



Social Media Apps Programming

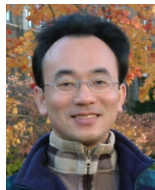
Developing Android Native Apps with Java (Eclipse)(MIT App Inventor)

1031SMAP03

TLMXM1A (8687) (M2143) (Fall 2014)

(MIS MBA) (2 Credits, Elective) [Full English Course]

Thu 8,9 (15:10-17:00) V201



Min-Yuh Day, Ph.D.

Assistant Professor

Department of Information Management

Tamkang University

<http://mail.tku.edu.tw/myday>



Course Schedule (1/3)

Week Date Subject/Topics

- 1 2014/09/17 Course Orientation and Introduction to Social Media and Mobile Apps Programming
- 2 2014/09/24 Introduction to Android / iOS Apps Programming
- 3 2014/10/01 Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)
- 4 2014/10/08 Developing iPhone / iPad Native Apps with Swift / Objective-C (XCode)
- 5 2014/10/15 Mobile Apps using HTML5/CSS3/JavaScript
- 6 2014/10/22 jQuery Mobile

Course Schedule (2/3)

Week Date Subject/Topics

- 7 2014/10/29 Create Hybrid Apps with Phonegap
- 8 2014/11/05 jQuery Mobile/Phonegap
- 9 2014/11/12 jQuery Mobile/Phonegap
- 10 2014/11/19 Midterm Exam Week
(Midterm Project Report)
- 11 2014/11/26 Case Study on Social Media Apps
Programming and Marketing in
Google Play and App Store
- 12 2014/12/03 Google Cloud Platform

Course Schedule (3/3)

Week	Date	Subject/Topics
• 13	2014/12/10	Google App Engine
• 14	2014/12/17	Google Map API
• 15	2014/12/24	Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)
• 16	2014/01/31	Twitter API
• 17	2015/01/07	Final Project Presentation
• 18	2015/01/14	Final Exam Week (Final Project Report)

Outline

- **Developing **Android Native Apps** with **Java****
 - **Eclipse**
 - **Android Developer Tools (**ADT**) Bundle**
 - **Building Your First Android App**
- **MIT App Inventor**

Android /iOS Apps Programming

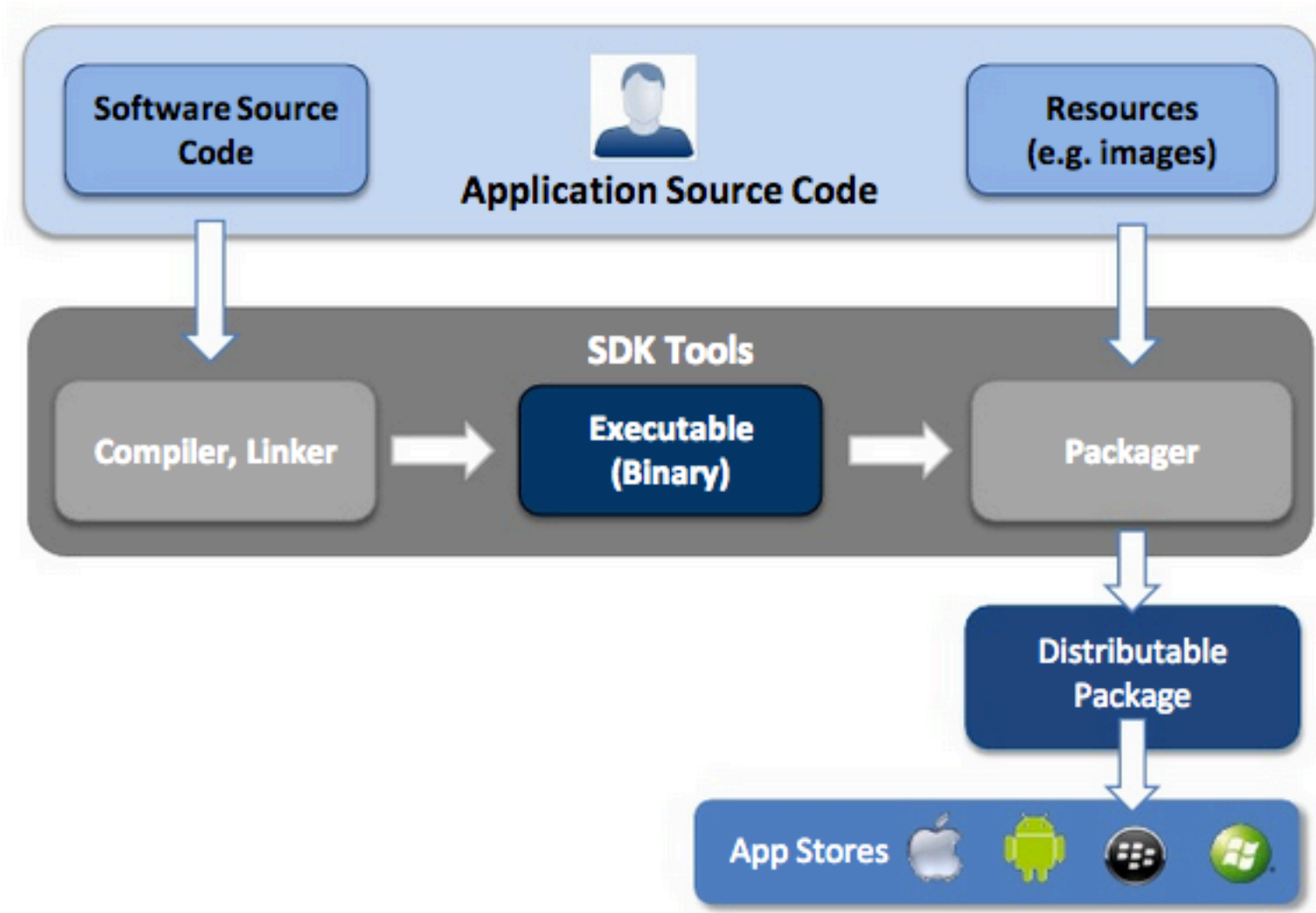
Native Apps

Hybrid Apps

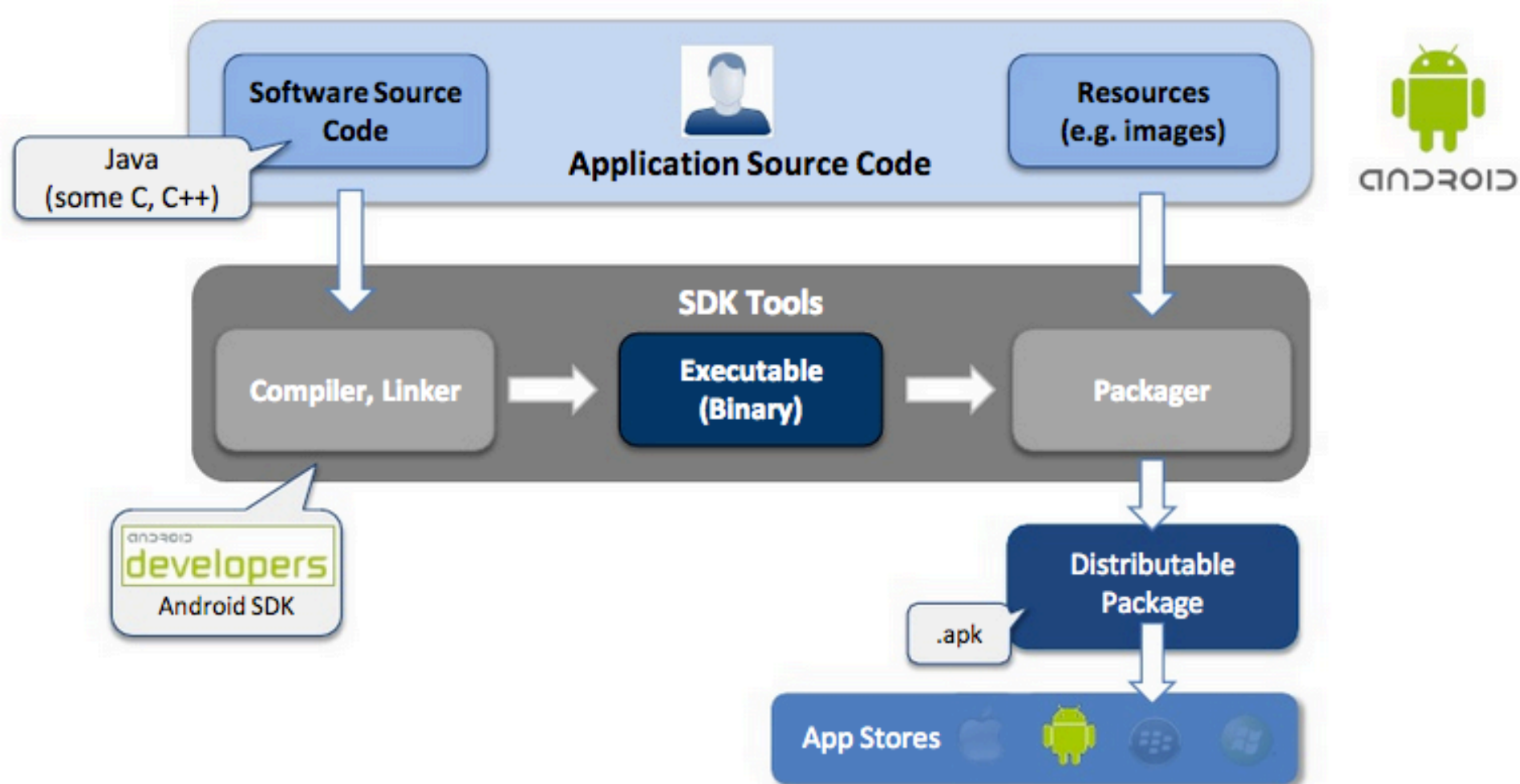
Mobile Web Apps



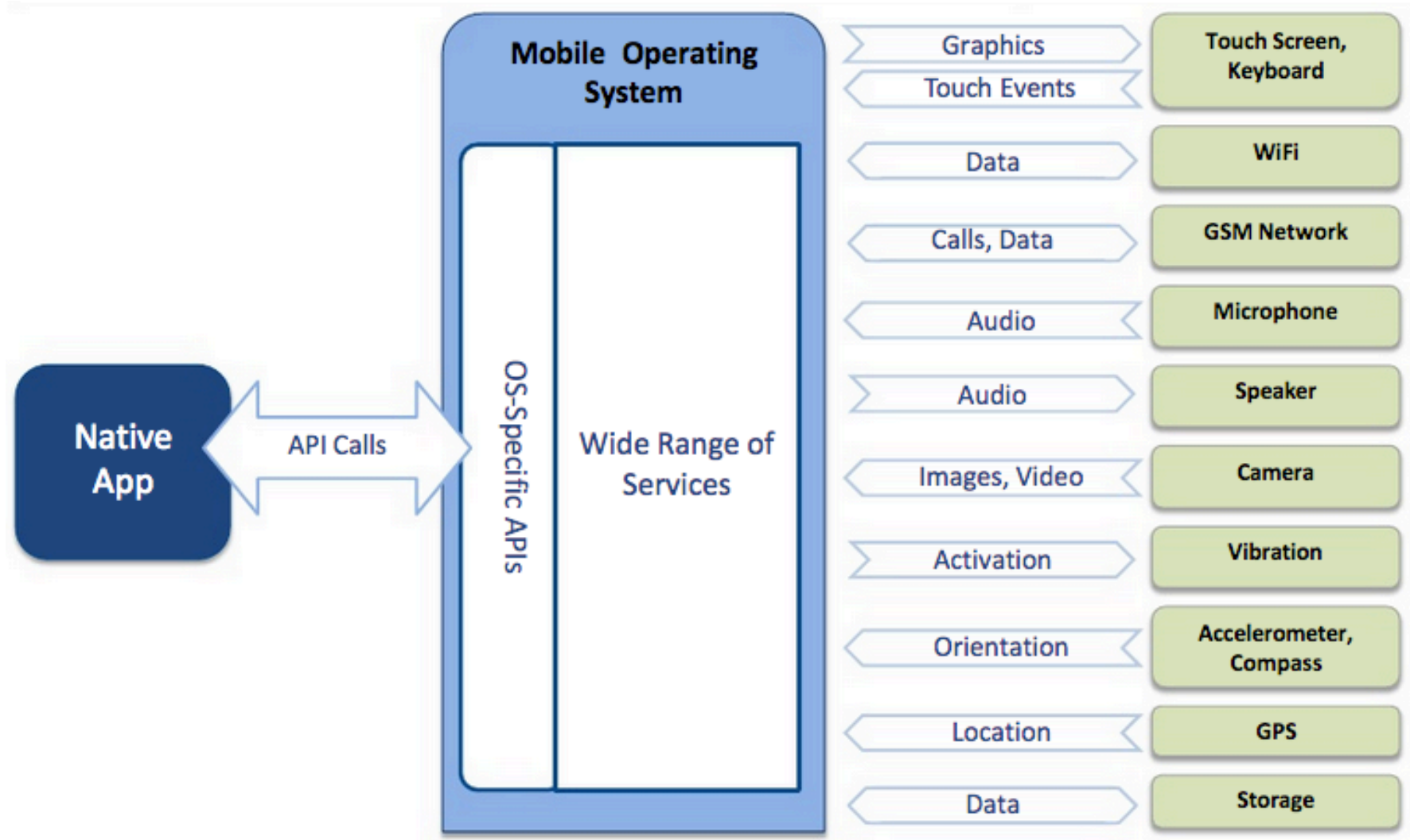
Native App Development



Android - Native App Development



Native App – Interaction with Mobile Device



android

Phones & Tablets

Wear

TV

Auto

One

Play



Coming to watches, phones, tablets,
TVs, and cars near you



[Android Wear >](#)



[Phones and Tablets >](#)



[Android TV >](#)



[Android Auto >](#)

Android 2.2, Froyo

Voice Typing lets you input text, and Voice Actions let you control your phone, just by speaking.



2.2

Android 2.3, Gingerbread

New sensors make Android great for gaming - so you can touch, tap, tilt, and play away.



2.3

Android 3.0, Honeycomb

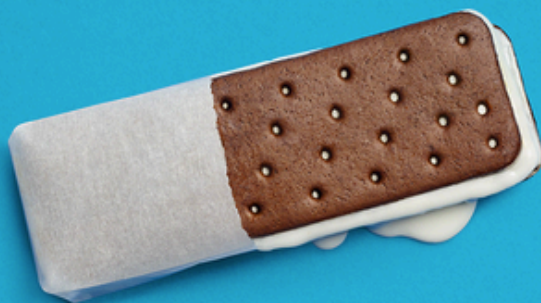
Optimized for tablets, this release opens up new horizons wherever you are.



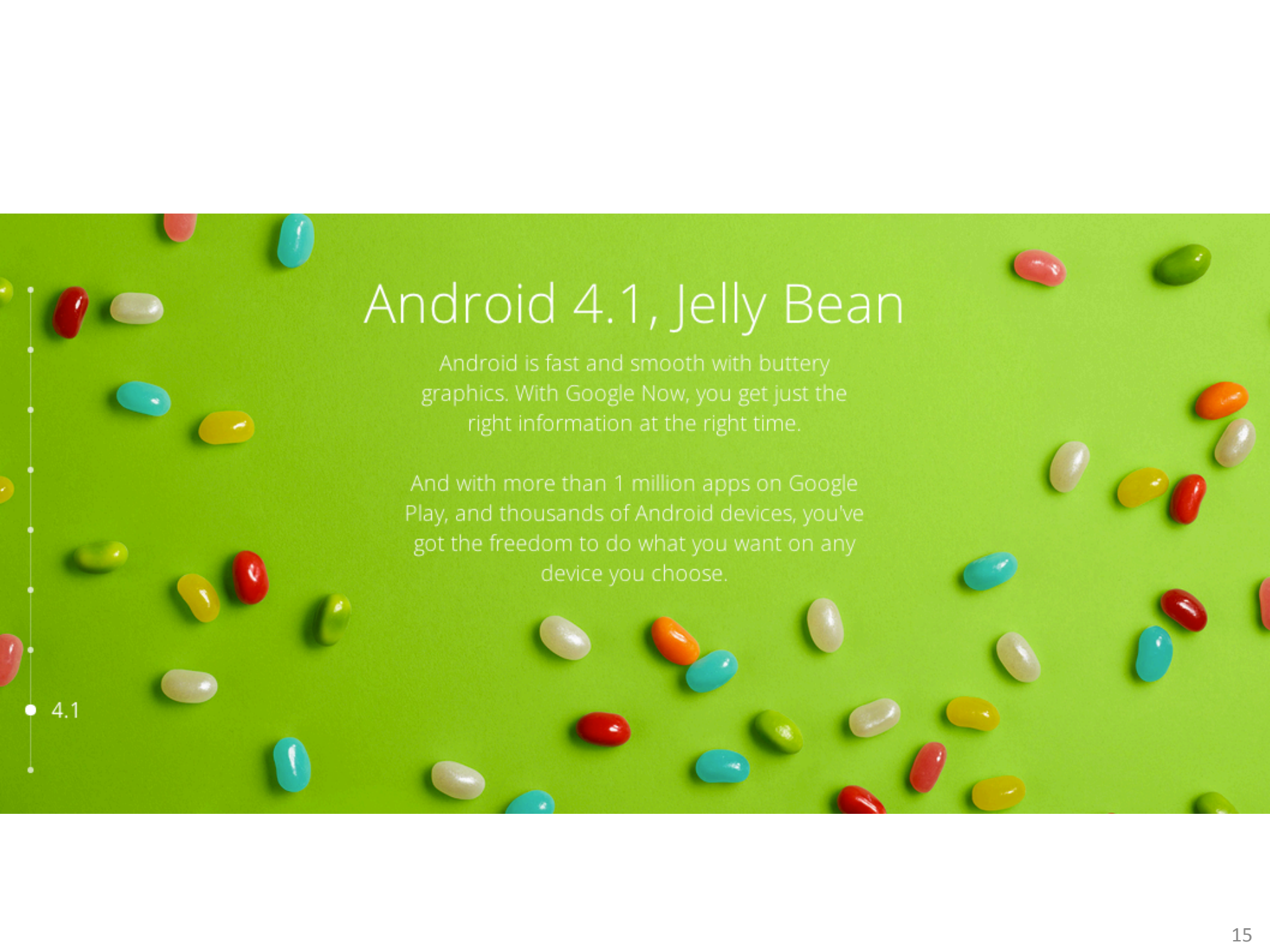
3.0

Android 4.0, Ice Cream Sandwich

Android comes of age with a new, refined design. Simple, beautiful and beyond smart.



4.0



Android 4.1, Jelly Bean

Android is fast and smooth with buttery graphics. With Google Now, you get just the right information at the right time.

And with more than 1 million apps on Google Play, and thousands of Android devices, you've got the freedom to do what you want on any device you choose.

4.1

Android 4.4, KitKat

It's our goal with Android KitKat to make an amazing Android experience available for everybody.



4.4

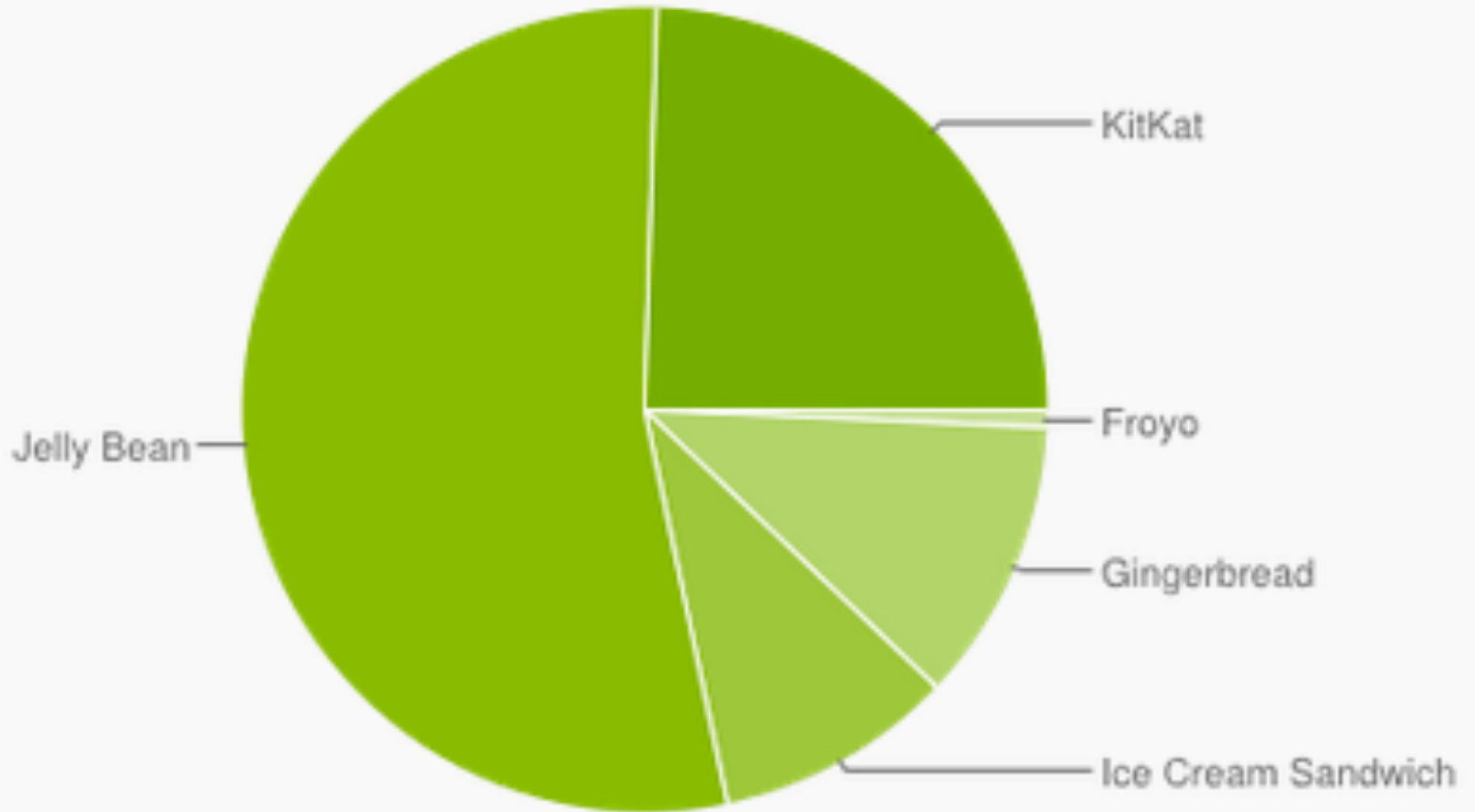


Android Platform

Version	Codename	API	Distribution
2.2	Froyo	8	0.7%
2.3.3 - 2.3.7	Gingerbread	10	11.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	9.6%
4.1.x	Jelly Bean	16	25.1%
4.2.x		17	20.7%
4.3		18	8.0%
4.4	KitKat	19	24.5%

*Data collected during a 7-day period ending on September 9, 2014.
Any versions with less than 0.1% distribution are not shown.*

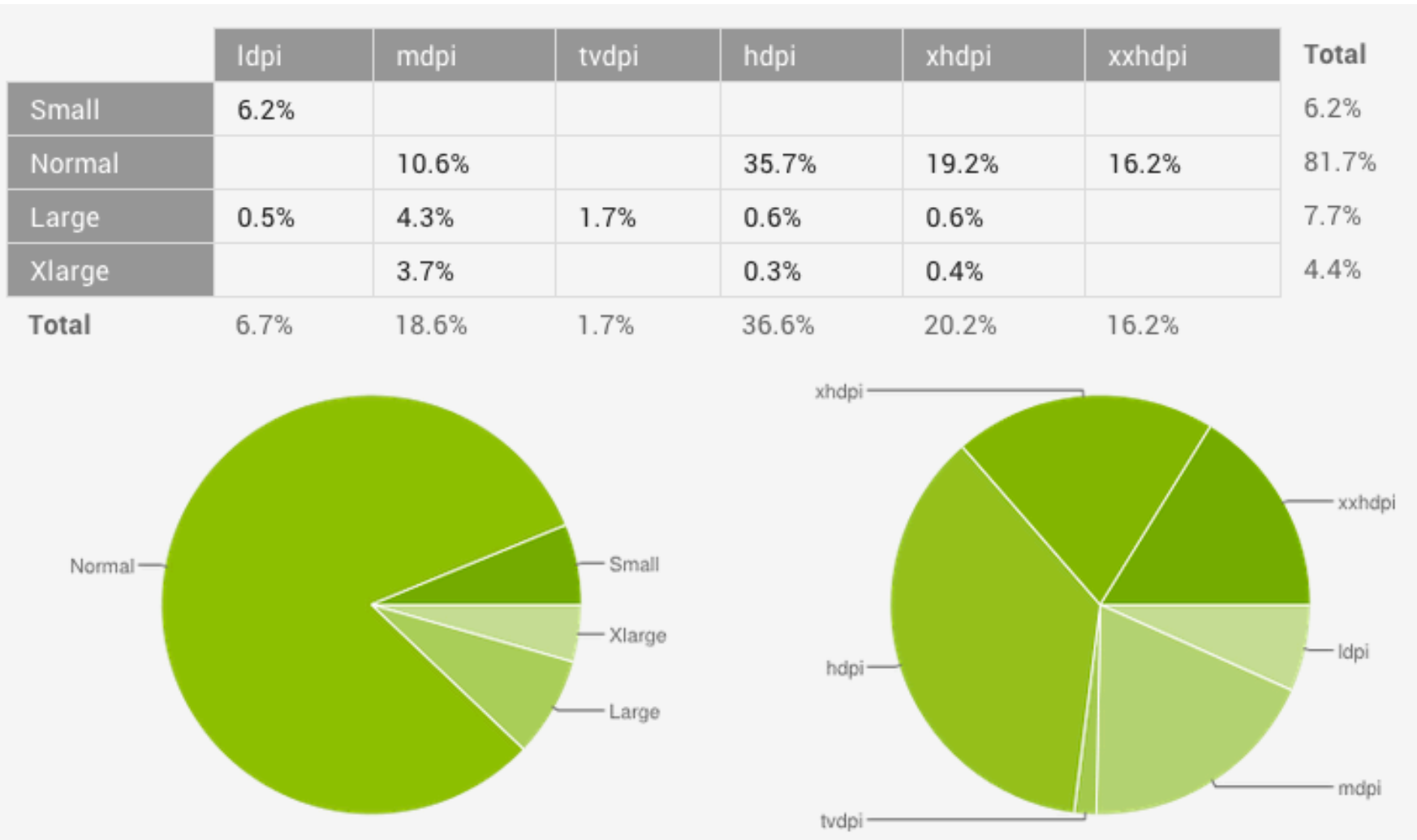
Android Platform



Data collected during a 7-day period ending on September 9, 2014.

