Ubuntu Linux

> IMPORTANTE: Usando sistemas Linux não é possível desenvolver apps com código nativo para a plataforma IOS.

Preparando ambiente para desenvolvimento Android com React Native

Vamos começar instalando todos os recursos necessários para rodar apps Android utilizando o React Native. Todos os passos dessa configuração estão disponíveis na documentação do React Native, que pode ser acessada através deste link.

Node js

Para começar precisaremos contar com o runtime do Node js para executar nosso código JavaScript. Podemos instalá-lo facilmente usando o `apt-get` no Ubuntu:

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
sudo apt-get install -y nodejs
```

React Native CLI

Para contar com algumas facilidades na construção do projeto, testar alterações, rodar nossas apps nas plataformas nativas e etc., contamos também com uma ferramenta de CLI do React Native. Podemos instalá-la usando o *Node Package Manager* que já vem junto com o node js.

```
npm install -g react-native-cli
```

> Recomenda-se a utilização do Node Package Manager na versão 4. Você também pode (e é recomendável) utilizar a ferramenta Yarn. Yarn é um gerenciador de pacotes criado também pelo Facebook que já conta com uma série de otimizações para facilitar o gerenciamento das dependências nos seus projetos que usam ferramentas da própria empresa como React, React Native, Jest, Watchman, etc. No Debian ou Ubuntu Linux, você pode instalar o Yarn através do repositório de pacotes Debian. Primeiro precisamos configurar o repositório: `curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add - echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list` . Em seguida digite: `sudo apt-get update && sudo apt-get install yarn` .

Java

Para podermos rodar nossa app no Android precisaremos do Java Development Kit (JDK) na versão 8 ou superior. Você pode baixar o JDK aqui.

Android Studio

Seguindo em frente precisaremos também do ambiente de desenvolvimento Android configurado, portanto, vamos baixar e instalar também a ferramenta Android Studio e as SDK Tools. Baixe e instale o Android Studio, selecione *"Custom"* quando perguntado sobre o tipo de instalação desejado e certifique-se de marcar as seguintes opções no instalador da ferramenta antes de clicar em *"Next"* e instalar efetivamente os componentes:

