

# Ambiente di sviluppo: guida installazione

...

Sviluppo React e React Native su Android

# Prima di cominciare

- Alcuni passi di installazione agiscono sui file di configurazione della bash: `.bashrc` per i sistemi linux e `.zshrc` per quelli mac.
- Sulle macchine del laboratorio sono già state apportate le modifiche a questi file e quindi alcuni passaggi (denotati con `*`) saranno necessari solo sulle macchine che userete a casa e non dovranno essere ripetuti in laboratorio
- Se a casa lavorate su sistemi macOS dovreste creare, se non già presente, nella home dell'utente del sistema un file `.zshrc` (*`touch ~/.zshrc`*)
- Per l'installazione di node avrete due possibilità di scelta:
  - la più consigliata nel settore è quella installare `nvm` che è un tool che permette di gestire le versioni di node
  - la più semplice e immediata, ma sconsigliata, è scaricare node dal sito ufficiale (non possibile sulle macchine del laboratorio)
- in laboratorio dovreste usare il comando `[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"` ogni volta che vorrete usare node, per avviare `nvm`

# Spiegazioni Unix

- in unix con ~ si intende la cartella home dell'utente, corrispondente, se esiste, alla variabile d'ambiente HOME. Verificare con il comando `echo $HOME` (stampa il valore della var. d'ambiente)
- i file puntati (che iniziano con .) sono file speciali e sono nascosti nei filesystem, per mostrarli serve qualcosa in più rispetto a quelli normali ad esempio l'opzione -h del comando ls da terminale (comando completo: `ls -h pathCartella`) oppure con qualche azione esplicita sul visualizzatore grafico delle cartelle:
  - sul finder mac usare la shortcut Command + Shift + .
  - sul file explorer di linux di solito basta cliccare con il tasto destro dentro la cartella e cliccare sulla voce mostra file nascosti
- quindi per modificare il file .zshrc/.bashrc o usate un editor da terminale (tipo vim) oppure mostrate i file nascosti nel finder/file explorer e aprite un file con un qualsiasi editor

# Parte 1: React

# NVM: node version manager

Tool che ci servirà per installare e gestire le versioni di nodejs e npm (gestore di pacchetti JavaScript)

Se si desidera installare node dal sito saltare questa e la prossima slide (non possibile in lab)

## Installazione nvm:

- `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash`
- aggiungere al file `.bashrc/.zshrc` la riga: `export NVM_DIR="$HOME/.nvm"` (\* non in lab)
- aggiungere anche la riga: `[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"` (\* in lab dovreste eseguire questa linea ogni volta che aprirete un terminale, per avviare nvm)

## Verifica:

- chiudere e riaprire il terminale oppure eseguire il comando `source ~/.zshrc` sul mac o `source ~/.bashrc` su linux
- `nvm --version`

Link utile per qualsiasi problema: <https://github.com/nvm-sh/nvm>

# Node ed NPM

## Installazione tramite nvm:

- `nvm install --lts`

## Verifica:

- `node -v`            `--->`        `Now using node v16.13.0`
- `npm -v`            `--->`        `npm 8.1.0`

**Nota:** sulle macchine del lab, visto che dovremo avviare nvm ogni volta che vogliamo utilizzare un nuovo terminale, verificare sempre di usare una versione di node ed npm aggiornata, se la versione risulta vecchia (v12 ad esempio) eseguire nvm con il comando `[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"` e ricontrollare la versione di node

# Installazione tramite sito (non possibile in lab)

Molto semplice:

- vai sul sito ufficiale di node <https://nodejs.org/en/> e scarica la versione lts (lastest stable)
- segui i passaggi per installare ;)

## Verifica:

- `node -v`            --->        Now using node v16.13.0
- `npm -v`            --->        npm 8.1.0

**Note:** 1) l'installazione da sito installa node in una cartella che necessita di permessi di amministratore per scriverla, quindi quando installerete qualsiasi cosa globalmente con npm dovrete usare i permessi di amministratore

2) anche con installazione da sito, sarà necessario avere il file `.zshrc` su mac e `.bashrc` su linux

# Settare un path per le installazioni globali

Per l'installazione di molti moduli node sarà conveniente un'installazione globale.