<http://developer.android.com/about/dashboards/index.html>

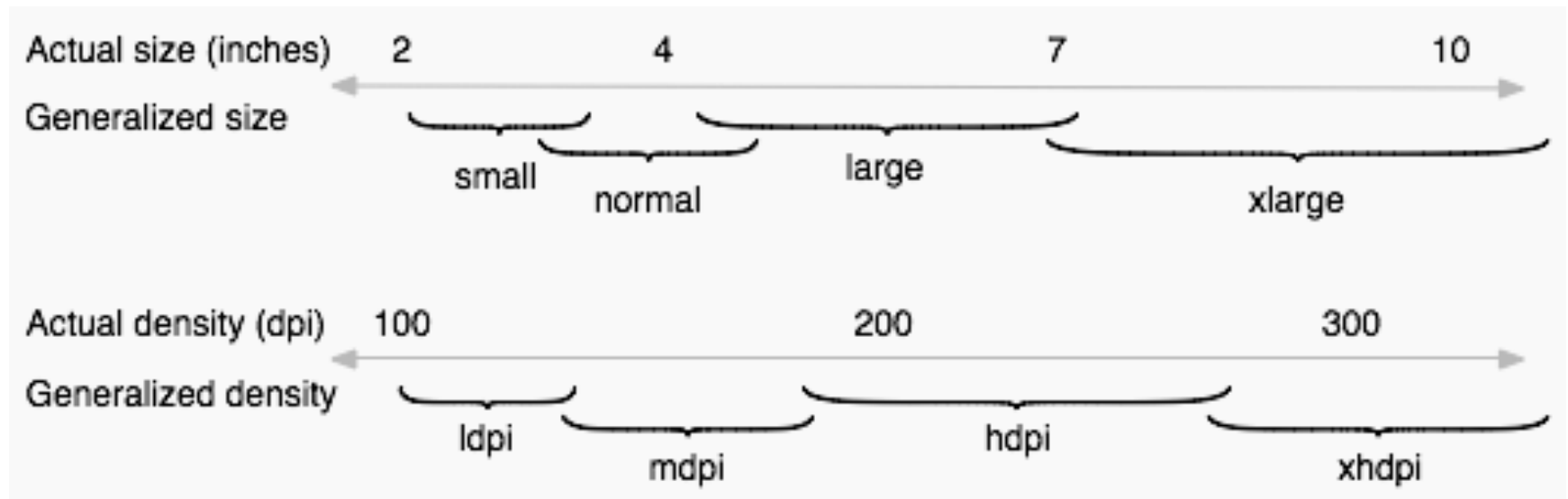
Android Screen Sizes and Densities



Data collected during a 7-day period ending on September 9, 2014.

<http://developer.android.com/about/dashboards/index.html>

Android Screen Sizes and Densities



Android Development Environment

Android Native Apps

1

JDK

(Java Development Kit)

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

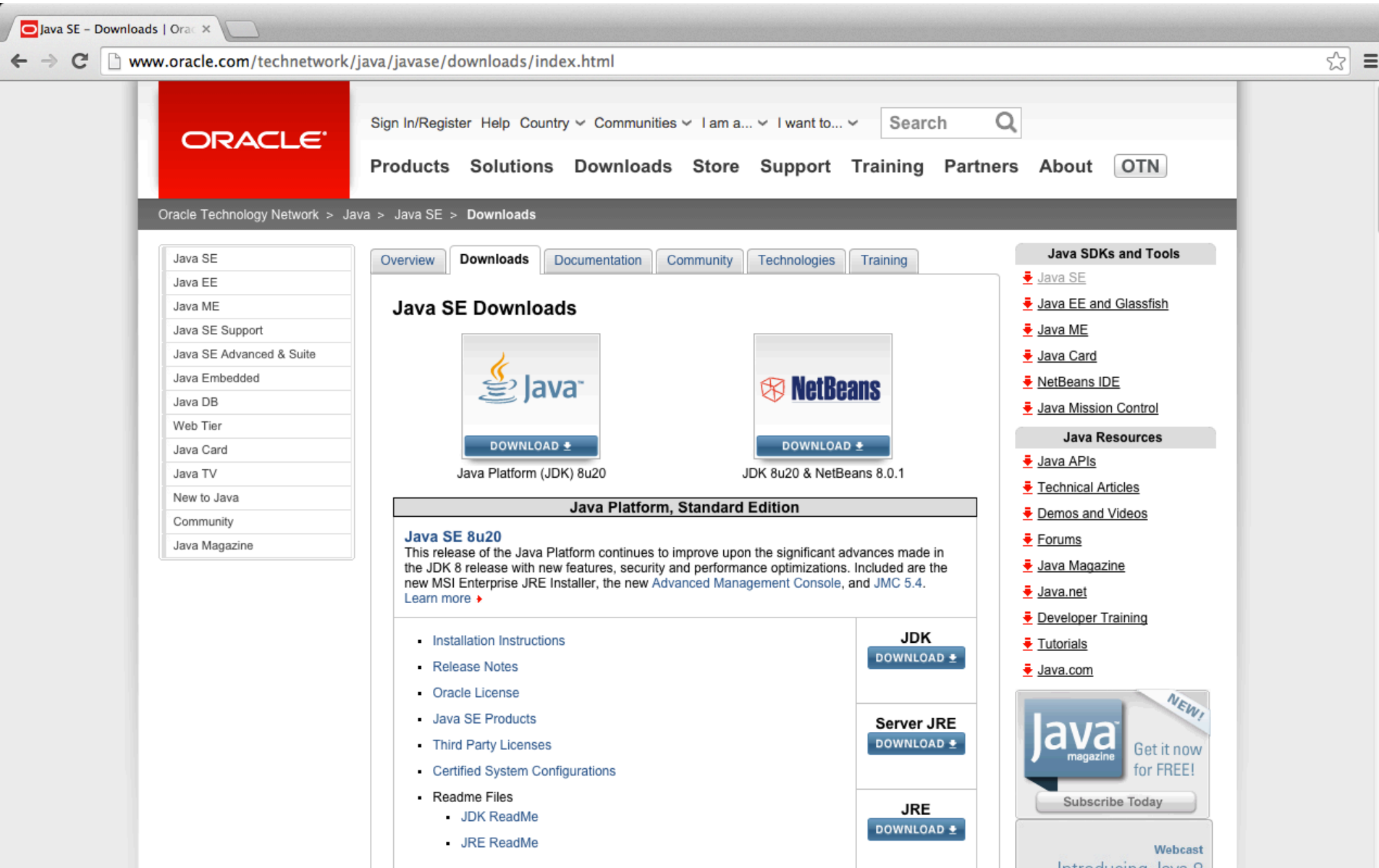
2

ADT Bundle

(Android Developer Tools Bundle)
(Eclipse + ADT plugin + Android SDK
+ Android Platform + emulator)

<http://developer.android.com/sdk/index.html>

1 JDK (Java Development Kit)



The screenshot shows the Oracle Java SE Downloads page. The browser address bar displays www.oracle.com/technetwork/java/javase/downloads/index.html. The Oracle logo is in the top left. The navigation bar includes links for Products, Solutions, Downloads, Store, Support, Training, Partners, and About, along with an OTN button. A search bar is located on the right. The breadcrumb trail reads: Oracle Technology Network > Java > Java SE > Downloads. On the left, a sidebar lists various Java-related links. The main content area has tabs for Overview, Downloads, Documentation, Community, Technologies, and Training. The 'Downloads' tab is active, showing 'Java SE Downloads'. It features two download buttons: 'Java Platform (JDK) 8u20' and 'JDK 8u20 & NetBeans 8.0.1'. Below these, a section titled 'Java Platform, Standard Edition' describes the 'Java SE 8u20' release, highlighting new features, security, and performance optimizations. It includes links to 'Learn more', 'Installation Instructions', 'Release Notes', 'Oracle License', 'Java SE Products', 'Third Party Licenses', 'Certified System Configurations', and 'Readme Files' (including JDK and JRE ReadMe). On the right, a sidebar lists 'Java SDKs and Tools' and 'Java Resources'. At the bottom right, there is a 'Java magazine' subscription banner and a 'Webcast' section.

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

Overview Downloads Documentation Community Technologies Training

Java SE Downloads

 **DOWNLOAD**

Java Platform (JDK) 8u20

 **DOWNLOAD**

JDK 8u20 & NetBeans 8.0.1

Java Platform, Standard Edition

Java SE 8u20

This release of the Java Platform continues to improve upon the significant advances made in the JDK 8 release with new features, security and performance optimizations. Included are the new MSI Enterprise JRE Installer, the new Advanced Management Console, and JMC 5.4. [Learn more](#)

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
 - JDK ReadMe
 - JRE ReadMe

JDK
DOWNLOAD

Server JRE
DOWNLOAD

JRE
DOWNLOAD

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

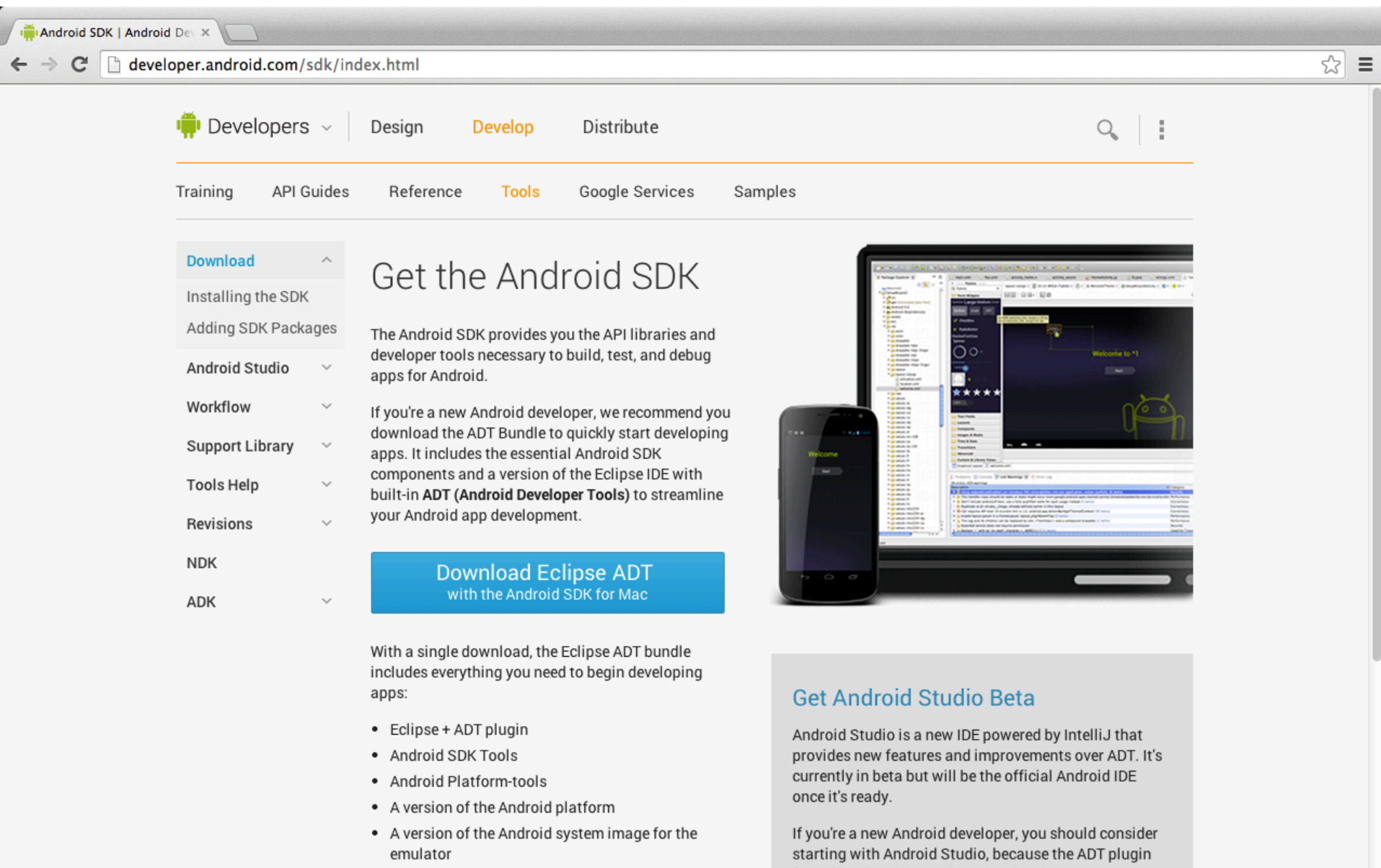
 **NEW!**
Get it now for FREE!
Subscribe Today

Webcast
Introducing Java 8

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2

ADT Bundle



The screenshot shows the 'Get the Android SDK' page on the Android developer website. The page has a navigation bar with 'Developers', 'Design', 'Develop', and 'Distribute'. Below this is a secondary navigation bar with 'Training', 'API Guides', 'Reference', 'Tools', 'Google Services', and 'Samples'. On the left, a sidebar menu lists various resources like 'Download', 'Installing the SDK', 'Adding SDK Packages', 'Android Studio', 'Workflow', 'Support Library', 'Tools Help', 'Revisions', 'NDK', and 'ADK'. The main content area features the heading 'Get the Android SDK' and a paragraph explaining that the SDK provides API libraries and developer tools. A blue button labeled 'Download Eclipse ADT with the Android SDK for Mac' is prominent. Below this, a list of included components is shown: Eclipse + ADT plugin, Android SDK Tools, Android Platform-tools, a version of the Android platform, and a version of the Android system image for the emulator. To the right, there is an image of a smartphone and a laptop displaying the Eclipse IDE with an Android project. Further down, a section titled 'Get Android Studio Beta' introduces the new IDE powered by IntelliJ, noting it is currently in beta and will be the official Android IDE.

Android SDK | Android Dev x

developer.android.com/sdk/index.html

Developers ▾ Design Develop Distribute

Training API Guides Reference Tools Google Services Samples

Download ▴
Installing the SDK
Adding SDK Packages
Android Studio ▾
Workflow ▾
Support Library ▾
Tools Help ▾
Revisions ▾
NDK
ADK ▾

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

Download Eclipse ADT
with the Android SDK for Mac

With a single download, the Eclipse ADT bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- A version of the Android platform
- A version of the Android system image for the emulator

Get Android Studio Beta

Android Studio is a new IDE powered by IntelliJ that provides new features and improvements over ADT. It's currently in beta but will be the official Android IDE once it's ready.

If you're a new Android developer, you should consider starting with Android Studio, because the ADT plugin for Eclipse is no longer in active development.

Option 1. Eclipse ADT Bundle

Option 2. Android Studio



Eclipse ADT Bundle



Android Studio

Building Your First Android App

1. Download the Android SDK.
 - <http://developer.android.com/sdk/index.html>
2. Install the ADT plugin for Eclipse
3. Download the latest SDK tools and platforms using the SDK Manager.

ADT Bundle

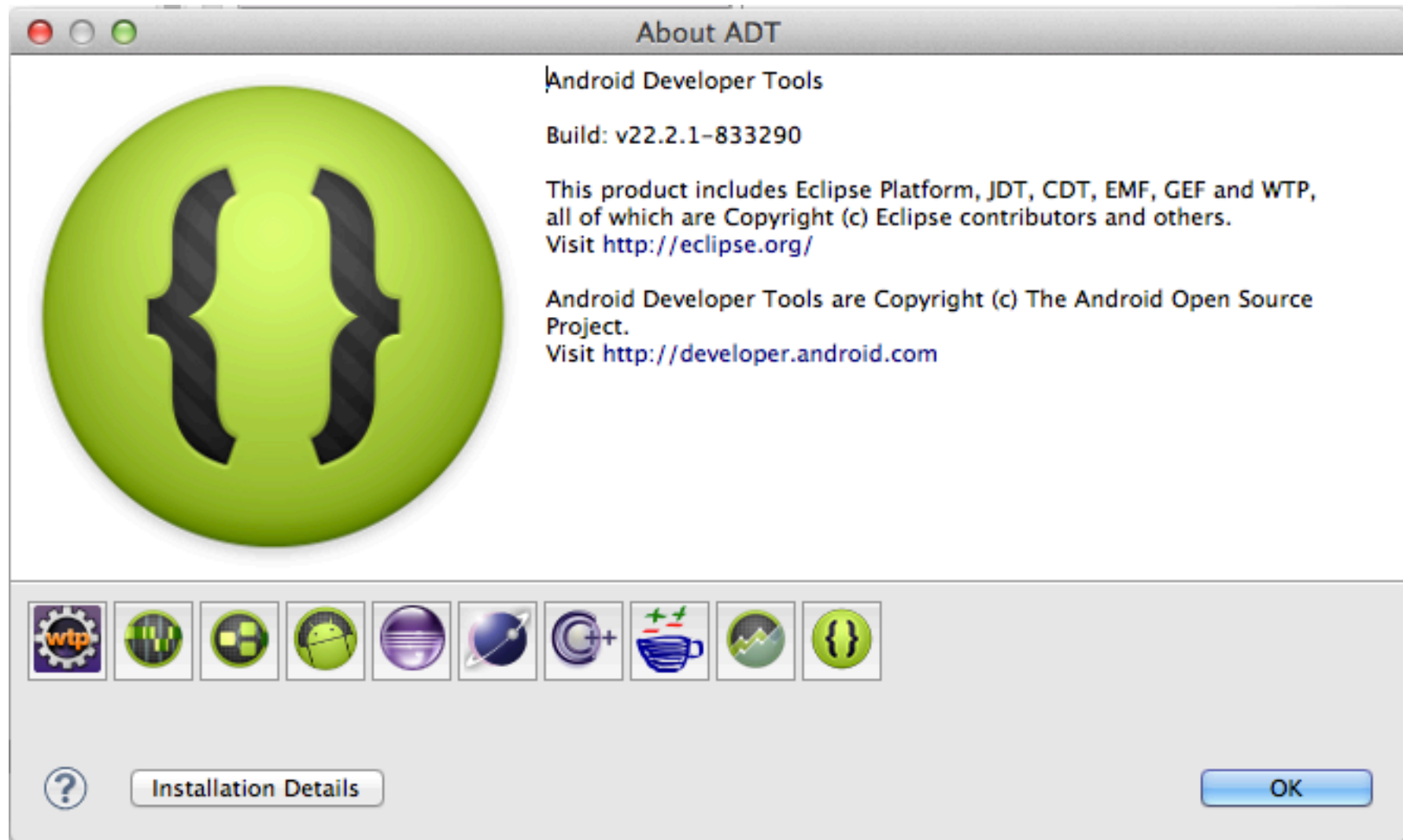
(Android Developer Tools Bundle)
(Eclipse + ADT plugin + Android SDK
+ Android Platform + emulator)

ADT Bundle

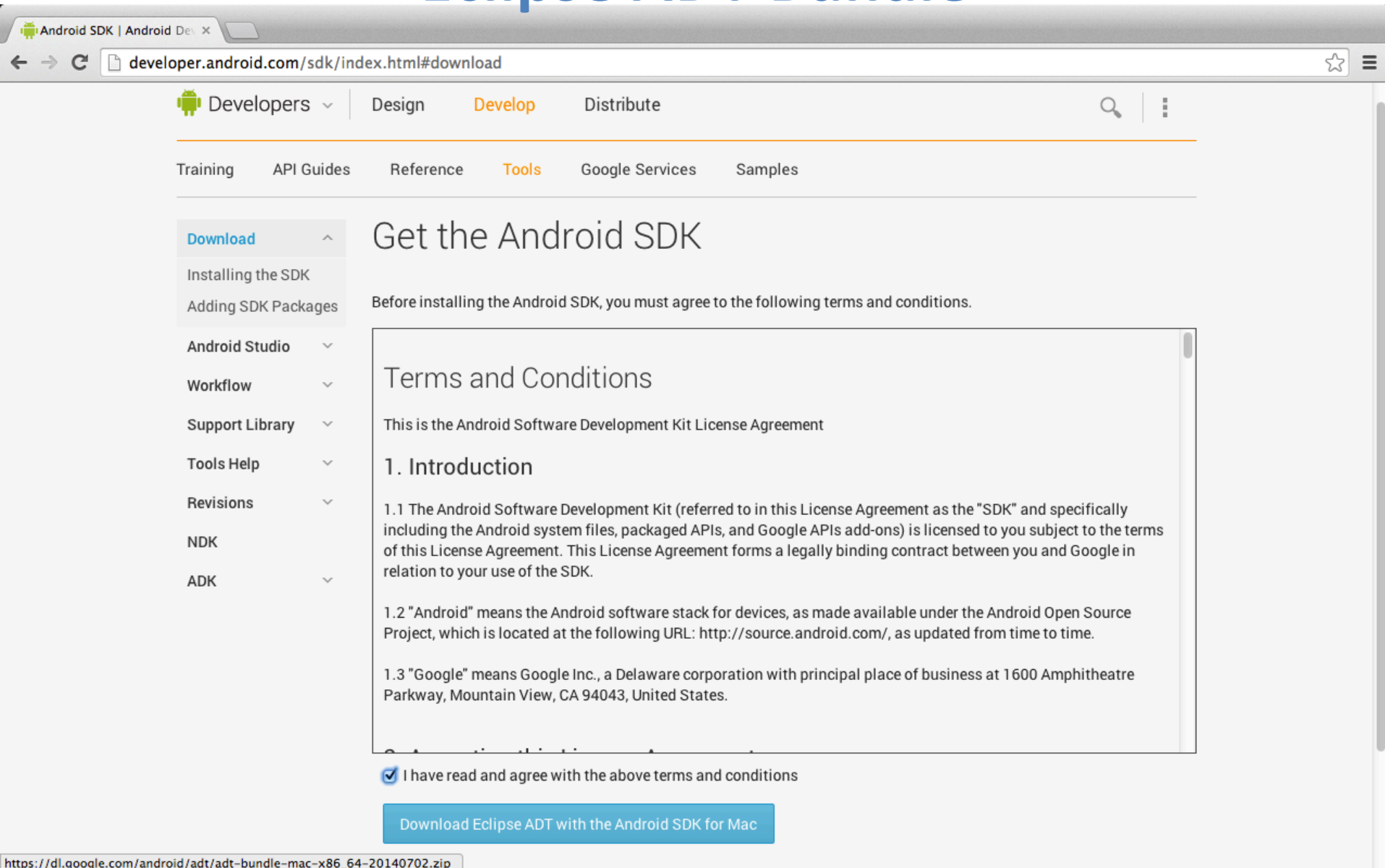
ADT (Android Developer Tools)

- With a single download, the ADT Bundle includes everything you need to begin developing apps:
 1. Eclipse + ADT plugin
 2. Android SDK Tools
 3. Android Platform-tools
 4. The latest Android platform
 5. The latest Android system image for the emulator

ADT



Get the Android SDK Eclipse ADT Bundle



The screenshot shows a web browser window with the URL `developer.android.com/sdk/index.html#download`. The page features a navigation bar with links for Developers, Design, Develop, and Distribute. Below this, there are links for Training, API Guides, Reference, Tools, Google Services, and Samples. A left sidebar contains a 'Download' menu with options like 'Installing the SDK', 'Adding SDK Packages', 'Android Studio', 'Workflow', 'Support Library', 'Tools Help', 'Revisions', 'NDK', and 'ADK'. The main content area is titled 'Get the Android SDK' and includes a section for 'Terms and Conditions' of the Android Software Development Kit License Agreement. This section contains an introduction and the first three numbered clauses (1.1, 1.2, 1.3) defining the SDK, Android, and Google. At the bottom, there is a checkbox for agreeing to the terms and a button to download the Eclipse ADT bundle for Mac.

