

# Android

Ambrosi Michele  
Rigo Francesco  
Marcantoni Francesco

mastergap@gmail.com

4 giugno 2009

# Indice

- 1 Ambiente di sviluppo
- 2 Sviluppo di applicazioni

# Installazione dell'Android SDK

- Effettuare il download dell'archivio. L'attuale versione stabile (per Windows, Mac OS X e Linux (i386)) è la 1.5 reperibile all'indirizzo *[http://developer.android.com/sdk/1.5\\_r1/index.html](http://developer.android.com/sdk/1.5_r1/index.html)*
- Scompattare l'archivio nella locazione desiderata, ad esempio `/opt/android`

```
unzip android-sdk-linux-x86-1.5_r1.zip /opt/android
```

- Esportare la directory tools contenuta nell'SDK aggiungendo nel vostro `/.bashrc`

```
export PATH=${PATH}:/opt/android/tools
```

# Installazione dell'Android SDK (continua)

La directory dell'SDK contiene le seguenti subdirectories

- tools - Strumenti di sviluppo (emulator, activitycreator ...ecc)
- samples - Esempi di applicazioni (utilizzati all'interno della documentazione ufficiale di Adroid)
- docs - Copia locale della documentazione (API, Tutorial ...ecc)

Per semplificare l'accesso alla documentazione locale aggiungete il seguente bookmark all'interno del vostro browser

- *file:///opt/android/documentation.html*

# Installazione del plugin ADT per Eclipse

Le maggiori distribuzioni Linux mettono a disposizione pacchetti precompilati di Eclipse 3.4 (Ganymede)

- Selezionare **Help** → **Software Updates**
- Selezionare il Tab **Available Software**
- Selezione **Add Site**
- Inserire come *Location*  
<https://dl-ssl.google.com/android/eclipse>  
oppure la versione priva di supporto ai socket SSL  
<http://dl-ssl.google.com/android/eclipse>  
Selezionare **OK**
- Tornare al Tab **Available Software**, spuntare **Developer Tools** e selezionare **Install**

# Installazione del plugin ADT per Eclipse (continua)

- Nella finestra seguente selezionare **Android Developer Tools** e **Android Editors** (facoltativo). Per quest'ultimo è necessario preinstallare il plugin WST (Eclipse Web Tools)
- Riavviare Eclipse
- Selezionare **Window** → **Preferences**
- Selezionare, dal pannello sulla sinistra, **Android**
- Selezionare **Browse** e localizzare la directory dell'SDK, contenente il file *android.jar*
- Selezionare **Apply** e successivamente **OK**

# Installazione del plugin ADT per Eclipse (continua)

E' possibile, nel caso la prima procedura non vada a buon fine, installare l'ADT plugin nel seguente modo

- Eseguire il download del plugin dal seguente indirizzo  
*[http://developer.android.com/sdk/adt\\_download.html](http://developer.android.com/sdk/adt_download.html)*
- Selezionare (in Eclipse) **Add Site** e successivamente **Archive**
- Selezionare il file ZIP scaricato precedentemente
- Seguire la seconda parte della precedente procedura

# Upgrade dell'SDK

- Eliminare i precedenti file di configurazione eseguendo su una console il comando

```
/opt/android-sdk/tools/emulator -wipe-data
```

- Modificare la direttiva export nel proprio `/.bashrc` in accordo con la nuova locazione
- Avviare Eclipse
- Selezionare **Window** → **Preferences**
- Selezionare, dal pannello sulla sinistra, **Android**
- Selezionare **Browse** e localizzare la nuova directory dell'SDK
- Selezionare **Apply** e successivamente **OK**



# Upgrade dell'ADT plug-in per Eclipse

- Avviare Eclipse
- Selezionare **Help** → **Software Updates**
- Selezionare il Tab **Installed Software**
- Selezionare **Update**
- Selezionare **Finish**
- Riavviare Eclipse

# Aspetti principali

- Gestione di *database*
- Creazione di *activity*
- Utilizzo degli *intent*

# Creazione di una nuova activity

I passi essenziali da compiere per la creazione di una nuova **Activity** sono i seguenti

- Creare una classe che estenda **Android.app.activity** o una delle sue sottoclassi dirette o indirette (*ListActivity*, *TabActivity* ... ecc)
- Reimplementare il metodo **onCreate** ed eventualmente gli altri metodi per la gestione degli eventi legati alle activity (*onStop*, *onRestart*, *onCreateOptionsMenu* ... ecc)
- Registrare la nuova activity all'interno dell'**Android Manifest** specificando eventualmente le azioni che è in grado di svolgere.

# Utilizzo degli Intent

Gli **Intent** hanno due scopi principali

- Lanciare l'esecuzione di una *activity* mediante la specifica dell'**azione da compiere**
- Permettere lo scambio di dati tra le activity, tra l'activity chiamante e l'activity chiamata e viceversa; nel primo caso per lo scambio dei parametri necessari all'esecuzione dell'azione richiesta, nel secondo caso per restituire il risultato dell'azione.