

# Programação para Dispositivos Móveis

**Leopoldo Teixeira**

[imt@cin.ufpe.br](mailto:imt@cin.ufpe.br) | [@leopoldomt](https://twitter.com/leopoldomt)

# Intents

O que são, onde vivem,  
como se reproduzem... :-)

# What's your intent?

- Como já visto, protocolo HTTP consiste de verbos + endereços (ex.: GET url...)
- Intent é algo similar, representa uma ação mais o contexto
  - existem bem mais verbos...
  - assim como um navegador processa requisições HTTP, Android sabe como lidar com Intents...

# Intents

- Basicamente, uma mensagem passada entre componentes
- Estrutura de dados que permite especificar uma operação a ser executada, ou um evento ocorrido
- Permite *late binding* entre código de diferentes aplicações Android

# Operações

- Podemos visualizar Intents como uma linguagem flexível, que permite especificar operações desejadas
- Um componente que deseja realizar uma operação por meio de outro componente tem apenas de expressar seu *intent*.
- Qualquer outro componente que exista e afirme que pode realizar esta operação por meio de *intent-filters*, é invocado pela plataforma.
- Os componentes não precisam estar cientes da existência uns dos outros e ainda assim podem trabalhar juntos para fornecer o resultado desejado para o usuário.

# Possibilidades

- Misturar e combinar (plug and play) componentes em tempo de execução
- Substituir aplicativos padrão Android com aplicações personalizadas
- Reuso em nível de componentes dentro da própria aplicação e entre aplicações

# Principais Casos de Uso

- **Iniciar uma atividade**

- `startActivity()` | `startActivityForResult()`

- Iniciar um serviço

- `startService()`...

- Transmitir uma mensagem (broadcast)

- `sendBroadcast()`...

# Tipos de Intent

- Explícitos
  - especificam o componente a ser iniciado pelo nome (fully-qualified)
  - tipicamente utilizado no contexto do próprio app
- Implícitos
  - não define um componente específico, apenas declara uma ação a ser realizada
  - outros aplicativos podem realizar esta ação



Como iniciar activities  
utilizando intents?