Android SDK | Android Dev x

developer.android.com/sdk/index.html#download

Developers ▾ Design Develop Distribute

Training API Guides Reference Tools Google Services Samples

Get the Android SDK

Before installing the Android SDK, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.

1.3 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

☒ I have read and agree with the above terms and conditions

Download Eclipse ADT with the Android SDK for Mac

https://dl.google.com/android/adt/adt-bundle-mac-x86_64-20140702.zip

Installing the Eclipse ADT Bundle

1. Unpack the ZIP file (named **adt-bundle-
<os_platform>.zip**) and save it to an appropriate location, such as a "**Development**" directory in your home directory.
2. Open the **adt-bundle-<os_platform>/eclipse/** directory and launch Eclipse.

Get Android Studio Beta

- **Android Studio** is a new IDE powered by IntelliJ that provides new features and improvements over ADT.
It's currently in beta but will be the official Android IDE once it's ready.
- If you're a new Android developer, you should consider starting with Android Studio, because the **ADT plugin for Eclipse** is no longer in active development.

Installing the Android SDK

developer.android.com/sdk/installing/index.html?pkg=adt

☆

 Developers

Design

Develop

Distribute

Training

API Guides

Reference

Tools

Google Services

Samples

Download

Installing the SDK

Adding SDK Packages

Android Studio

Workflow

Support Library

Tools Help

Revisions

NDK

ADK

Installing the Eclipse ADT Bundle

The Eclipse ADT Bundle provides everything you need to start developing apps, including the Android SDK tools and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

If you didn't download the Eclipse ADT bundle, go [download the Eclipse ADT bundle now](#), or switch to the [Android Studio install](#) or [stand-alone SDK Tools install](#) instructions.

To set up the ADT Bundle:

1. Unpack the ZIP file (named `adt-bundle-<os_platform>.zip`) and save it to an appropriate location, such as a "Development" directory in your home directory.
2. Open the `adt-bundle-<os_platform>/eclipse/` directory and launch **Eclipse**.

Caution: Do not move any of the files or directories from the `adt-bundle-<os_platform>` directory. If you move the `eclipse/` or `sdk/` directory, ADT will not be able to locate the SDK and you'll need to manually update the ADT preferences.

Eclipse with ADT is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.

[CONTINUE: ADDING SDK PACKAGES](#)

g+1

3.4k

Adding SDK Packages | An...

[←](#) [→](#) [↻](#)
developer.android.com/sdk/installing/adding-packages.html

☆ ☰

Developers ▾

Design
Develop
Distribute

Download ▴

Installing the SDK
Adding SDK Packages

Android Studio ▾
Workflow ▾
Support Library ▾
Tools Help ▾
Revisions ▾
NDK
ADK ▾

Adding SDK Packages

By default, the Android SDK does not include everything you need to start developing. The SDK separates tools, platforms, and other components into packages you can download as needed using the [Android SDK Manager](#). So before you can start, there are a few packages you should add to your Android SDK.

To start adding packages, launch the Android SDK Manager in one of the following ways:

- In Eclipse or Android Studio, click **SDK Manager** in the toolbar.
- If you're not using Eclipse or Android Studio:
 - Windows: Double-click the **SDK Manager.exe** file at the root of the Android SDK directory.
 - Mac/Linux: Open a terminal and navigate to the **tools/** directory in the Android SDK, then execute **android sdk**.

When you open the SDK Manager for the first time, several packages will be selected by default. Leave these selected, but be sure you have everything you need to get started by following these steps:

1. Get the latest SDK tools

As a minimum when setting up the Android SDK, you should download the latest tools and Android platform:

- Open the Tools directory and select:
 - Android SDK Tools**
 - Android SDK Platform-tools**
 - Android SDK Build-tools (highest version)**
- Open the first Android X.X folder (the latest version) and select:

Source: <http://developer.android.com/sdk/installing/adding-packages.html>

32

1. Get the latest SDK tools

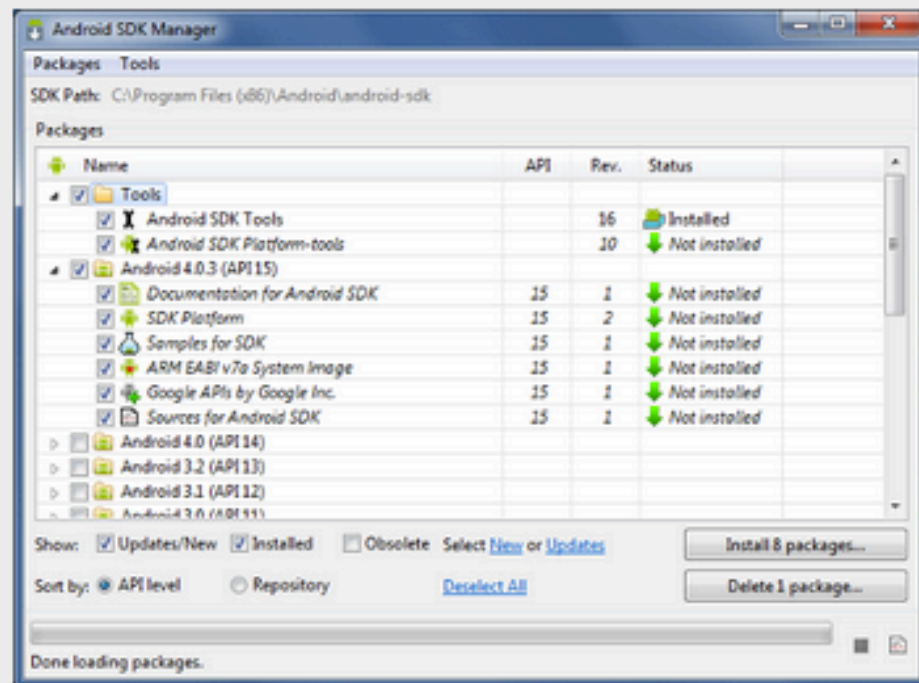
As a minimum when setting up the Android SDK, you should download the latest tools and Android platform:

1. Open the Tools directory and select:

- **Android SDK Tools**
- **Android SDK Platform-tools**
- **Android SDK Build-tools** (highest version)

2. Open the first Android X.X folder (the latest version) and select:

- **SDK Platform**
- A system image for the emulator, such as **ARM EABI v7a System Image**



2. Get the support library for additional APIs

The [Android Support Library](#) provides an extended set of APIs that are compatible with most versions of Android.

Open the **Extras** directory and select:

- **Android Support Repository**
- **Android Support Library**

The support library is required for:

- [Android Wear](#)
- [Android TV](#)
- [Google Cast](#)

It also provides these popular APIs:

- [Navigation drawer](#)
- [Swipe views](#)
- [Backward-compatible action bar](#)

3. Get Google Play services for even more APIs

To develop with Google APIs, you need the Google Play services package:

Open the **Extras** directory and select:

- **Google Repository**
- **Google Play services**

Note: Google Play services APIs are not available on all Android-powered devices, but are available on all devices with Google Play Store. To use these APIs in the Android emulator, you must also install the **Google APIs** system image from the latest Android X.X directory in the SDK Manager.

The Google Play services APIs provide a variety of features and services for your Android apps, such as:

- **User authentication**
- **Google Maps**
- **Google Cast**
- **Games achievements and leaderboards**
- **And much more**

4. Install the packages

Once you've selected all the desired packages, continue to install:

1. Click **Install X packages**.
2. In the next window, double-click each package name on the left to accept the license agreement for each.
3. Click **Install**.

The download progress is shown at the bottom of the SDK Manager window. **Do not exit the SDK Manager** or it will cancel the download.

5. Build something!

With the above packages now in your Android SDK, you're ready to build apps for Android. As new tools and other APIs become available, simply launch the SDK Manager to download the new packages for your SDK.

Here are a few options for how you should proceed:

Get started

If you're new to Android development, learn the basics of Android apps by following the guide to [Building Your First App](#).

Build for wearables

If you're ready to start building apps for Android wearables, see the guide to [Building Apps for Android Wear](#).

Use Google APIs

To start using Google APIs, such as Maps or Play Game services, see the guide to [Setting Up Google Play Services](#).

Building Your First App | A x

developer.android.com/training/basics/firstapp/index.html

Developers ▾ Design Develop Distribute

Training API Guides Reference Tools Google Services Samples

Building Your First App

Welcome to Android application development!

This class teaches you how to build your first Android app. You'll learn how to create an Android project and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.

Before you start this class, be sure you have your development environment set up. You need to:

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse (if you'll use the Eclipse IDE).
3. Download the latest SDK tools and platforms using the SDK Manager.

Note: Make sure you install the most recent versions of the ADT plugin and the Android SDK before you start this class. The procedures described in this class may not apply to earlier versions.

If you haven't already done these tasks, start by downloading the [Android SDK](#) and following the install steps. Once you've finished the setup, you're ready to begin this class.

This class uses a tutorial format that incrementally builds a small Android app that teaches you some fundamental concepts about Android development, so it's important that you follow each step.

[Start the first lesson](#)

GETTING STARTED

- Building Your First App
 - Creating an Android Project
 - Running Your Application
 - Building a Simple User Interface
 - Starting Another Activity
- Adding the Action Bar ▾
- Supporting Different Devices ▾
- Managing the Activity Lifecycle ▾
- Building a Dynamic UI with Fragments ▾
- Saving Data ▾
- Interacting with Other Apps ▾
- Building Apps with Content Sharing ▾
- Building Apps with Multimedia ▾
- Building Apps with Graphics & Animation ▾

DEPENDENCIES AND PREREQUISITES

- [Android SDK](#)
- [ADT Plugin](#) 22.6.2 or higher (if you're using Eclipse)

Android App Building Blocks

1

Activity

2

Service

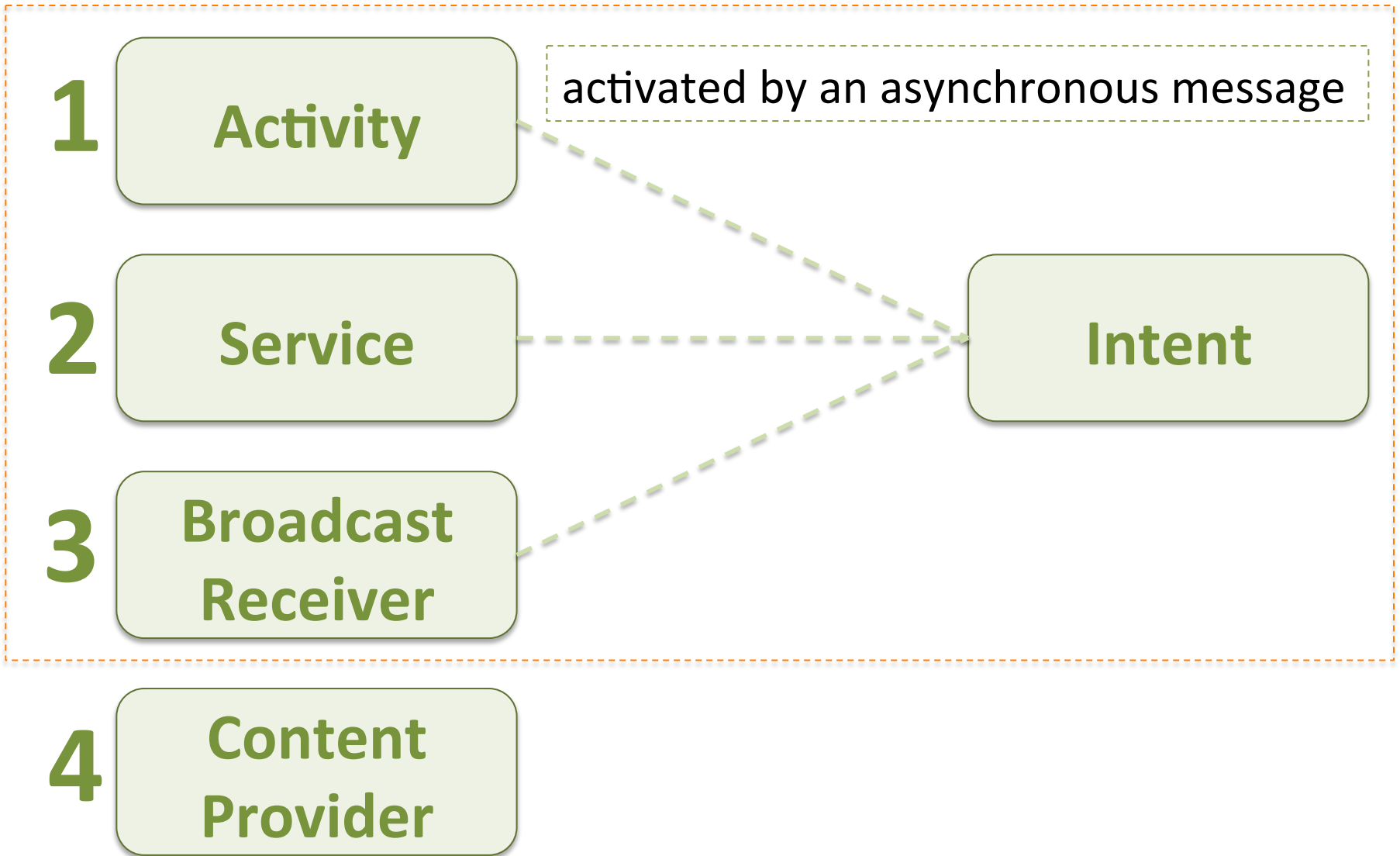
3

**Broadcast
Receiver**

4

**Content
Provider**

Android App Building Blocks



Android App Building Blocks

- 1 **Activity** a **single screen** with a user interface
- 2 **Service**
- 3 **Broadcast Receiver**
- 4 **Content Provider**

Android App Building Blocks

1

Activity

2

Service

runs in the **background** to perform long-running operations or to perform work for remote processes

3

Broadcast
Receiver

4

Content
Provider

Android App Building Blocks

- 1 Activity
- 2 Service
- 3 Broadcast Receiver
responds to system-wide broadcast
announcements
- 4 Content Provider

Android App Building Blocks

1

Activity

2

Service

3

**Broadcast
Receiver**

4

**Content
Provider**

manages a shared set of application data

Developing Android Apps

1. Screen Layout Design: Views and Layouts

- Graphical Layout
- `activity_main.xml`

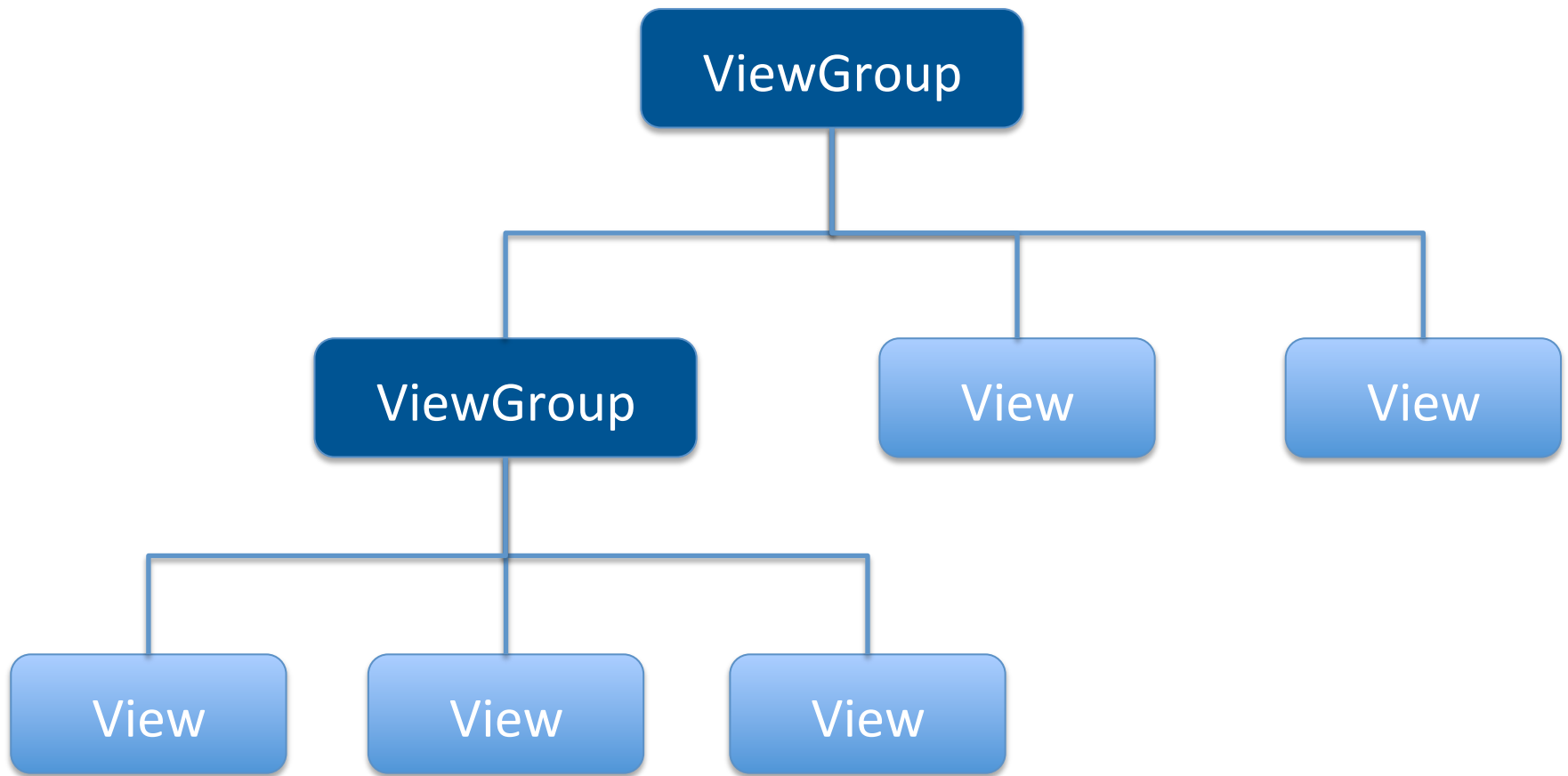
2. App Components (Activity) Programming

- `MainActivity.java`

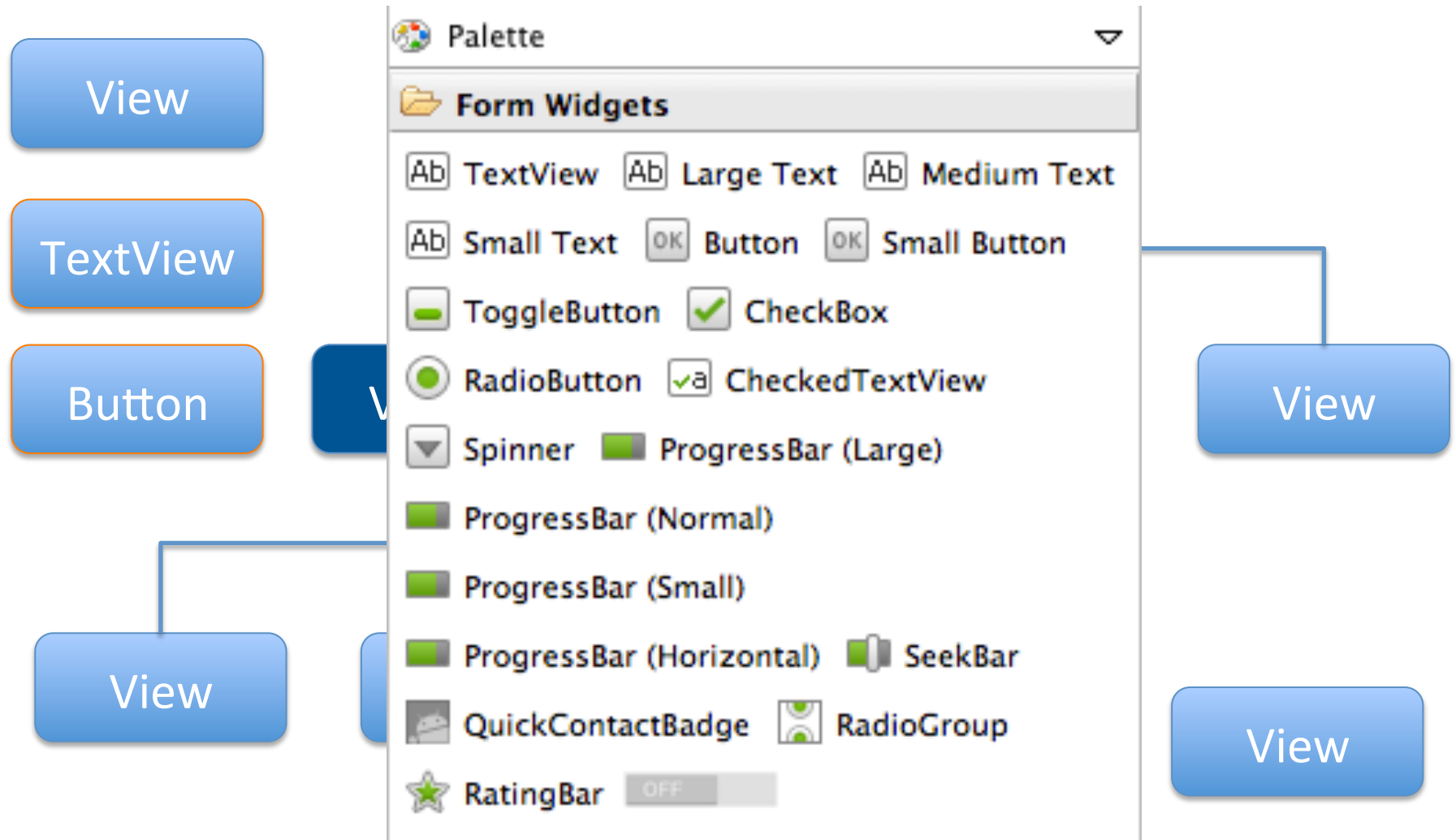
Building a Simple User Interface

- Create a Linear Layout
- Add a Text Field
- Add String Resources
- Add a Button
- Make the Input Box Fill in the Screen Width

