***Course: 305-02: Mobile Application Development - 1***

***Unit-2: Setting up Android Environment:***

*2.1 Android Emulator*

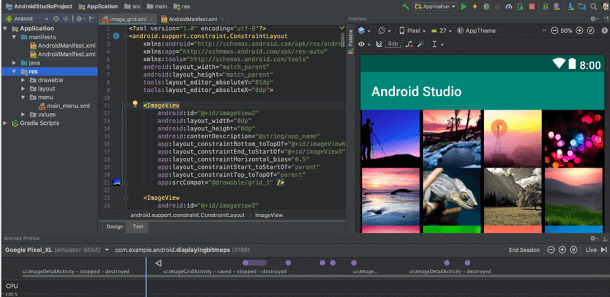
*2.1.1 Setting up JDK and Android Studio*

*Android Studio*

*Android Studio is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.*

*Android Studio was announced on 16th May 2013 at the Google I/O conference as an official IDE for Android app development. It started its early access preview from version 0.1 in May 2013. The first stable built version was released in December 2014, starts from version 1.0.*

*Since 7th May 2019, Kotlin is Google's preferred language for Android application development. Besides this, other programming languages are supported by Android Studio.*



***Features of Android Studio***

* *It has a flexible Gradle-based build system.*
* *It has a fast and feature-rich emulator for app testing.*
* *Android Studio has a consolidated environment where we can develop for all Android devices.*
* *Apply changes to the resource code of our running app without restarting the app.*
* *Android Studio provides extensive testing tools and frameworks.*
* *It supports C++ and NDK.*
* *It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google Cloud Messaging and App Engine.*

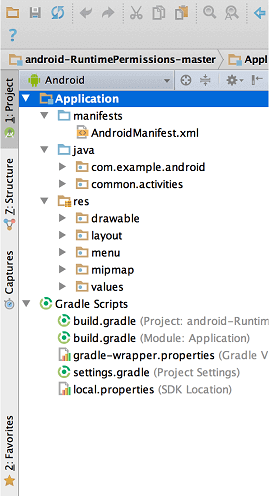
***Android Studio Version History***

|  |  |
| --- | --- |
| **Version** | **Release date** |
| 1.0 | December 2014 |
| 1.1 | February 2015 |
| 1.2 | April 2015 |
| 1.3 | July 2015 |
| 1.4 | September 2015 |
| 1.5 | November 2015 |
| 2.0 | April 2016 |
| 2.1 | April 2016 |
| 2.2 | September 2016 |
| 2.3 | March 2017 |
| 3.0 | October 2017 |
| 3.1 | March 2018 |
| 3.2 | September 2018 |
| 3.3 | January 2019 |
| 3.4 | April 2019 |
| 3.5 | August 2019 |

***Android Studio Project Structure***

*The Android Studio project contains one or more modules with resource files and source code files. These include different types of modules-*

* *Android app modules*
* *Library modules*
* *Google App Engine modules*



*By default, Android Studio displays our project files in the Android project view, as shown in the above image. This view is formed by modules to provide quick access to our project's key source files.*

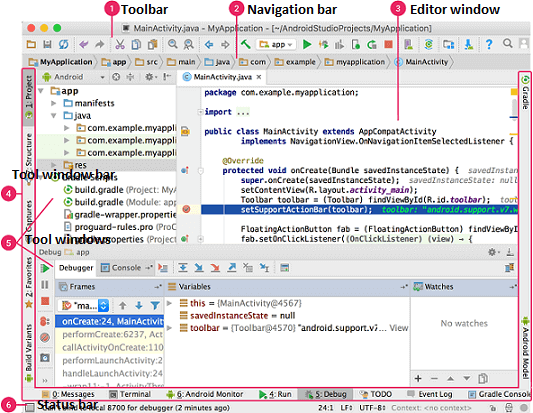
*These build files are visible to the top-level under Gradle Scripts. And the app module contains the following folders:*

* ***manifests****: It contains the AndroidManifest.xml file.*
* ***java****: It contains the source code of Java files, including the JUnit test code.*
* ***res****: It contains all non-code resources, UI strings, XML layouts, and bitmap images.*

*We will see the actual file structure of the project by selecting the Project from the Project dropdown.*

***Android Studio User Interface***

*The Android Studio main window contains the several logical areas which are shown in the below figure:*



