MARSHMALLOW FOR DEVELOPERS

Athila Santos

athilahs@gmail.com

github: athilahs

linkedin: athilasantos

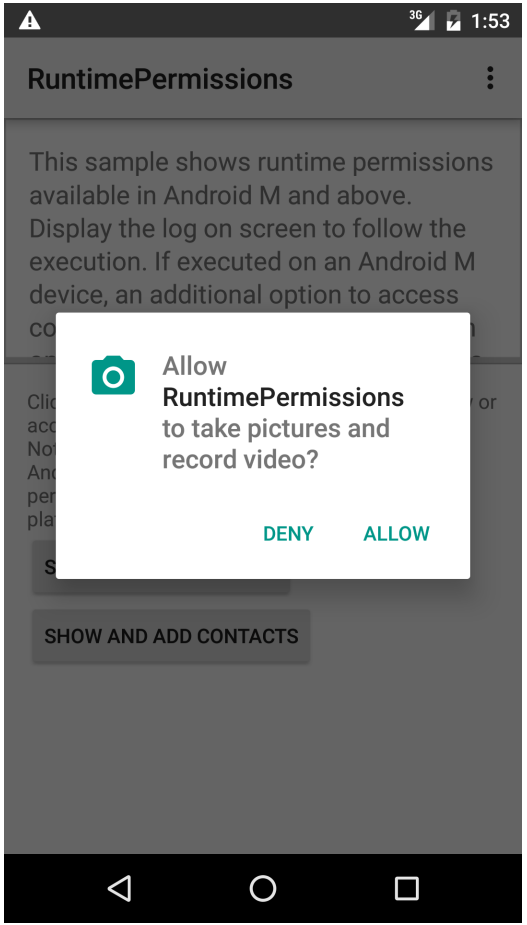
RUNTIME PERMISSIONS

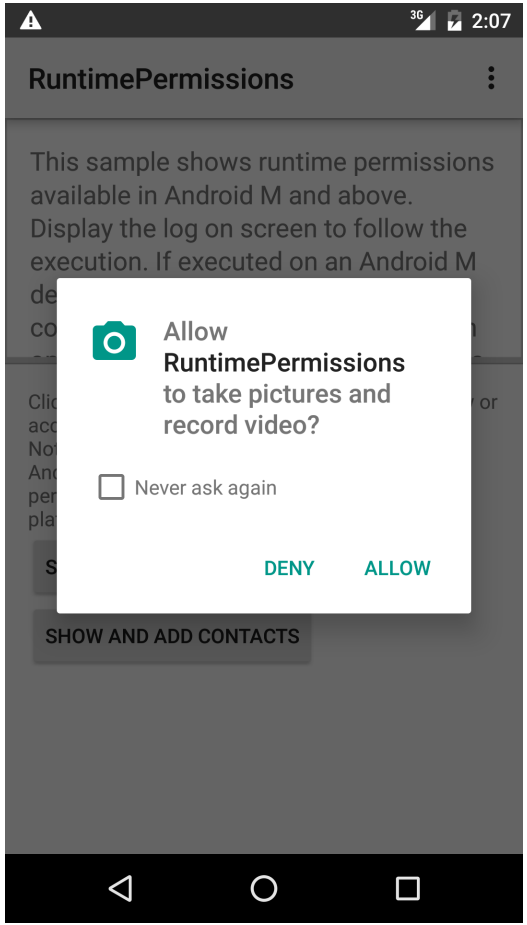
RUNTIME PERMISSIONS

- Permissões solicitadas **on demand** pelo aplicativo
- Introduzido na **API 23**
- Oferece maior **controle e segurança** para o usuário

RUNTIME PERMISSIONS

- Agrupamento das permissões, evitando que o usuário seja **sobrecarregado** com solicitações de permissões muito semelhantes
- Permissões podem ser **revogadas ou adicionadas** a qualquer momento pelo usuário através dos Settings do telefone
- **Novas responsabilidades** para o desenvolvedor


1:53
RuntimePermissions
This sample shows runtime permissions available in Android M and above. Display the log on screen to follow the execution. If executed on an Android M device, an additional option to access
Allow RuntimePermissions to take pictures and record video?
DENY
ALLOW
SHOW AND ADD CONTACTS


2:07
RuntimePermissions
This sample shows runtime permissions available in Android M and above. Display the log on screen to follow the execution. If executed on an Android M
Allow RuntimePermissions to take pictures and record video?
Never ask again
DENY
ALLOW
SHOW AND ADD CONTACTS

PERMISSIONS GROUPS

Permission Group	Permissions
android.permission-group.CALENDAR	• android.permission.READ_CALENDAR • android.permission.WRITE_CALENDAR
android.permission-group.CAMERA	• android.permission.CAMERA
android.permission-group.CONTACTS	• android.permission.READ_CONTACTS • android.permission.WRITE_CONTACTS • android.permission.GET_ACCOUNTS
android.permission-group.LOCATION	• android.permission.ACCESS_FINE_LOCATION • android.permission.ACCESS_COARSE_LOCATION
android.permission-group.MICROPHONE	• android.permission.RECORD_AUDIO

PERMISSIONS GROUPS

android.permission-group.PHONE	• android.permission.READ_PHONE_STATE • android.permission.CALL_PHONE • android.permission.READ_CALL_LOG • android.permission.WRITE_CALL_LOG • com.android.voicemail.permission.ADD_VOICEMAIL • android.permission.USE_SIP • android.permission.PROCESS_OUTGOING_CALLS
android.permission-group.SENSORS	• android.permission.BODY_SENSORS
android.permission-group.SMS	• android.permission.SEND_SMS • android.permission.RECEIVE_SMS • android.permission.READ_SMS • android.permission.RECEIVE_WAP_PUSH • android.permission.RECEIVE_MMS • android.permission.READ_CELL_BROADCASTS
android.permission-group.STORAGE	• android.permission.READ_EXTERNAL_STORAGE • android.permission.WRITE_EXTERNAL_STORAGE

Passo 1: Checagem de permissão

```
public void showCamera() {
    if (checkSelfPermission(Manifest.permission.CAMERA)
            != PackageManager.PERMISSION_GRANTED) {
        // Camera permission has not been granted.
        requestPermissions(new String[]{Manifest.permission.CAMERA},
                REQUEST_CAMERA);
    } else {
        // Camera permissions is already available, show the camera preview.
        showCameraPreview();
    }
}
```

Passo 2: Tratar retorno da requisição de permissão

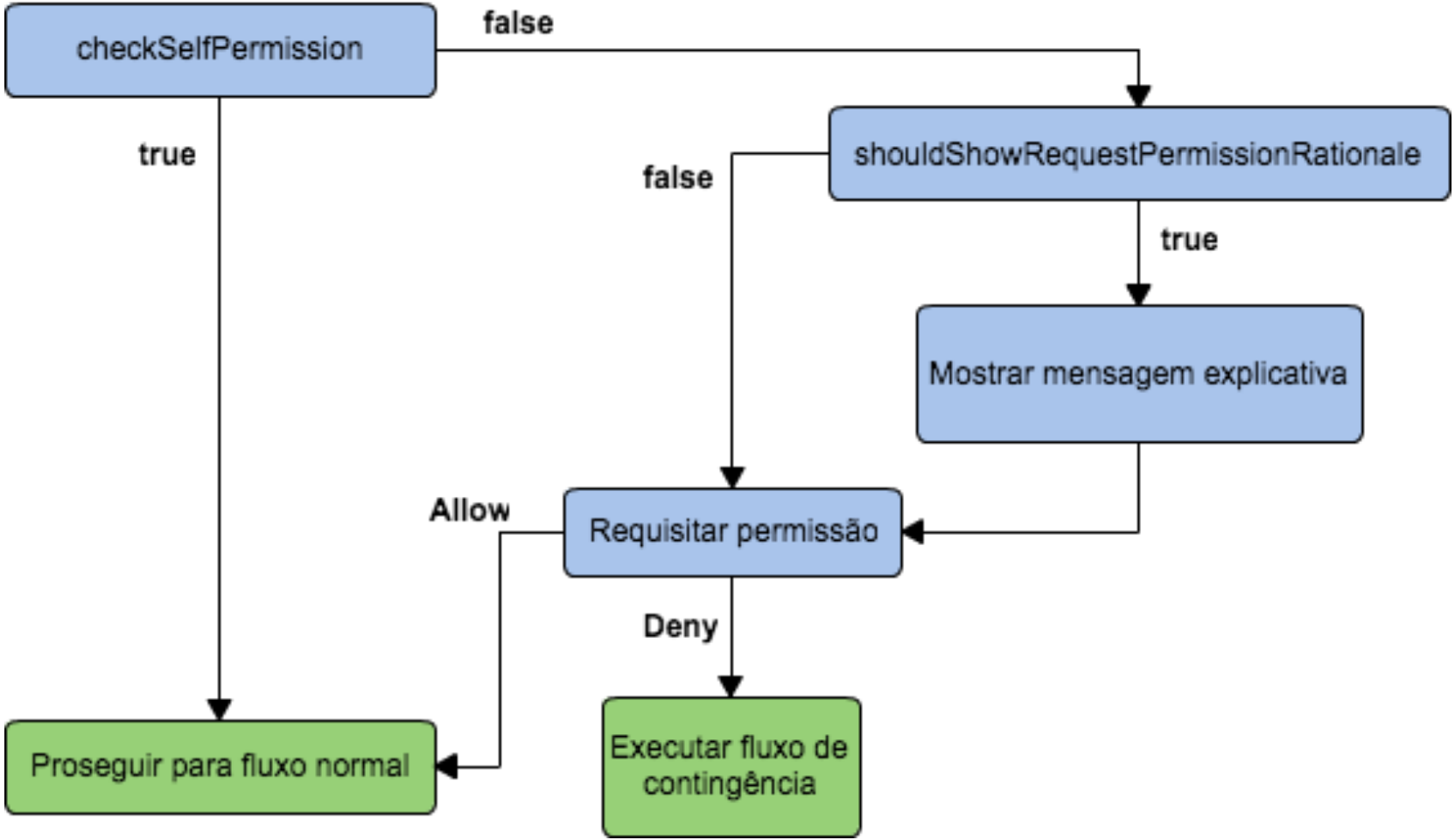
```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                       @NonNull int[] grantResults) {

    if (requestCode == REQUEST_CAMERA) {
        // Received permission result for camera permission.
        // Check if the only required permission has been granted
        if (grantResults.length == 1
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            // Camera permission has been granted, preview can be displayed
            showCameraPreview();
        } else {
            Snackbar.make(mLayout, R.string.permissions_not_granted,
                    Snackbar.LENGTH_SHORT).show();

        }
    } else {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    }
}
```

Se o usuário negar a permissão...

- **Falhe suavemente** ou continue de forma limitada
- Você pode descobrir se o usuário negou anteriormente uma permissão através do método **shouldShowRequestPermissionRationale** da Activity

  Se este método retornar **true**, a aplicação deve mostrar uma mensagem explicando a **necessidade** daquela permissão
- Se o usuário escolheu **"Não perguntar novamente"** no diálogo de requisição de permissão este método sempre retornará **false**

```
private void requestCameraPermission() {
    if (shouldShowRequestPermissionRationale(Manifest.permission.CAMERA)) {
        // Provide an additional rationale to the user if the permission was not granted
        // and the user would benefit from additional context for the use of the permission.
        // For example if the user has previously denied the permission.
        Snackbar.make(mLayout, "Camera permission is needed to show the camera preview.",
                Snackbar.LENGTH_INDEFINITE)
                .setAction(R.string.ok, new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        requestPermissions(new String[]{Manifest.permission.CAMERA},
                                REQUEST_CAMERA);
                    }
                })
                .show();
    } else {
        // Camera permission has not been granted yet. Request it directly.
        requestPermissions(new String[]{Manifest.permission.CAMERA},
                REQUEST_CAMERA);
    }
}
```


checkSelfPermission
false
true
shouldShowRequestPermissionRationale
false
true
Mostrar mensagem explicativa
Allow
Requisitar permissão
Deny
Proseguir para fluxo normal
Executar fluxo de contingência

Múltiplas permissões

Requisição:

```
requestPermissions(new String[]{Manifest.permission.CAMERA},
        REQUEST_CAMERA);
```

Array de permissões

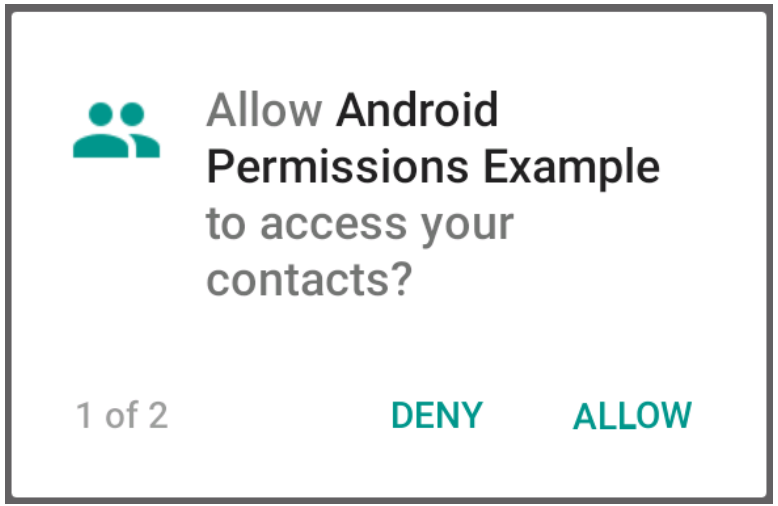
Resposta:

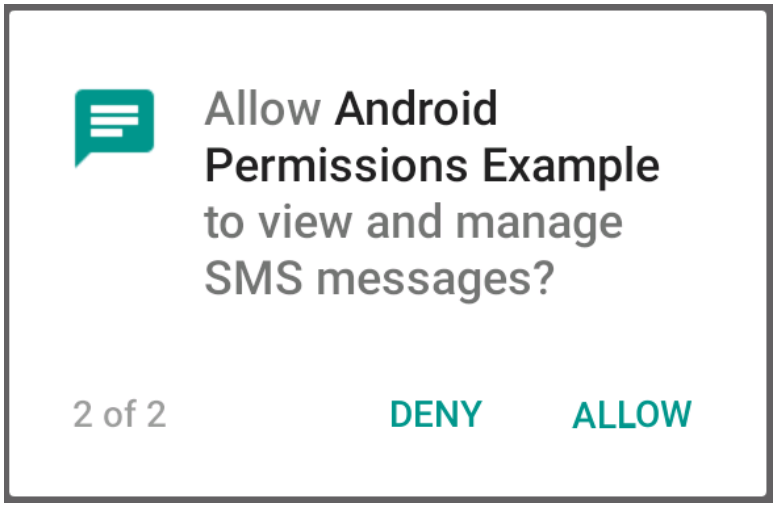
```
@Override
public void onRequestPermissionsResult(
        int requestCode, @NonNull String[] permissions,
        @NonNull int[] grantResults) {
```

Array de resultados

Múltiplas permissões

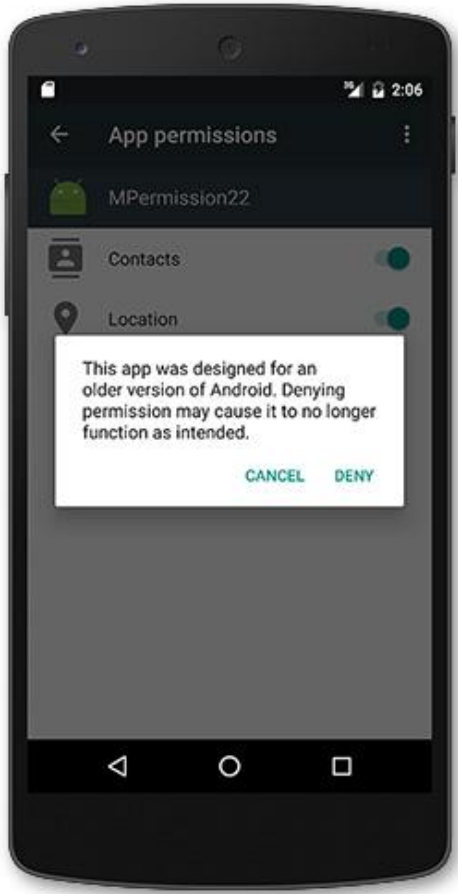
Possíveis problemas de usabilidade:





Compatibilidade

- Se **targetSDK < 23**, permissões são requisitadas na instalação
- Se o device estiver rodando API >= 23, usuário pode revogar permissões, **mesmo que sua app não suportar permissões seletivas**



Compatibilidade

Você pode suportar permissões seletivas **mesmo se sua app não suportar**

Android Marshmallow+:

appcompat v4-r23

ContextCompat.checkSelfPermission()

ActivityCompat.requestPermissions()

ActivityCompat.

shouldShowRequestPermissionRationale()

appcompat v13-r23

FragmentCompat.requestPermissions()

FragmentCompat.

shouldShowRequestPermissionRationale()

Dicas

- Requisite permissões **imediatamente** antes de usá-la. Isso ajuda o usuário a entender o motivo da permissão e reduz a chance dele negá-la
- Cuidado com requisição de multiplas permissões ao mesmo tempo
- Atente-se para o cenário em que o usuário selecionou **"Não pergunte novamente"** no diálogo. Você não saberá quando isso aconteceu e futuras requisições irão retornar **DECLINED**
- Teste sua app contra o Marshmallow. Utilize o emulador

DATA BINDING

Data Binding

- Views são atualizadas automaticamente de acordo com mudanças nos modelos
- Simplificação da lógica de "**binding**" do layout
- Model View ViewModel (MVVM)

O que eu preciso para começar

- **API mínima: 7** e **Android Studio 1.3** ou superior
- Gradle plugin **1.3.0-beta4** ou superior e **databinding** plugin definidos no seu top-level build.gradle:

```
dependencies {
        classpath "com.android.tools.build:gradle:1.3.0-beta4"
        classpath "com.android.databinding:dataBinder:1.0-rc1"
}
```

- Para cada módulo utilizando databinding, aplicar plugin do databinding

```
apply plugin: 'com.android.databinding’
```

Sintaxe

```
// activity_main.xml
<layout xmlns:android="http://schemas.android.com/apk/res/android">
    <data>
        <variable name="user" type="com.example.User"/>
    </data>
    <LinearLayout
        ...>
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{user.firstName}"/>
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{user.lastName}"/>
    </LinearLayout>
</layout>
```

Sintaxe

```
// activity_main.xml
<layout xmlns:android="http://schemas.android.com/apk/res/android">
    <data>
        <variable name="user" type="com.example.User"/>
    </data>
    <LinearLayout
        ...>
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{user.firstName}"/>
        <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{user.lastName}"/>
    </LinearLayout>
</layout>
```

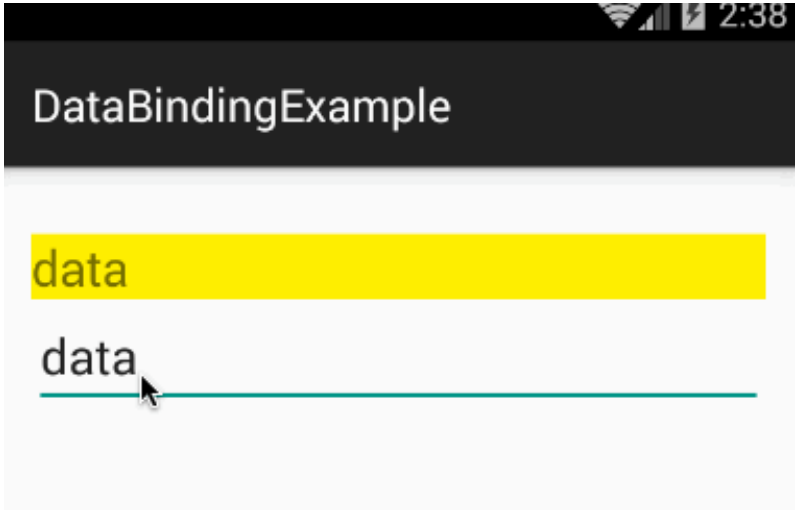
Sintaxe

```java
// User.java
public class User {
    public final String firstName;
    public final String lastName;
    public User(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
}
```

Binding Data

```java
// MainActivity.java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    MainActivityBinding binding =
    DataBindingUtil.setContentView(this, R.layout.main_activity);
    User user = new User("Test", "User");
    binding.setUser(user);
}
```

Atualização automática de views




```
private User mUser;

editTextUsername.addTextChangedListener(
new TextWatcher() {
    @Override
    public void afterTextChanged(Editable s) {
        mUser.setUsername(s.toString());
    }
}
```

Atualização automática de views

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@{user.username}"
    android:background="#fff000"
    android:textSize="22sp"/>

<EditText
    android:id="@+id/editText_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="22sp"/>
```

Atualização automática de views

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@{user.username}"
    android:background="#fff000"
    android:textSize="22sp"/>

<EditText
    android:id="@+id/editText_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="22sp"/>
```

Atualização automática de views

```
// User.java
public class User extends BaseObservable {
    private String username;

    @Bindable
    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
        notifyPropertyChanged(BR.username);
    }
}
```

Mais poder!!!!

```
<data>
    <import type="com.example.MyStringUtils"/>
    <variable name="user" type="com.example.User"/>
</data>
...
<TextView
   android:text="@{MyStringUtils.capitalize(user.lastName)}"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"/>
```

Mais poder!!!!

```
<data>
    <import type="android.view.View"/>
</data>
...
<TextView
   android:text="@{user.lastName}"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:visibility="@{user.isAdult ? View.VISIBLE : View.GONE}"/>
```

Mais poder!!!!

```
<data>
    <import type="android.util.SparseArray"/>
    <import type="java.util.Map"/>
    <import type="java.util.List"/>
    <variable name="list" type="List"/>
    <variable name="sparse" type="SparseArray<String>"/>
    <variable name="map" type="Map<String, String>"/>
    <variable name="index" type="int"/>
    <variable name="key" type="String"/>
...
android:text="@{list[index]}"
...
android:text="@{sparse[index]}"
...
android:text="@{map[key]}"
```


WITH GREAT POWER
COMES GREAT RESPONSABILITY
http://newsccdream.tistory.com
imgflip.com

AUTO BACKUP

Auto backup dos dados do aplicativo

- Android Marshmallow habilita **backup automático** dos dados do aplicativo (menos diretorios de **cache** e **armazenamento externo**) para todas as apps instaladas no dispositivo
- Rotina de backup ocorre **a cada 24hs**, quando o dispositivo está **idle**, **carregando** e **conectado à WiFi**
- Backup automático pode ser **desabilitado pelo usuário** ou **desabilitado/limitado pelo programador**

BATERIA

Doze Mode - Power Saving

- Estratégia agressiva da Google para reduzir **consumo de bateria** nos dispositivos Android
- Critérios para que o Doze Mode seja ativado:
  - Device desplugado
  - Tela apagada
  - Device imóvel
  - Manutenção dos estados acima por um tempo (1 hora no M Preview)

Doze Mode - Recursos Desligados

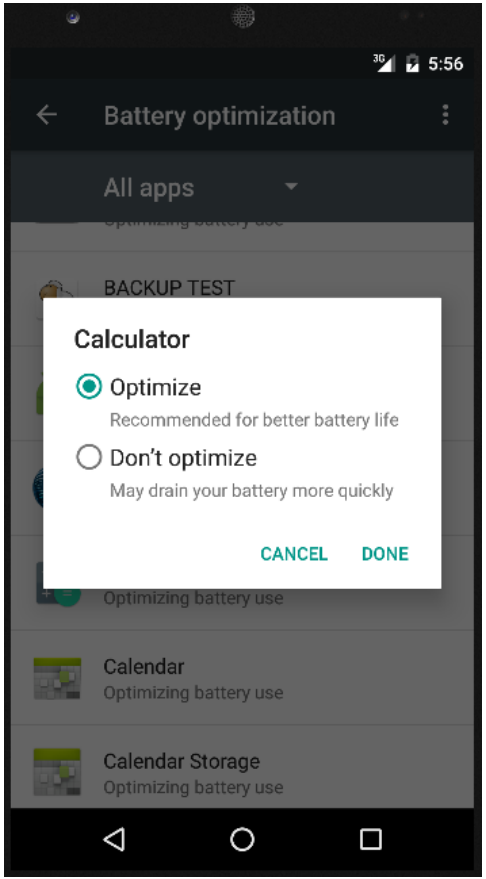
- ⊘ Acesso à rede
- ⊘ Wake locks
- ⊘ Alarms usando AlarmManager
  - a menos que `setAndAllowWhileIdle()` seja chamado (várias limitações para este método)
- ⊘ WiFi scans
- ⊘ JobScheduler
- ⊘ SyncAdapter

Doze Mode - DeviceIdleControler

- Novo serviço de sistema que irá gerenciar inatividade do dispositivo
- Mantém uma **whitelist de apps** que poderão operar normalmente mesmo com o dispositivo em Doze Mode
- Abre **"janelas" de atividade** periodicamente quando o device está em Doze Mode. Eventos pendentes serão entregues durante estas janelas

Contornando o Doze

- **System apps** podem ser adicionadas à whitelist pelo fabricante
- Mensagens GCM de **alta prioridade** (novo esquema de prioridades de mensagens GCM introduzido no Marshmallow)
- Usuário pode adicionar ou remover apps da whitelist pelas **configs do telefone**



SEGURANÇA

Identificação de Hardware

- MAC Address do device não pode mais ser obtido através das **APIs de WiFi e bluetooth**.
  - `WifiInfo.getMacAddress()` e `BluetoothAdapter.getAddress()` agora retornam o valor constante `02:00:00:00:00:00`
- Para obter MAC address dos dispositivos nas proximidades, a app precisará de **permissões**:
  - `WifiManager.getScanResults()`: Permissões: `ACCESS_FINE_LOCATION` ou `ACCESS_COARSE_LOCATION`
  - `BluetoothDevice.ACTION_FOUND`: Permissão `ACCESS_COARSE_LOCATION`
  - `BluetoothLeScanner.startScan()`: Permissões: `ACCESS_FINE_LOCATION` ou `ACCESS_COARSE_LOCATION`

Apache HttpClient removido!

- HttpClient foi **removido** da API. Use HttpUrlConnection / HttpsUrlConnection
- Atenção para **libs incluídas no seu projeto** que usem o HttpClient

Se você realmente precisar...:

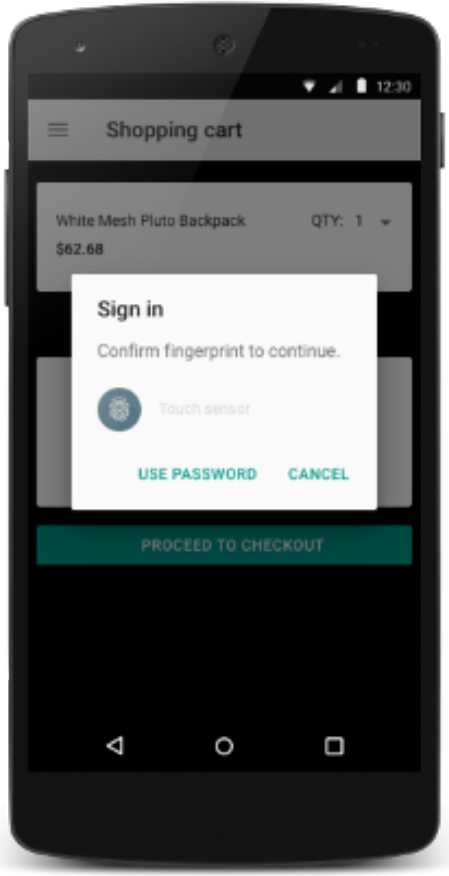
```
// build.gradle
android {
    useLibrary 'org.apache.http.legacy'
}
```

FINGERPRINT AUTHENTICATION

Fingerprint authentication

- Limitado a dispositivos que possuem o sensor biométrico
- Nova classes adicionadas à API:

  `FingerprintManager`
- Nova permissão: `android.permission.USE_FINGERPRINT`



Fingerprint authentication

Passo 1: Verificar se o dispositivo possui o leitor biométrico:

```
mFingerprintManager.isHardwareDetected();
```

Passo 2: Verificar se o usuário configurou uma lockscreen: (fingerprint authentication funciona em conjunto com a segurança do dispositivo):

```
mFingerprintManager.isDeviceSecure();
```

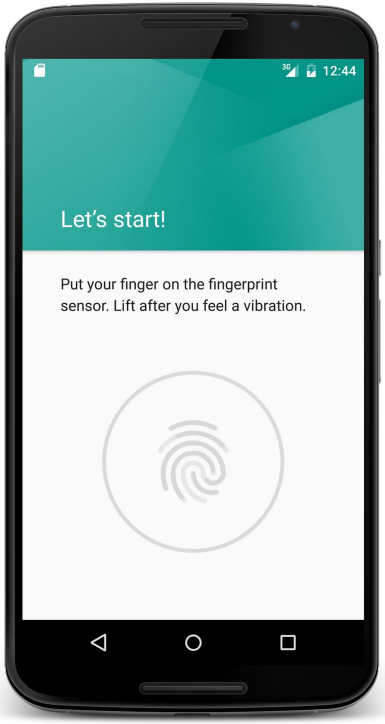
Fingerprint authentication

Passo 3: Verificar se o usuário cadastrou ao menos uma impressão digital:

```
mFingerprintManager.hasEnrolledFingerprints()
```

Orientar o usuário como agir, em caso negativo:

```
Toast.makeText(this, "Go to 'Settings -> Security -> Fingerprint' and register at least one fingerprint", Toast.LENGTH_LONG).show();
```



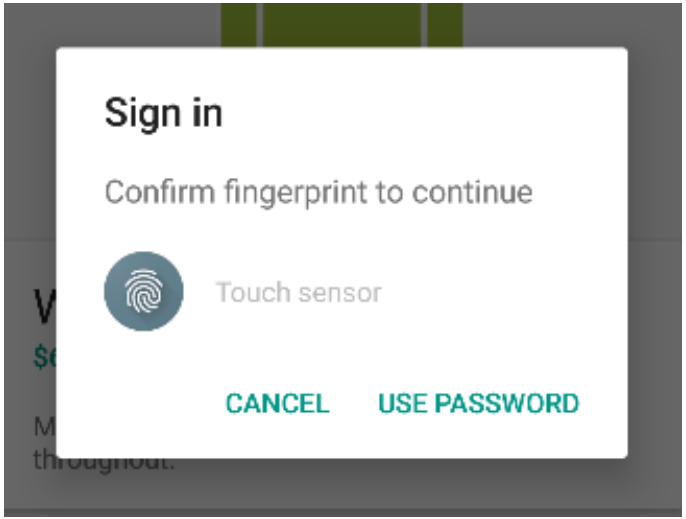
Fingerprint authentication

Passo 4: Mostre o diálogo de autenticação

(criado e configurado pelo desenvolvedor) e

comece a "ouvir" o leitor biométrico:

```
mFingerprintManager.authenticate
(cryptoObject, mCancellationSignal, 0
/* flags */, this, null);
```



Fingerprint authentication

```
authenticate (cryptoObject, mCancellationSignal, 0 /* flags */,
this, null);
```

android.hardware.fingerprint.FingerprintManager.CryptoObject

android.hardware.fingerprint.FingerprintManager.AuthenticationCallback

Fingerprint authentication

Passo 5: Tratar resposta do leitor

android.hardware.fingerprint.FingerprintManager.AuthenticationCallback

Public Methods	
void	onAuthenticationError (int errorCode, CharSequence errString) Called when an unrecoverable error has been encountered and the operation is complete.
void	onAuthenticationFailed () Called when a fingerprint is valid but not recognized.
void	onAuthenticationHelp (int helpCode, CharSequence helpString) Called when a recoverable error has been encountered during authentication.
void	onAuthenticationSucceeded (FingerprintManager.AuthenticationResult result) Called when a fingerprint is recognized.

Dicas:

- Sempre dê a opção para o usuário se autenticar por senha, **mesmo que o leitor biométrico esteja disponível** e o usuário tenha cadastrado uma impressão digital
- Utilize o **Android Keystore System** em conjunto com a API de autenticação biométrica para aumentar ainda mais a segurança (detalhes na app de exemplo do sdk)
- Use o **ícone padrão** (incluso na app de exemplo do sdk): 

E MUITO MAIS!

Muito mais!!!

- App Links
- Text Selection
- Direct Share
- Voice Interactions
- Android For Work
- Assist API (Google Now on tap)
  - Informação contextual de dentro da sua app!
- Novas APIs para câmera
  - Torch mode

REFERÊNCIAS

Android M API Overview: http://bit.ly/1UNmJyi

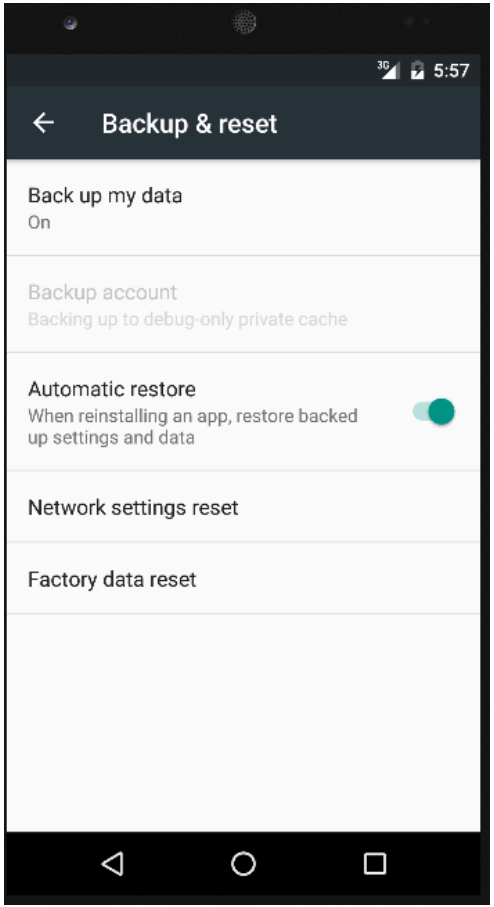
Diving Into Android 'M' Doze: http://bit.ly/1UPzHGt

Android Marshmallow Samples: http://bit.ly/1iH5cGG

OBRIGADO!

Desabilitar auto-backup

```
<manifest xmlns:android="http://schemas.android.
com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
package="com.my.appexample">
    <uses-sdk android:minSdkVersion="MNC"/>
    <uses-sdk android:targetSdkVersion="MNC"/>
    <app ...
        android:allowBackup="false">
    </app>
    ...
</manifest>
```



Configurar Auto Backup

```
// AndroidManifest.xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    <app ...
        android:fullBackupContent="@xml/mybackupscheme">
    </app>
</manifest>

// res/xml/mybackupscheme.xml
<?xml version="1.0" encoding="utf-8"?>
<full-backup-content>
    <exclude domain="database" path="device_info.db"/>
<!-- domains can be: ["file" | "database" | "sharedpref" | "external" | "root"] -->
</full-backup-content>
```