

An Introduction to Android

John Sonmez

www.pluralsight.com



Course Overview

- **What we will cover**
 - What is Android?
 - Setting up development environment
 - Project structure
 - Activity
 - UI and layouts
 - Preferences
 - Menus
 - Dialogs
 - Deployment

Course Overview

- **After this course you will be able to**
 - Create a simple Android application
 - Publish that application to the Android Market
- **We will create a real application during this course**

Protein consumed:	<input type="text" value="50g"/>
	<input type="button" value="Enter"/>
Total for day:	100g
Goal:	175g
Needed: 75g	
<input type="button" value="Reset"/>	

Introduction

- Android Background
- Android Architecture
- Android Applications
- Application Lifecycle
- APIs
- Security and Permissions
- The Android Market
- Summary

Android Background



Mobile operating system developed by Android Inc

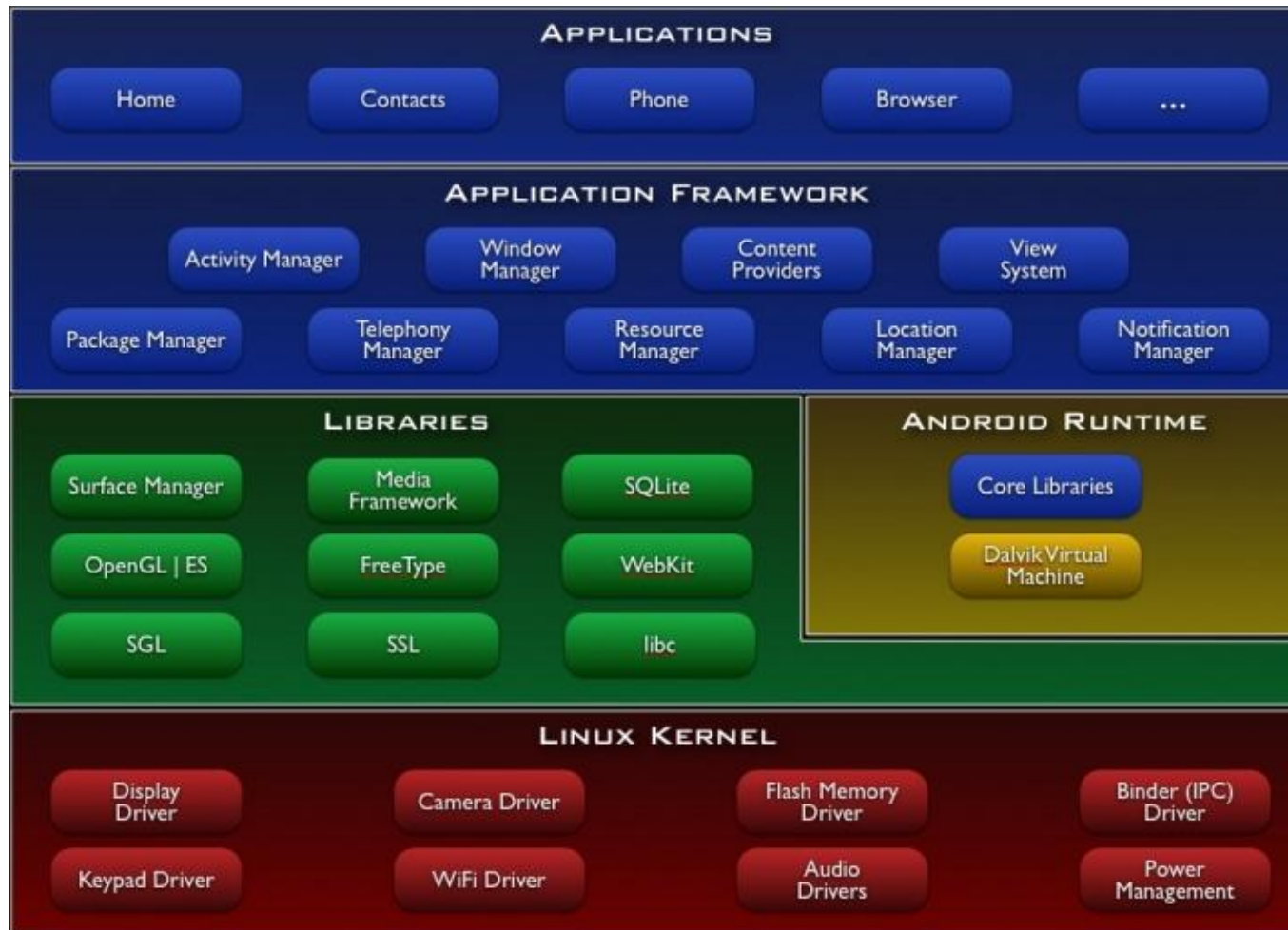
- Google bought it in 2005
- **Based in the Linux kernel**
- **Open source**
- **Big! (About 12 million lines of code)**
- **Application layer is Java**
- **Has a built in Google Market for application distribution**
- **Built on the idea of being “Open”**
 - All applications are equal
 - Everything is swappable

