

Aula 39

Permissões

Permissões

O objetivo de uma permissão é proteger a privacidade de um usuário do Android. Os aplicativos Android devem solicitar permissão para acessar dados confidenciais do usuário (como contatos e SMS), além de certos recursos do sistema (como câmera e internet).

Permissões - Princípios básicos

- Solicite permissões no contexto, quando o usuário começar a interagir com o recurso que requer a permissão.
- Não bloqueie o usuário. Sempre ofereça a opção de cancelar um fluxo de IU educacional relacionado a permissões.
- Se o usuário negar ou revogar uma permissão necessária para um recurso, faça uma degradação suave para que ele possa continuar usando o app, possivelmente desativando o recurso que requer a permissão.
- Não presuma nenhum comportamento do sistema.

Permissões

Para solicitar uma permissão o aplicativo deve declarar as permissões necessárias, incluindo a tag **<uses-permission>** no AndroidManifest do aplicativo. Por exemplo, um aplicativo que precisa de permissão de internet teria esta linha AndroidManifest:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.snazzyapp">

    <uses-permission android:name="android.permission.INTERNET"/>
    <!-- other permissions go here -->

    <application ...>
        ...
    </application>
</manifest>
```

Permissões

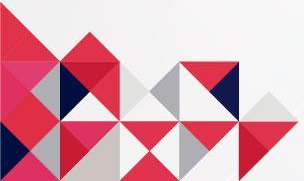
O comportamento do sistema após a declaração de uma permissão depende da importância dela. Algumas permissões são consideradas "normais", então o sistema as concede na instalação. Outras permissões são consideradas "perigosas", então o usuário precisa conceder acesso ao app de forma explícita.

Lista de permissões:

<https://developer.android.com/reference/android/Manifest.permission?hl=pt-br>



Mas como verificamos a existência de uma permissão?




```
when {
    ContextCompat.checkSelfPermission(
        CONTEXT,
        Manifest.permission.REQUESTED_PERMISSION
    ) == PackageManager.PERMISSION_GRANTED -> {
        // You can use the API that requires the permission.
        performAction(...)
    }
    shouldShowRequestPermissionRationale(...) -> {
        // In an educational UI, explain to the user why your app requires this
        // permission for a specific feature to behave as expected. In this UI,
        // include a "cancel" or "no thanks" button that allows the user to
        // continue using your app without granting the permission.
        showInContextUI(...)
    }
    else -> {
        // You can directly ask for the permission.
        requestPermissions(...)
    }
}
```

então é só isso???

Nãooo tem mais....



Quando o usuário responde à solicitação de permissão do seu app, o sistema invoca a implementação de `onRequestPermissionsResult()` do app.

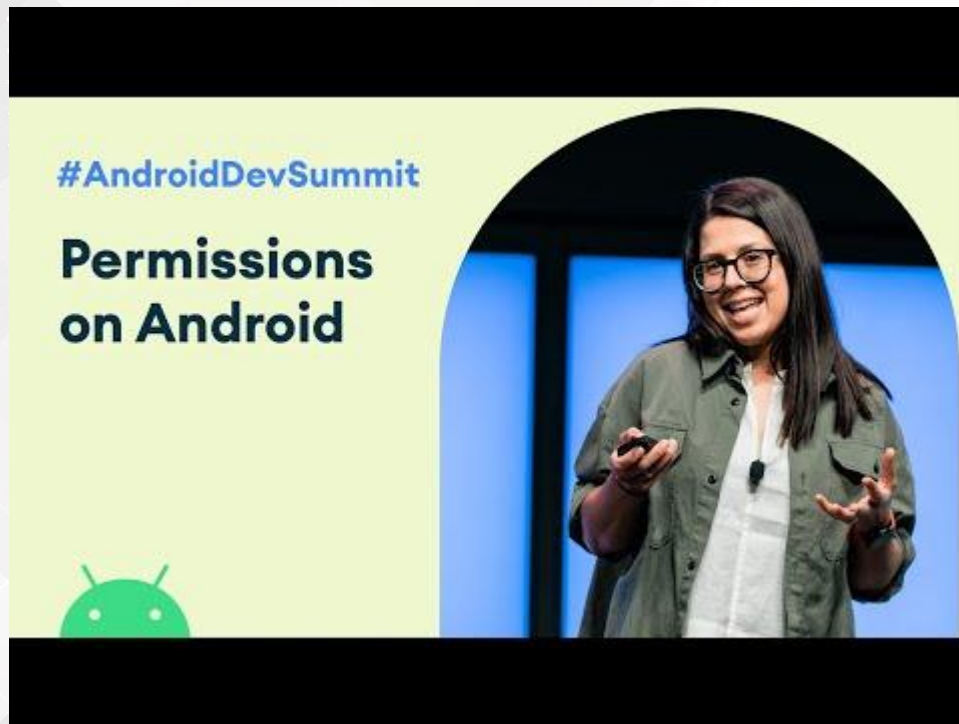
```
override fun onRequestPermissionsResult(requestCode: Int,
permissions: Array<String>, grantResults: IntArray) {
    when (requestCode) {
        PERMISSION_REQUEST_CODE -> {
            // If request is cancelled, the result arrays are empty.
            if ((grantResults.isNotEmpty() &&
                grantResults[0] == PackageManager.PERMISSION_GRANTED)) {
                // Permission is granted. Continue the action or workflow
                // in your app.
            } else {
                // Explain to the user that the feature is unavailable because
                // the features requires a permission that the user has denied.
                // At the same time, respect the user's decision. Don't link to
                // system settings in an effort to convince the user to change
                // their decision.
            }
            return
        }

        // Add other 'when' lines to check for other
        // permissions this app might request.
        else -> {
            // Ignore all other requests.
        }
    }
}
```

Práticas recomendadas ao processar negação de permissão

- Direcione a atenção do usuário.
 - Mostrar uma mensagem em que os resultados ou dados do recurso seriam exibidos.
 - Exibir um botão diferente que contenha um ícone e uma cor de erro.
- Seja específico. Não exiba uma mensagem genérica. Em vez disso, mencione quais recursos não estão disponíveis porque o app não tem a permissão necessária.
- Não bloqueie a interface do usuário.

Veja em casa...



Exercícios!