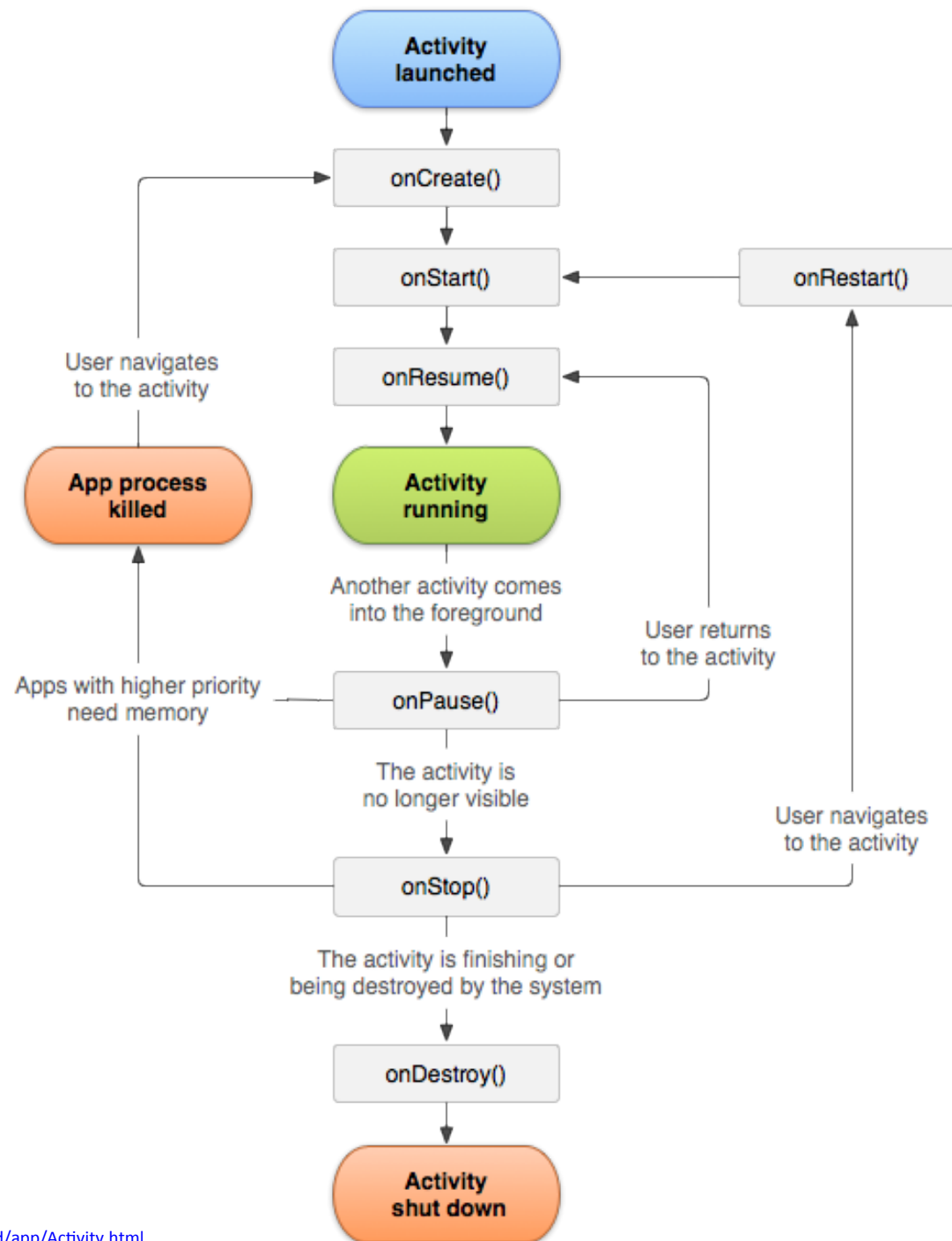
Building a Simple User Interface



Building a Simple User Interface



Android App Activity Lifecycle



Source:

<http://developer.android.com/reference/android/app/Activity.html>

Demo:
Building Your First Android App
with
Android Developer Tools
(ADT)

emulator64-arm Window

Java - MyFirstApp/src/tw/edu/tku/im/smap2013/imyday/myfirstapp/MainActivity.java - ADT - /Users/imyday/Development/AndroidWorkspace

Package Explorer

- tw.edu.tku.im.smap2013.imyday.myfirsta
 - DisplayMessageActivity.java
 - MainActivity.java
- gen [Generated Java Files]
- Android 4.3
- Android Private Libraries
- assets
- bin
 - dexedLibs
 - res
 - AndroidManifest.xml
 - classes.dex
 - jarlist.cache
 - MyFirstApp.apk
 - resources.ap_
- libs
- res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - activity_display_message.xml
 - activity_main.xml
 - menu
 - values
 - dimens.xml
 - strings.xml
 - styles.xml
 - values-en
 - strings.xml
 - values-sw600dp
 - values-sw720dp-land
 - values-v11
 - values-v14
 - values-zh-rTW

MainActivity.java

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Intent;
import android.widget.EditText;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.my";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the menu.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /** Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
```

strings.xml

activity_display_message.xml

DisplayMessageActivity.java

Outline

- tw.edu.tku.im.smap2013.imyday.myfirsta
 - MainActivity
 - EXTRA_MESSAGE : String
 - onCreate(Bundle) : void
 - onCreateOptionsMenu(Menu) : boolean
 - sendMessage(View) : void

5554:AVD43

My Message

Hello World
Myday

Basic Controls

Hardware Buttons

DPAD not enabled in AVD

Hardware Keyboard
Use your physical keyboard to provide input

Problems @ Javadoc Declaration Console

Android

Writable Smart Insert 11 : 89 135M of 249M Android SDK Content Loader

Building Your First App | A x

← → ↻ developer.android.com/training/basics/firstapp/index.html ☆ ☰

 Developers ▾

Design

Develop

Distribute

🔍 | ⋮

Training API Guides Reference Tools Google Services

Getting Started ^

Building Your First App ^

Creating an Android Project

Running Your Application

Building a Simple User Interface

Starting Another Activity

Adding the Action Bar ▾

Supporting Different Devices ▾

Managing the Activity Lifecycle ▾

Building a Dynamic UI with Fragments ▾

Saving Data ▾

Interacting with Other Apps ▾

Building Apps with Content Sharing ▾

Building Apps with Multimedia ▾

Building Apps with Graphics & Animation ▾

Building Your First App

Welcome to Android application development!

This class teaches you how to build your first Android app. You'll learn how to create an Android project and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.

Before you start this class, be sure you have your development environment set up. You need to:

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse (if you'll use the Eclipse IDE).
3. Download the latest SDK tools and platforms using the SDK Manager.

If you haven't already done these tasks, start by downloading the [Android SDK](#) and following the install steps. Once you've finished the setup, you're ready to begin this class.

This class uses a tutorial format that incrementally builds a small Android app that teaches you some fundamental concepts about Android development, so it's important that you follow each step.

[Start the first lesson >](#)

 +1  1.4k

Except as noted, this content is licensed under [Creative Commons Attribution 2.5](#). For details and restrictions, see the [Content License](#).

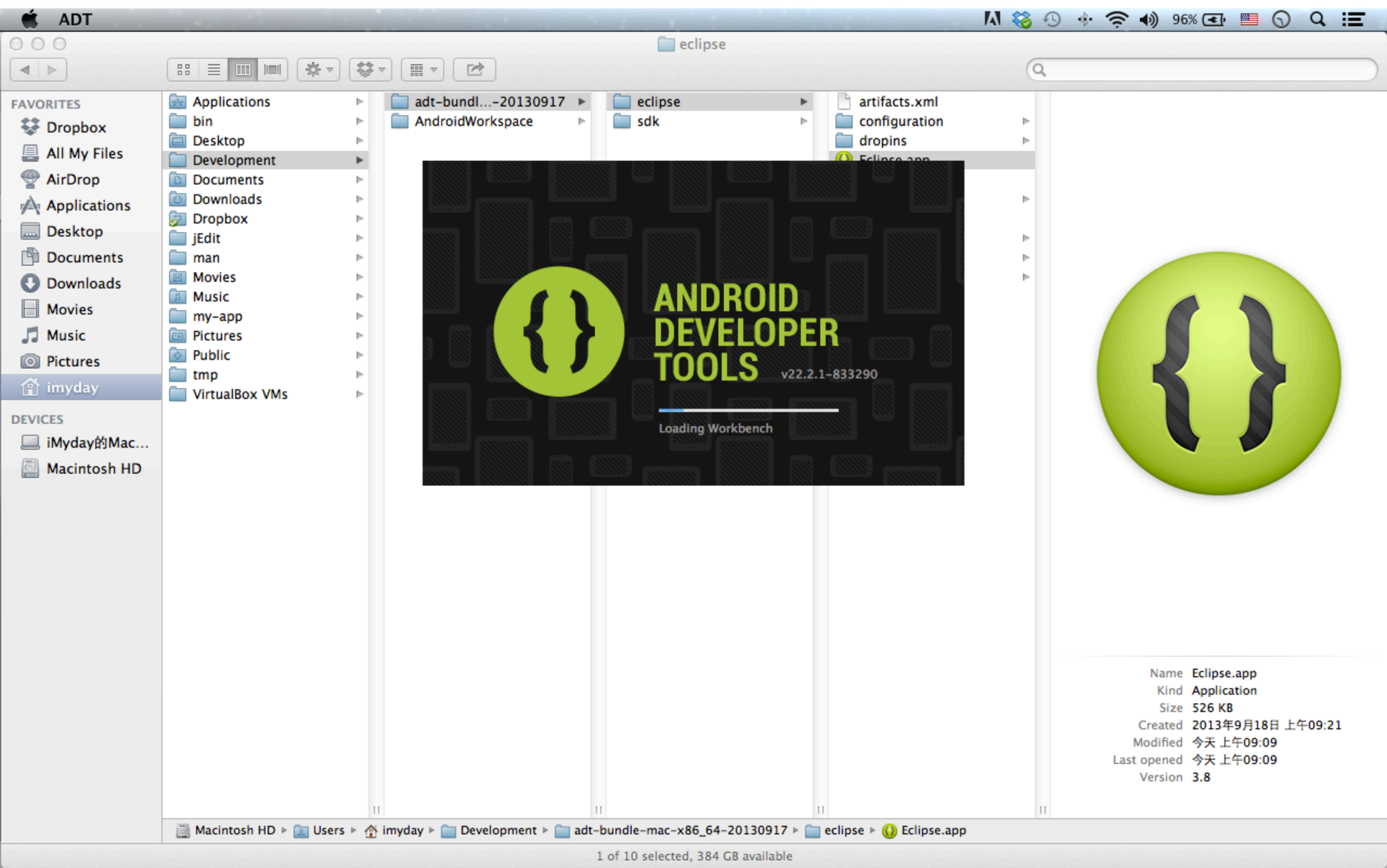
GET STARTED >

DEPENDENCIES AND PREREQUISITES

- [Android SDK](#)
- [ADT Plugin](#) 20.0.0 or higher (if you're using Eclipse)

<http://developer.android.com/training/basics/firstapp/index.html>

52



Creating an Android Project

developer.android.com/training/basics/firstapp/creating-project.html

Developers

Design

Develop

Distribute

Training

API Guides

Reference

Tools

Google Services

Getting Started

Building Your First App

Creating an Android Project

Running Your Application

Building a Simple User Interface

Starting Another Activity

Adding the Action Bar

Supporting Different Devices

Managing the Activity Lifecycle

Building a Dynamic UI with Fragments

Saving Data

Interacting with Other Apps

Building Apps with Content Sharing

Building Apps with Multimedia

Building Apps with Graphics & Animation

Creating an Android Project

An Android project contains all the files that comprise the source code for your Android app. The Android SDK tools make it easy to start a new Android project with a set of default project directories and files.

This lesson shows how to create a new project either using Eclipse (with the ADT plugin) or using the SDK tools from a command line.

Note: You should already have the Android SDK installed, and if you're using Eclipse, you should also have the [ADT plugin](#) installed (version 21.0.0 or higher). If you don't have these, follow the guide to [Installing the Android SDK](#) before you start this lesson.

Create a Project with Eclipse

- Click **New** in the toolbar.
- In the window that appears, open the **Android** folder, select **Android Application Project**, and click **Next**.
- Fill in the form that appears:
 - Application Name** is the app name that appears to users. For this project, use "My First App."
 - Project Name** is the name of your project directory and the name visible in Eclipse.

< PREVIOUS

NEXT >

THIS LESSON TEACHES YOU TO

- Create a Project with Eclipse
- Create a Project with Command Line Tools

YOU SHOULD ALSO READ

- Installing the SDK
- Managing Projects

New Android Application

New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used

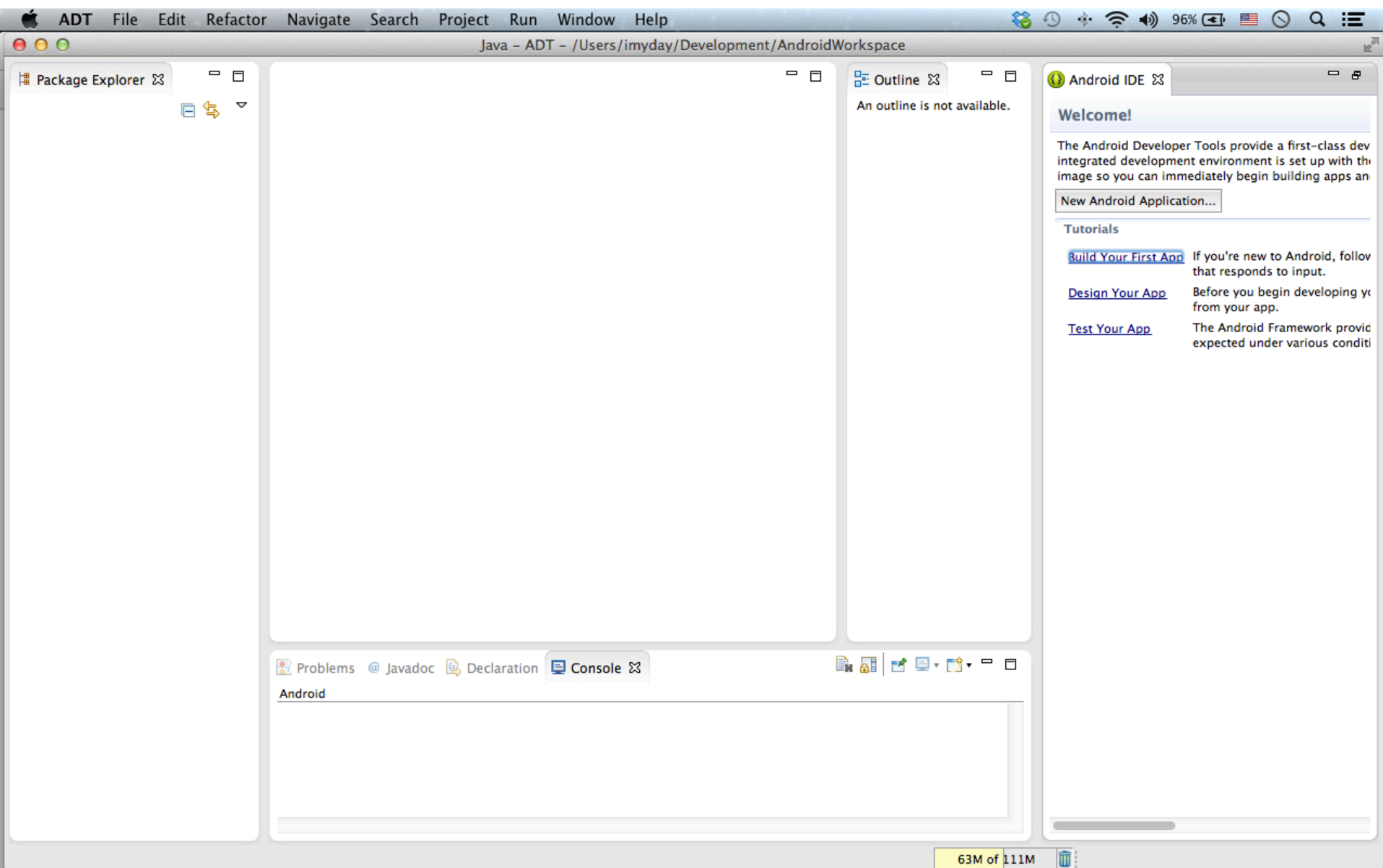
Application Name: My First App

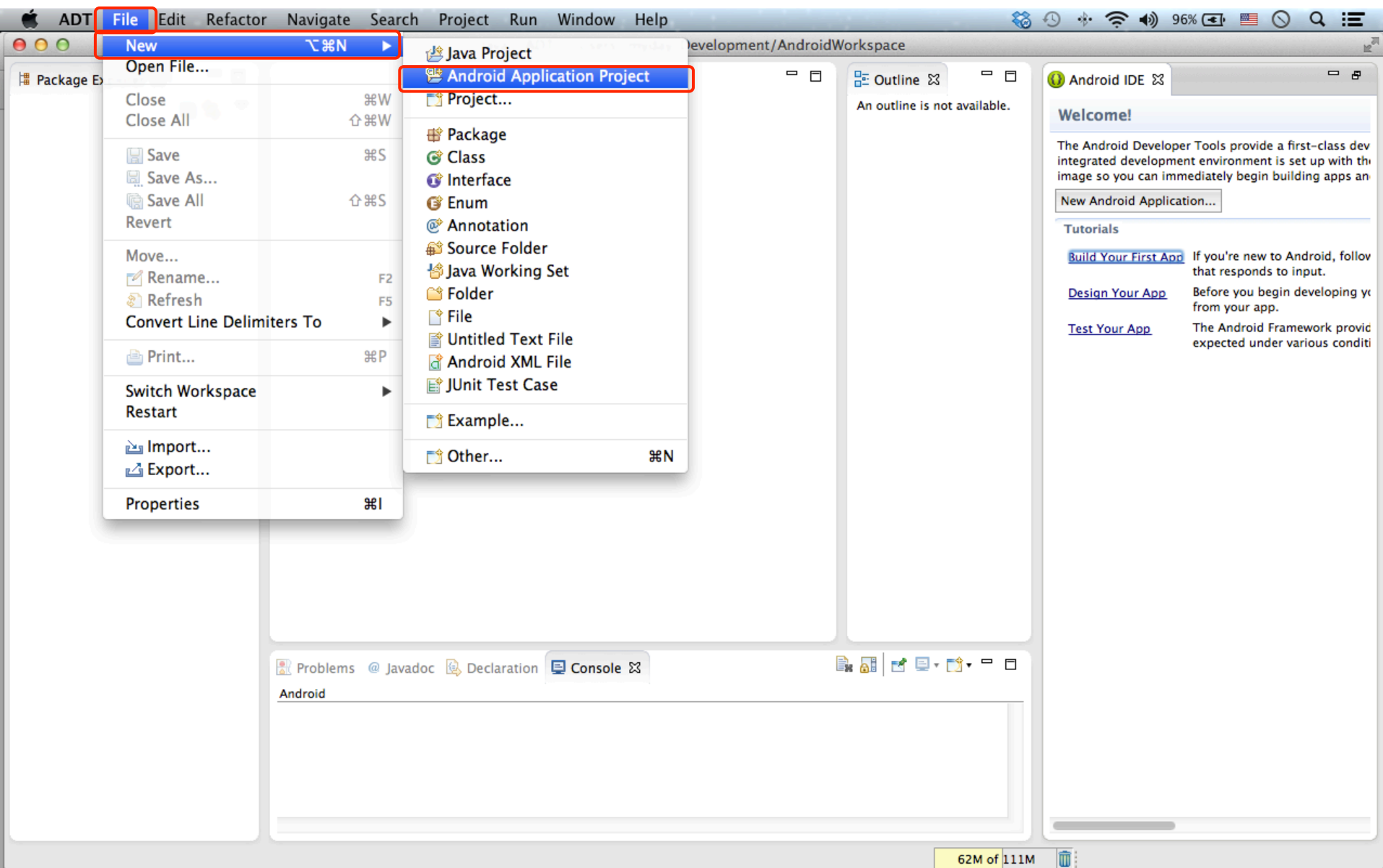
Project Name: MyFirstApp

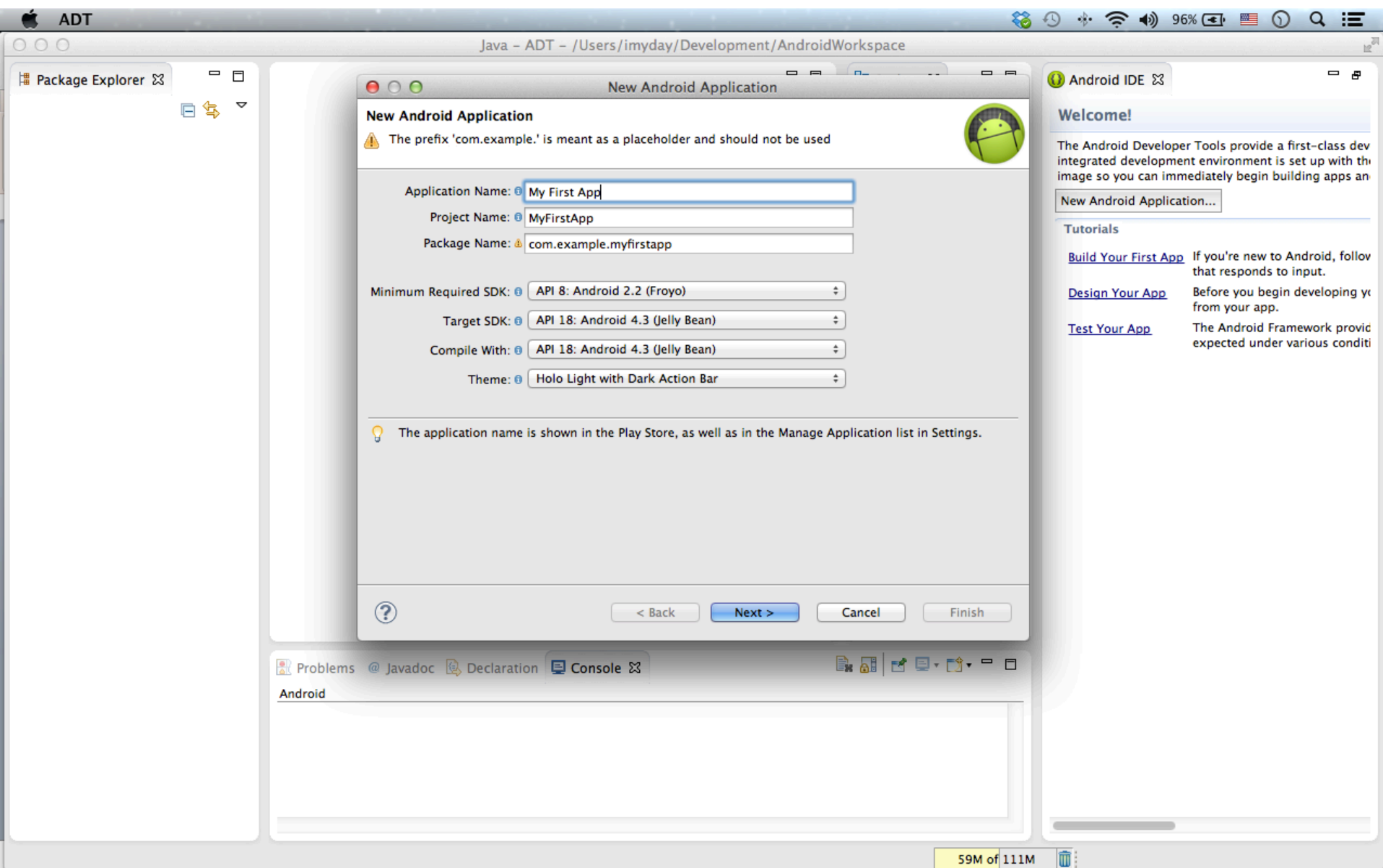
Package Name: com.example.myfirstapp

<http://developer.android.com/training/basics/firstapp/creating-project.html>

54







ADT

Java - ADT - /Users/imyday/Development/AndroidWorkspace

Package Explorer

New Android Application

Creates a new Android Application

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

 The package name must be a unique identifier for your application. It is typically not shown to users, but it *must* stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.



Android IDE

Welcome!

The Android Developer Tools provide a first-class dev integrated development environment is set up with the image so you can immediately begin building apps and

Tutorials

[Build Your First App](#) If you're new to Android, follow that responds to input.

[Design Your App](#) Before you begin developing your from your app.

[Test Your App](#) The Android Framework provides expected under various conditions

Problems @ Javadoc Declaration Console

Android

56M of 111M

ADT

Java - ADT - /Users/imyday/Development/AndroidWorkspace

Package Explorer

New Android Application

Creates a new Android Application

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme: ☒ Holo Light with Dark Action Bar

 The package name must be a unique identifier for your application. It is typically not shown to users, but it *must* stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the "same app". This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.

Android IDE

Welcome!

The Android Developer Tools provide a first-class dev integrated development environment is set up with the image so you can immediately begin building apps and

[New Android Application...](#)

Tutorials

[Build Your First App](#) If you're new to Android, follow that responds to input.

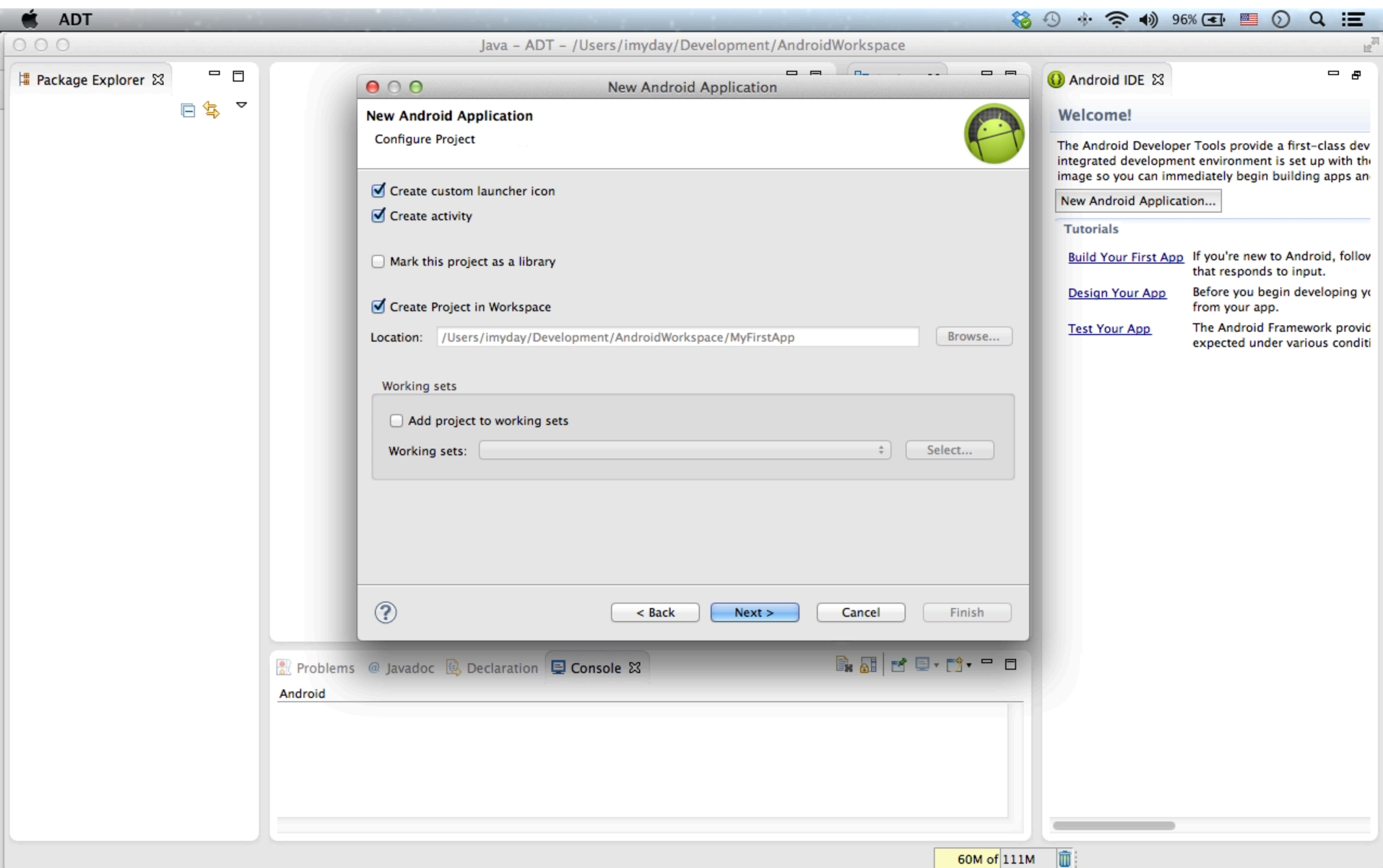
[Design Your App](#) Before you begin developing your app.

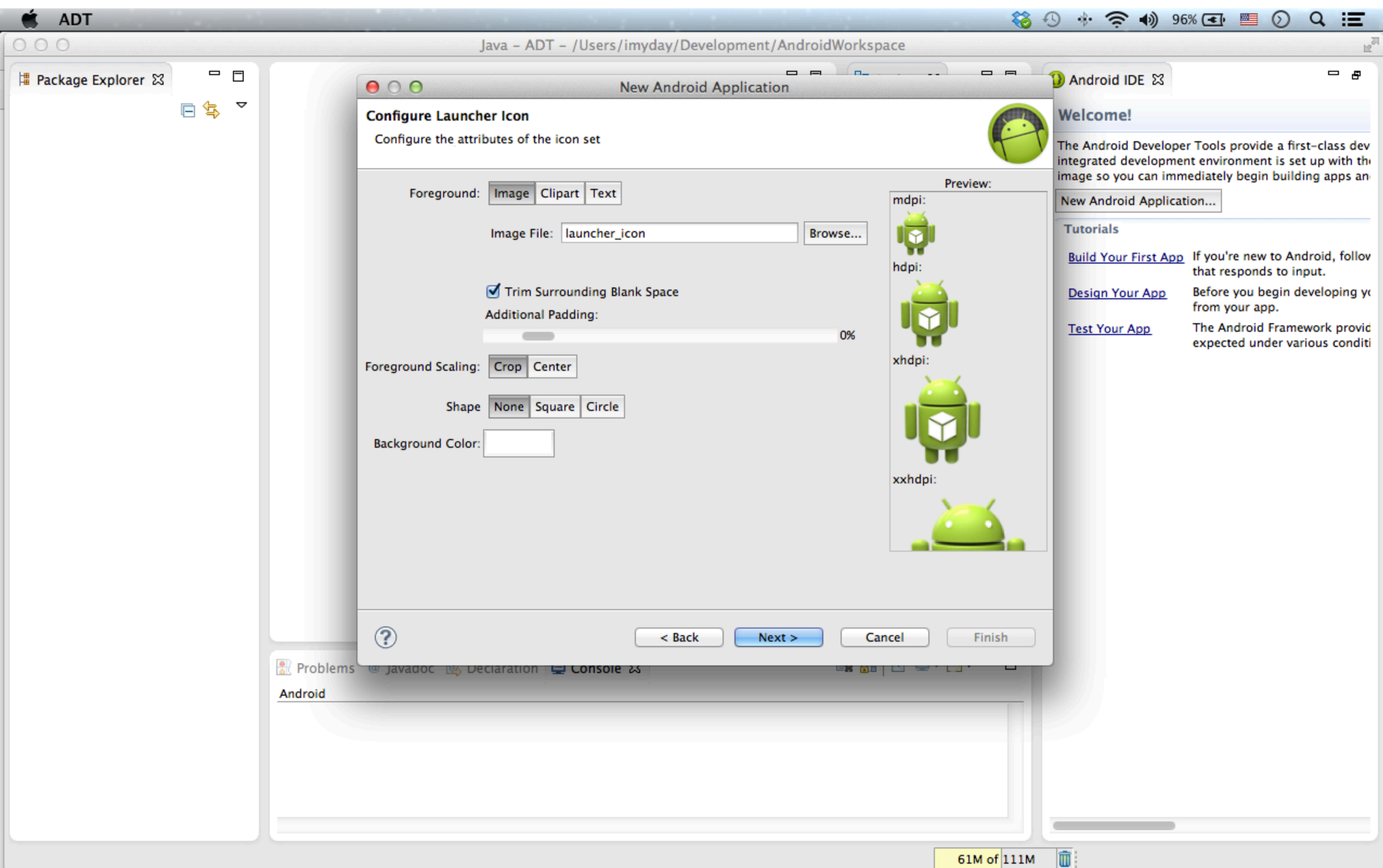
[Test Your App](#) The Android Framework provides expected under various conditions.

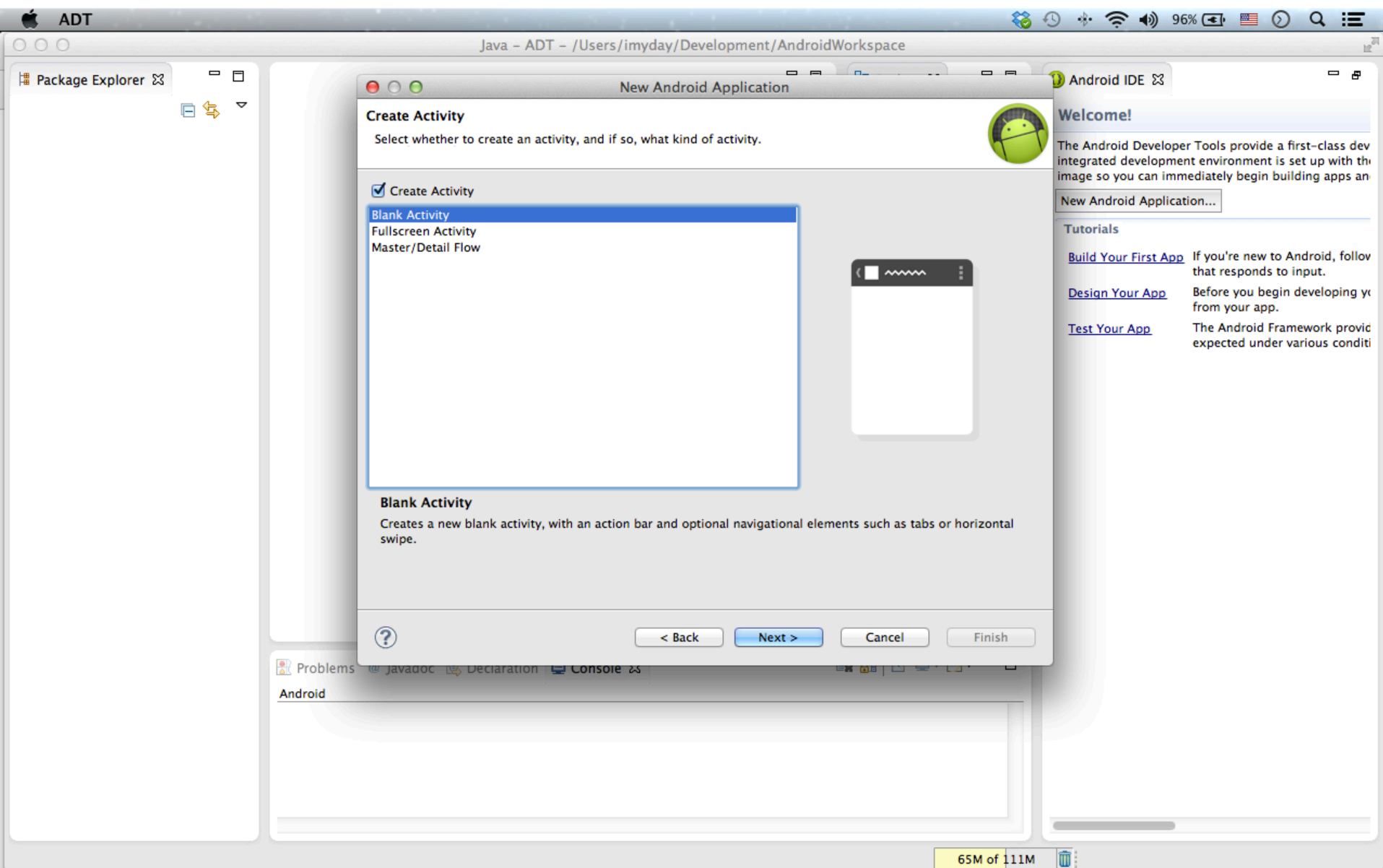
Problems @ Javadoc Declaration Console

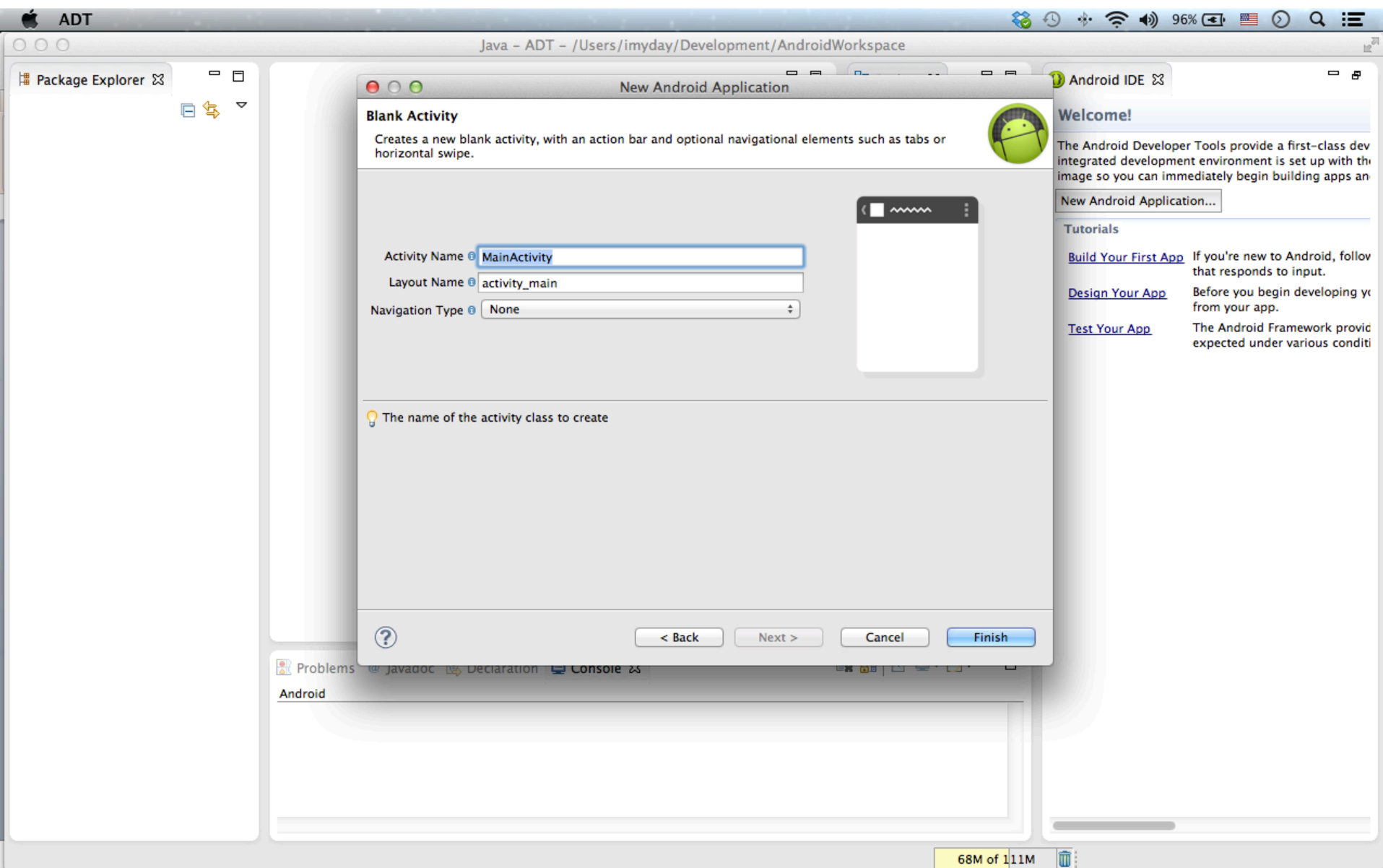
Android

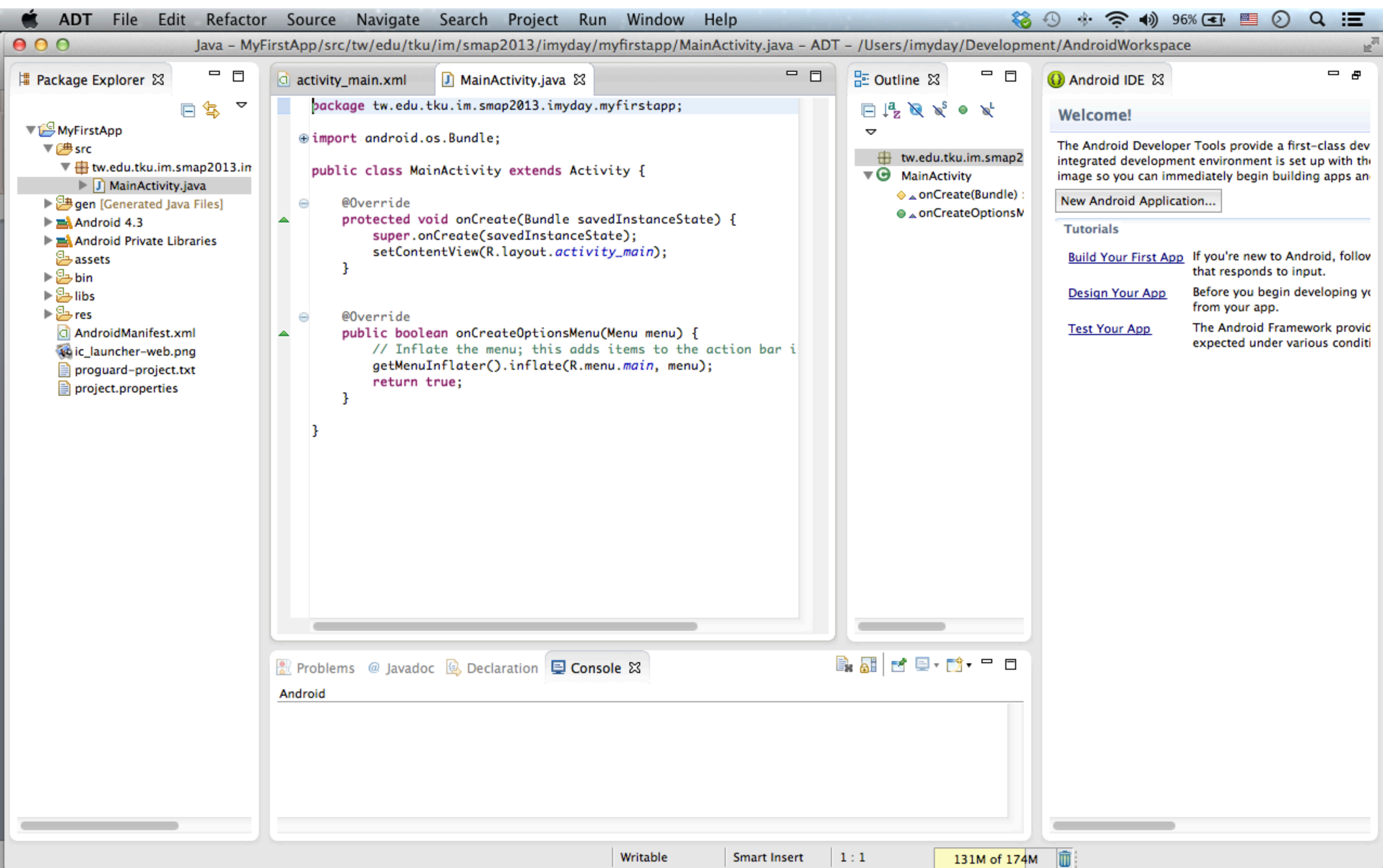
62M of 111M

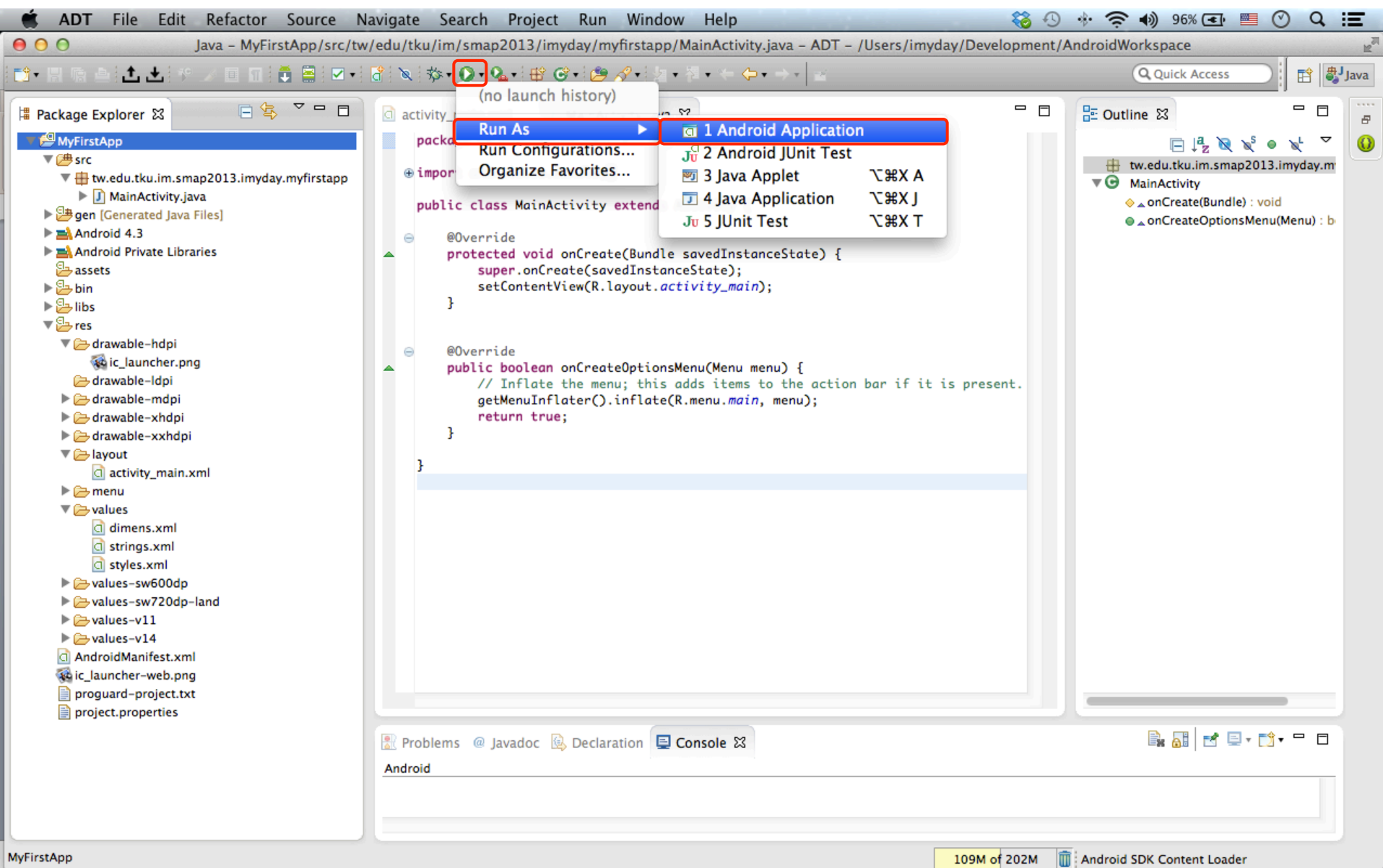












ADT File Edit Refactor Source Navigate Search Project Run Window Help

Java - MyFirstApp/src/tw/edu/tku/im/smap2013/imyday/myfirstapp/MainActivity.java - ADT - /Users/imyday/Development/AndroidWorkspace

Quick Access

Package Explorer

- MyFirstApp
 - src
 - tw.edu.tku.im.smap2013.imyday.myfirstapp
 - MainActivity.java
 - gen [Generated Java Files]
 - Android 4.3
 - Android Private Libraries
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - ic_launcher.png
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - activity_main.xml
 - menu
 - values
 - dimens.xml
 - strings.xml
 - styles.xml
 - values-sw600dp
 - values-sw720dp-land
 - values-v11
 - values-v14
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt
 - project.properties

activity_main.xml

MainActivity.java

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

Outline

- tw.edu.tku.im.smap2013.imyday.m
 - MainActivity
 - onCreate(Bundle) : void
 - onCreateOptionsMenu(Menu) : b

Problems @ Javadoc Declaration Console

DDMS

MyFirstApp 153M of 202M Launching MyFirstApp: (100%)

ADT

Java - MyFirstApp/src/tw/edu/tku/im/smap2013/imyday/myfirstapp/MainActivity.java - ADT - /Users/imyday/Development/AndroidWorkspace

Package Explorer

- MyFirstApp
 - src
 - tw.edu.tku.im.smap2013.imyday.myfirstapp
 - MainActivity.java
 - gen [Generated Java Files]
 - Android 4.3
 - Android Private Libraries
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - ic_launcher.png
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - activity_main.xml
 - menu
 - values
 - dimens.xml
 - strings.xml
 - styles.xml
 - values-sw600dp
 - values-sw720dp-land
 - values-v11
 - values-v14
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt
 - project.properties

activity_main.xml

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
```