Di default la cartella in cui vengono effettuate le installazioni globali è accessibile solo con permessi di root ma secondo alcune fonti è sconsigliabile fare delle installazioni globali con permessi di root e un approccio migliore è quello di cambiare la cartella delle installazioni globali in questo modo:

- `npm config set prefix ~/.npm` (mostra un problema di compatibilità con nvm ma non sembra inficiare le installazioni)
- aggiungere al file di configurazione (.bashrc o .zshrc) la riga:
  - `export PATH=$PATH:$HOME/.npm/bin` (\*)

Gli asterischi denotano che sulle macchine del laboratorio sono già presenti queste modifiche



# Creazione App React

**Consiglio:** crea una tua directory da usare come workspace per le tue app

## Creazione App:

- installare globalmente il comando di creazione app: `npm i -g create-react-app`
- creare l'app nella cartella in cui vi trovate: `create-react-app myApp`

## Esecuzione:

- Entrare nella cartella dell'applicazione: `cd myApp`
- Eseguire l'app: `npm start`
- Si aprirà un browser con il simbolo di react -> è l'applicazione che react crea di default

**Altri consigli:** installare i plugin di vscode *Prettier* e *Simple react snippet*, aggiungere il format on save nelle impostazioni di vscode e mettere Prettier come default formatter

# Bootstrap: toolkit per front end

**Installazione:** eseguire all'interno della cartella del progetto il comando: `npm install bootstrap`

Il modulo verrà salvato, come tutti i moduli installati, nella cartella node\_modules

**Verifica:** importare bootstrap nel file App.js e notare il cambio di font e colorazione del testo nel browser

codice in App.js:

```
import "bootstrap/dist/css/bootstrap.css"
```

**Bootstrap:** <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

# Parte 2: React Native

# EXPO Cli: tool per eseguire App React Native

## Installazione comando di creazione app:

- `npm i -g expo-cli`

## Creazione App:

- `expo init nomeApp`
- selezionare il template dell'app, nel nostro caso blank, per un'app vuota

## Esecuzione:

- `cd nomeApp`
- `npm start` → apre un web browser che fornisce alternative per l'esecuzione
- varie alternative:
  - installazione su dispositivo fisico (non ci interessa per il corso)
  - emulazione (guarda le prossime slide)

# Emulazione Android

Per testare le proprie applicazioni senza usare un dispositivo reale si possono usare gli emulatori, nel nostro caso useremo quello fornito da Android Studio. Immagino che abbiate già settato l'SDK android e android studio in generale, se qualcosa non dovesse funzionare comunque potete seguire [questa guida](#) per risolvere i problemi.

Dopo aver settato l'ambiente android e relativa SDK c'è un altro passaggio necessario all'emulazione:

- aggiungere le seguenti righe al file di configurazione della bash (\* non da fare in lab):
  - export ANDROID\_SDK=XXX
  - export PATH=XXX/platform-tools:\$PATH

Dove XXX è la posizione dell'Android SDK che di solito è \$HOME/Library/Android/sdk.

Per essere sicuri della locazione dell'sdk android seguire il primo passaggio della guida linkata sopra dove parla di Android SDK Location

Per verificare usare il comando: `adb`

se esce un messaggio di errore (del tipo command not found) qualcosa è andato storto)  
potrebbe essere necessario riavviare la bash prima di eseguire adb

# Emulazione Android (2)

## Settare l'emulatore:

- Aprite l'AVD Manager nello stesso modo in cui avete aperto l'SDK Manager
- create virtual device -> scelgo il dispositivo (ad esempio Pixel 3a) -> scarico l'immagine con nome della release 'Q' (ultima release stable)
- una volta installato, posso avviare l'emulatore

## Emulazione con EXPO:

- entrare nella cartella dell'app react native ed eseguire il comando *npm start*
- dal browser (aperto in automatico sul metro bundler) cliccare su Run on Android device/emulator
- in alternativa dal terminale da cui avete eseguito npm start, digitare il carattere "a"