Flutter: innovazione tecnologica per lo sviluppo multipiattaforma

Durante il seminario ci sarà una breve introduzione su Flutter e poi si passerà allo sviluppo di un'applicazione. È consigliato configurare la propria macchina per poter partecipare attivamente.

Per impostare la propria macchina:

1. Installa Flutter sul tuo sistema operativo:

2. Configura la piattaforma per cui sviluppare:

3. Configura l'editor per sviluppare con Flutter.

Ora hai tutto il necessario per sviluppare la tua prima app con Flutter.
Per maggiori dettagli visita Flutter.

Windows

Requisiti di sistema

- **OS:** Windows 7 SP1 o superiore (64 bit)
- **Spazio su disco:** 400 MB (non include lo spazio richiesto per l'IDE)
- **Strumenti:** Flutter necessita che questi strumenti siano installati
    - Windows Power Shell 5.0 o superiore (preinstallato con W10)
    - Git per Windows 2.x, con l'opzione "**Use Git from the Windows Command Prompt**"

      Se Git per Windows è già installato, assicurati che puoi eseguire i comandi *git* dal prompt dei comandi o da PowerShell

Ottieni Flutter SDK

1. Scarica il seguente pacchetto di installazione:

   flutter_windows_v1.2.1-stable.zip

2. Estrai il file zip e sposta la cartella *flutter* in una posizione desiderata (es. *C:\src\flutter*; non installare Flutter in una directory che richiede permessi come *C:\Program Files*)
3. Trova il file *flutter_console.bat* dentro la cartella *flutter*. Esegui il file.

Ora sei pronto per eseguire i comandi Flutter nella console di Flutter!

Aggiorna il tuo path (raccomandato)

Se desideri eseguire i comandi di Flutter direttamente dalla tua console windows (prompt or PowerShell), esegui questi step per settare il PATH come variabile d'ambiente.

- Dalla barra di ricerca Start, scrivi 'env' e seleziona **Modifica variabili d'ambiente per l'account**
- Sotto **Variabili dell'utente** controlla se c'è una voce chiamata **Path:**
    - Se la voce esiste, aggiungere il percorso completo a flutter/bin (es. *C:\src\flutter/bin*) usando *;* come separatore dai valori esistenti
    - Se la voce non esiste, crea una nuova variabile d'ambiente chiamata Path con il percorso completo a flutter/bin come suo valore

Sarà necessario chiudere e riaprire qualsiasi finestra della console esistente affinché queste modifiche abbiano effetto.

Esegui *flutter doctor*

Se hai correttamente settato la variabile d'ambiente potrai eseguire questo comando da una qualunque console, altrimenti dovrai eseguirlo da una finestra della console che ha la directory Flutter nel percorso.
Il seguente comando server a vedere se ci sono delle dipendenze della piattaforma necessarie per completare la configurazione:

```
flutter doctor
```

Questo comando controlla il tuo ambiente e visualizza un rapporto sullo stato della tua installazione Flutter. Controllare attentamente l'output per altri software che potrebbe essere necessario installare (mostrato in grassetto).

Ad esempio:

```
[-] Android toolchain - develop for Android devices
    • Android SDK at D:\Android\sdk
    ✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
    • Try re-installing or updating your Android SDK,
      visit https://flutter.dev/setup/#android-setup for detailed instructions.
```

Le seguenti sezioni descrivono come eseguire queste attività e completare il processo di installazione. Dopo aver installato eventuali dipendenze mancanti, è possibile eseguire di nuovo il comando *flutter doctor* per verificare di aver impostato correttamente tutto.

MacOS

Requisiti di sistema

- **OS:** macOS (64 bit)
- **Spazio su disco:** 700 MB (non include lo spazio richiesto per l'IDE)
- **Strumenti:** Flutter necessita che questi strumenti da linea di comando siano installati
    - bash
    - curl
    - git 2.x
    - mkdir
    - rm
    - unzip
    - which

Ottieni Flutter SDK

1. Scarica il seguente pacchetto di installazione:

   flutter_macos_v1.2.1-stable.zip

2. Estrai il file zip nella directory desiderata, ad esempio:

```
cd ~/development
$ unzip ~/Downloads/flutter_macos_v1.2.1-stable.zip
```

3. Aggiungi lo strumento flutter al tuo percorso:

```
export PATH="$PATH:`pwd`/flutter/bin"
```

Questo comando imposta la variabile PATH solo per la finestra del terminale corrente. Per aggiungere il tuo percorso in modo permanente, vedi **Aggiorna il tuo path**.

