Social Media Apps Programming

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

1051SMAP03
TLMXM1A (8648) (M2143) (Fall 2016)
(MIS MBA) (2 Credits, Elective) [Full English Course]
Wed 8,9 (15:10-17:00) B310

Min-Yuh Day, Ph.D.
Assistant Professor
Department of Information Management
Tamkang University

http://mail.tku.edu.tw/myday
2016-09-28
<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2016/09/14</td>
<td>Course Orientation and Introduction to Social Media and Mobile Apps Programming</td>
</tr>
<tr>
<td>2</td>
<td>2016/09/21</td>
<td>Introduction to Android / iOS Apps Programming</td>
</tr>
<tr>
<td>3</td>
<td>2016/09/28</td>
<td>Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)</td>
</tr>
<tr>
<td>4</td>
<td>2016/10/05</td>
<td>Developing iPhone / iPad Native Apps with Swift (XCode)</td>
</tr>
<tr>
<td>5</td>
<td>2016/10/12</td>
<td>Mobile Apps using HTML5/CSS3/JavaScript</td>
</tr>
<tr>
<td>6</td>
<td>2016/10/19</td>
<td>jQuery Mobile</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>2016/10/26</td>
<td>Create Hybrid Apps with Phonegap</td>
</tr>
<tr>
<td>8</td>
<td>2016/11/02</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>9</td>
<td>2016/11/09</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>10</td>
<td>2016/11/16</td>
<td>Midterm Exam Week (Midterm Project Report)</td>
</tr>
<tr>
<td>11</td>
<td>2016/11/23</td>
<td>Case Study on Social Media Apps Programming and Marketing in Google Play and App Store</td>
</tr>
<tr>
<td>12</td>
<td>2016/11/30</td>
<td>Invited Speaker: Prof. Yoshinobu Kano, Associate Professor, Faculty of Informatics, Shizuoka University</td>
</tr>
</tbody>
</table>
# Course Schedule (3/3)

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>2016/12/07</td>
<td>Google Cloud Platform</td>
</tr>
<tr>
<td>14</td>
<td>2016/12/14</td>
<td>Google App Engine and Google Map API</td>
</tr>
<tr>
<td>15</td>
<td>2016/12/21</td>
<td>Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)</td>
</tr>
<tr>
<td>16</td>
<td>2016/12/28</td>
<td>Twitter API</td>
</tr>
<tr>
<td>17</td>
<td>2017/01/04</td>
<td>Final Project Presentation</td>
</tr>
<tr>
<td>18</td>
<td>2017/01/11</td>
<td>Final Exam Week (Final Project Presentation)</td>
</tr>
</tbody>
</table>
Android / iOS Apps Programming

Native Apps

Hybrid Apps

Mobile Web Apps
App Development Comparison

- **Native Apps**
  - Device Access: Full
  - Speed: Very Fast
  - Development Cost: Expensive
  - App Store: Available
  - Approval Process: Mandatory

- **Hybrid Apps**
  - Device Access: Full
  - Speed: Native Speed as Necessary
  - Development Cost: Reasonable
  - App Store: Available
  - Approval Process: Low Overhead

- **Web Apps**
  - Device Access: Partial
  - Speed: Fast
  - Development Cost: Reasonable
  - App Store: Not Available
  - Approval Process: None

Outline

• Developing **Android Native Apps with Java**
  – Android Studio
  – Eclipse
  – Android Developer Tools (**ADT**) Bundle
  – Building Your First Android App

• **MIT App Inventor**
Native App Development

Android - Native App Development

Native App – Interaction with Mobile Device

Android

Powering screens of all sizes

Android is the customizable, easy to use operating system that powers more than a billion devices across the globe — from phones and tablets to watches, TV, cars and more to come.

http://www.android.com/
Android 5.1 Lollipop

Source: http://developer.android.com/about/versions/lollipop.html
## Android Platform

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>0.2%</td>
</tr>
<tr>
<td>2.3.3 - 2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>4.1%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>3.7%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>12.1%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>15.2%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>4.5%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>39.2%</td>
</tr>
<tr>
<td>5.0</td>
<td>Lollipop</td>
<td>21</td>
<td>15.9%</td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td>22</td>
<td>5.1%</td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on September 7, 2015.

