

Programação Para Dispositivos Móveis I

PREFERENCES

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Índice

- SharedPreferences;
- Preference Screen;
- Preference Fragment;
- Leitura Adicional.

Utilização de Preferências

As preferências podem ser definidas para uma única atividade, ou serem partilhadas entre todas as atividades numa aplicação. Outros componentes, como serviços, também podem utilizar as preferências partilhadas (`sharedPreferences`).

Para aceder às preferências podemos escolher uma de três APIs:

- **`getPreferences()`**: Para aceder a preferências específicas de uma atividade;
- **`getSharedPreferences()`**: Para aceder às preferências partilhadas por toda a aplicação;
- **`getDefaultSharedPreferences()`**: utilizando um `PreferenceManager`, permite aceder a preferências que também estão relacionadas com a framework global de preferências do Android.

Utilização de Preferências

A utilização das APIs `getPreferences()` e `getSharedPreferences()` requerem um parâmetro de modo de segurança. Deve ser utilizado o `MODE_PRIVATE` para que nenhuma outra aplicação tenha acesso a este ficheiro de preferências.

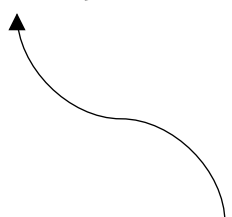
- Tipicamente é utilizada a API `getDefaultSharedPreferences()`.

Utilização de Preferências

Editor

O editor permite interagir com a lista de preferências presente no objeto `SharedPreferences`. Cada preferência é declarada definida por um par **chave** -> **valor** (chave é uma `String` e valor pode ser de qualquer tipo primitivo)

```
SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(this);  
SharedPreferences.Editor editor = preferences.edit();
```



Instanciação do Editor de
`SharedPreferences`

Utilização de Preferências

Editor

Para além de permitir adicionar preferências, o editor também permite efetuar os seguintes métodos:

- `remove()`: para remover uma preferência;
- `clear()`: para remover todas as preferências;
- `apply()` or `commit()`: para persistir as alterações feitas pelo editor.

Utilização de Preferências

Obter preferência através da chave (KEY)

```
SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(this);  
preferences.getString("CHAVE", "VALOR_POR_DEFEITO");
```

Para além do método `getString()`, também temos disponíveis os métodos `getFloat()`, `getInt()`, `getBoolean()`, `getLong()`.