Aggiorna il tuo path (raccomandato)

È possibile aggiornare la variabile PATH per la sessione corrente solo sulla riga di comando, come mostrato sopra. Probabilmente vorrai aggiornare questa variabile in modo permanente, in modo da poter eseguire i comandi di flutter in qualsiasi sessione di terminale.

I passaggi per la modifica permanente di questa variabile per tutte le sessioni del terminale sono specifici della macchina. In genere si aggiunge una riga a un file che viene eseguito ogni volta che si apre una nuova finestra.

1. Determina la directory dove hai posizionato SDK di Flutter. Avrai bisogno di questo percorso nello step 3.
2. Apri (o crea) $HOME/.bash_profile. Il percorso del file potrebbero essere differenti nella tua macchina.
3. Aggiungi la seguente linea e cambia [PATH_TO_FLUTTER_GIT_DIRECTORY] con il percorso in cui hai posizionato SDK di Flutter.

```
export PATH="$PATH:[PATH_TO_FLUTTER_GIT_DIRECTORY]/flutter/bin"
```

4. Esegui source $HOME/.bash_profile per aggiornare la finestra corrente.
5. Verifica che la directory flutter/bin sia nel tuo PATH eseguendo:

```
echo $PATH
```

Per maggiori dettagli, vedere questa domanda StackExchange.

Linux

Requisiti di sistema

- **OS:** Linux (64 bit)
- **Spazio su disco:** 600 MB (non include lo spazio richiesto per l'IDE)
- **Strumenti:** Flutter necessita che questi strumenti da linea di comando siano installati
  - bash
  - curl
  - git 2.x
  - mkdir
  - rm
  - unzip
  - which
  - xz-utils
- Librerie condivise: il comando test di Flutter dipende dalla disponibilità di questa libreria.
  - libGLU.so.1 – fornita dai pacchetti mesa ad es. libglu1-mesa su Ubuntu/Debian

Ottieni Flutter SDK

4. Scarica il seguente pacchetto di installazione:

   flutter_linux_v1.2.1-stable.tar.xz

5. Estrai il file zip nella directory desiderata, ad esempio:

```
cd ~/development
$ tar xf ~/Downloads/flutter_linux_v1.2.1-stable.tar.xz
```

6. Aggiungi lo strumento flutter al tuo percorso:

```
export PATH="$PATH:`pwd`/flutter/bin"
```

Questo comando imposta la variabile PATH solo per la finestra del terminale corrente. Per aggiungere il tuo percorso in modo permanente, vedi **Aggiorna il tuo path** (stessa procedura utilizzata per MacOs).

Impostazioni Piattaforma

Con Windows sarà possibile testare ed eseguire le app sviluppate con Flutter solo su Android. MacOS invece, supporta lo sviluppo di app Flutter sia per iOS che per Android.

Android setup

Installazione Android Studio

1. Scarica e installa Android Studio
2. Avvia Android Studio, e clicca 'Android Studio Setup Wizard'. Questo installa l'ultimo Android SDK, Android SDK Platform-Tools, and Android SDK Build-Tools, che sono richiesti da Flutter quando si sviluppa per Android

Preparazione del tuo dispositivo fisico Android

Per eseguire e testare la tua applicazione Flutter su un dispositivo fisico Android, avrai bisogno di un device con Android 4.1 (API level 16) o superiore.

1. Abilita 'Opzioni sviluppatore' e 'USB Debugging' sul tuo dispositivo. Istruzioni dettagliate sono disponibili nella documentazione Android.
2. Solo per Windows: installare Google USB Driver
3. Usando un cavo USB, collega il dispositivo al tuo computer. Autorizza il tuo computer ad avere accesso al tuo dispositivo (ti apparirà un pop-up da dover accettare sul tuo dispositivo)
4. Da terminale, esegui *flutter devices* per verificare che Flutter riconosce il tuo dispositivo Android

Per impostazione predefinita, Flutter utilizza la versione di Android SDK sui cui è basato lo strumento ADB. Se si desidera che Flutter utilizzi un'installazione diversa di Android SDK, è necessario impostare la variabile di ambiente ANDROID_HOME su tale directory di installazione.

Impostazione di un emulatore Android

Per eseguire e testare la tua applicazione Flutter su un emulatore Android, segui questi step:

1. Abilita VM acceleration sul tuo computer
2. Lancia **Android Studio > Tools > Android > AVD Manager** e seleziona **Create Virtual Device**
3. Scegli un device e seleziona **Next**
4. Seleziona uno o più immagini di sistema per la versione Android che vuoi emulare, e seleziona **Next**. È raccomandata un'immagine *x86 o x86_64*.
5. Sotto Emulated Performance, seleziona **Hardware – GLES 2.0** per abilitare l'accelerazione hardware
6. Verifica che la configurazione AVS è corretta e seleziona **Finish**.

Per ulteriori dettagli, vedi Managing AVDs

7. In Android Virtual Device Manager, clicca **Run** nella toolbar. L'emulatore si avvierà mostrando il device e sistema operativo selezionato.

iOS setup

Installazione Xcode

Per sviluppare app per iOS, hai bisogno di un Mac con Xcode 9.0 o superiore.

1. Installa Xcode 9.0 o superiore (via web download o Mac App Store).
2. Configura gli struementi da linea di comdando di Xcode per usare la nuova versione di Xcode installata eseugendo:

```
sudo xcode-select --switch /Applications/Xcode.app/Contents/Developer
```

3. Assicurati che il contratto di licenza di Xcode sia firmato aprendo Xcode una volta e confermando o eseguendo sudo xcodebuild –license da linea di comando.

Con Xcode, sarai in grado di eseguire le app Flutter su un dispositivo iOS o sul simulatore.

Configura il tuo simulatore

Per eseguire e testare la tua app Flutter sul simulatore iOS, procedi nel seguente modo:

1. Sul tuo Mac, trova "Simulator" via Spotlight o usando la seguente linea di comando:

```
open -a Simulator
```

2. Assicurati che il simulatore usi un device a 64 bit (iPhone 5s o superiore) controllando nel menù del simulatore in **Hardware > Device**.

Configura il deploy per il dispositivo fisico

Per distribuire l'app Flutter su un dispositivo iOS fisico, sono necessari alcuni strumenti aggiuntivi e un account Apple (solo per la distribuzione nello store). Avrai anche bisogno di configurare la distribuzione dei dispositivi fisici in Xcode.

1. Installa Homebrew
2. Assicurati che Homebrew sia aggiornato

```
brew update
```

3. Installa gli strumenti per la distribuzione delle app Flutter sui dispositivi iOS eseguendo i seguenti comandi:

```
brew install --HEAD usbmuxd
brew link usbmuxd
brew install --HEAD libimobiledevice
brew install ideviceinstaller ios-deploy cocoapods
pod setup
```

Se uno di questi comandi fallisce, esegui brew doctor e segui le istruzioni per risolvere eventuali problemi.

Impostazione Editor

È possibile creare app con Flutter usando qualsiasi editor di testo combinato con i strumenti da riga di comando. Tuttavia, è consigliato di utilizzare uno dei plug-in editor per un'esperienza migliore. Questi plugin forniscono auto-completamento del codice, evidenziazione della sintassi, assistenza per la modifica dei widget, supporto di esecuzione, debug e altro.

È altamente consigliato utilizzare Android Studio come editor in quanto il seminario si svolgerà utilizzando questo strumento.

Android Studio (Raccomandato)

- Android Studio, versione 3.0 o superiore

Installazione dei plugin Flutter e Dart

Per installare seguire questi passi:

1. Esegui Android Studio.
2. Apri **Preferences > Plugins** su macOS, **File > Settings > Plugins** su Windows & Linux.
3. Seleziona **Browse repositories,** seleziona il plugin Flutter e clicca **Install.**
4. Clicca **Yes** quando viene richiesto di installare il plugin Dart.
5. Clicca **Restart** quando richiesto.

Visual Studio Code

- VS Code, l'ultima versione stabile.

Installazione dei plugin Flutter e Dart

1. Esegui VS Code
2. Invoca **View > Command Palette…**.
3. Scrivi "install", e seleziona **Extensions: Install Extensions**.
4. Scrivi "flutter" nel campo di ricerca delle estensioni, seleziona **Flutter** dalla lista, e clicca **Install**. Questo installerà anche il plugin Dart.

Valida le tue impostazioni con Flutter Doctor

1. Invoca **View > Command Palette…**.
2. Scrivi "doctor", e seleziona **Flutter: Run Flutter Doctor**.
3. Per ogni problema revisiona l'output nel pannello **OUTPUT**.