# Iniciando activities

```
startActivity(Intent intent, ...)
```

```
startActivityForResult(Intent intent, ...)
```

# Intent Explícito

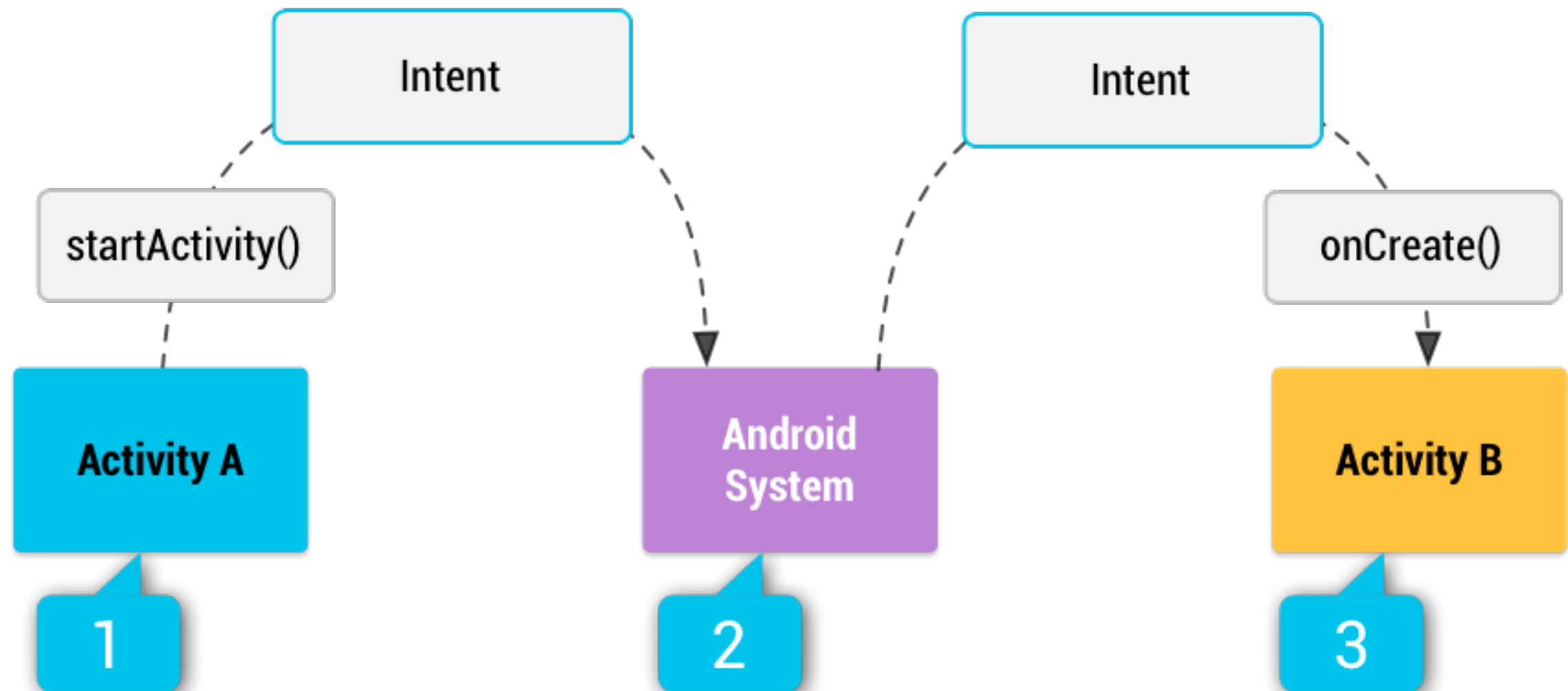
`//fileUrl é uma string como "http://www.example.com/image.png"`

```
Intent downloadIntent = new Intent(this,  
                                   DownloadService.class);  
  
downloadIntent.setData(Uri.parse(fileUrl));  
  
startService(downloadIntent);
```

# Intens Implícitos

- Android encontra o componente apropriado por meio dos *intent filters*
- Expressões declaradas no manifest, que especificam o(s) tipo(s) de intent(s) que os componentes do app recebem

# Intens Implícitos



Quais os possíveis  
resultados?

# Definindo um Intent

- Component
- Action
- Data
- Type
- Category
- Extras
- Flags

# Component

- O nome do componente que deve receber o Intent
- Distingue um intent explícito de um implícito
- Se precisamos iniciar um componente específico, devemos especificar o nome do mesmo
  - ex.: `com.example.ExampleActivity`
- Ao iniciar serviços, devemos sempre especificar o nome do componente, por qual razão?



# Definindo

```
Intent i = new Intent(Context packageContext,  
                      Class<?> cls);
```

ou...

```
Intent i = new Intent();
```

```
setComponent(...), setClass(...), setClassName(...)
```

# Action

- Uma String que especifica a ação genérica a ser executada
- No caso de broadcasts, é a ação que aconteceu e está sendo relatada
- Este campo determina como o restante do Intent é estruturado, particularmente quais os dados e campos extra que estarão contidos
- Podemos especificar nossas ações, mas em geral devemos usar as constantes definidas na classe `Intent`

# Exemplos

`ACTION_DIAL` - discar um número

`ACTION_EDIT` - mostrar dados para editar

`ACTION_VIEW` - mostrar dados ao usuário

`ACTION_SYNC` - sincronizar dados c/ um servidor

`ACTION_MAIN` - disparar uma atividade como atividade inicial de uma aplicação

`ACTION_SEND` - "share" intent

# Definindo

```
Intent i = new Intent(Intent.ACTION_DIAL);
```

ou...

```
Intent i = new Intent();
```

```
i.setAction(Intent.ACTION_DIAL);
```