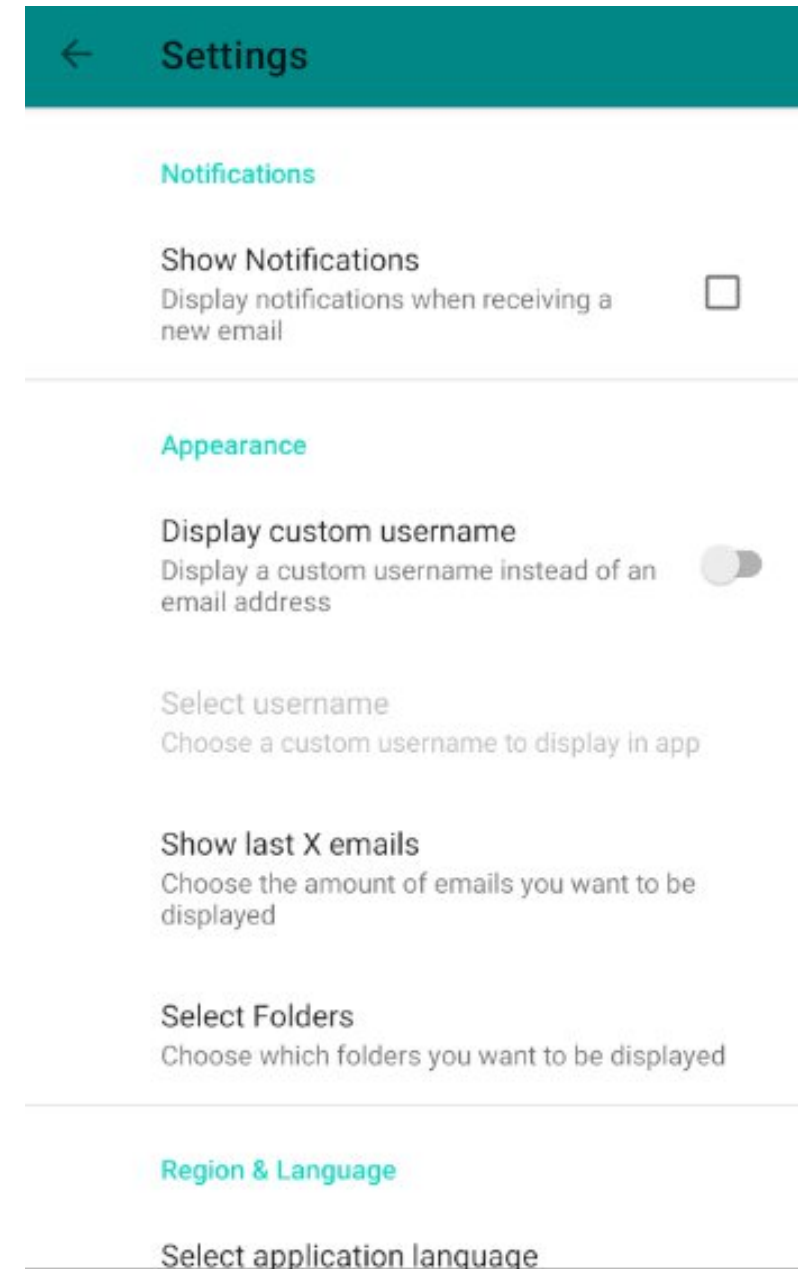
Utilização de Preferências

PreferenceFragment

Podemos também definir um conjunto de preferências específicas da nossa aplicação e associadas ao contexto da mesma.

Para isso, temos de indicar ao Android quais as preferências que estamos a tentar recolher.

Esse processo é descrito de seguida.