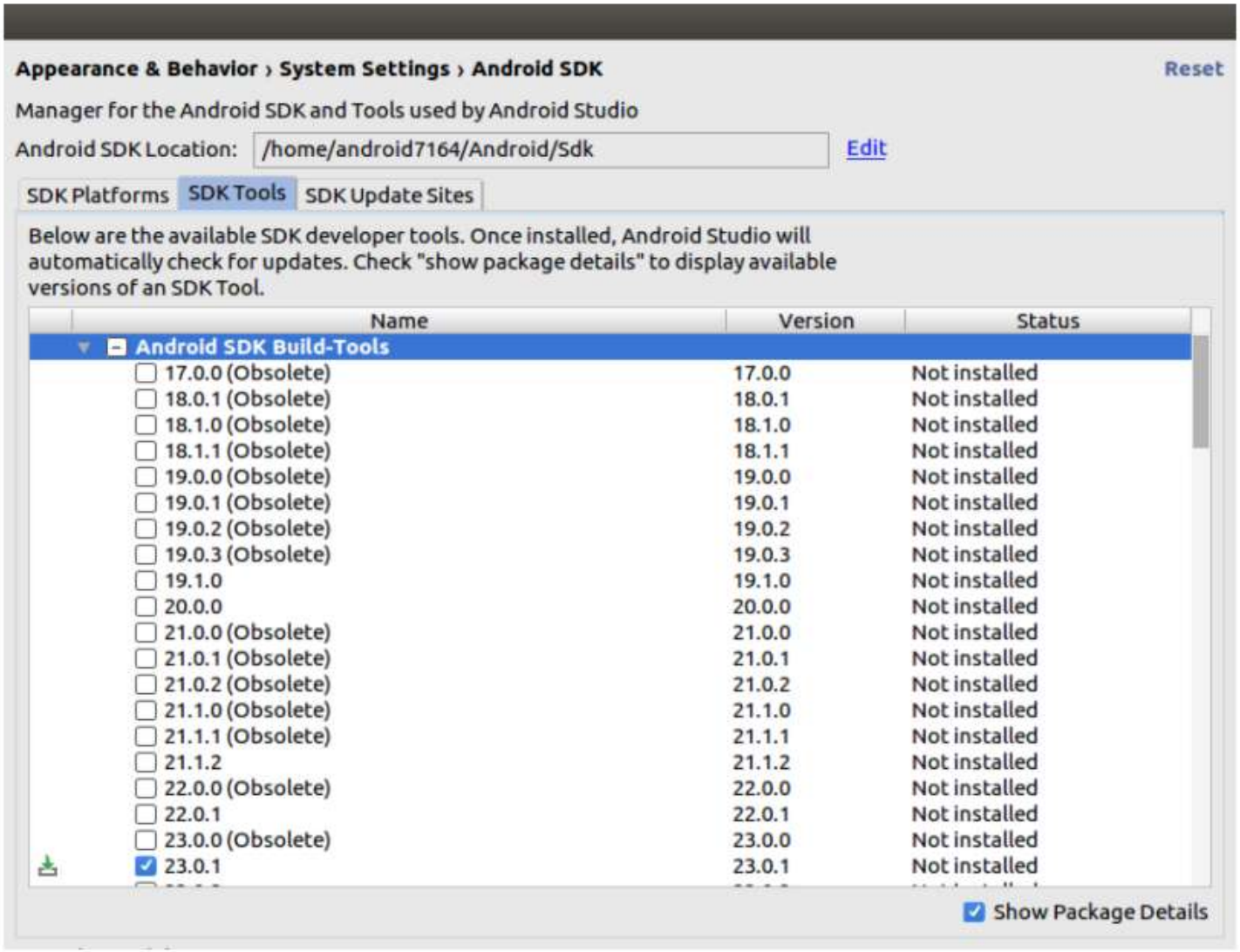
- Android SDK
- Android SDK Platform
- Android Virtual Device

O Android Studio já instala por padrão a última versão do SDK do Android, porém para desenvolver apps nativas para o Android com React Native, precisaremos instalar adicionalmente o SDK na versão Android 6.0 (Marshmallow). Podemos adicionar SDKs adicionais no SDK Manager do Android Studio. Para acessá-lo, clique em *"Configure"* na tela de boas vindas, e então selecione *"SDK Manager"*.

Selecione a aba *"SDK Platforms"* e marque o checkbox *"Show Package Details"* no canto inferior direito. Abra a seção *"Android 6.0 (Marshmallow)"*, e certifique-se de selecionar os seguintes items:

- Google APIs
- Android SDK Platform 23
- Intel x86 Atom_64 System Image
- Google APIs Intel x86 Atom_64 System Image

Agora selecione a aba *"SDK Tools"* e marque o checkbox *"Show Package Details"* no canto inferior direito. Abra a seção *"Android SDK Build-Tools"* e selecione a opção *"23.0.1"*.



Por fim, clique em *"Apply"* para baixar e instalar o SDK e as Build Tools.

Agora precisaremos configurar a variável de ambiente `ANDROID_HOME`, para que o ambiente do React Native consiga enxergar o SDK do Android no momento de instalar e rodar nossas apps no Android.

Adicione as seguintes linhas ao seu arquivo de configuração bash `$HOME/.bash_profile`:

```
export ANDROID_HOME=$HOME/Android/Sdk
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

Adicionalmente, rode o seguinte comando para recarregar as configurações no terminal: `source $HOME/.bash_profile`. Você pode também verificar se o valor de ANDROID_HOME foi adicionado corretamente à variável de ambiente PATH executando: `echo $PATH`.

Preparando um emulador Android

Um último passo importante é que precisamos preparar um Android Virtual Device (AVD) para podermos testar nossas aplicações. Você pode ver a lista com os emuladores configurados acessando o *"AVD Manager"* do Android Studio. Procure por um ícone como o que segue na barra de ferramentas do Android Studio:

Selecione *"Create Virtual Device"*, escolha um modelo de dispositivo disponível (Nexus 5X, por exemplo) e clique em *"Next"*. Selecione a aba *"x86 Images"*, e então procure por *"Marshmallow API Level 23, x86_64 ABI image"* com *"Android 6.0 (Google APIs)"*.