MainActivity.java

Outline

- tw.edu.tku.im.smap2013.imyday.m
 - MainActivity
 - onCreate(Bundle) : void
 - onCreateOptionsMenu(Menu) : b

Android AVD Error

No compatible targets were found. Do you wish to add a new Android Virtual Device?

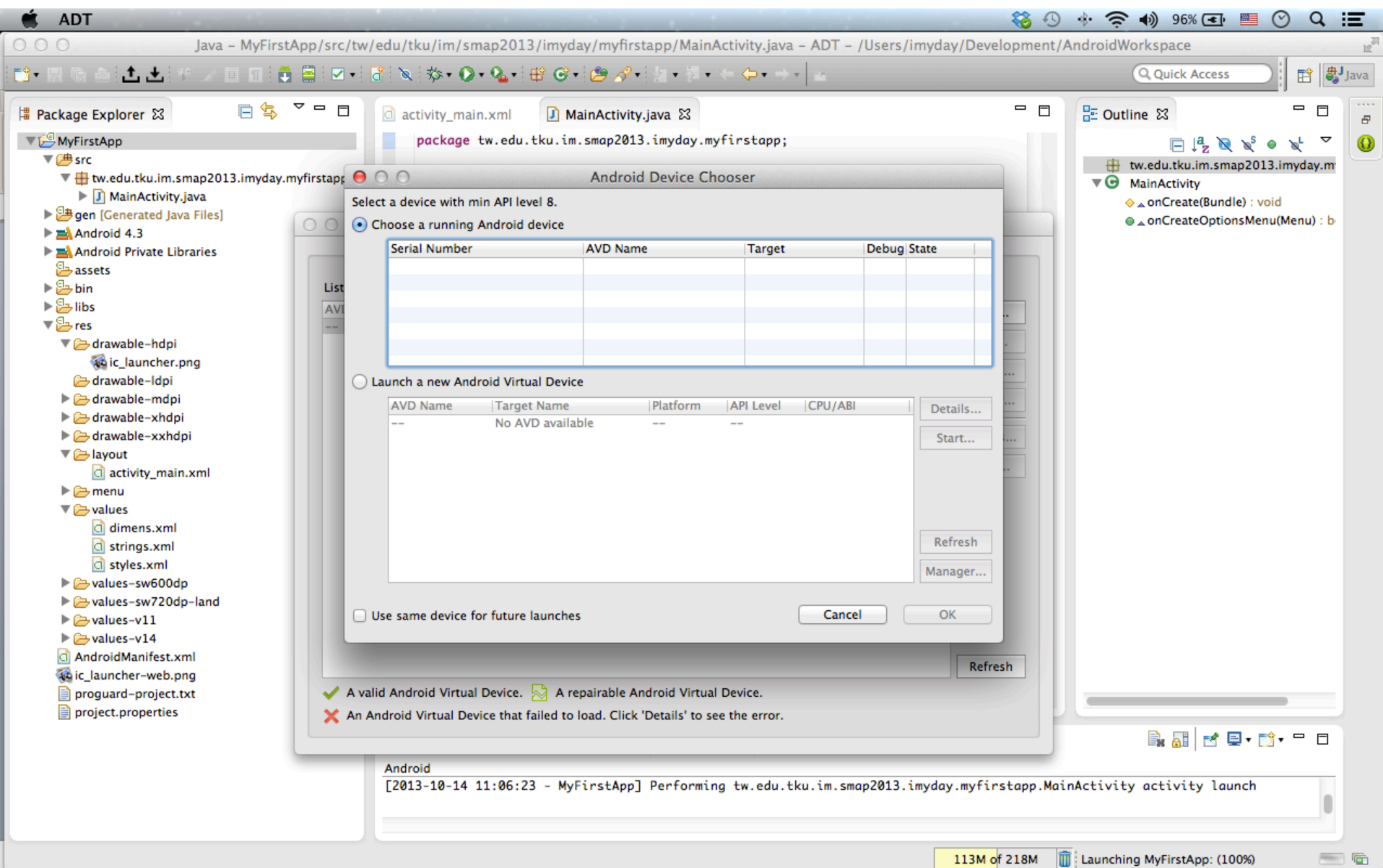
No Yes

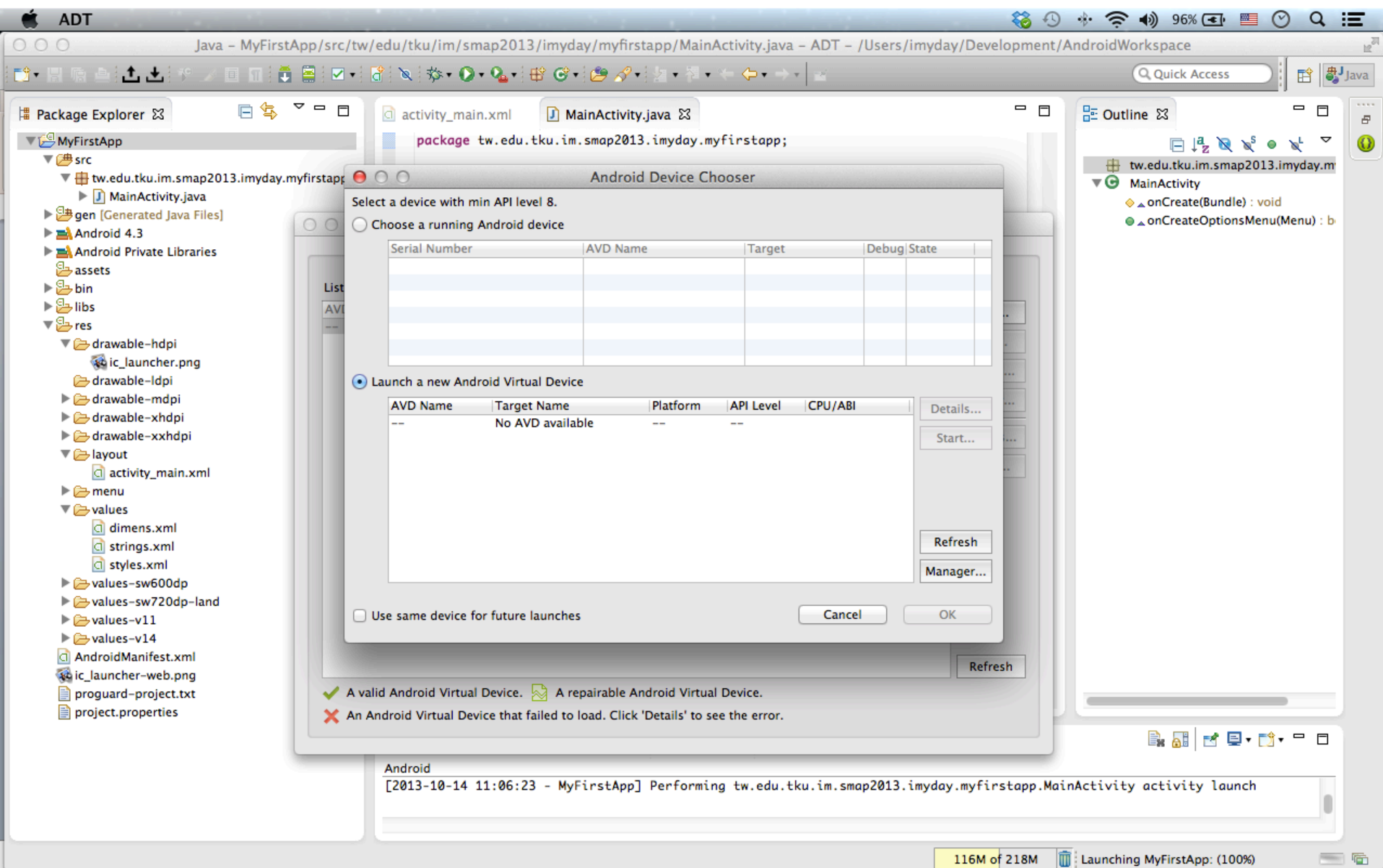
Problems @ Javadoc Declaration Console

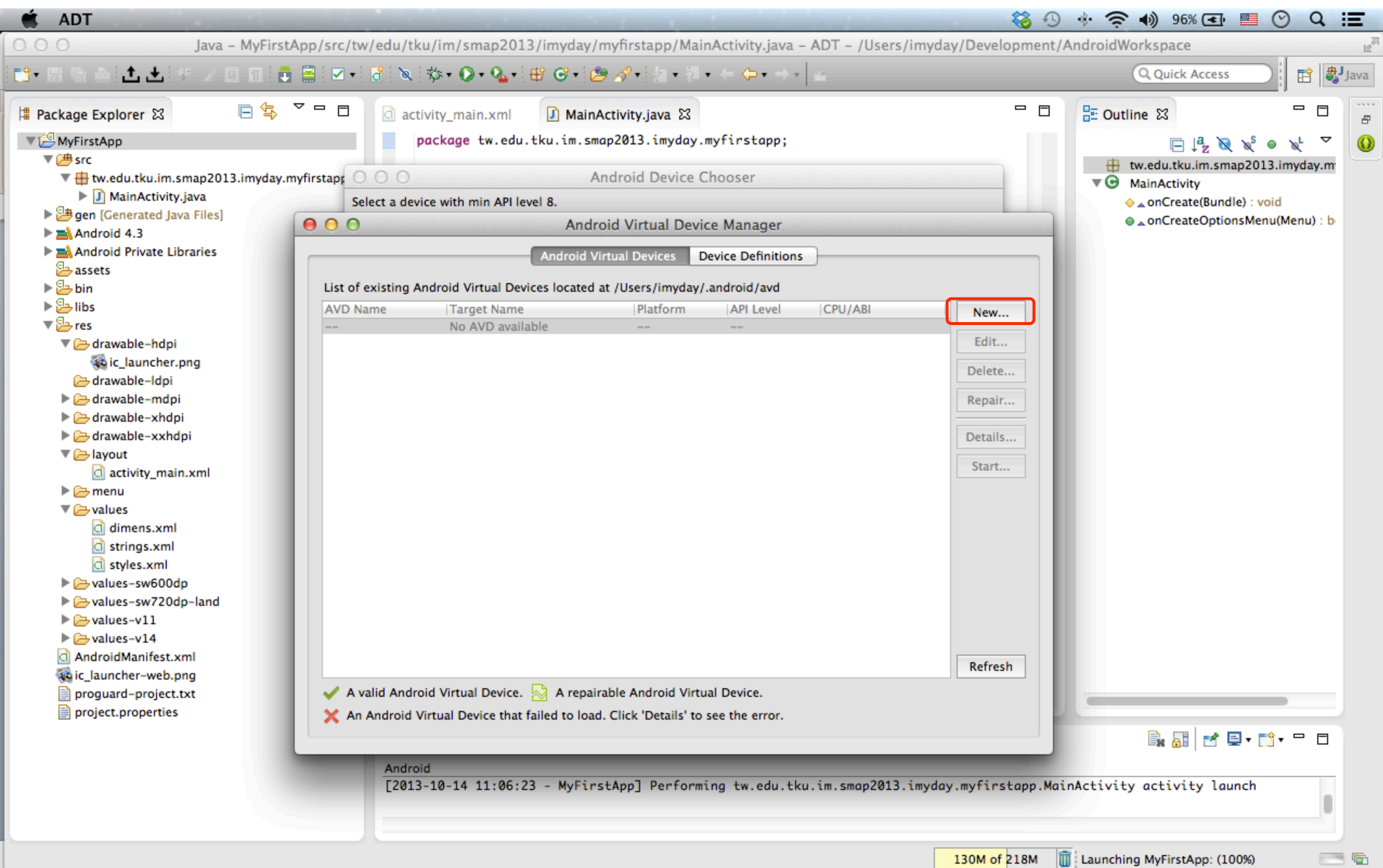
DDMS

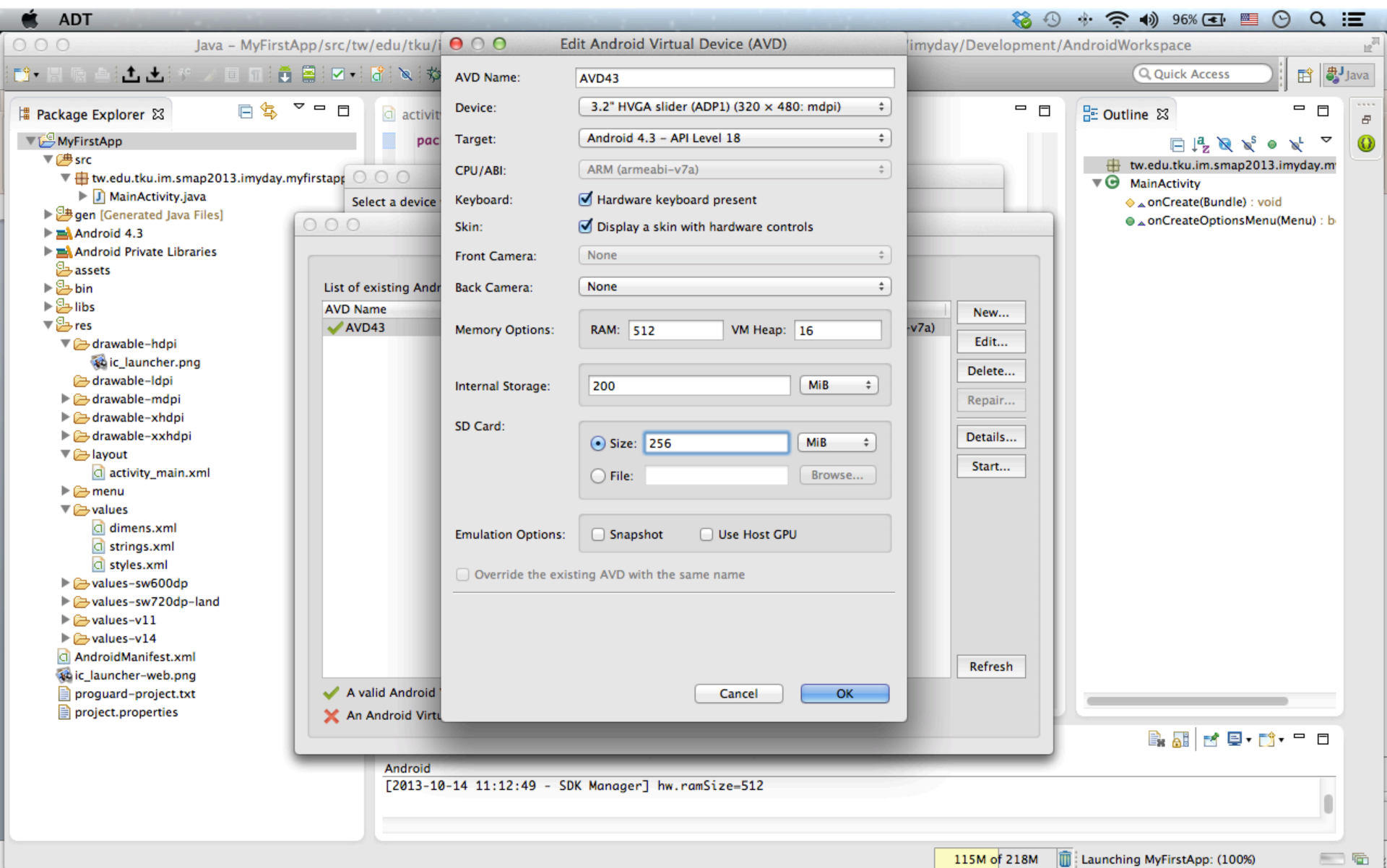
MyFirstApp

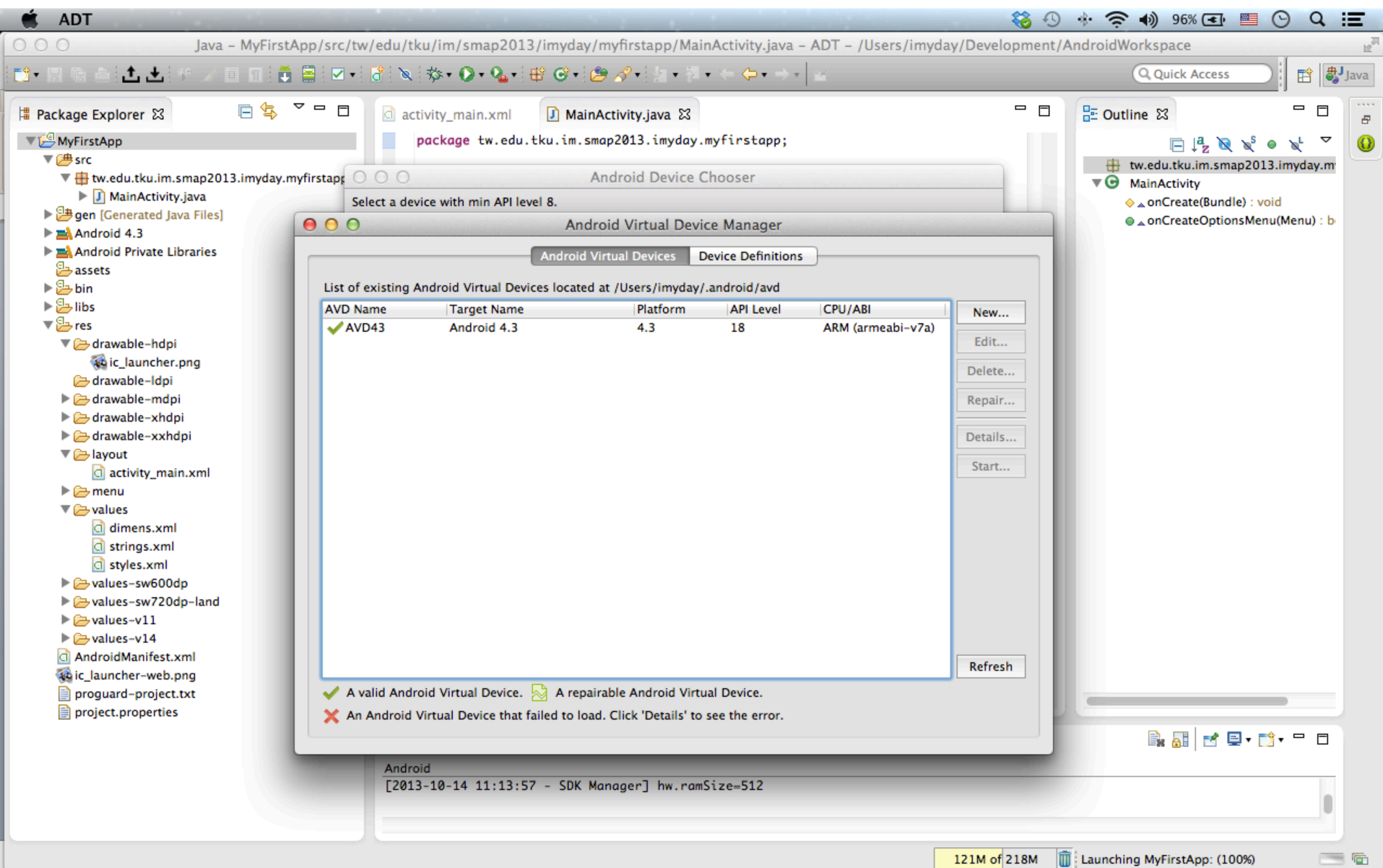
153M of 202M Building workspace: (75%)