1. *The toolbar provides us a wide range of actions, which includes running apps and launching Android tools.*
2. *The navigation bar helps in navigating our project and open files for editing. It gives a compact view of structure visible in the Project window.*
3. *The editor window is a space where we can create and modify our code. On the basis of the current file type, the editor can change. While viewing a layout file, the editor displays the Layout Editor.*
4. *The tool window bar runs around the outside the IDE window and contains buttons that allow as to expand and collapse individual tool windows.*
5. *The tool windows provide us access specific tasks like search, project management, version control, and more. We can able expand and collapse them.*
6. *The status bar displays the status of our project and IDE itself, as well as any messages or warnings.*

*We are willing to organize the main window to give ourselves more screen space by moving or hiding toolbars and tool windows. We can also use keyboard shortcuts to access most of the IDE features.*

***Android Studio Tool window***

*We can use keyboard shortcuts to open tool windows. The below table provides the list of shortcuts for the most common windows.*

|  |  |  |
| --- | --- | --- |
| **Tool window** | **Windows and Linux** | **Mac** |
|  |  |  |
| Project | Alt+1 | Command+1 |
| Version Control | Alt+9 | Command+9 |
| Run | Shift+F10 | Control+R |
| Debug | Shift+F9 | Control+D |
| Logcat | Alt+6 | Command+6 |
| Return to Editor | Esc | Esc |
| Hide all Tool Windows | Control+Shift+F12 | Command+Shift+F12 |

***Gradle build system***

*Gradle build used as the foundation of the build system in Android Studio. It uses more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs independently from the command line and integrated tool from the Android Studio menu. We can use build features for the following purpose:*

* *Configure, customize, and extend the build process.*
* *We can create multiple APKs from our app, with different features using the same project and modules.*
* *Reuse resource and code across source sets.*

*2.1.2 Android SDK manager*

*Android tutorial or Android Studio tutorial covers basic and advanced concepts of android technology. Our Android development tutorial is developed for beginners and professionals.*

*Android is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc.*

*It contains a linux-based Operating System, middleware and key mobile applications.*

*It can be thought of as a mobile operating system. But it is not limited to mobile only. It is currently used in various devices such as mobiles, tablets, televisions etc.*

***What is Android?***

*Before learning all topics of android, it is required to know what is android.*

*Android is a software package and linux based operating system for mobile devices such as tablet computers and smartphones.*

*It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used.*

*The goal of android project is to create a successful real-world product that improves the mobile experience for end users.*

*There are many code names of android such as Lollipop, Kitkat, Jelly Bean, Ice cream Sandwich, Froyo, Ecliar, Donut etc which is covered in next page.*

***What is Open Handset Alliance (OHA)***

*It's a consortium of 84 companies such as google, samsung, AKM, synaptics, KDDI, Garmin, Teleca, Ebay, Intel etc.*

*It was established on 5th November, 2007, led by Google. It is committed to advance open standards, provide services and deploy handsets using the Android Plateform.*

***Features of Android***

*After learning what is android, let's see the features of android. The important features of android are given below:*

*1) It is open-source.*

*2) Anyone can customize the Android Platform.*

*3) There are a lot of mobile applications that can be chosen by the consumer.*

*4) It provides many interesting features like weather details, opening screen, live RSS (Really Simple Syndication) feeds etc.*

*It provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, Wi-Fi etc.), media, handset layout etc.*

***Categories of Android applications***

*There are many android applications in the market. The top categories are:*

* *Entertainment*
* *Tools*
* *Communication*
* *Productivity*
* *Personalization*
* *Music and Audio*
* *Social*
* *Media and Video*
* *Travel and Local etc.*

***History of Android***

*The history and versions of android are interesting to know. The code names of android ranges from A to J currently, such as Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwitch, Jelly Bean, KitKat and Lollipop. Let's understand the android history in a sequence.*

*1) Initially, Andy Rubin founded Android Incorporation in Palo Alto, California, United States in October, 2003.*

*2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.*

*3) The key employees of Android Incorporation are Andy Rubin, Rich Miner, Chris White and Nick Sears.*

*4) Originally intended for camera but shifted to smart phones later because of low market for camera only.*