Android Background

- Supports a multitude of different devices
- Android 3.0 (Honeycomb) will be for tablet devices



Android Architecture



Android Architecture

- **Linux kernel**
 - Basic hardware abstraction layer



Android Architecture

■ Libraries

- Standard C library
- Media libraries
- Surface managers
- LibWebCore
- SGL
- 3D Libraries
- FreeType
- SQLite



Android Architecture

■ Android Runtime

- Core libraries that make up most of the standard Java libraries
- Doesn't use a JVM, uses a Dalvik VM
 - Runs .dex files
 - Each app runs in its own VM



Android Architecture

■ Application Framework

- Full set of services built in Java
- Views and windows
- Resources, content providers
- Phone services and APIs
- Notifications
- Application life-cycle management
- All applications directly interact with this layer



Android Architecture

■ Applications

- Base system applications are built off the same APIs all developers have access to
- Homescreen, Contacts, Phone, Browser can all be replaced by user versions
- All written in Java



Android Applications

- **What are they?**

- Loosely coupled set of services
- Consist of one or more
 - Activities
 - Services
 - Broadcast receivers
 - Content providers
- Can use code from other applications without directly linking to them
- Most applications are composed of a series of screens or Activities



Gmail Example

The image displays four sequential screenshots of the Gmail mobile application interface, illustrating the flow from the main menu to a specific email and its reply.

- Labels:** The first screenshot shows the Gmail 'Labels' screen. It lists various folders: 'Inbox' (1 email, sync 4 days), 'Starred' (star icon), 'Chats', 'Sent' (sync 4 days), 'Outbox', 'Drafts' (sync all), 'All Mail', and 'Spam' (1472 emails).
- Inbox:** The second screenshot shows the 'Inbox' screen. It lists three emails: 'From Eva K and other Agil...' (5:20pm), 'Boise Code Camp Schedule - A...' (1:36pm, from David Starr), and 'Jan 2011 PaceMaker Sales - Jo...' (Feb 14, from Me .. annette (4)).
- Boise Code Camp Schedule:** The third screenshot shows the details of the email 'Boise Code Camp Schedule' from David Starr (david@elegantcode.com). The email body contains information about the current schedule for the Boise Code Camp, mentioning a reversion to a single day due to fewer session submissions. It also mentions a 'Speaker's Dinner' at 6:30 at Fuddruckers.
- Reply:** The fourth screenshot shows the 'Reply' screen for the email. It includes a 'Compose Mail' text area, a 'Respond inline' button, and a preview of the email content. The preview shows the email body text, including the 'Schedule' section and the 'Speaker's Dinner' information.

Application Lifecycle



Controlled by Tasks

- A task is a set of Activities arranged in a stack
 - Example: going to the Gmail application and selecting a contact
- **Launching an application makes it the root of a task**
 - New activities get pushed on the task stack
 - Only the default behavior
- **All applications are run in their own process**
- **Android manages memory by prioritizing processes and killing least important ones**
- **Activities on a task stack can belong to processes that are killed (uh oh)**
 - Need to be able to restore state

Task Stack Example

The image displays three sequential screenshots of a mobile application interface, illustrating a task stack for a user named David Starr.