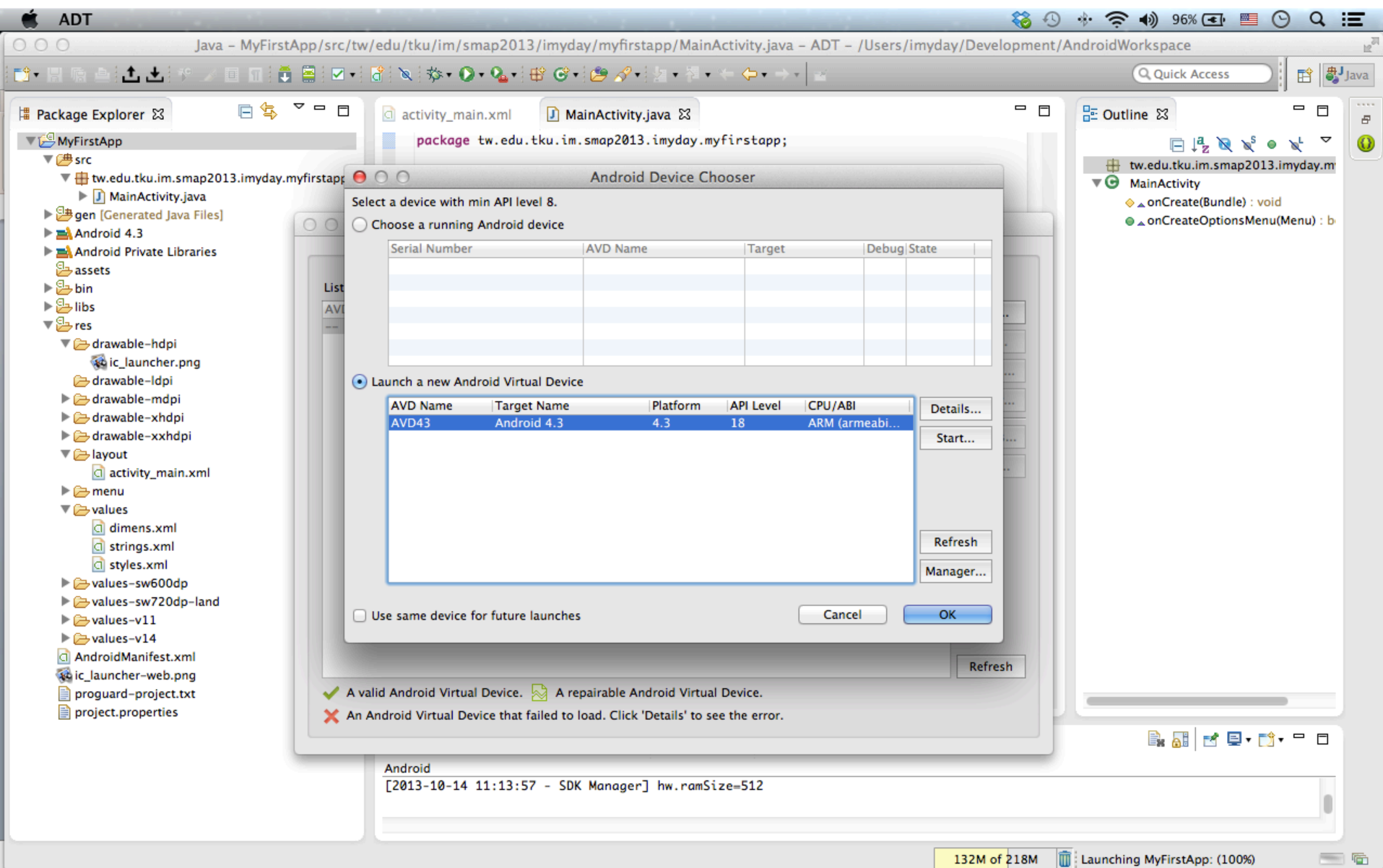


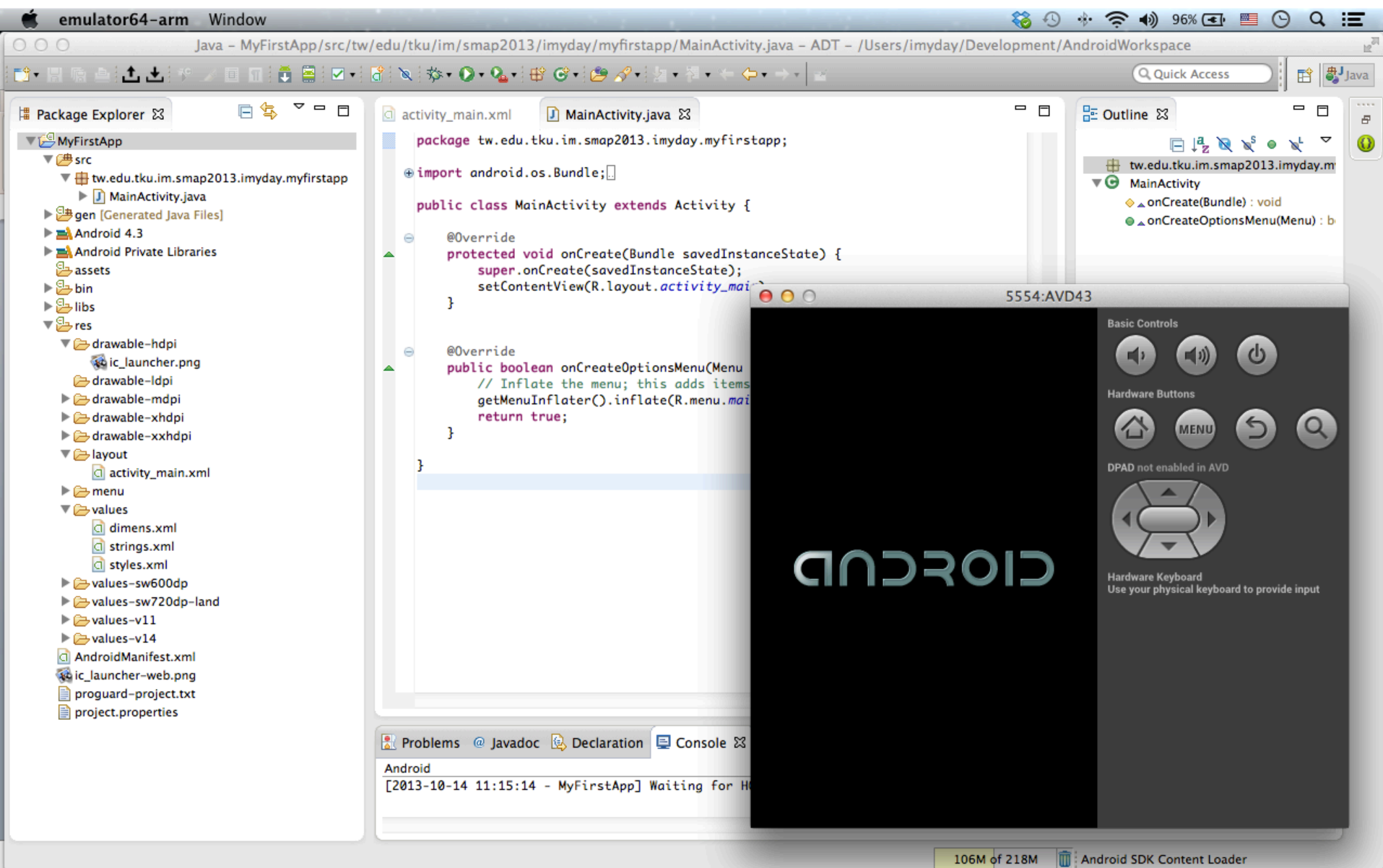


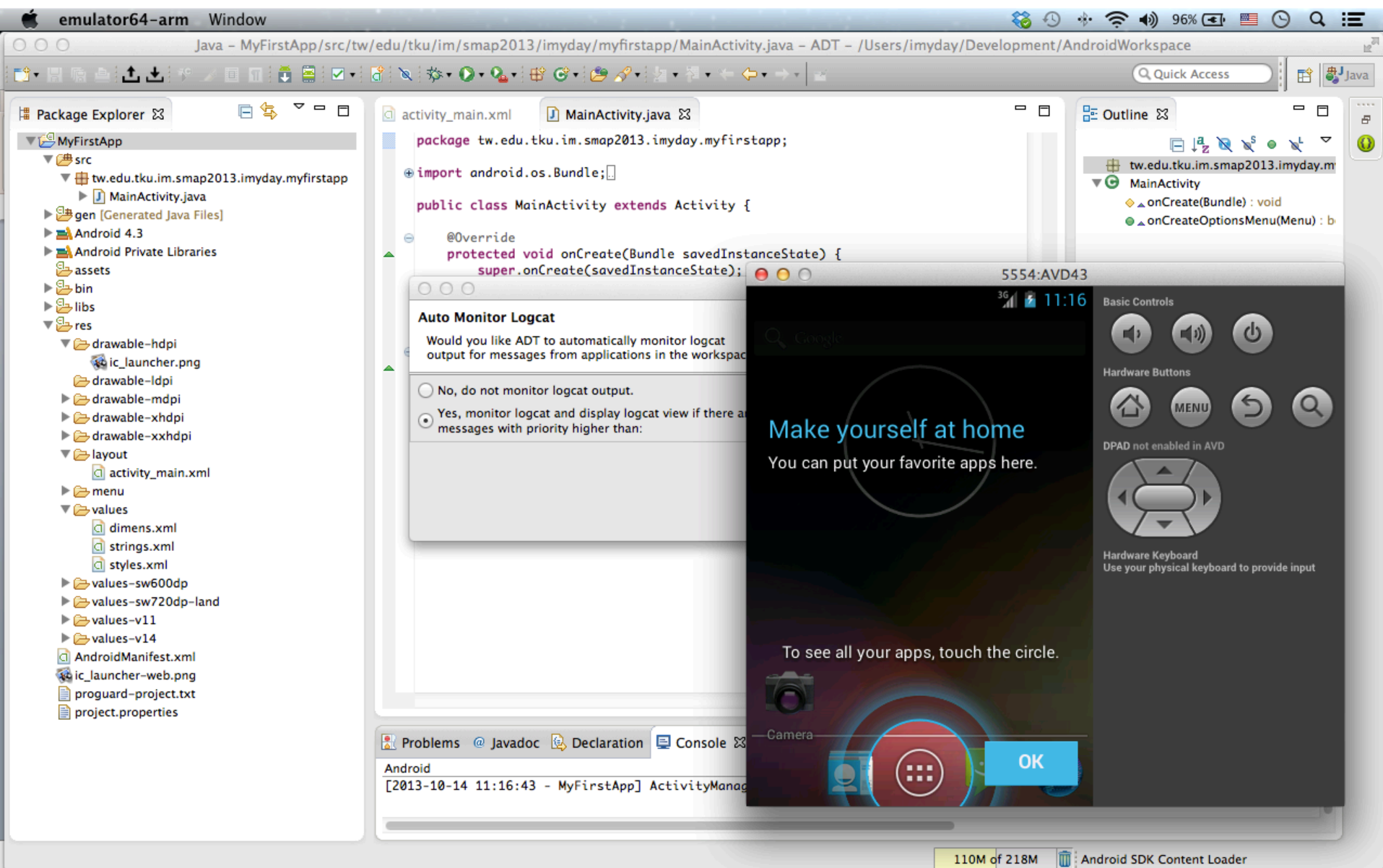


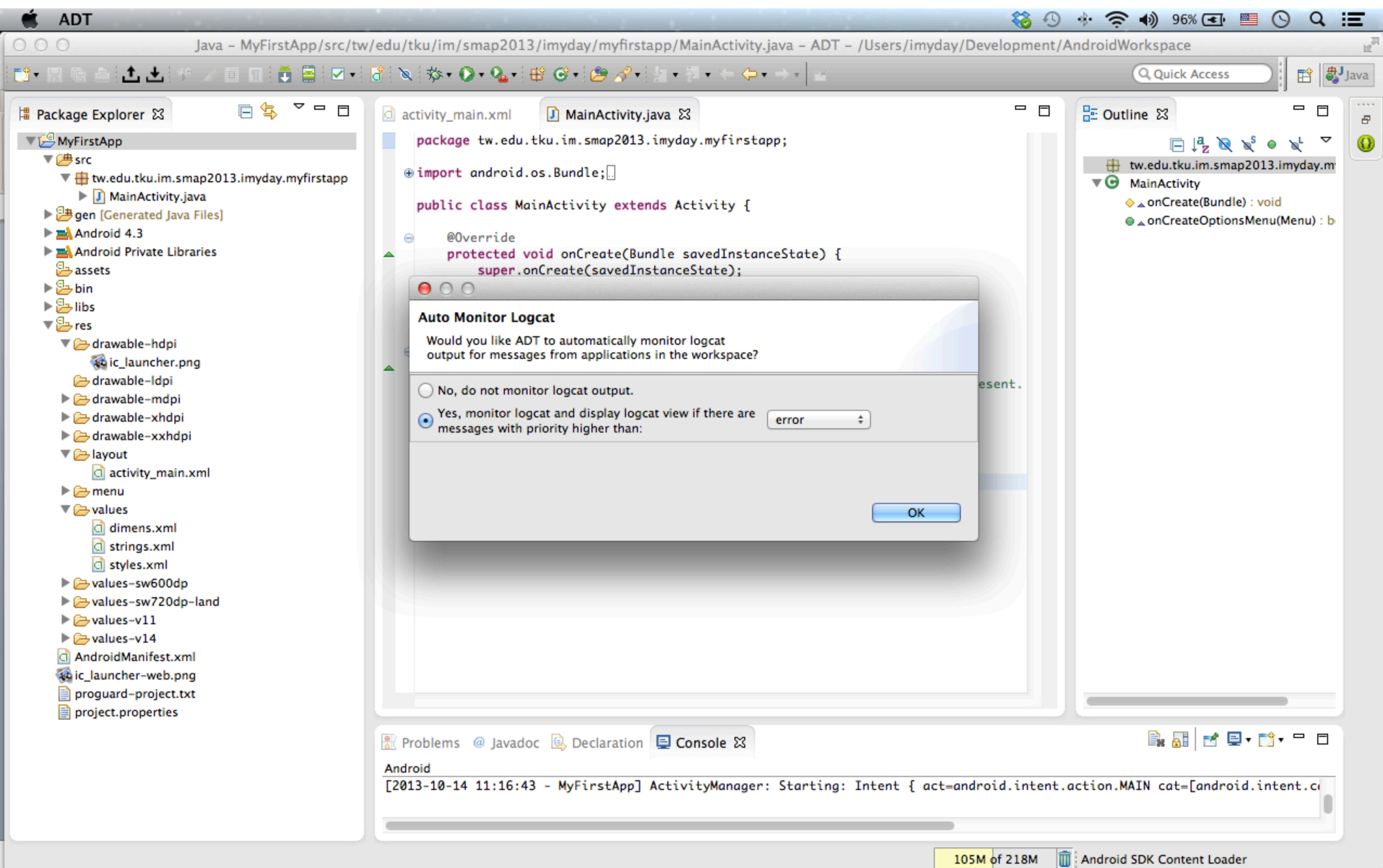


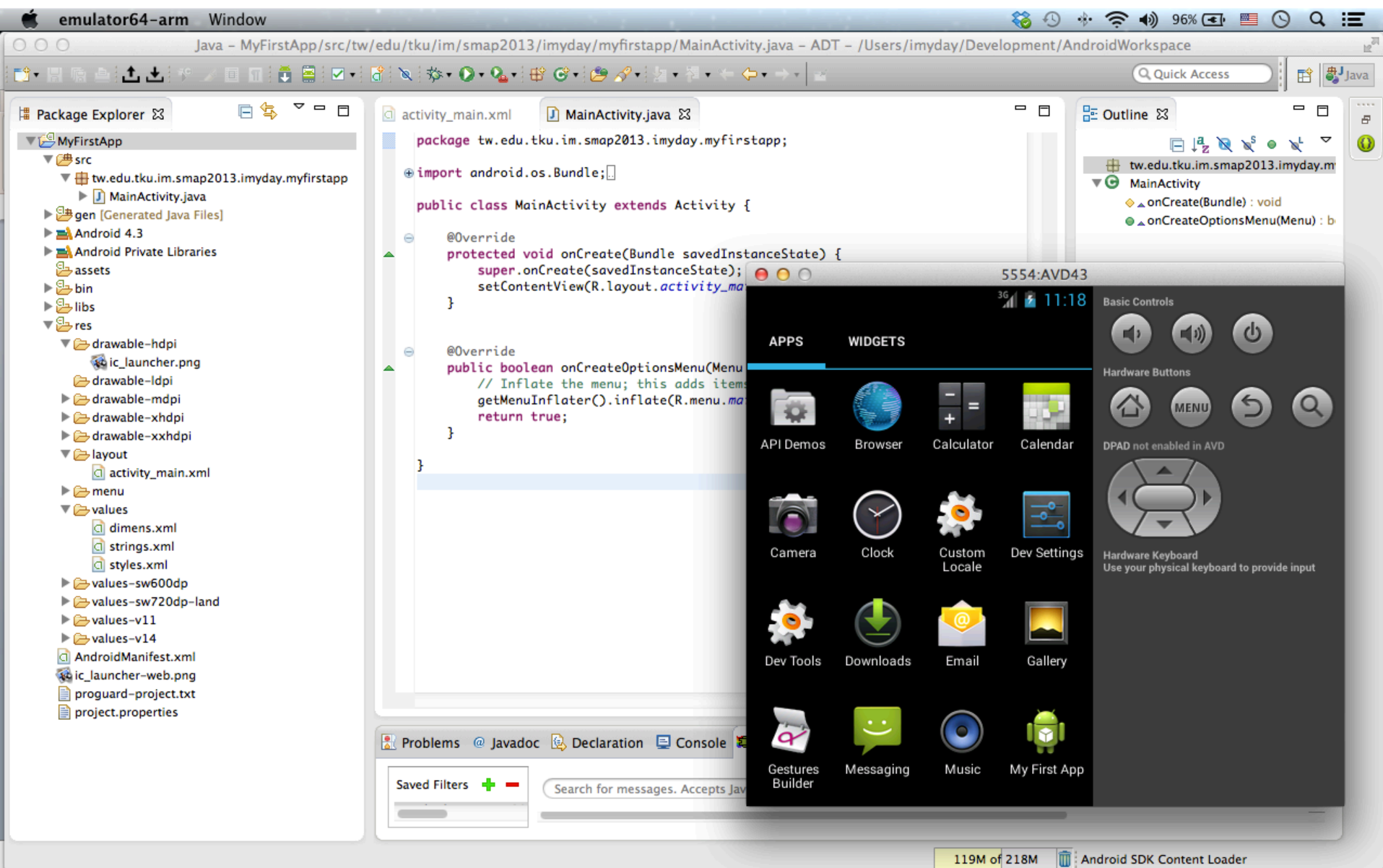












emulator64-arm Window

Java - MyFirstApp/src/tw/edu/tku/im/smap2013/imyday/myfirstapp/MainActivity.java - ADT - /Users/imyday/Development/AndroidWorkspace

Package Explorer

- MyFirstApp
 - src
 - tw.edu.tku.im.smap2013.imyday.myfirstapp
 - MainActivity.java
 - gen [Generated Java Files]
 - Android 4.3
 - Android Private Libraries
 - assets
 - bin
 - libs
 - res
 - drawable-hdpi
 - ic_launcher.png
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - activity_main.xml
 - menu
 - values
 - dimens.xml
 - strings.xml
 - styles.xml
 - values-sw600dp
 - values-sw720dp-land
 - values-v11
 - values-v14
 - AndroidManifest.xml
 - ic_launcher-web.png
 - proguard-project.txt
 - project.properties

activity_main.xml

MainActivity.java

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the menu.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
}
```

Outline

- tw.edu.tku.im.smap2013.imyday.m
 - MainActivity
 - onCreate(Bundle) : void
 - onCreateOptionsMenu(Menu) : b

5554:AVD43

11:19

My First App

Hello world!

Basic Controls

Hardware Buttons

DPAD not enabled in AVD

Hardware Keyboard

Use your physical keyboard to provide input

Problems @ Javadoc Declaration Console

Saved Filters + -

Search for messages. Accepts Java

123M of 218M Android SDK Content Loader

Starting Another Activity

developer.android.com/training/basics/firstapp/starting-activity.html

Getting Started

Building Your First App

Creating an Android Project

Running Your Application

Building a Simple User Interface

Starting Another Activity

Adding the Action Bar

Supporting Different Devices

Managing the Activity Lifecycle

Building a Dynamic UI with Fragments

Saving Data

Interacting with Other Apps

Building Apps with Content Sharing

Building Apps with Multimedia

Building Apps with Graphics & Animation

Building Apps with Connectivity & the Cloud

Building Apps with User Info & Location

Best Practices for

on the second activity.

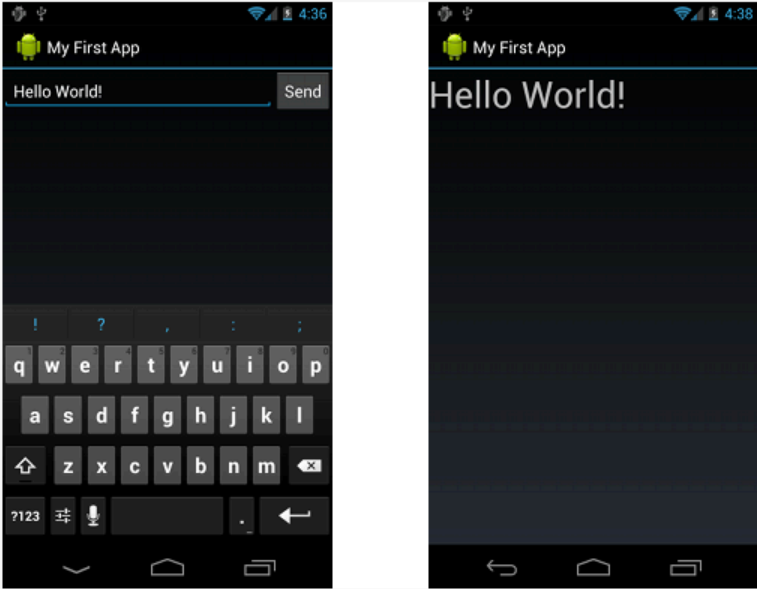


Figure 2. Both activities in the final app, running on Android 4.0.

That's it, you've built your first Android app!

To learn more, follow the link below to the next class.

 +1

350

< PREVIOUS

NEXT >

NEXT CLASS: ADDING THE ACTION BAR >

Except as noted, this content is licensed under [Creative Commons Attribution 2.5](#). For details and restrictions, see the [Content License](#).

[About Android](#) | [Legal](#) | [Support](#)

emulator64-arm Window

Java - MyFirstApp/src/tw/edu/tku/im/smap2013/imyday/myfirstapp/MainActivity.java - ADT - /Users/imyday/Development/AndroidWorkspace

Package Explorer

- tw.edu.tku.im.smap2013.imyday.myfirsta
 - DisplayMessageActivity.java
 - MainActivity.java
- gen [Generated Java Files]
- Android 4.3
- Android Private Libraries
- assets
- bin
 - dexedLibs
 - res
 - AndroidManifest.xml
 - classes.dex
 - jarlist.cache
 - MyFirstApp.apk
 - resources.ap_
- libs
- res
 - drawable-hdpi
 - drawable-ldpi
 - drawable-mdpi
 - drawable-xhdpi
 - drawable-xxhdpi
 - layout
 - activity_display_message.xml
 - activity_main.xml
 - menu
 - values
 - dimens.xml
 - strings.xml
 - styles.xml
 - values-en
 - strings.xml
 - values-sw600dp
 - values-sw720dp-land
 - values-v11
 - values-v14
 - values-zh-rTW

MainActivity.java

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Intent;
import android.widget.EditText;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.my";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /** Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
```

strings.xml

activity_display_message.xml

DisplayMessageActivity.java

Outline

- tw.edu.tku.im.smap2013.imyday.myfirsta
 - MainActivity
 - EXTRA_MESSAGE : String
 - onCreate(Bundle) : void
 - onCreateOptionsMenu(Menu) : boolean
 - sendMessage(View) : void

