

Minicurso Spring e Android

Criado por Gabriel Schaidhauer - Aula 4



Framework Android

Framework Android

O framework Android é um framework desenvolvido especialmente para atuar em conjunto com o sistema operacional Android. Este framework permite diversas interações com o sistema operacional, tais como acesso a outros aplicativos, acesso a recursos como ligações, agenda, banco de dados, armazenamento, som, etc.

Framework Android

Para desenvolvimento no android podem ser usadas três linguagens. Java, Kotlin, e C++. Java e Kotlin são usados para desenvolvimentos comuns. C++ é utilizado para coisas que exigem alta performance, tais como comunicação em rede, etc. em casos muito específicos.

Framework Android

Além das linguagens, e das possibilidades que o framework oferece, ele possui uma forma única de trabalhar. Não só é necessário conhecer as linguagens de programação, como também as formas de configurá-lo, sendo que para tanto existem arquivos específicos, como é o caso do Manifest.xml, da árvore de recursos, etc.

Manifest.xml

Manifest.xml

O Manifest é um arquivo do tipo XML, e nele são descritos e configurados pontos críticos do aplicativo, havendo uma infinidade de possibilidades.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.gabriel.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Tá, mas tipo o que?

So what can **I** do?



Configurando dados da aplicação

<application></application>

- allowBackup: Informa se pode ser feito backup do aplicativo;
- icon: Informa o ícone da aplicação;
- label: Indica o nome da aplicação a ser lido pelo usuário;
- roundIcon: São ícones alternativos para que o usuário configure;
- supportsRtl: Permite que o aplicativo seja utilizado da esquerda para a direita em culturas onde isto é o padrão, como no japonês.
- theme: Indica o tema visual do aplicativo;

Activities

<activity></activity>

Activities representam de forma geral as telas do aplicativo. No Manifest, elas devem ser declaradas a fim de ser conhecidas pela aplicação e poder agir em cima delas.

Services

<service></service>

Services são classes que implementam rotinas de longa duração que não importem à utilização do usuário. Todas as services devem ser declaradas no Manifest.xml

Receivers

<receiver></receiver>

Receiver é uma subclasse de BroadcastReceiver, a qual implementa comportamentos a ser executados sem a necessidade da interação com o usuário. Ao contrário dos services, um BroadcastReceiver deve ser de curta duração, entretanto, pode ser utilizado mesmo sem a necessidade da aplicação.

Receivers podem ser declarados no manifest, ou ativados de forma programática na aplicação.

Providers

<provider></provider>

Provider é uma subclasse de ContentProvider. Content provider é uma classe que fornece acesso a dados controlados pelo aplicativo desenvolvido, como por exemplo a uma lista de contatos. Um ContentProvider deve ser declarado no Manifest.xml a fim de ser identificado por outras aplicações.

É muita coisa!



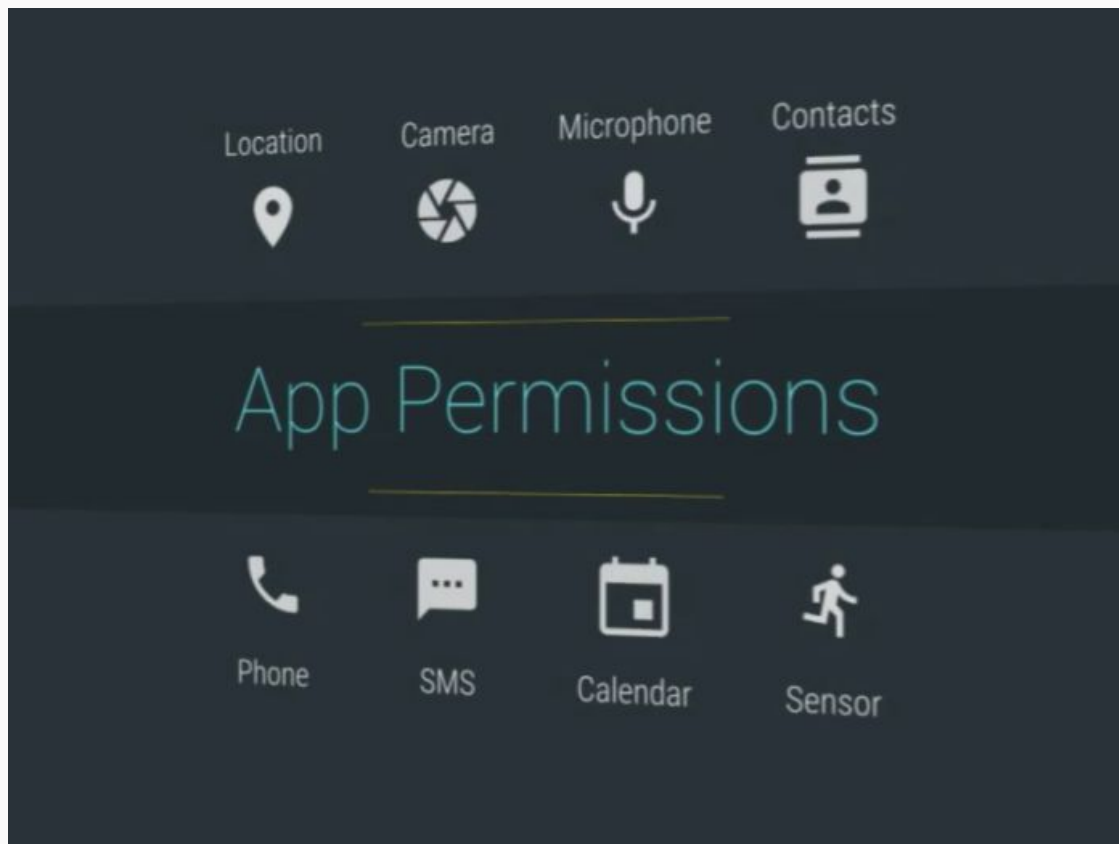
Desculpa, tem mais!



Permissões

Permissões

Permissões são um ponto muito importante em aplicativos Android. é necessário declarar no Manifest.xml todas as permissões que o aplicativo irá precisar.



<uses-permission></uses-permission>

Todas as permissões que a aplicação utiliza devem ser declaradas no Manifest.xml. As permissões devem ser declaradas com o nome pré-definido no framework, seguindo o seguinte formato:

"android.permission.WRITE_EXTERNAL_STORAGE"

Permissões

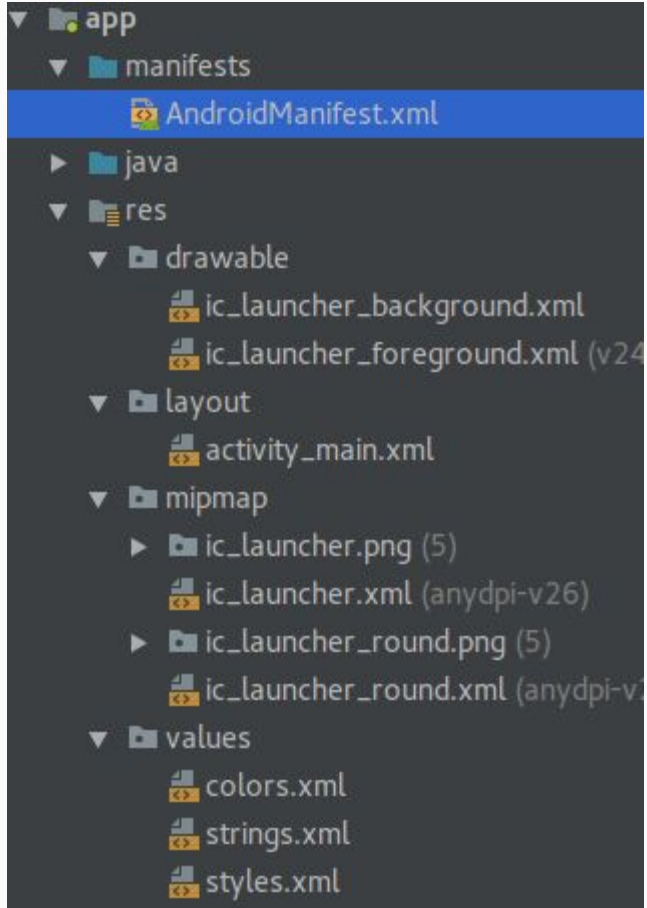
A partir da API 23 (Android 6), as permissões não só devem ser declaradas, como devem ser solicitadas de forma programática no momento em que precisarem ser usadas. Esta requisição deve ser feita da seguinte forma:

```
String[] permissions = new String[] {  
    ACCESS_COARSE_LOCATION  
};  
ActivityCompat.requestPermissions(this, permissions, PERMISSION_LOCATION);
```


Recursos

Recursos

O Android possui um sistema de gerência de recursos, onde são gerenciados os textos da aplicação, cores, estilos, formas, imagens de diferentes tamanhos, atividades, fragmentos, etc.



Drawable

Drawable são os recursos de imagens e formas da aplicação. Dentro desta árvore de recursos podem ser inseridos imagens bmp, jpg, png, etc. separadas por pastas indicando tamanhos (mmdpi, mdpi, hdpi, etc.) onde o sistema operacional irá definir qual irá utilizar de acordo com a resolução da tela.

Layout

Layout é onde é realizada a gerência das activities, fragments, etc. Elementos visuais da tela das aplicações.

Mipmap

Mipmap é a pasta de recursos que tem como objetivo conter os ícones da aplicação.

Values

Values é a pasta que contém os arquivos de chave e valor da aplicação.

Colors

Colors é um arquivo de recursos de chave e valor que nomeia cores e seus valores para uso em uma activity ou fragment.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

Strings

Strings é um arquivo de recursos chave e valor que nomeia variáveis de texto. Este arquivo é criado para internacionalização podendo ser criado um por idioma, e os textos serão indicados de acordo com o idioma do dispositivo.

```
<resources>  
    <string name="app_name">My Application</string>  
</resources>
```


Styles

Arquivo de definição de um tema. Temas são definidos utilizando variáveis pré definidas. Os valores acionados com '@color/...' estão acessando os valores do arquivo colors.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

</resources>
```

Dúvidas?