Screenshot 1 (Left): Shows an email inbox titled "Boise Code Camp Schedule". The email is from David Starr (david@elegantcode.com) to a group of recipients, dated Feb 16 at 1:36pm. The email content includes a "Schedule" section and a "Speaker's Dinner" section. The interface includes standard mobile status bar icons at the top and action buttons (Archive, Delete, navigation arrows) at the bottom.

Screenshot 2 (Middle): Shows a detailed view of the email from David Starr. The header includes his profile picture and name. The body of the email is displayed in a list format with sections: "Email" (david@guild3.com), "Email" (david@elegantcode.com), "Email" (david-starr@pluralsight.com), "Nickname" (ElegantCoder), "Twitter profile" (trying dotCover from JetBrains. 4 hours ago via Twitter), and "Facebook profile" (wife: why don't you just put the dishes... February 8 via Facebook). The interface includes standard mobile status bar icons at the top and a star icon for favorites at the bottom right.

Screenshot 3 (Right): Shows a social media profile page for David Starr (@ElegantCoder). The header includes his profile picture and name. The page features a "Bio" section (Software Craftsman | Pluralsight Instructor), a "Web" section (http://elegantcode.com), a "Location" section (Boise, ID), a "Following" section (252), and a "Followers" section (1,136). The interface includes standard mobile status bar icons at the top and action buttons (Unfollow, settings gear) at the bottom.

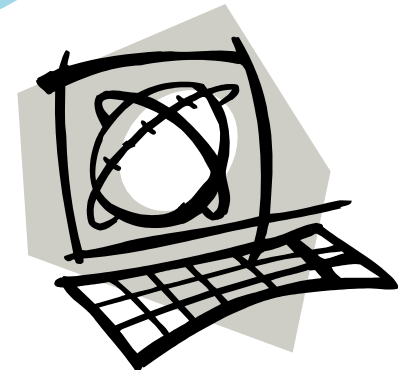
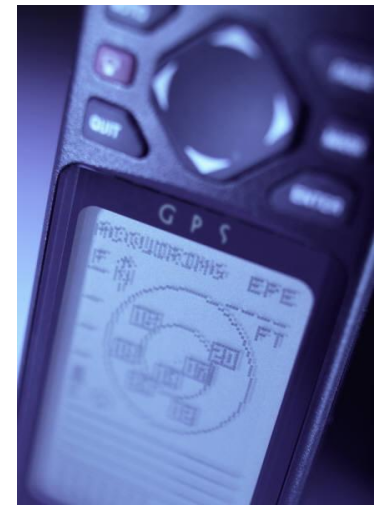
APIs

- Basic Android API levels correspond with the version of the Android OS.
- Typically forward compatible

Platform Version	API Level
Android 2.3.3	10
Android 2.3	9
Android 2.2	8
Android 2.1	7
Android 2.0.1	6
Android 2.0	5
Android 1.6	4
Android 1.5	3
Android 1.1	2
Android 1.0	1

APIs

- Other kinds of APIs
 - Location services
 - Telephony services
 - Audio and video
 - Web browser
 - Google maps



Security and Permissions



Each application runs with its own Linux user ID

- **All applications are sandboxed by default**
- **Permissions**
 - Applications use to declare what they need access to
 - Declared in the Android manifest file
 - Users are prompted at install time to allow app permissions
- **Applications are signed locally (APK files)**



The Android Market



Easy way to publish and find applications

- **Directly accessible from phone and now from web**
- **Open**
 - Not policed
 - No approval process
- **You don't have to distribute through Google's Android Market**
- **Licensing service**
- **Costs**
 - One time \$25 fee to become an Android developer
 - 30% fee for applications sold

Summary

- What Android is and what it is about
- Basic architecture of the platform
- Applications and their lifecycles
- APIs
- Security and permissions
- The Android Market



For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**