5554:AVD43

My First App

Hello World Myday

Send

Basic Controls

Hardware Buttons

DPAD not enabled in AVD

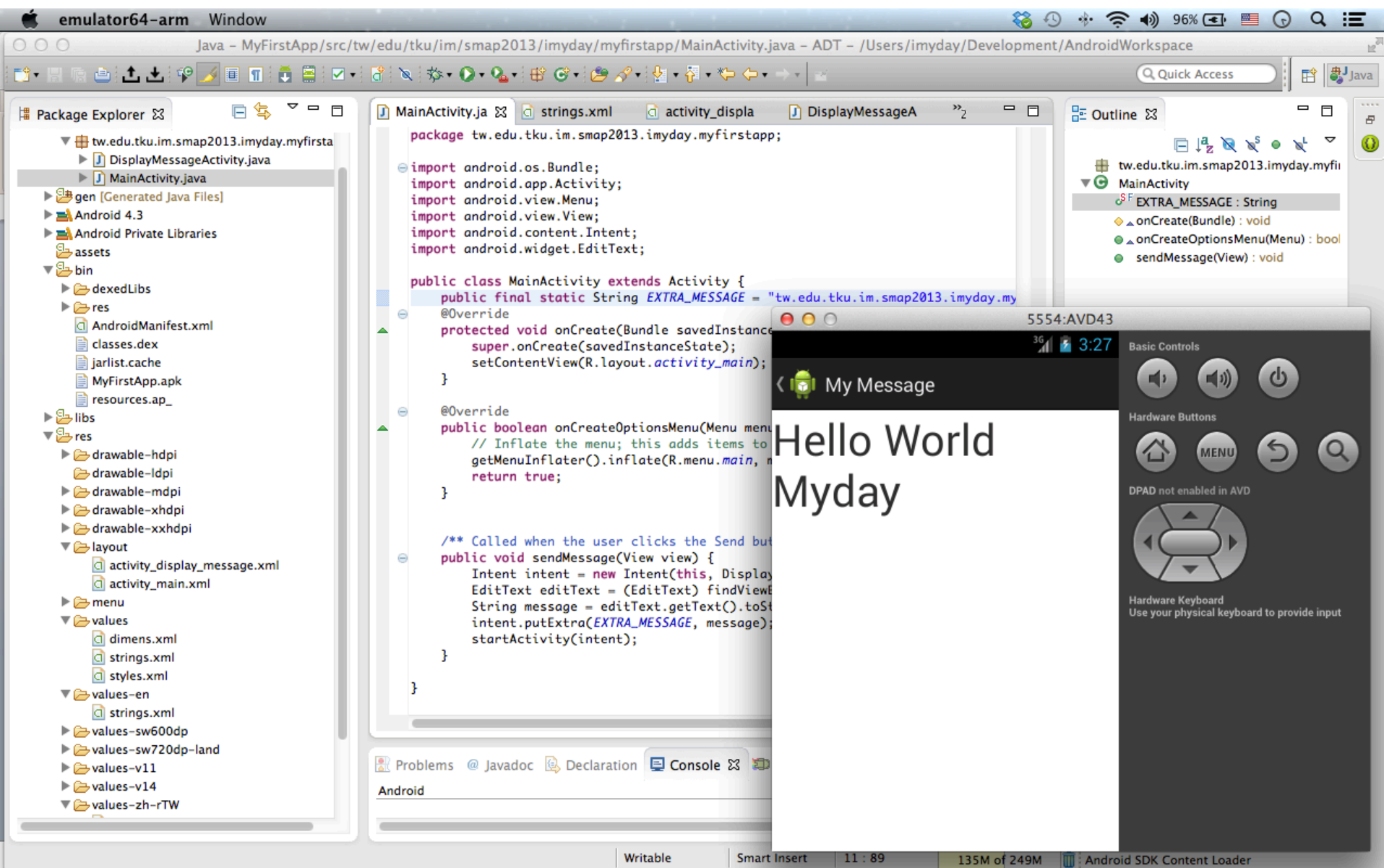
Hardware Keyboard

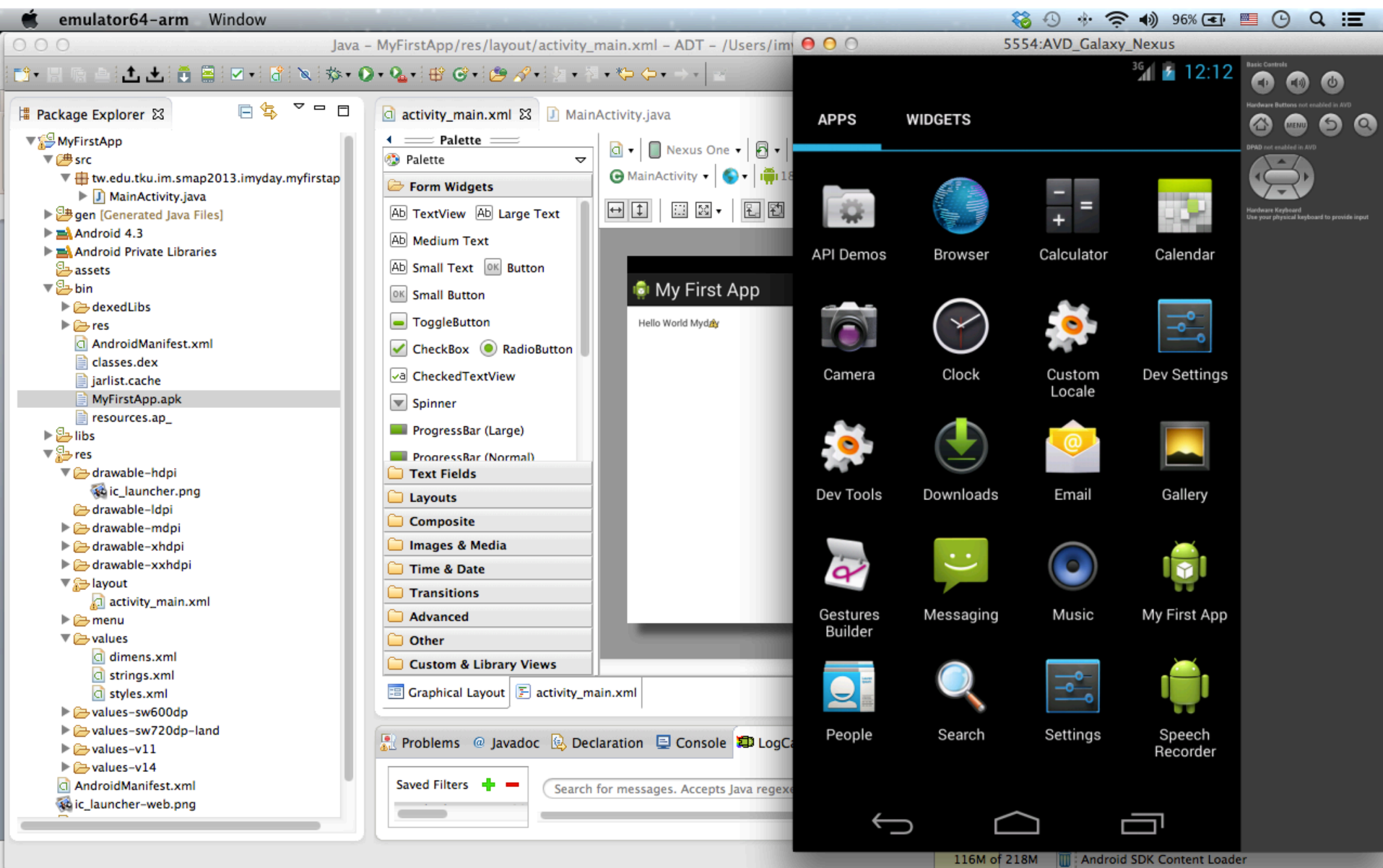
Use your physical keyboard to provide input

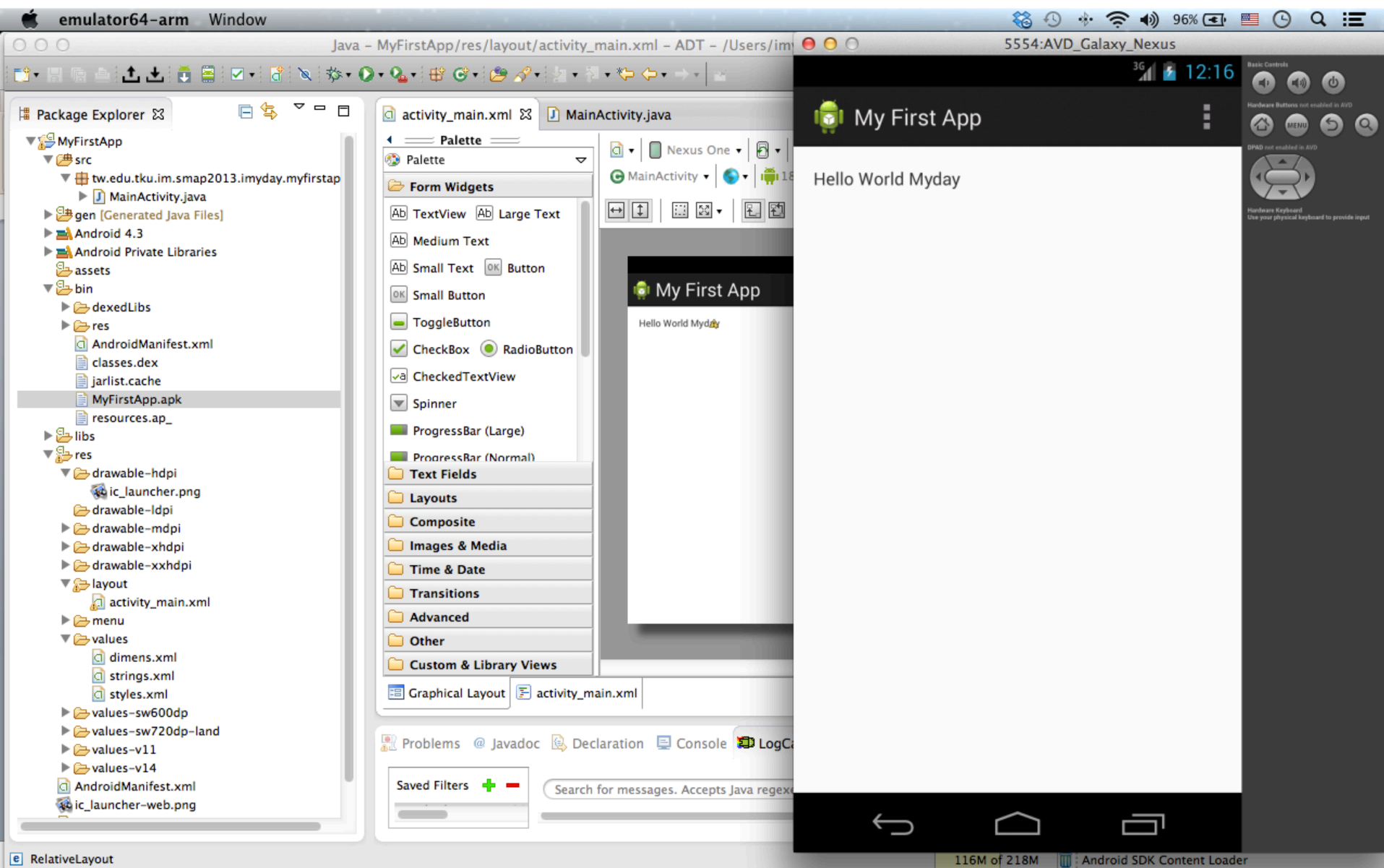
Problems @ Javadoc Declaration Console

Android

Writable Smart Insert 11 : 89 133M of 249M Android SDK Content Loader







activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />
</LinearLayout>
```

activity_display_message.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".DisplayMessageActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">My First App</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="edit_message">Enter a message</string>
    <string name="button_send">Send</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="title_activity_display_message">My Message</string>

</resources>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="tw.edu.tku.im.smap2013.imyday.myfirstapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="tw.edu.tku.im.smap2013.imyday.myfirstapp.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="tw.edu.tku.im.smap2013.imyday.myfirstapp.DisplayMessageActivity"
            android:label="@string/title_activity_display_message"
            android:parentActivityName="tw.edu.tku.im.smap2013.imyday.myfirstapp" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="tw.edu.tku.im.smap2013.imyday.myfirstapp" />
            </activity>
    </application>

</manifest>
```

DisplayMessageActivity.java

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.os.Build;
import android.content.Intent;
import android.widget.TextView;

public class DisplayMessageActivity extends Activity {

    @SuppressWarnings("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_message);
        // Show the Up button in the action bar.
        setupActionBar();

        //Get the message from the intent
        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

        //Great the text view
        TextView textView = new TextView(this);
        textView.setTextSize(40);
        textView.setText(message);

        setContentView(textView);
    }
    ...
}
```

MainActivity.java

```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Intent;
import android.widget.EditText;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myfirstapp.MESSAGE";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /** Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
```

DisplayMessageActivity.java

```
/**
 * Set up the {@link android.app.ActionBar}, if the API is available.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
private void setupActionBar() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.display_message, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            // This ID represents the Home or Up button. In the case of this
            // activity, the Up button is shown. Use NavUtils to allow users
            // to navigate up one level in the application structure. For
            // more details, see the Navigation pattern on Android Design:
            //
            // http://developer.android.com/design/patterns/navigation.html#up-vs-back
            //
            NavUtils.navigateUpFromSameTask(this);
            return true;
    }
    return super.onOptionsItemSelected(item);
}
```

Alternatives for Developing Android Apps

- MIT App Inventor
 - drag-and-drop Android app development tool.
 - <http://appinventor.mit.edu/>
- Appery.io
 - develop apps for Android (iOS / Windows Phone).
 - <http://appery.io>
- Appnotch
 - drag-and-drop service that allows you to develop apps for Android (iOS).
 - <http://www.appnotch.com/>

MIT App Inventor

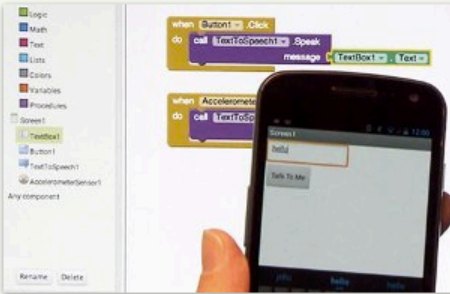
MIT App Inventor

[About Us](#) [Blog](#) [Support](#) [Create](#)

Follow Us:     

Google™ Custom Search

Weekly Active Users: **87K** Total Registered Users: **1.9M** Countries: **195** Apps Built: **4.7M**



Your ideas.
Your designs.
Your apps.

[Invent Now](#)

Recent News

[AUGUST: Congratulations to co-App of the Month: Sparki Bluetooth Controller](#)

[AUGUST: Congratulations to co-App of the Month: Let Me Hear Again PRO](#)

[JULY: Congratulations to co-App of the Month: KidRescue!](#)

[The creation of live programming in App Inventor](#)

[More...](#)

Tweets

 21 Sep

 **Arjun S**
@Arjuninventor

Wish I was able to attend this great event with iSafeGuard - my women safety app @MITAppInventor #MITAppInvSummit pic.twitter.com/A8d5AWKFz0

Retweeted by LateralLogics



Get Started



Follow these simple steps to build your first app.

[Start](#)

Tutorials



Step-by-step guides show you how to build all kinds of apps.

[Tutorials](#)

Teach



Teachers, find out about curriculum and teaching

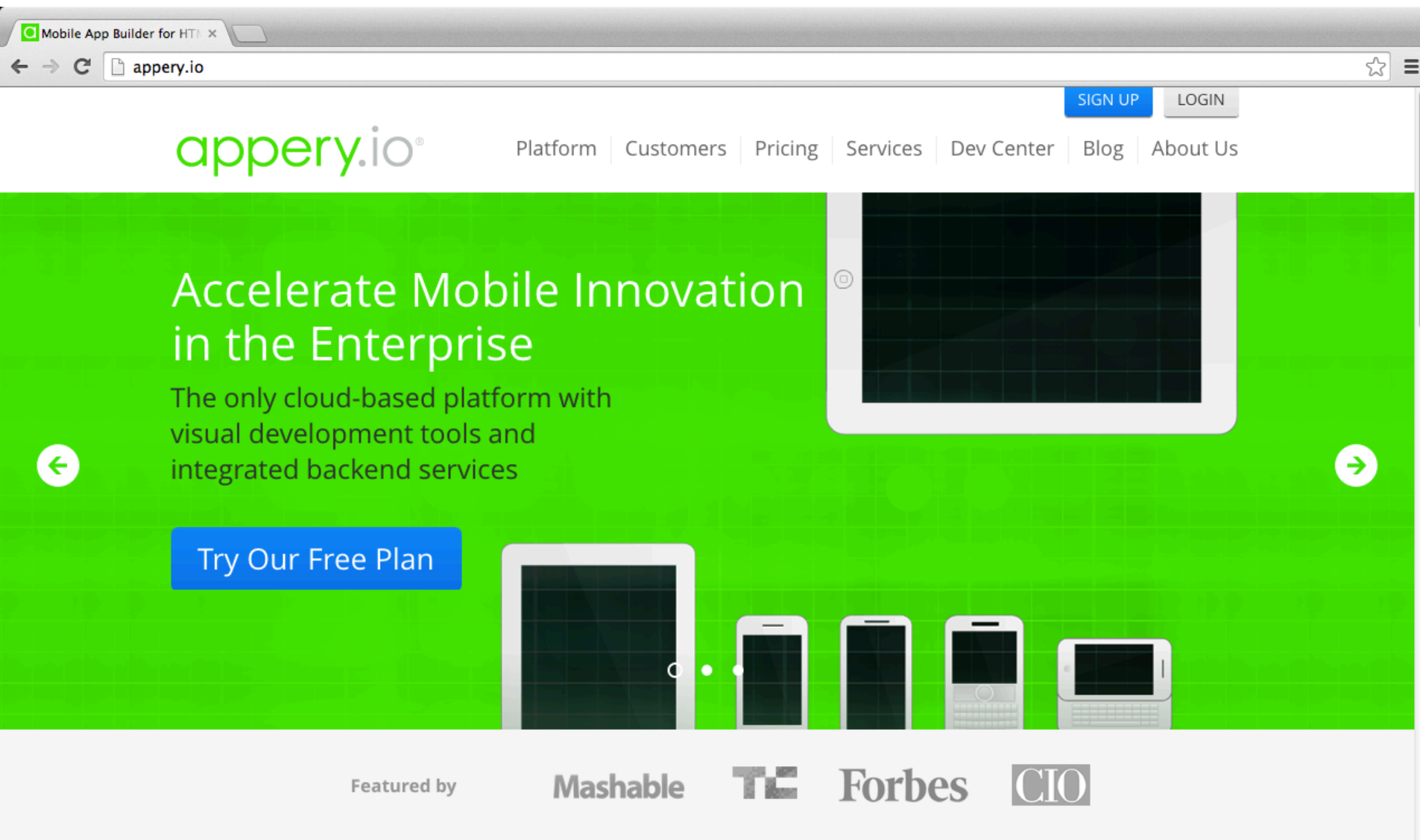
Forums



Join community forums to get answers to your questions

<http://appinventor.mit.edu/>

appery.io



The screenshot shows the appery.io website homepage. At the top, there's a browser window with the address bar showing 'appery.io'. The website header includes the 'appery.io' logo, navigation links for 'Platform', 'Customers', 'Pricing', 'Services', 'Dev Center', 'Blog', and 'About Us', and buttons for 'SIGN UP' and 'LOGIN'. The main content area has a green background with the text 'Accelerate Mobile Innovation in the Enterprise' and 'The only cloud-based platform with visual development tools and integrated backend services'. A blue button labeled 'Try Our Free Plan' is present. Below this, there are illustrations of various mobile devices (tablet, smartphones, and a smartwatch). At the bottom, a section titled 'Featured by' lists logos for Mashable, TechCrunch, Forbes, and CIO.

Mobile App Builder for HTML5

appery.io

SIGN UP LOGIN

appery.io®

Platform Customers Pricing Services Dev Center Blog About Us

Accelerate Mobile Innovation in the Enterprise

The only cloud-based platform with visual development tools and integrated backend services

Try Our Free Plan

Featured by

Mashable TechCrunch Forbes CIO

appery.io

The screenshot displays the Appery.io web interface for editing a mobile application named 'iMydayApp'. The browser address bar shows the URL: `appery.io/app/project/f044ff9d-24e0-4f68-9962-f855b35bff3e/editor`. The interface is divided into several sections:

- Left Sidebar:** Contains a 'CREATE NEW' button and a tree view of the project structure, including 'Project', 'App settings', 'Model and Storage', 'Pages' (with 'Screen1' and 'startScreen'), 'Dialogs', 'Templates', 'Themes', 'CSS', 'Services', 'JavaScript', and 'Custom components'.
- Top Bar:** Includes the 'appery.io' logo, 'CLOSE', 'SAVE', 'EXPORT', and 'TEST' buttons, along with status indicators for 'Database', 'Push', 'Server Code', and user counts.
- Design Palette:** A central panel with 'DESIGN' and 'DATA' tabs. The 'DESIGN' tab is active, showing a 'PALETTE' of controls such as Button, Group Buttons, Input, Textarea, Datepicker, Label, Link, Radio, Checkbox, Slider, Toggle, Select, List, Image, Grid, and Navbar.
- Canvas:** The main workspace showing a mobile device mockup. A green header bar at the top of the mockup is labeled 'Caption'. The title bar above the mockup reads 'Screen1 / mobileheader'.
- Properties Panel:** Located on the right, it shows settings for the selected 'Header' element. It includes sections for 'OPERATIONS' (Zoom Level set to 100), 'CUSTOM' (Text set to 'Caption', Swatch set to 'Inherit', Back Button set to 'Back', Position set to 'fixed'), and 'COMMON' (Name set to 'mobileheader', Class Name set to 'Enter Class Name'). A 'More properties' link is also visible.

<http://appery.io/>

Appnotch

Mobile App Builder, Develop

www.appnotch.com

☆

≡

 **APPNOTCH**

Home Tour Support Pricing Features **Login**

Easiest way to create Apps.

- ✓ Build HTML5, iOS and Android apps.
- ✓ Just drag & drop. No coding required.
- ✓ Easy. Professional. Powerful.
- ✓ Nothing to install. Build on Cloud.
- ✓ Get your live app in 3 easy steps.

Start Building for Free!





See in Your Phone

Build in Cloud

Reasons AppNotch is getting popular

Transform your Ideas to Apps !

Build interactive mockups, wireframes and clickable working prototypes in HTML5, Android, iPhone and iPad.

Did we say this is 100% free?

Start Prototyping

Beautiful, Flexible and Powerful !

Create topnotch apps with drag & drop templates, themes, widgets, controls, forms with instant hosting.

Did we say this is affordably priced?

Start Exploring

Personal Tour and Private Demo !

We'll show you an addictively easy way to create beautiful web, native and hybrid apps, without coding.

Did we say you will love the magic?

Schedule a Demo

<http://www.appnotch.com/>

95

References

- Android Developer: <http://developer.android.com/>
- MIT App Inventor: <http://appinventor.mit.edu/>
- Native, Web or Hybrid Mobile Apps?,
<https://www.youtube.com/watch?v=Ns-JS4amITc>