PreferenceFragmentCompat

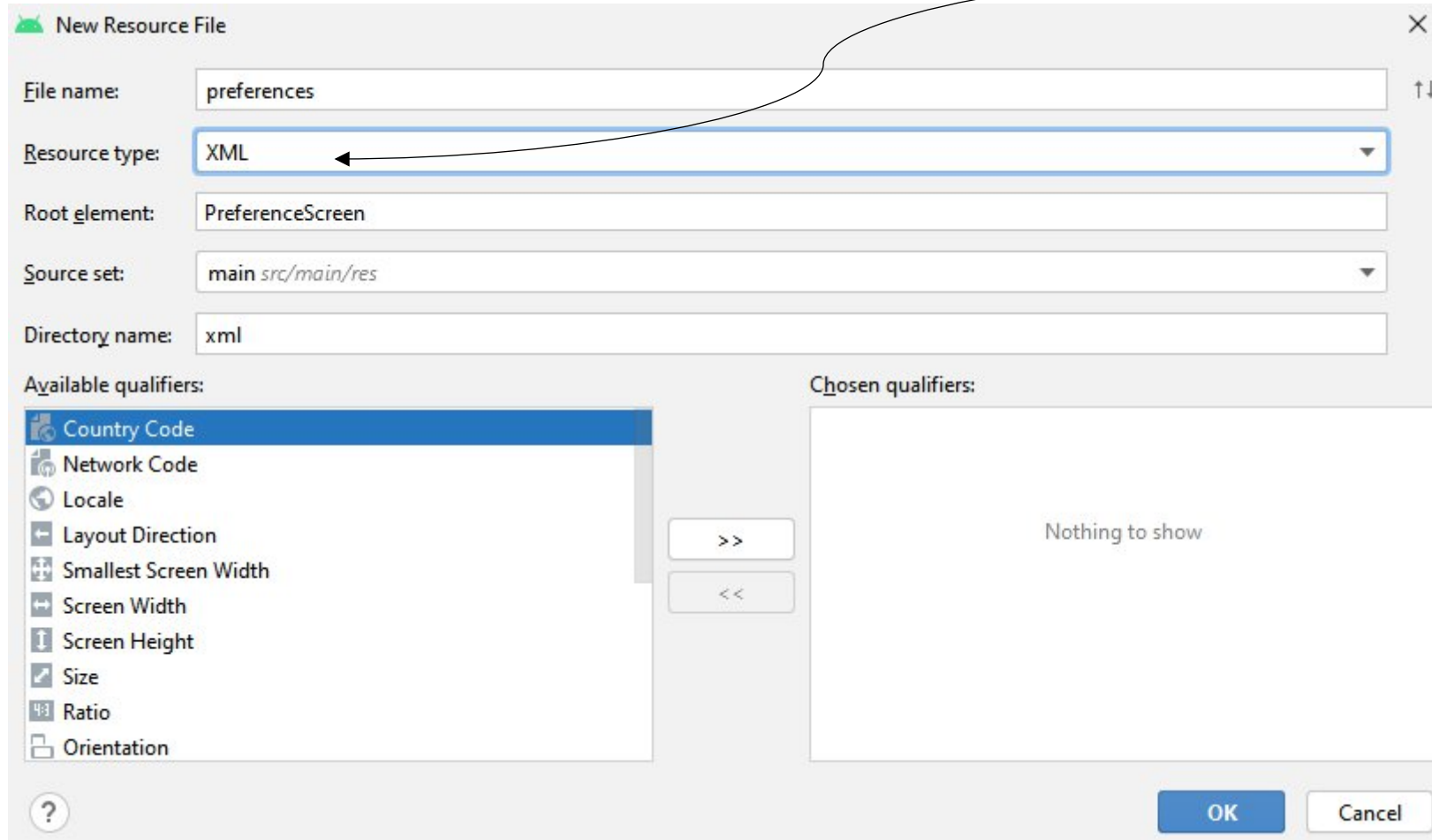
1. Adicionar as dependências ao ficheiro `build.gradle(module)`

```
dependencies {  
    ( ... )  
    implementation 'androidx.preference:preference:1.2.1'  
    ( ... )  
}
```

PreferenceFragmentCompat

2. Criar o ficheiro de preferências [xml/preferences.xml]

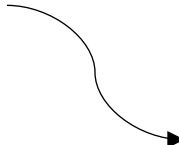
Garantir que escolhemos XML como valor de Resource Type



PreferenceFragmentCompat

2. Adicionar preferências[xml/preferences.xml]

PreferenceScreen é o
element ROOT do ficheiro
de preferências



```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">

    <CheckBoxPreference
        android:key="@string/pref_notification_key"
        android:title="@string/pref_notification_title"
        android:summary="@string/pref_notification_summary"
        android:defaultValue="false" />

</PreferenceScreen>
```



Exemplo de uma
preferência

PreferenceFragmentCompat

3. Adicionar as preferências ao Fragment[PreferencesFragment.java]

```
public class PreferencesFragment extends PreferenceFragmentCompat
```

Não necessita de um layout em XML, mas sim de um XML com a especificação das preferências

O fragment estende PreferenceFragmentCompat

```
@Override
```

```
public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {  
    setPreferencesFromResource(R.xml.preferences, rootKey);  
}
```

Especificação das preferências

Tipos de Preferência

CheckBoxPreference

Show Notifications

Display notifications when receiving a new email



<CheckBoxPreference

`android:key="@string/pref_notification_key"`
`android:title="@string/pref_notification_title"`
`android:summary="@string/pref_notification_summary"`
`android:defaultValue="false" />`

