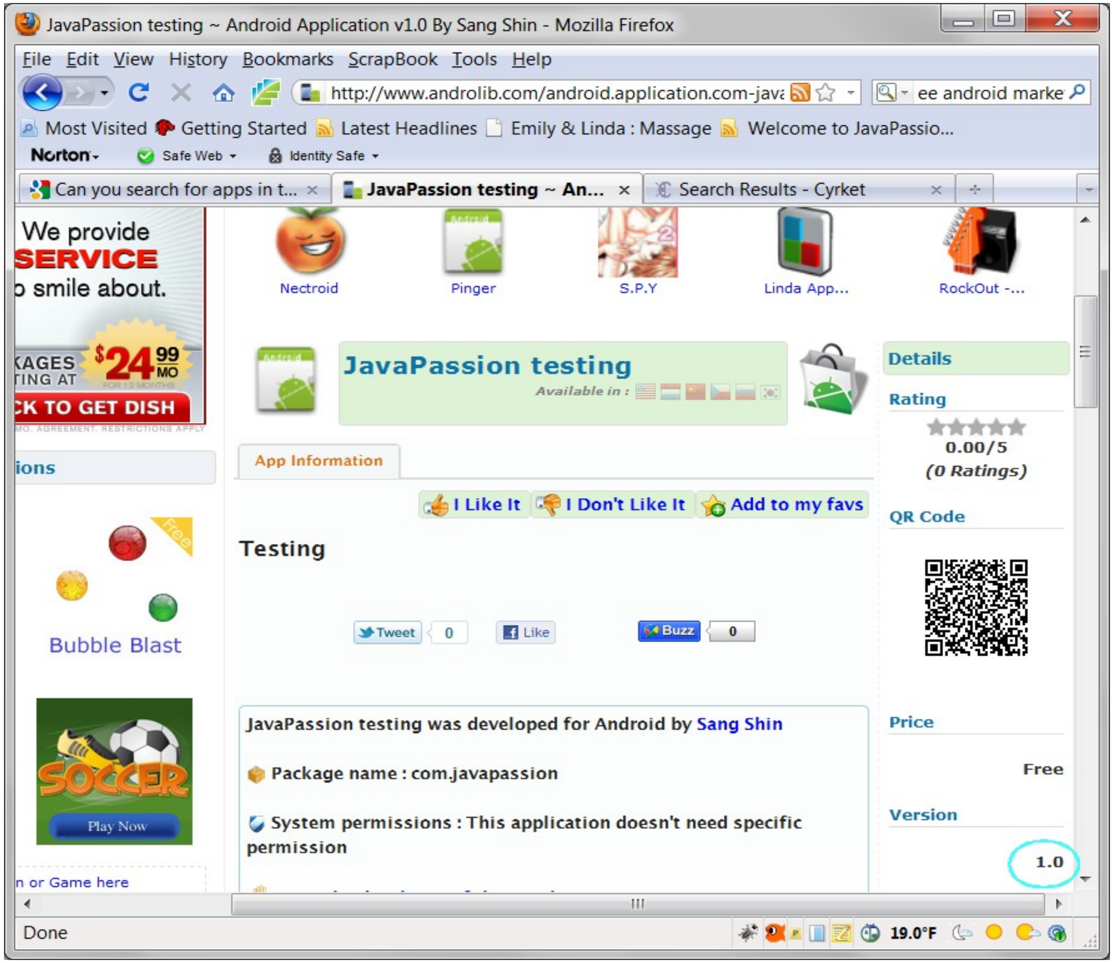
Screenshots
at least 2

Add a screenshot: Browse... Upload

Screenshots:
320w x 480h, 480w x 800h, or 480w x 854h
24 bit PNG or JPEG (no alpha)
Full bleed, no border in art
Landscape thumbnails are cropped

# Checking Your Published Application

- http://www.androlib.com/



JavaPassion testing ~ Android Application v1.0 By Sang Shin - Mozilla Firefox

http://www.androlib.com/android.application.com-jav

JavaPassion testing

Available in :

App Information

Testing

JavaPassion testing was developed for Android by Sang Shin

Package name : com.javapassion

System permissions : This application doesn't need specific permission

Details

Rating

0.00/5

(0 Ratings)

QR Code

Price

Free

Version

1.0

Done

# Thank you!

Sang Shin
Michèle Garoche
http://www.javapassion.com
“Learn with Passion!”