Data collected during a 7-day period ending on September 7, 2015.

http://developer.android.com/about/dashboards/index.html
Android Screen Sizes and Densities

<table>
<thead>
<tr>
<th></th>
<th>ldpi</th>
<th>mdpi</th>
<th>tvdpi</th>
<th>hdpi</th>
<th>xhdpi</th>
<th>xxhdpi</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>3.3%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.3%</td>
</tr>
<tr>
<td>Normal</td>
<td>6.5%</td>
<td>0.1%</td>
<td></td>
<td>41.0%</td>
<td>20.9%</td>
<td></td>
<td>84.1%</td>
</tr>
<tr>
<td>Large</td>
<td>0.3%</td>
<td>4.8%</td>
<td>2.3%</td>
<td>0.6%</td>
<td>0.6%</td>
<td></td>
<td>8.6%</td>
</tr>
<tr>
<td>Xlarge</td>
<td>3.0%</td>
<td></td>
<td>0.3%</td>
<td>0.7%</td>
<td></td>
<td></td>
<td>4.0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3.6%</strong></td>
<td><strong>14.3%</strong></td>
<td><strong>2.4%</strong></td>
<td><strong>41.9%</strong></td>
<td><strong>22.2%</strong></td>
<td><strong>15.6%</strong></td>
<td></td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on September 7, 2015.

http://developer.android.com/about/dashboards/index.html
Android Screen Sizes and Densities

http://developer.android.com/guide/practices/screens_support.html
Android Development Environment

1. **JDK**
   (Java Development Kit)
   

2. **Android Studio**
   
1 JDK (Java Development Kit)

Android Studio

- Android Studio IDE
- Android SDK tools
- Android 5.0 (Marshmallow) Platform
- Android 6.0 emulator system image with Google APIs

DOWNLOAD ANDROID STUDIO FOR MAC

- System Requirements
- Other Download Options
- Migrating to Android Studio
- Take a Survey

Android Development Environment

1. JDK  
   (Java Development Kit)  
   [Link to JDK Download](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)  
   [Link to ADT Bundle](http://developer.android.com/sdk/index.html)
1. JDK (Java Development Kit)

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 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.

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

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

Android Developer Tools

Build: v22.2.1-833290

This product includes Eclipse Platform, JDT, CDT, EMF, GEF and WTP, all of which are Copyright (c) Eclipse contributors and others. Visit http://eclipse.org/

Android Developer Tools are Copyright (c) The Android Open Source Project. Visit http://developer.android.com
Get the Android SDK

Eclipse ADT Bundle

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 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 >

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 adk`.

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:
   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:
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 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:

<table>
<thead>
<tr>
<th>Get started</th>
<th>Build for wearables</th>
<th>Use Google APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you're new to Android development, learn the basics of Android apps by following the guide to <a href="http://developer.android.com/sdk/installing/adding-packages.html">Building Your First App.</a></td>
<td>If you're ready to start building apps for Android wearables, see the guide to <a href="http://developer.android.com/sdk/installing/adding-packages.html">Building Apps for Android Wear.</a></td>
<td>To start using Google APIs, such as Maps or Play Game services, see the guide to <a href="http://developer.android.com/sdk/installing/adding-packages.html">Setting Up Google Play Services.</a></td>
</tr>
</tbody>
</table>
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**
Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

Android App Building Blocks

1. **Activity**
   - activated by an asynchronous message

2. **Service**

3. **Broadcast Receiver**

4. **Content Provider**

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

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Building a Simple User Interface

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Building a Simple User Interface

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Android App Activity Lifecycle

Source:
The development process for Android applications

Setup
- **Set up your development environment**
  - Install the Android SDK, Android Development Tools, and Android platforms.

Development
- **Create your application**
  - Create an Android project with your source code, resource files, and Android manifest file.

Debugging and Testing
- **Build and run your application**
  - Build and run your application in debug mode.
- **Debug your application**
  - Debug your application using the Android debugging and logging tools.
- **Test your application**
  - Test your application using the Android testing and instrumentation framework.

Publishing
- **Prepare your application for release**
  - Configure, build, and test your application in release mode.
- **Release your application**
  - Publicize, sell, and distribute your application to users.