Key e Title são os valores obrigatórios para todas as preferências

Tipos de Preferência

PreferenceCategory

Notifications

Show Notifications

Display notifications when receiving a new email



```
<PreferenceCategory
  android:title="@string/category_notifications">

  <CheckBoxPreference
    android:key="@string/pref_notification_key"
    android:title="@string/pref_notification_title"
    android:summary="@string/pref_notification_summary"
    android:defaultValue="false" />

</PreferenceCategory>
```

Podemos categorizar um conjunto de preferências através deste elemento. O único valor obrigatório é o android:title

Tipos de Preferência

SwitchPreferenceCompat

Display custom username
Display a custom username instead of an email address



```
<SwitchPreferenceCompat  
    android:key="@string/pref_username_key"  
    android:title="@string/pref_username_title"  
    android:summary="@string/pref_username_summary"  
    android:defaultValue="false" />
```

Semelhante à CheckBoxPreference mas utiliza um slider para classificar como ativo ou inativo

Podemos definir um valor por defeito para cada preferência

Tipos de Preferência

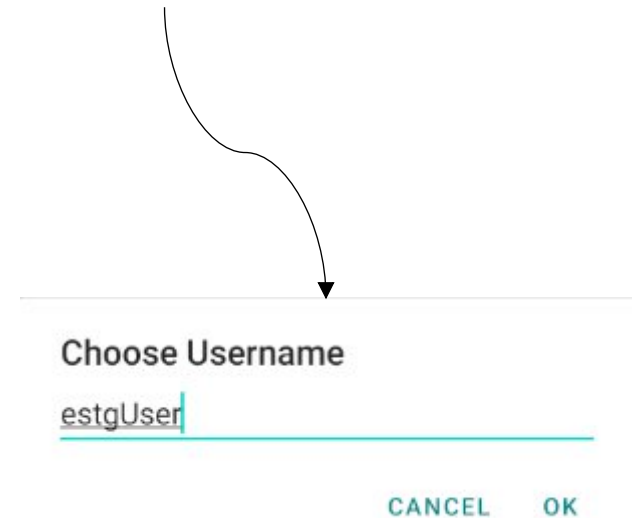
EditTextPreference

Select username

Choose a custom username to display in app

```
<EditTextPreference  
    android:key="@string/pref_setusername_key"  
    android:title="@string/pref_setusername_title"  
    android:summary="@string/pref_setusername_summary"  
    android:dialogTitle="@string/pref_setusername_dialog" />
```

Podemos definir o valor de alguns elementos da dialog (ex. Título)



Tipos de Preferência

EditTextPreference

Display custom username

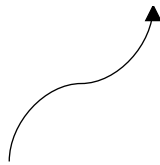
Display a custom username instead of an email address



Select username

Choose a custom username to display in app

```
<EditTextPreference
    android:key="@string/pref_setusername_key"
    android:title="@string/pref_setusername_title"
    android:summary="@string/pref_setusername_summary"
    android:dialogTitle="@string/pref_setusername_dialog"
    android:dependency="@string/pref_username_key" />
```



Podemos criar dependências entre preferências.
Neste exemplo esta preferência apenas está ativa
se a opção da preferência indicada for True

Tipos de Preferência

ListPreference

```
<ListPreference  
    android:key="@string/pref_language_key"  
    android:title="@string/pref_language_title"  
    android:summary="@string/pref_language_summary"  
    android:entries="@array/language_options"  
    android:entryValues="@array/language_values"  
    android:defaultValue="@string/default_language_value"  
    android:dialogTitle="@string/pref_language_dialog" />
```

Select application language
Select your preferred language

Language

- ☒ EN - English
☐ PT - Português

CANCEL

Lista de valores que aparece ao utilizador

Lista de valores que a aplicação vai utilizar

Tipos de Preferência

ListPreference [res/values/strings.xml]

```
<string-array name="language_options">  
    <item>EN - English</item>  
    <item>PT - Português</item>  
</string-array>
```

Lista de valores que aparece ao utilizador

```
<string-array name="language_values">  
    <item>EN</item>  
    <item>PT</item>  
</string-array>
```