*5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.*

*6) In 2007, Google announces the development of android OS.*

*7) In 2008, HTC launched the first android mobile.*

***Android Versions, Codename and API***

*Let's see the android versions, codenames and API Level provided by Google.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Code name** | **Version numbers** | **API level** | **Release date** |
| No codename | 1.0 | 1 | September 23, 2008 |
| No codename | 1.1 | 2 | February 9, 2009 |
| Cupcake | 1.5 | 3 | April 27, 2009 |
| Donut | 1.6 | 4 | September 15, 2009 |
| Eclair | 2.0 - 2.1 | 5 - 7 | October 26, 2009 |
| Froyo | 2.2 - 2.2.3 | 8 | May 20, 2010 |
| Gingerbread | 2.3 - 2.3.7 | 9 - 10 | December 6, 2010 |
| Honeycomb | 3.0 - 3.2.6 | 11 - 13 | February 22, 2011 |
| Ice Cream Sandwich | 4.0 - 4.0.4 | 14 - 15 | October 18, 2011 |
| Jelly Bean | 4.1 - 4.3.1 | 16 - 18 | July 9, 2012 |
| KitKat | 4.4 - 4.4.4 | 19 - 20 | October 31, 2013 |
| Lollipop | 5.0 - 5.1.1 | 21- 22 | November 12, 2014 |
| Marshmallow | 6.0 - 6.0.1 | 23 | October 5, 2015 |
| Nougat | 7.0 | 24 | August 22, 2016 |
| Nougat | 7.1.0 - 7.1.2 | 25 | October 4, 2016 |
| Oreo | 8.0 | 26 | August 21, 2017 |
| Oreo | 8.1 | 27 | December 5, 2017 |
| Pie | 9.0 | 28 | August 6, 2018 |
| Android 10 | 10.0 | 29 | September 3, 2019 |
| Android 11 | 11 | 30 | September 8, 2020 |
| Android 12 | 12 | 31 | February 18, 2021 |

*Android Architecture*

*android architecture or Android software stack is categorized into five parts:*

*linux kernel*

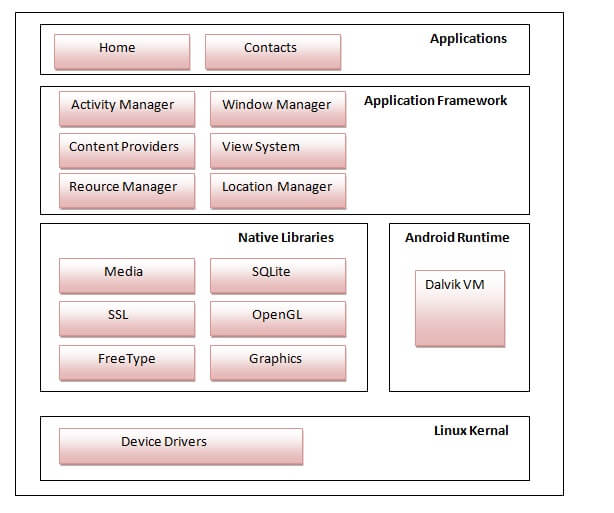
*native libraries (middleware),*

*Android Runtime*

*Application Framework*

*Applications*

*Let's see the android architecture first.*



*1) Linux kernel*

*It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.*

*2) Native Libraries*

*On the top of linux kernel, their are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc.*

*The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats.*

*3) Android Runtime*

*In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.*

*4) Android Framework*

*On the top of Native libraries and android runtime, there is android framework. Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.*

*5) Applications*

*On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernal.*

***Android Core Building Blocks***

***Android Components***

*An android component is simply a piece of code that has a well defined life cycle e.g. Activity, Receiver, Service etc.*

*The core building blocks or fundamental components of android are activities, views, intents, services, content providers, fragments and AndroidManifest.xml.*

***Activity***

*An activity is a class that represents a single screen. It is like a Frame in AWT.*

***View***

*A view is the UI element such as button, label, text field etc. Anything that you see is a view.*

***Intent***

*Intent is used to invoke components. It is mainly used to:*

* *Start the service*
* *Launch an activity*
* *Display a web page*
* *Display a list of contacts*
* *Broadcast a message*
* *Dial a phone call etc.*

*For example, you may write the following code to view the webpage.*

*Intent intent=new Intent(Intent.ACTION\_VIEW);*

*intent.setData(Uri.parse("http://www.nehalpatel.in"));*

*startActivity(intent);*

***Service***

*Service is a background process that can run for a long time.*

*There are two types of services local and remote. Local service is accessed from within the application whereas remote service is accessed remotely from other applications running on the same device.*

***Content Provider***

*Content Providers are used to share data between the applications.*

***Fragment***

*Fragments are like parts of activity. An activity can display one or more fragments on the screen at the same time.*

***AndroidManifest.xml***

*It contains informations about activities, content providers, permissions etc. It is like the web.xml file in Java EE.*

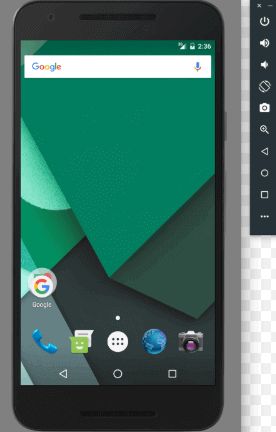
***Android Virtual Device (AVD)***

*It is used to test the android application without the need for mobile or tablet etc. It can be created in different configurations to emulate different types of real devices.*

*2.2 Creating Android Virtual Device (AVD)*

*Android Emulator*

*The Android emulator is an Android Virtual Device (AVD), which represents a specific Android device. We can use the Android emulator as a target device to execute and test our Android application on our PC. The Android emulator provides almost all the functionality of a real device. We can get the incoming phone calls and text messages. It also gives the location of the device and simulates different network speeds. Android emulator simulates rotation and other hardware sensors. It accesses the Google Play store, and much more*



*Testing Android applications on emulator are sometimes faster and easier than doing on a real device. For example, we can transfer data faster to the emulator than to a real device connected through USB.*

*The Android emulator comes with predefined configurations for several Android phones, Wear OS, tablet, Android TV devices.*

*Requirement and recommendations*

*The Android emulator takes additional requirements beyond the basic system requirement for Android Studio. These requirements are given below:*

*SDK Tools 26.1.1 or higher*

*64-bit processor*

*Windows: CPU with UG (unrestricted guest) support*

*HAXM 6.2.1 or later (recommended HAXM 7.2.0 or later)*

*Install the emulator*

*The Android emulator is installed while installing the Android Studio. However some components of emulator may or may not be installed while installing Android Studio. To install the emulator component, select the Android Emulator component in the SDK Tools tab of the SDK Manager.*

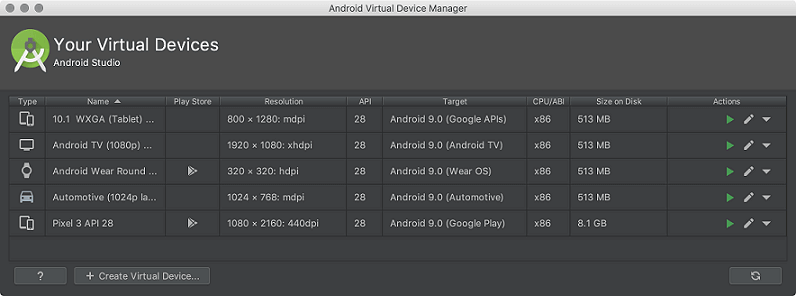
*Run an Android app on the Emulator*

*We can run an Android app form the Android Studio project, or we can run an app which is installed on the Android Emulator as we run any app on a device.*

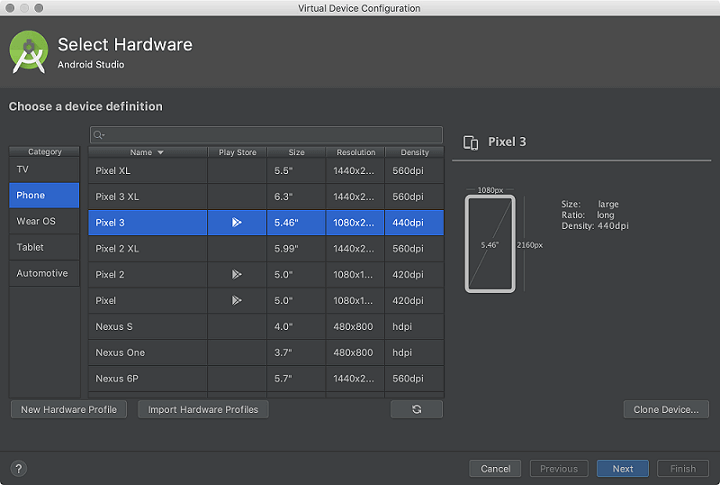
*To start the Android Emulator and run an application in our project:*

*1. In Android Studio, we need to create an Android Virtual Device (AVD) that the emulator can use to install and run your app. To create a new AVD:-*

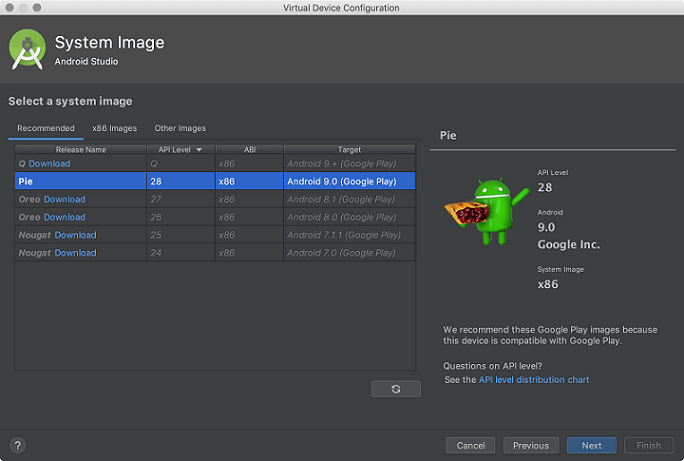
* 1. *Open the AVD Manager by clicking Tools > AVD Manager.*



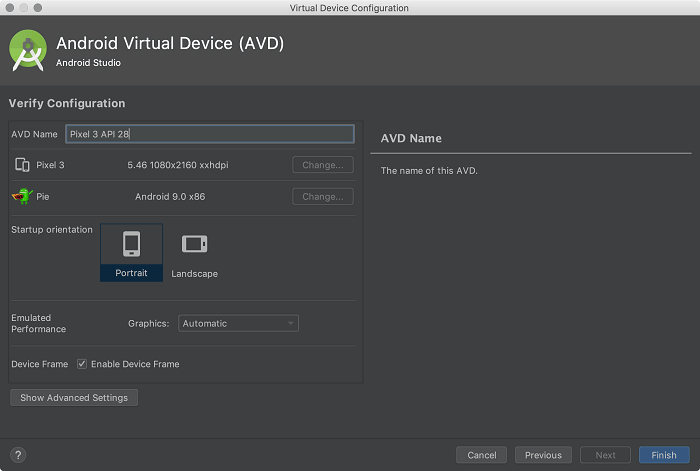
* 1. *Click on Create Virtual Device, at the bottom of the AVD Manager dialog. Then Select Hardware page appears.*



* 1. *Select a hardware profile and then click Next. If we don?t see the hardware profile we want, then we can create or import a hardware profile. The System Image page appears.*

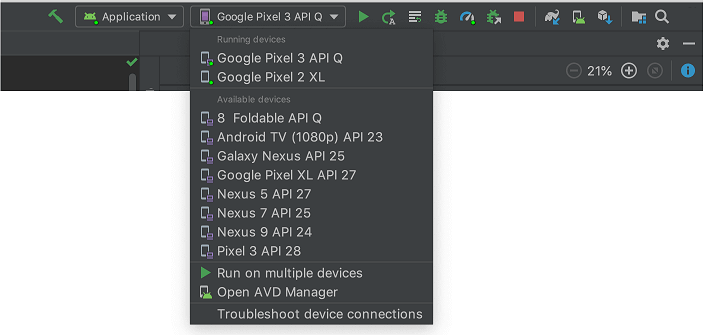


* 1. *Select the system image for the particular API level and click Next. This leads to open a Verify Configuration page.*



* 1. *Change AVD properties if needed, and then click Finish.*

1. *In the toolbar, choose the AVD, which we want to run our app from the target device from the drop-down menu.*



1. *Click Run.*

*Launch the Emulator without first running an app*

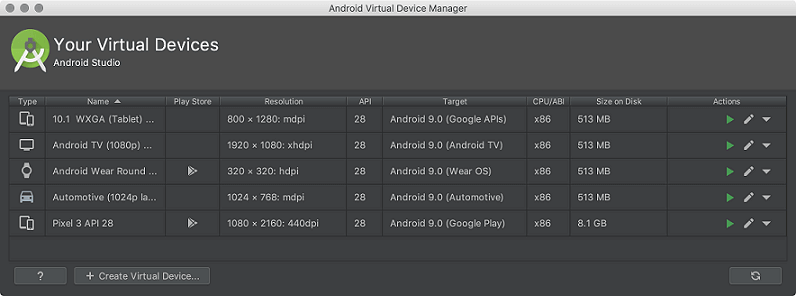
*To start the emulator:*

* *Open the AVD Manager.*
* *Double-click an AVD, or click Run*

*While the emulator is running, we can run the Android Studio project and select the emulator as the target device. We can also drag an APKs file to install on an emulator, and then run them.*

*Run and stop an emulator, and clear data*

*From the Virtual Device page, we can perform the following operation on emulator:*



* *To run an Android emulator that uses an AVD, double-click the AVD, or click Launch*
* *To stop the running emulator, right-click and select Stop, or click Menu ▼ and select Stop.*
* *If we want to clear the data from an emulator and return it to the initial state when it was first defined, then right-click an AVD and select Wipe Data. Or click menu ▼ and select Wipe Data.*

*2.3 Creating first App:*

*2.3.1 Activity*

***How to make android apps***

*In this page, you will know how to create the simple hello android application. We are creating the simple example of android using the Eclipse IDE. For creating the simple example:*

1. *Create the new android project*
2. *Write the message (optional)*
3. *Run the android application*

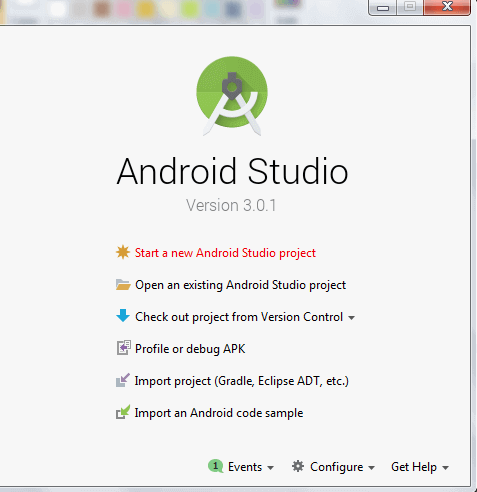
*Hello Android Example*

*You need to follow the 3 steps mentioned above for creating the Hello android application.*

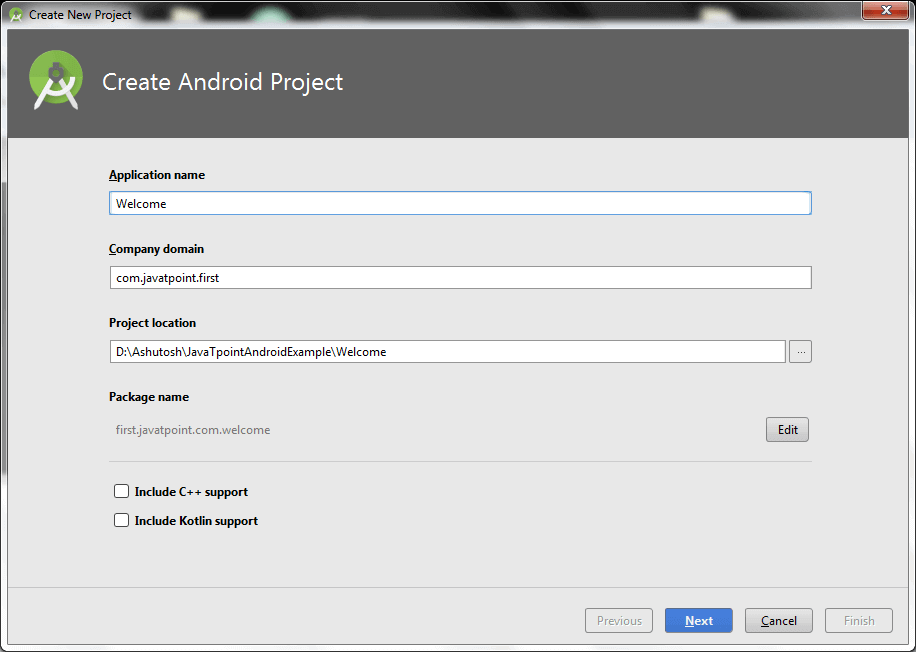
*1) Create the New Android project*

*For creating the new android studio project:*

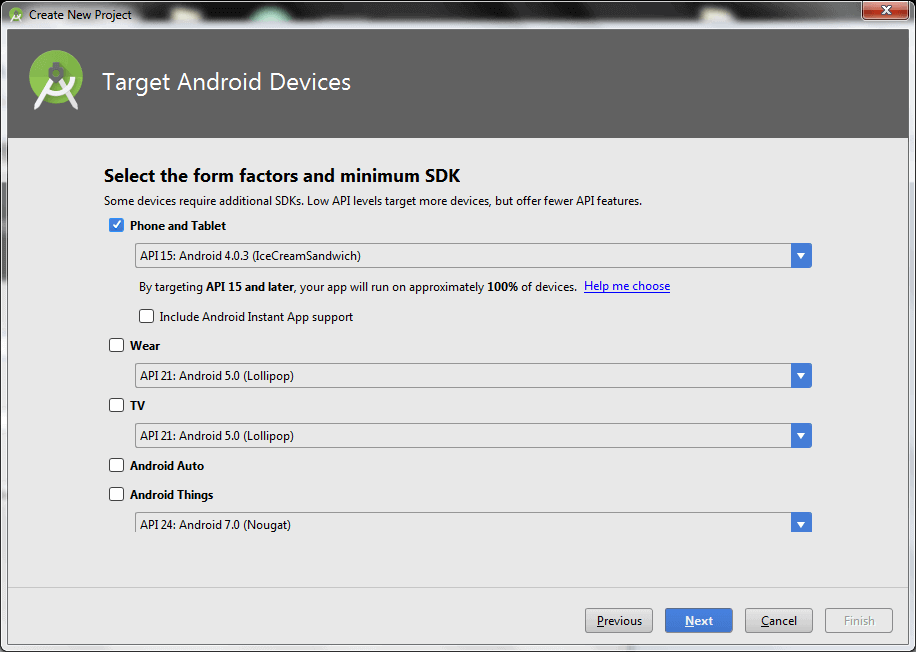
*1) Select Start a new Android Studio project*



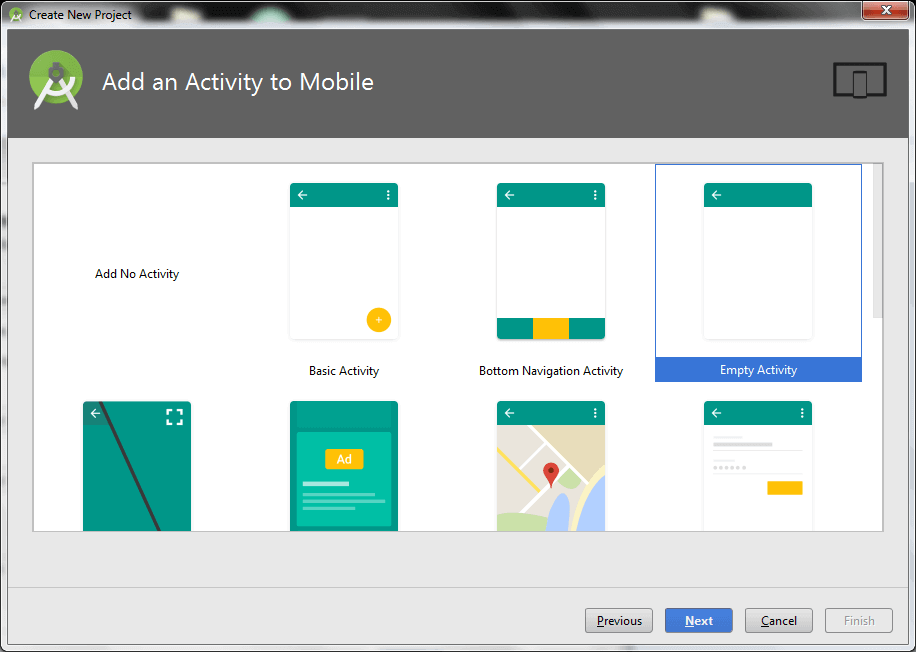
*2) Provide the following information: Application name, Company domain, Project location and Package name of application and click next.*



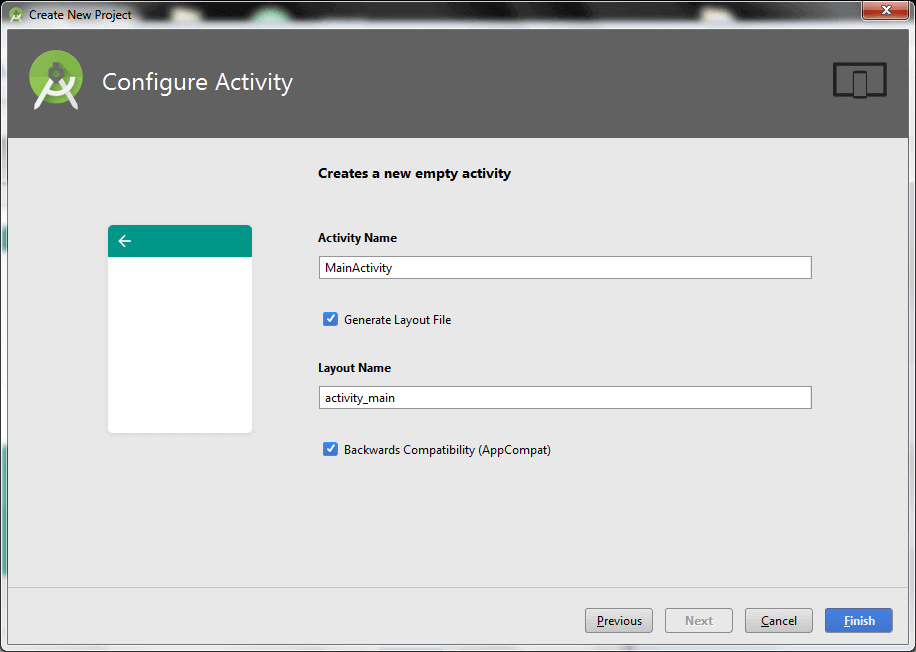
*3) Select the API level of application and click next.*



*4) Select the Activity type (Empty Activity).*

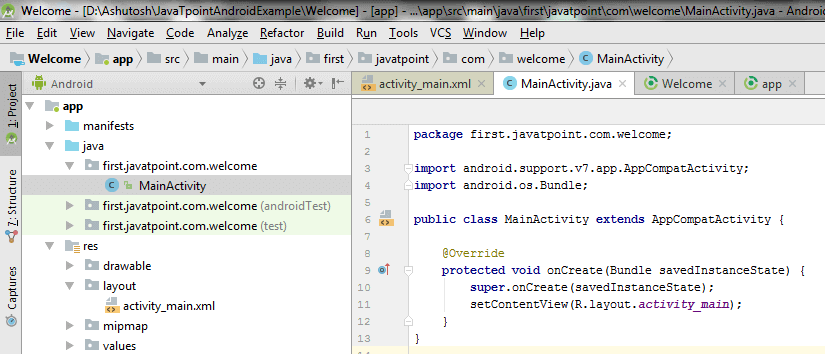


*5) Provide the Activity Name and click finish.*



*After finishing the Activity configuration, Android Studio auto generates the activity class and other required configuration files.*

*Now an android project has been created. You can explore the android project and see the simple program, it looks like this:*



*2) Write the message*

*File: activity\_main.xml*

*Android studio auto generates code for activity\_main.xml file. You may edit this file according to your requirement.*

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

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context="first.javatpoint.com.welcome.MainActivity">

    <TextView

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="Hello Android!"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintLeft\_toLeftOf="parent"

        app:layout\_constraintRight\_toRightOf="parent"

        app:layout\_constraintTop\_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

*File: MainActivity.java*

**package** first.test.com.welcome;

**import** android.support.v7.app.AppCompatActivity;

**import** android.os.Bundle;

**public** **class** MainActivity **extends** AppCompatActivity {

    @Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

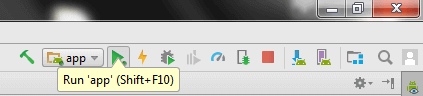
        setContentView(R.layout.activity\_main);

    }

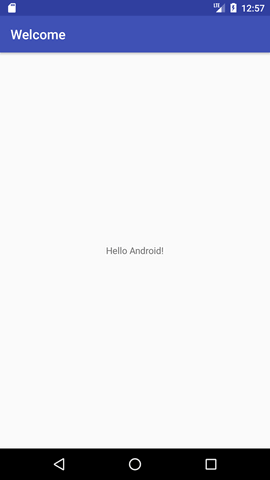
}

*3) Run the android application*

*To run the android application, click the run icon on the toolbar or simply press Shift + F10.*



*The android emulator might take 2 or 3 minutes to boot. So please have patience. After booting the emulator, the android studio installs the application and launches the activity. You will see something like this:*



*2.3.2 Layout*