1. Setup

- Set up your development environment
- Install the Android SDK, Android Development Tools, and Android platforms.
- Set up AVDs and devices for testing
- Create Android Virtual Devices and connect hardware devices that will be used for testing.
2. Development

Create your application

Create an Android project with your source code, resource files, and Android manifest file.
3. Debugging and Testing

- Build and run your application
- Debug your application using the Android debugging and logging tools
- Test your application using the Android testing and instrumentation framework.
4. Publishing

- **Publishing**
  - Prepare your application for release
  - Configure, build, and test your application in release mode.
  - Release your application
  - Publicize, sell, and distribute your application to users.
Demo:
Building Your First Android App with Android Studio
My First Android App
Java SE Development Kit (JDK)

Java SE Development Kit (JDK)

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:
- Java Developer Newsletter (ticked the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u60 Checksum

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

<table>
<thead>
<tr>
<th>Product / File Description</th>
<th>File Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux ARM v6/v7 Hard Float ABI</td>
<td>77.59 MB</td>
<td>jdk-8u60-linux-arm32-xf86-elf.tar.gz</td>
</tr>
<tr>
<td>Linux ARM v8 Hard Float ABI</td>
<td>74.84 MB</td>
<td>jdk-8u60-linux-arm64-xf86-elf.tar.gz</td>
</tr>
<tr>
<td>Linux x86</td>
<td>154.55 MB</td>
<td>jdk-8u60-linux-x86_64.rpm</td>
</tr>
<tr>
<td>Linux x64</td>
<td>174.83 MB</td>
<td>jdk-8u60-linux-x86_64.tar.gz</td>
</tr>
<tr>
<td>Mac OS X x64</td>
<td>152.97 MB</td>
<td>jdk-8u60-macosx-x64.dmg</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit (SparcOS package)</td>
<td>172.84 MB</td>
<td>jdk-8u60-solaris-x86_64.tar.gz</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit</td>
<td>227.07 MB</td>
<td>jdk-8u60-solaris-sparcv9.tar.gz</td>
</tr>
<tr>
<td>Solaris x64 (SparcOS package)</td>
<td>393.02 MB</td>
<td>jdk-8u60-solaris-x64.tar.gz</td>
</tr>
<tr>
<td>Solaris x64</td>
<td>139.07 MB</td>
<td>jdk-8u60-solaris-x86_64.tar.gz</td>
</tr>
<tr>
<td>Windows x86</td>
<td>193.16 MB</td>
<td>jdk-8u60-windows-i586.exe</td>
</tr>
<tr>
<td>Windows x64</td>
<td>185.15 MB</td>
<td>jdk-8u60-windows-x86_64.exe</td>
</tr>
</tbody>
</table>
Java SE Development Kit (JDK)

Java SE Development Kit 8 Downloads
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:
- Java Developer Newsletter (click the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u60 Checksum

Java SE Development Kit 8u60
You must accept the Oracle Binary Code License Agreement for Java SE to download this software. Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

<table>
<thead>
<tr>
<th>Product / File Description</th>
<th>File Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux ARM v6/v7 Hard Float ABI</td>
<td>77.69 MB</td>
<td>jdk-8u60-linux-arm32-vfp-hft.tar.gz</td>
</tr>
<tr>
<td>Linux ARM v8 Hard Float ABI</td>
<td>74.64 MB</td>
<td>jdk-8u60-linux-arm64-vfp-hft.tar.gz</td>
</tr>
<tr>
<td>Linux x86</td>
<td>154.66 MB</td>
<td>jdk-8u60-linux-x86.tar.gz</td>
</tr>
<tr>
<td>Linux x864</td>
<td>174.83 MB</td>
<td>jdk-8u60-linux-x864.tar.gz</td>
</tr>
<tr>
<td>Linux x64</td>
<td>152.67 MB</td>
<td>jdk-8u60-linux-x64.tar.gz</td>
</tr>
<tr>
<td>Linux x644</td>
<td>172.84 MB</td>
<td>jdk-8u60-linux-x64.tar.gz</td>
</tr>
<tr>
<td>Mac OS X x64</td>
<td>227.07 MB</td>
<td>jdk-8u60-macosx-x64.tar.gz</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit (SVR4 package)</td>
<td>139.67 MB</td>
<td>jdk-8u60-solaris-sparcv9.tar.Z</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit</td>
<td>99.02 MB</td>
<td>jdk-8u60-solaris-sparcv9.tar.Z</td>
</tr>
<tr>
<td>Solaris x64 (SVR4 package)</td>
<td>140.18 MB</td>
<td>jdk-8u60-solaris-x64.tar.Z</td>
</tr>
<tr>
<td>Solaris x64</td>
<td>96.71 MB</td>
<td>jdk-8u60-solaris-x64.tar.Z</td>
</tr>
<tr>
<td>Windows x86</td>
<td>180.82 MB</td>
<td>jdk-8u60-windows-x86.exe</td>
</tr>
<tr>
<td>Windows x64</td>
<td>186.16 MB</td>
<td>jdk-8u60-windows-x64.exe</td>
</tr>
</tbody>
</table>
Java SE Development Kit (JDK)

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:
- Java Developer Newsletter (click the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

Java SE Development Kit 8u60

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

<table>
<thead>
<tr>
<th>Product / File Description</th>
<th>File Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux ARM v6/v7 Hard Float ABI</td>
<td>77.69 MB</td>
<td>jdk-8u60-linux-armv7-vfp-hft.tar.gz</td>
</tr>
<tr>
<td>Linux ARM v6 Hard Float ABI</td>
<td>74.64 MB</td>
<td>jdk-8u60-linux-armv6-float.tar.gz</td>
</tr>
<tr>
<td>Linux x86</td>
<td>154.86 MB</td>
<td>jdk-8u60-linux-i586.rpm</td>
</tr>
<tr>
<td>Linux x86</td>
<td>174.83 MB</td>
<td>jdk-8u60-linux-x86.tar.gz</td>
</tr>
<tr>
<td>Linux x64</td>
<td>152.67 MB</td>
<td>jdk-8u60-linux-x64.rpm</td>
</tr>
<tr>
<td>Linux x64</td>
<td>172.84 MB</td>
<td>jdk-8u60-linux-x64.iar.gz</td>
</tr>
<tr>
<td>Mac OS X x64</td>
<td>227.07 MB</td>
<td>jdk-8u60-macosx-x64.dmg</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit (SVR4 package)</td>
<td>139.67 MB</td>
<td>jdk-8u60-solaris-sparcv9.tar.Z</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit (SVR4 package)</td>
<td>99.02 MB</td>
<td>jdk-8u60-solaris-sparcv9.tar.gz</td>
</tr>
<tr>
<td>Sun Solaris UltraSPARC 64-bit (SVR4 package)</td>
<td>140.18 MB</td>
<td>jdk-8u60-solaris-ultrasparc-v9.tar.Z</td>
</tr>
<tr>
<td>Sun SPARC Ultra64 64-bit (SVR4 package)</td>
<td>96.71 MB</td>
<td>jdk-8u60-solaris-ultrasparc-v9.tar.gz</td>
</tr>
</tbody>
</table>
Java Development Kit

Double-click on icon to install

JDK 8 Update 60.pkg
Install JDK 8

Welcome to the JDK 8 Update 60 Installer

The Java Development Kit is a development environment for building applications, applets, and components using the Java programming language.

The Java Mission Control profiling and diagnostic tools suite is now available as part of JDK.
Install JDK 8

This will take 533.4 MB of space on your computer.

Click Install to perform a standard installation of this software for all users of this computer. All users of this computer will be able to use this software.
Install JDK 8

This will take 533.4 MB of space on your computer.

Click Install to perform a standard installation of this software for all users of this computer. All users of this computer will be able to use this software.

Installer is trying to install new software. Type your password to allow this.
The installation was completed successfully.

Access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

Next Steps?
Android Studio

The official Android IDE

- Android Studio IDE
- Android SDK tools
- Android 5.0 (Marshmallow) Platform
- Android 6.0 emulator system image with Google APIs

DOWNLOAD ANDROID STUDIO FOR MAC

- System Requirements
- Other Download Options
- Migrating to Android Studio
- Take a Survey

Android Studio

- Android Studio IDE
- Android SDK tools
- Android 6.0 (Marshmallow) Platform
- Android 6.0 emulator system image with Google APIs

DOWNLOAD ANDROID STUDIO FOR MAC
Android Studio

Download

Before installing Android Studio or the standalone SDK tools, 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.

I have read and agree with the above terms and conditions

DOWNLOAD ANDROID STUDIO FOR MAC
Installing Android Studio

Android Studio provides everything you need to start developing apps for Android, including the Android Studio IDE and the Android SDK tools.

If you didn't download Android Studio, go download Android Studio now, or switch to the stand-alone SDK Tools install instructions.

Before you set up Android Studio, be sure you have installed JDK 6 or higher (the JRE alone is not sufficient)—JDK 7 is required when developing for Android 5.0 and higher. To check if you have JDK installed (and which version), open a terminal and type `javac -version`. If the JDK is not available or the version is lower than 6, go download JDK.

To set up Android Studio on Mac OSX:

1. Launch the .dmg file you just downloaded.
2. Drag and drop Android Studio into the Applications folder.
3. Open Android Studio and follow the setup wizard to install any necessary SDK tools.

Depending on your security settings, when you attempt to open Android Studio, you might see a warning that says the package is damaged and should be moved to the trash. If this happens, go to System Preferences > Security & Privacy and under Allow applications downloaded from, select Anywhere. Then open Android Studio again.
Installing Android Studio

Android Studio provides everything you need to start developing apps for Android, including the Android Studio IDE and the Android SDK tools.

If you didn’t download Android Studio, go download Android Studio now, or switch to the stand-alone SDK Tools install instructions.

Before you set up Android Studio, be sure you have installed JDK 6 or higher (the JRE alone is not sufficient)—JDK 7 is required when developing for Android 5.0 and higher. To check if you have JDK installed (and which version), open a terminal and type `javac -version`. If the JDK is not available or the version is lower than 6, go download JDK.

To set up Android Studio on Mac OSX:

1. Launch the .dmg file you just downloaded.
2. Drag and drop Android Studio into the Applications folder.
3. Open Android Studio and follow the setup wizard to install any necessary SDK tools.

   Depending on your security settings, when you attempt to open Android Studio, you might see a warning that says the package is damaged and should be moved to the trash. If this happens, go to System Preferences > Security & Privacy and under Allow applications downloaded from, select Anywhere. Then open Android Studio again.

If you need use the Android SDK tools from a command line, you can access them at:

```
/Users/<user>/Library/Android/sdk/
```

Android Studio is now ready and loaded with the Android developer tools, but there are still a couple packages you should...
Android Studio
Welcome! This wizard will set up your development environment for Android Studio. Additionally, the wizard will help port existing Android apps into Android Studio or create a new Android application project.
Choose the type of setup you want for Android Studio:

- **Standard**
  - Android Studio will be installed with the most common settings and options.
  - Recommended for most users.

- **Custom**
  - You can customize installation settings and components installed.
If you want to review or change any of your installation settings, click Previous.

Current Settings:

<table>
<thead>
<tr>
<th>Total Download Size:</th>
<th>752 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sdk Components to Download:</strong></td>
<td></td>
</tr>
<tr>
<td>Android SDK Build-tools, revision 23.0.1</td>
<td>36.3 MB</td>
</tr>
<tr>
<td>Android SDK Platform-tools, revision 23.0.1</td>
<td>2.37 MB</td>
</tr>
<tr>
<td>Android SDK Tools, revision 24.4.0</td>
<td>97.1 MB</td>
</tr>
<tr>
<td>Android Support Repository, revision 20</td>
<td>131 MB</td>
</tr>
<tr>
<td>Google APIs Intel x86 Atom System Image, Google Inc. API 23, revision 7</td>
<td>331 MB</td>
</tr>
<tr>
<td>Google APIs, Android API 23, revision 1</td>
<td>176 KB</td>
</tr>
<tr>
<td>Google Repository, revision 22</td>
<td>56.7 MB</td>
</tr>
<tr>
<td>Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0</td>
<td>218 KB</td>
</tr>
<tr>
<td>SDK Platform Android 6.0, API 23, revision 1</td>
<td>67.1 MB</td>
</tr>
<tr>
<td>Sources for Android SDK, API 23, revision 1</td>
<td>30.3 MB</td>
</tr>
</tbody>
</table>
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (1%)
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (94%)
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (95%)
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (99%)
Installed Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0
Installing Android SDK Platform-tools, revision 23.0.1
Stopping ADB server failed (code -1).
  Installed Android SDK Platform-tools, revision 23.0.1
Installing Sources for Android SDK, API 23, revision 1
  Installed Sources for Android SDK, API 23, revision 1
Installing Google APIs Intel x86 Atom System Image, Google Inc. API 23, revision 7
  Installed Google APIs Intel x86 Atom System Image, Google Inc. API 23, revision 7
Installing Android SDK Tools, revision 24.4.0
  Installed Android SDK Tools, revision 24.4.0
  Updated ADB to support the USB devices declared in the SDK add-ons.
Stopping ADB server succeeded.
Starting ADB server succeeded.
Done. 10 packages installed.
Android SDK is up to date.
Running Intel® HAXM installer
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (1%)
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (94%)
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (95%)
Unzipping Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0 (99%)
Installed Intel x86 Emulator Accelerator (HAXM installer), revision 5.4.0
Installing Android SDK Platform-tools, revision 23.0.1
Stopping ADB server failed (code -1).
  Installed Android SDK Platform-tools, revision 23.0.1
Installing Sources for Android SDK, API 23, revision 1
  Installed Sources for Android SDK, API 23, revision 1
Installing Google APIs Intel x86 Atom System Image, Google Inc. API 23, revision 7
  Installed Google APIs Intel x86 Atom System Image, Google Inc. API 23, revision 7
Installing Android SDK Tools, revision 24.4.0
  Installed Android SDK Tools, revision 24.4.0
  Updated ADB to support the USB devices declared in the SDK add-ons.
  Stopping ADB server succeeded.
  Starting ADB server succeeded.
Done. 10 packages installed.
Android SDK is up to date.
Running Intel® HAXM installer
Silent installation Pass!
Creating Android virtual device
Android virtual device Nexus_5_API_23_x86 was successfully created
Welcome to Android Studio

Recent Projects

Quick Start

- Start a new Android Studio project
- Open an existing Android Studio project
- Import an Android code sample
- Check out project from Version Control
- Import project (Eclipse ADT, Gradle, etc.)
- Configure
- Docs and How-Tos
SDK Location

Android SDK location:
The directory where the Android SDK is located. This location will be used for new projects, and for existing projects that do not have a local.properties file with a sdk.dir property.

/Users/imyday/Library/Android/sdk

JDK location:
The directory where the Java Development Kit (JDK) is located.

/Library/Java/JavaVirtualMachines/jdk1.8.0_60.jdk/Contents/Home

Android NDK location:
The directory where the Android NDK is located. This location will be saved as ndk.dir property in the local.properties file.

Download Android NDK.
### Welcome to Android Studio

#### Recent Projects

- No Project Open Yet

#### Quick Start

1. **Start a new Android Studio project**
2. **Open an existing Android Studio project**
3. **Import an Android code sample**
4. **Check out project from Version Control**
5. **Import project (Eclipse ADT, Gradle, etc.)**
6. **Configure**
7. **Docs and How-Tos**

---

*Android Studio 1.3.2 Build 141.2178183. Check for updates now.*
Configure your new project

Application name: My Application

Company Domain: myday.example.com

Package name: com.example.myday.myapplication

Project location: /Users/myday/AndroidStudioProjects/MyApplication
Configure your new project

Application name: MyFirstAndroidApp
Company Domain: imyday.example.com
Package name: com.example.imyday.myfirstandroidapp
Project location: /Users/imyday/AndroidStudioProjects/MyFirstAndroidApp
# Select the form factors your app will run on

Different platforms may require separate SDKs

<table>
<thead>
<tr>
<th>Form Factor</th>
<th>Minimum SDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone and Tablet</td>
<td>API 15: Android 4.0.3 (IceCreamSandwich)</td>
</tr>
<tr>
<td>Wear</td>
<td>API 21: Android 5.0 (Lollipop)</td>
</tr>
<tr>
<td>TV</td>
<td>API 21: Android 5.0 (Lollipop)</td>
</tr>
<tr>
<td>Android Auto</td>
<td></td>
</tr>
<tr>
<td>Glass (Not Installed)</td>
<td>Download</td>
</tr>
</tbody>
</table>

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately 94.0% of the devices that are active on the Google Play Store. [Help me choose](#)
Select the form factors your app will run on

Different platforms may require separate SDKs

- **Phone and Tablet**
  - Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)
  - API 15: Android 4.0.3 (IceCreamSandwich)
  - API 16: Android 4.1 (Jelly Bean)
  - API 17: Android 4.2 (Jelly Bean)
  - API 18: Android 4.3 (Jelly Bean)
  - API 19: Android 4.4 (KitKat)
  - API 21: Android 5.0 (Lollipop)
  - API 22: Android 5.1 (Lollipop)
  - API 23: Android 6.0 (Marshmallow)

- **Wear**
  - Minimum SDK

- **TV**
  - Minimum SDK: API 21: Android 5.0 (Lollipop)

- **Android Auto**

- **Glass (Not Installed)**

Minimum SDK
Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

- **Phone and Tablet**
  - Minimum SDK: API 8: Android 2.2 (Froyo)
  - Wear
    - Minimum SDK
  - TV
  - Android Auto
  - Glass (Not Installed)
    - Minimum SDK

---

Download

---

Cancel  Previous  Next  Finish
Add an activity to Mobile

- Add No Activity
- Blank Activity
- Blank Activity with Fragment
- Fullscreen Activity
- Google AdMob Ads Activity
- Google Maps Activity
- Google Play Services Activity
- Login Activity
- Master/Detail Flow
- Navigation Drawer Activity
Creates a new blank activity with an action bar.

Activity Name: MainActivity
Layout Name: activity_main
Title: MainActivity
Menu Resource Name: menu_main

The name of the activity class to create
Hello world!
package com.example.imyday.myfirstandroidapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    @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.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        // Handle action bar item clicks here.
        // The action bar will automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
<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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".MainActivity">

    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"/>

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:layout_alignBaseline="@id/textView"
        android:layout_alignBottom="@id/textView"
        android:layout_centerHorizontal="true"/>

</RelativeLayout>
Edit translations for all locales in the translations editor.

```xml
<resources>
  <string name="app_name">MyFirstAndroidApp</string>
  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>
</resources>
```
<resources>
  <string name="app_name">MyFirstAndroidApp</string>
  <string name="hello_world">Hello world Myday!</string>
  <string name="action_settings">Settings</string>
</resources>
Edit translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">MyFirstAndroidApp</string>
    <string name="hello_world">Hello world Myday!</string>
    <string name="action_settings">Settings</string>
</resources>
```
Edit translations for all locales in the translations editor.

```
<resources>
  <string name="app_name">MyFirstAndroidApp</string>
  <string name="hello_world">Hello world Myday!</string>
  <string name="action_settings">Settings</string>
</resources>
```
My First Android App
Demo:
Building Your First Android App with Android Developer Tools (ADT)
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

Except as noted, this content is licensed under Creative Commons Attribution 2.5. For details and restrictions, see the Content License.
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](https://developer.android.com/studio/index.html) before you start this lesson.

Create a Project with Eclipse

1. Click **New** in the toolbar.
2. In the window that appears, open the **Android** folder, select **Android Application Project**, and click **Next**.
3. 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.
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

Minimum Required SDK: API 8: Android 2.2 (Froyo)
Target SDK: API 18: Android 4.3 (Jelly Bean)
Compile With: API 18: Android 4.3 (Jelly Bean)
Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.
New Android Application

Creates a new Android Application

- Application Name: My First App
- Project Name: MyFirstApp
- Package Name: tw.edu.tku.im.smap2013.imyday.myfirstapp
- Minimum Required SDK: API 8: Android 2.2 (Froyo)
- Target SDK: API 18: Android 4.3 (Jelly Bean)
- Compile With: API 18: Android 4.3 (Jelly Bean)
- 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.
New Android Application
Creates a new Android Application

Application Name: My First App
Project Name: MyFirstApp
Package Name: tw.edu.tku.im.smap2013.imyday.myfirstapp
Minimum Required SDK: API 8: Android 2.2 (Froyo)
Target SDK: None
Compile With: Holo Dark, Holo Light
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.
Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Activity Name: MainActivity
Layout Name: activity_main
Navigation Type: None

The name of the activity class to create

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

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
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
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;
}
}
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.menu_main, menu);
        return true;
    }
}
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.menu_activity_main, menu);
        return true;
    }
}

Hello world!
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.

http://developer.android.com/training/basics/firstapp/starting-activity.html
public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myfirstapp";

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    public void onMessageClicked(View view) {
        String message = editText.getText().toString();
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}

Hello World Myday
Hello World Myday
<?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>
<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>
<?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>
<?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>
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 {
    
    @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);

    }

    ...
}
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);
    }

}
/**
 * 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
  – 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/
Accelerate Mobile Innovation in the Enterprise

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

Try Our Free Plan

http://appezy.io/
http://appery.io/
Easiest way to create Apps.

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

Start Building for Free

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/
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-JS4amlTc