Lista de valores que a aplicação vai utilizar

Tipos de Preferência

DropDownPreference

Semelhante à ListPreference mas utiliza uma dropdown em vez de uma dialog

```
<DropDownPreference
    android:key="@string/pref_language_dropdown_key"
    android:title="@string/pref_language_title"
    android:summary="@string/pref_language_summary"
    android:entries="@array/language_options"
    android:entryValues="@array/language_values"
    android:defaultValue="@string/default_language_value"/>
```

Select application language
Select your preferred language

Select application language
Select your preferred language

EN - English

PT - Português

Lista de valores que aparece ao utilizador

Lista de valores que a aplicação vai utilizar

Tipos de Preferência

MultiSelectListPreference

Só está disponível em API's superiores à versão 11 (onze) (Android 3.0)

```
<MultiSelectListPreference
    android:key="@string/pref_show_folder_key"
    android:title="@string/pref_show_folder_title"
    android:summary="@string/pref_show_folder_summary"
    android:entries="@array/folder_options"
    android:entryValues="@array/folder_values"
    android:dialogTitle="@string/pref_show_folder_dialog" />
```

Select Folders
Choose which folders you want to be displayed

Which folder(s) you want to be displayed?

- ☒ Inbox
- ☒ Drafts
- ☒ Sent Email
- ☐ Spam
- ☒ Trash

CANCEL OK

Tipos de Preferência

SeekBarPreference

Notification Volume

Select the volume of the applications sound



```
<SeekBarPreference  
    android:key="@string/pref_volume_key"  
    android:title="@string/pref_volume_title"  
    android:summary="@string/pref_volume_summary" />
```

PreferenceFragmentCompat

4. Registar o listener [PreferencesFragment.java]

```
public class PreferencesFragment extends PreferenceFragmentCompat implements SharedPreferences.OnSharedPreferenceChangeListener
```

Implementar o listener que verifica se existiram alterações nas preferências

PreferenceFragmentCompat

4. Registrar o listener [PreferencesFragment.java]

```
@Override
public void onResume() {
    super.onResume();
    getPreferenceScreen().getSharedPreferences().registerOnSharedPreferenceChangeListener(this);
}
```

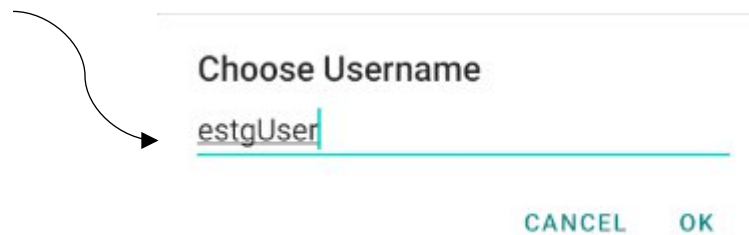
```
@Override
public void onPause() {
    super.onPause();
    getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(this);
}
```


PreferenceFragmentCompat

4. Registrar o listener [PreferencesFragment.java]

Select username

Choose a custom username to display in app



Select username

Choose a custom username to display in app.
Chosen username: estgUser

`@Override`

```
public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {  
    if(key.equals(getResources().getString(R.string.pref_setusername_key))){  
        EditTextPreference preferenceCustomUsername = findPreference(key);  
        preferenceCustomUsername.setSummary(  
            preferenceCustomUsername.getSummary() + ". Chosen username: " +  
            sharedPreferences.getString(key, ""));  
    }  
}
```

Comparação de
keys

PreferenceFragmentCompat

Aceder aos objetos de preferências

```
ListPreference languagesPref =  
    (ListPreference) findPreference(getResources().getString(R.string.pref_language_key));  
  
languagesPref.setEntries(R.array.language_options);  
languagesPref.setEntryValues(R.array.language_values);
```

Definição de
valores através
de código

Leitura Adicional

Settings:

<https://developer.android.com/guide/topics/ui/settings>

Programação Para Dispositivos Móveis I

PREFERENCES

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt