Android
Android

- Open software platform for mobile development
- A complete stack
  - OS, Middleware, Applications
- An Open Handset Alliance (OHA) project
- Powered by Linux operating system
- Open source under the Apache 2 license

- Android ships with a rich set of applications
  - Email, calendar, browser, maps, text messaging, contacts, camera, dialer, music player, settings and others
Android 6 (Marshmallow)
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.
Many different versions

<table>
<thead>
<tr>
<th>Platform Version</th>
<th>API Level</th>
<th>VERSION_CODE</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android 6.0</td>
<td>23</td>
<td>M</td>
<td>API Changes</td>
</tr>
<tr>
<td>Android 5.1</td>
<td>22</td>
<td>LOLLIPOP_MR1</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 5.0</td>
<td>21</td>
<td>LOLLIPOP</td>
<td></td>
</tr>
<tr>
<td>Android 4.4W</td>
<td>20</td>
<td>KITKAT_WATCH</td>
<td>KitKat for Wearables Only</td>
</tr>
<tr>
<td>Android 4.4</td>
<td>19</td>
<td>KITKAT</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 4.3</td>
<td>18</td>
<td>JELLY_BEAN_MR2</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 4.2, 4.2.2</td>
<td>17</td>
<td>JELLY_BEAN_MR1</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 4.1, 4.1.1</td>
<td>16</td>
<td>JELLY_BEAN</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 4.0.3, 4.0.4</td>
<td>15</td>
<td>ICE_CREAM_SANDWICH_MR1</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 4.0, 4.0.1, 4.0.2</td>
<td>14</td>
<td>ICE_CREAM_SANDWICH</td>
<td></td>
</tr>
<tr>
<td>Android 3.2</td>
<td>13</td>
<td>HONEYCOMB_MR2</td>
<td></td>
</tr>
<tr>
<td>Android 3.1.x</td>
<td>12</td>
<td>HONEYCOMB_MR1</td>
<td>Platform Highlights</td>
</tr>
<tr>
<td>Android 3.0.x</td>
<td>11</td>
<td>HONEYCOMB</td>
<td>Platform Highlights</td>
</tr>
</tbody>
</table>

Set up

• Set up your development environment
  – Android Studio (v 1.4)

• Set up AVDs and devices for testing
  – Create Android Virtual Devices (AVDs)
  – Connect hardware devices
  – Use external emulators (Genymotion)
Development

• Create your application
  – Source code, resource files, and Android manifest file
• Build and run your application
• Debug your application
• Test your application
Dalvik Virtual Machine (<= 4.4)
Android Runtime (>= 5.0)
ART

• Introduces the use of ahead-of-time (AOT) compilation
• Compiles entire applications into native machine code upon their installation
• Eliminates Dalvik's interpretation and trace-based JIT compilation
• Improves overall execution efficiency and reduces power consumption
• Brings faster execution of applications, improved memory allocation and garbage collection
Packaging

- An application is a single APK (application package) file
- An APK file roughly has three main components
  - Dalvik executable: all your Java source code compiled down to Dalvik executable
  - Resources: everything that is not code (images, audio/video clips, XML files describing layouts, language packs, and so on)
  - Native libraries: e.g. C/C++ libraries
Manifest

- Permissions the application requires
- Minimum API Level required by the application
- Hardware and software features used or required by the application
- API libraries the application needs to be linked with
- ... and more
Android Security Model

- Application Isolation
- Application Signing
- Filesystem Isolation
Security model

• Each application has a unique Linux user ID
  — Permissions for all the files in an application are assigned to particular IDs

• Every application runs in its own Linux process

• Each process has its own virtual machine (VM)
  — An application runs in isolation with respect to other applications

• Different UIDs can not access each other’s data
High-level Concepts
## Key terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Component for composing UI</td>
</tr>
<tr>
<td>Service</td>
<td>Component that runs on the background</td>
</tr>
<tr>
<td>Intent</td>
<td>Message element that sends actions and data to components</td>
</tr>
<tr>
<td>Intent filter</td>
<td>Defines components by setting receiving intents</td>
</tr>
<tr>
<td>Broadcast receiver</td>
<td>Receives/responds to a certain broadcast (e.g., low battery)</td>
</tr>
<tr>
<td>Content provider</td>
<td>Provides standardized interfaces for sharing data among applications</td>
</tr>
<tr>
<td>Notification</td>
<td>Notifies events to users</td>
</tr>
</tbody>
</table>
Android through MVC

View
- View and related classes

Control
- Activity
- Service
- Broadcast receiver

Model
- SQLite
- Files and content providers
Applications and activities

- Any application can start another application’s component
- When the system starts a component, it starts the process for that application and instantiates the classes needed for the component
  - If one starts the activity in the camera application that captures a photo, that activity runs in the process that belongs to the camera application, not in your application’s process
- An application cannot directly activate a component from another application
  - To activate a component in another application, you must deliver a message to the system that specifies your intent to start a particular component
  - The system then activates the component for you