# Data

- Dados associados com o Intent
- Formatados como Uniform Resource Identifier (`Uri`)
- O tipo de dado fornecido geralmente é ditado pela ação do Intent
  - Por exemplo, se a ação é `ACTION_EDIT`, o campo Data deve incluir a URI do documento a ser editado
- Ao criar um Intent, também é importante definir o tipo de dados (MIME type), além da URI

# Exemplos

Dados para mapa:

- `Uri.parse("geo:0,0?q=Av.+Jornalista+Anibal+Fernandes,+s/n+Recife+PE");`

Telefone a ser discado:

- `Uri.parse("tel:+558198761234")`

# Definindo

```
Intent i = new Intent(Intent.ACTION_DIAL,  
                        Uri.parse("tel:+5581..."));
```

ou...

```
Intent i = new Intent(Intent.ACTION_DIAL);  
  
i.setData(Uri.parse("tel:+5581..."));
```

# MIME Type

- Ao criar um Intent, também é importante definir o tipo de dados (MIME type), além da URI
- Especificar o MIME type ajuda Android a encontrar o melhor componente para receber o intent
- Em alguns casos, o MIME type pode ser inferido a partir da URI
  - Ex.: no caso de content:, que indica dados localizados no dispositivo, que tornam visível o MIME type



# Exemplos

`image/*, image/png, image/jpeg, image/jpg`

`text/html, text/plain`

# Definindo

```
Intent i = new Intent();  
i.setType(String type);
```

ou...

```
Intent i = new Intent();  
i.setDataAndType(Uri data, String type);
```

# Category

- Informação adicional sobre o tipo de componente que deve receber o intent
- Um intent pode ter várias categorias

# Exemplos

`CATEGORY_BROWSABLE`

invocada por um navegador para mostrar dados referenciados por um link (imagem, email...)

`CATEGORY_LAUNCHER`

atividade inicial de uma tarefa, listada como 'app launcher'

`CATEGORY_TAB | TEST | UNIT_TEST | PREFERENCE | ...`

# Definindo

```
Intent i = new Intent();
```

```
i.addCategory(Intent.CATEGORY_XYZ);
```

# Até o momento

- Estas propriedades listadas, representam as características de um Intent
- Ao ler estas propriedades, o sistema Android é capaz de definir qual componente deve ser iniciado
- Um intent pode carregar informação adicional, que não afeta a resolução do componente que será iniciado

# Extras

- Informação adicional associada com o Intent
- Um mapa de pares chave-valor
- Algumas ações utilizam dados extras
- Existem várias constantes já definidas para chaves, padronizando vários tipos de dados
- Podemos definir as nossas próprias

# Exemplo - EXTRA\_EMAIL

```
Intent i = new Intent(Intent.ACTION_SEND);  
  
i.putExtra(android.content.Intent.EXTRA_EMAIL,  
            new String[]{  
                "lmt@cin.ufpe.br",  
                "posgrad@cin.ufpe.br"  
            })  
;
```



# putExtra()

`putExtra(String name, double[] value)`

`putExtra(String name, int value)`

`putExtra(String name, CharSequence value)`

`putExtra(String name, char value)`

`putExtra(String name, Bundle value)`

`putExtra(String name, Parcelable[] value)`

`putExtra(String name, Serializable value)`

`putExtra(String name, int[] value)`

`putExtra(String name, float value)`

`putExtra(String name, byte[] value)`

`putExtra(String name, long[] value)`

`putExtra(String name, Parcelable value)`

`putExtra(String name, float[] value)`

`putExtra(String name, long value)`

`putExtra(String name, String[] value)`

`putExtra(String name, boolean value)`

`putExtra(String name, boolean[] value)`

`putExtra(String name, short value)`

`putExtra(String name, double value)`

`putExtra(String name, short[] value)`

`putExtra(String name, String value)`

`putExtra(String name, byte value)`

`putExtra(String name, char[] value)`

`putExtra(String name, CharSequence[] value)`

# Flags

- Funcionam como metadados para o Intent
- Podem instruir o sistema Android sobre como iniciar uma atividade...
  - a qual *task* a atividade vai pertencer
- ...ou como tratá-la após ser iniciada
  - se vai aparecer na lista de atividades recentes

# Exemplos

`FLAG_ACTIVITY_NO_HISTORY`

não coloca atividade na pilha

`FLAG_DEBUG_LOG_RESOLUTION`

imprime informação extra de log quando o intent está sendo processado

# Exemplos

FLAG\_GRANT\_READ\_URI\_PERMISSION

FLAG\_GRANT\_WRITE\_URI\_PERMISSION

FLAG\_DEBUG\_LOG\_RESOLUTION

FLAG\_FROM\_BACKGROUND

FLAG\_ACTIVITY\_BROUGHT\_TO\_FRONT

FLAG\_ACTIVITY\_CLEAR\_TASK

FLAG\_ACTIVITY\_CLEAR\_TOP

FLAG\_ACTIVITY\_CLEAR\_WHEN\_TASK\_RESET

FLAG\_ACTIVITY\_EXCLUDE\_FROM\_RECENTS

FLAG\_ACTIVITY\_FORWARD\_RESULT

FLAG\_ACTIVITY\_LAUNCHED\_FROM\_HISTORY

FLAG\_ACTIVITY\_MULTIPLE\_TASK

FLAG\_ACTIVITY\_NEW\_TASK

FLAG\_ACTIVITY\_NO\_ANIMATION

FLAG\_ACTIVITY\_NO\_HISTORY

FLAG\_ACTIVITY\_NO\_USER\_ACTION

FLAG\_ACTIVITY\_PREVIOUS\_IS\_TOP

FLAG\_ACTIVITY\_RESET\_TASK\_IF\_NEEDED

FLAG\_ACTIVITY\_REORDER\_TO\_FRONT

FLAG\_ACTIVITY\_SINGLE\_TOP

FLAG\_ACTIVITY\_TASK\_ON\_HOME

FLAG\_RECEIVER\_REGISTERED\_ONLY

# Definindo

```
Intent i = new Intent(Intent.ACTION_SEND);  
  
i.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
```

# Intents Implícitos

- Especificam ações que podem invocar qualquer app no dispositivo que seja capaz de executar a ação desejada
- Útil quando o aplicativo não é capaz de executar uma ação, mas outros aplicativos podem e você gostaria de oferecer a possibilidade ao usuário
- Por exemplo, podemos desejar compartilhar conteúdo com outras pessoas, por meio de uma ação **ACTION\_SEND**
- Ao chamar `startActivity()` com este intent, o usuário pode escolher o aplicativo que utilizará para compartilhar o conteúdo

# Exemplo de Intent Implícito

```
// txt = "..."  
Intent i = new Intent();  
i.setAction(Intent.ACTION_SEND);  
i.putExtra(Intent.EXTRA_TEXT, txt);  
i.setType(HTTP.PLAIN_TEXT_TYPE);  
  
// Checa que alguma activity pode receber o intent  
if (i.resolveActivity(getPackageManager()) != null)  
{  
    startActivity(sendIntent);  
}
```

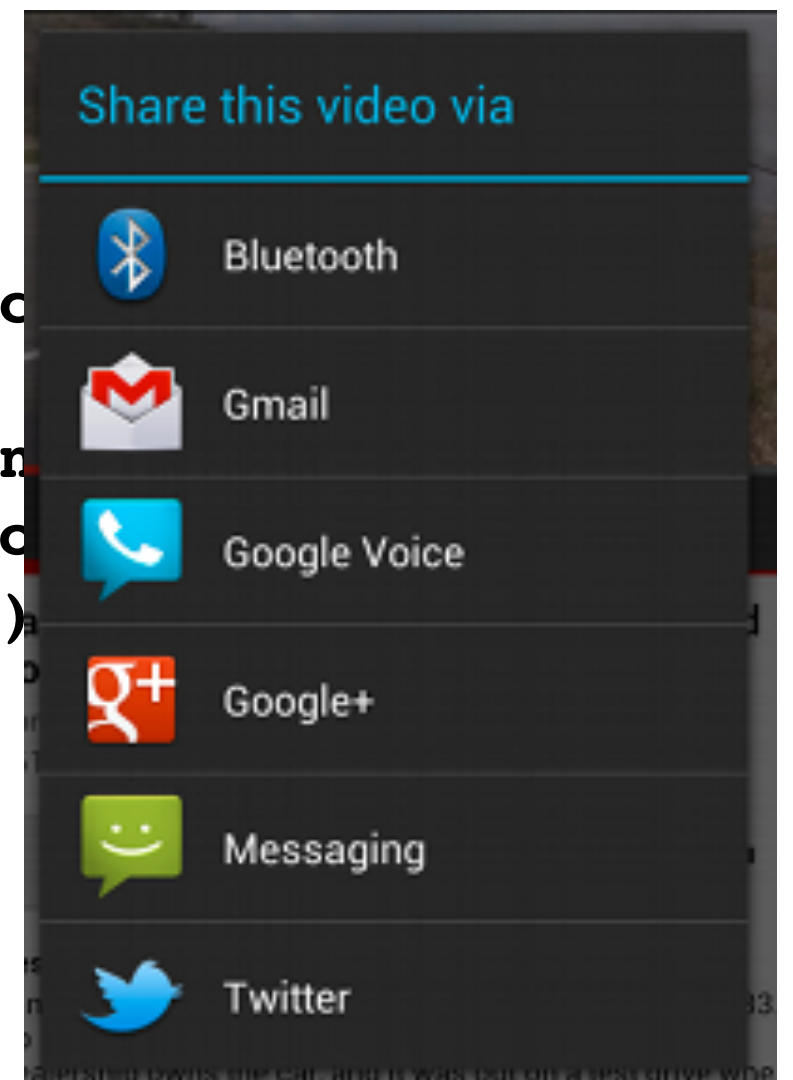
# Forçando a janela de escolha de um app

- Quando há mais de um aplicativo que responde ao intent implícito, o usuário pode selecionar qual deseja utilizar e marcar como escolha padrão para aquela ação
- No entanto, se vários aplicativos podem responder ao intent e o usuário deseja ser capaz de escolher um aplicativo diferente a cada vez, devemos mostrar uma janela de escolha
- Por exemplo, ao clicarmos em “share”...



# Exemplo

```
Intent i = new Intent(Intent.ACTION_SEND);  
...  
// Um recurso string para texto da interface  
// Algo como "Compartilhar esta foto com"  
String t = getResources().getString(R.string...  
// Criar Intent para mostrar janela de escolha  
Intent chooser = Intent.createChooser(i, t);  
  
if (i.resolveActivity(getPackageManager()) != null)  
    startActivity(sendIntent);  
}
```



# Intent Resolution

- Quando não especificamos um componente explicitamente, Android tenta descobrir qual Activity casa com o Intent
- O processo é chamado de Intent Resolution
  - Intent descrevendo operação
  - IntentFilters descrevendo que operações uma Activity pode receber

# Intent Filters

- Para declarar que intents implícitos seu aplicativo pode receber, declaramos um ou mais elementos do tipo `<intent-filter>` no manifesto
- Especifica os tipos baseados em action, data, e category
- Intents só são recebidos pela sua aplicação se passarem por um dos intent filters
- Um filter para cada tarefa que o componente da aplicação pode fazer.
  - Ex.: Visualizar e Editar uma imagem

# Especificando Intent Filters

```
<activity ...>  
  <intent-filter ...>  
    ...  
    <action ... />  
    <data ... />  
    <category ... />  
    ...  
  </intent-filter>  
  ...  
</activity>
```

# Exemplo

```
<activity android:name="ShareActivity">  
  <intent-filter>  
    <action  
      android:name="android.intent.action.SEND" />  
    <category  
      android:name="android.intent.category.DEFAULT" />  
    <data android:mimeType="text/plain" />  
  </intent-filter>  
</activity>
```

# Intent.ACTION\_DIAL

```
<activity ...>
  <intent-filter ...>
    ...
    <action android:name=
      "android.intent.action.DIAL" />
    ...
  </intent-filter>
  ...
</activity>
```

# Dados em um Intent Filter

```
<intent-filter ...>
    ...
    <data
        android:mimeType="string"
        android:scheme="string"
        android:host="string"
        android:port="string"
        android:path="string"
        android:pathPattern="string"
        android:pathPrefix="string"
    />
    ...
</intent-filter>
```

# Recebendo intents associados com mapas

```
<intent-filter ...>
```

```
...
```

```
<data android:scheme="geo" />
```

```
...
```

```
</intent-filter>
```



```
<activity android:name="MainActivity">
    <!-- This activity is the main entry, should appear in app launcher -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<activity android:name="ShareActivity">
    <!-- This activity handles "SEND" actions with text data -->
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
    <!-- This activity also handles "SEND" and "SEND_MULTIPLE" with media data -->
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <action android:name="android.intent.action.SEND_MULTIPLE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="application/vnd.google.panorama360+jpg" />
        <data android:mimeType="image/*" />
        <data android:mimeType="video/*" />
    </intent-filter>
</activity>
```

# Processo de Resolução

- Baseado em três aspectos
  - Action
  - Category
  - Data (URI e MIME type)

# Action

- Um intent filter pode declarar zero ou mais elementos `<action>`
- Para passar por um filtro, a ação especificada no Intent deve casar com uma das ações listadas no filter
- Se um intent filter não listar nenhuma ação, todos os intents vão falhar no teste

# Category

- Um intent filter pode declarar zero ou mais elementos `<category>`
- Para passar, todas as categorias declaradas no objeto Intent devem casar com categorias declaradas no filtro
- O inverso não é necessário

# Atenção

- Para receber intents implícitos, devemos incluir a categoria `CATEGORY_DEFAULT` no manifesto
- Android automaticamente aplica esta categoria aos intents implícitos passados para os métodos que chamam atividades

# Data

- Um intent filter pode declarar zero ou mais elementos `<data>`
- Cada elemento pode especificar uma estrutura URI e um tipo de dados (MIME)
- Existem atributos separados para cada parte da URI

`<scheme>://<host>:<port>/<path>`

`content://com.example.project:200/folder/subfolder/etc`

# Data

- `<scheme>://<host>:<port>/<path>`  
`content://com.example.project:200/folder/subfolder/etc`
- Cada atributo é opcional, mas há dependências
  - Se `scheme` não é especificado, `host` é ignorado
  - Se `host` não é especificado, `port` é ignorado
  - Se `host` e `scheme` não são especificados, `path` é ignorado

# Regras

- Um intent que não contém URI nem tipo MIME passa no teste apenas se o filtro não especificar quaisquer URIs ou tipos MIME.
- Um intent que contém URI, mas nenhum MIME (explícito ou inferido a partir da URI) passa no teste apenas se o seu URI corresponde ao formato URI do filtro e o filtro também não especifica um tipo de MIME.



# Regras

- Um intent que contém um tipo MIME, mas não um URI, passa no teste apenas se o filtro lista o mesmo tipo MIME e não especifica um formato de URI.
- Um intent que contém um URI e um tipo MIME (explícito ou inferido a partir da URI) passa a parte “MIME” do teste somente se esse tipo corresponde a um tipo listado no filtro. Só passa a “parte URI” do teste se a sua URI corresponde a um URI no filtro ou se tem uma URI `content:` ou `file:` e o filtro não especifica um URI.

# Intent Matching

- O Home app popula o launcher procurando por todas as activities com intent filters com action MAIN + category LAUNCHER
- O PackageManager tem um conjunto de métodos do tipo **query..()** que retornam componentes que aceitam um intent particular

if you want to share something in any app you want or open a url via every action, just use this method:

```
private void shareOrViewUrlViaThisApp(String appPackageName, String url) {
    boolean found = false;
    Intent intent = new Intent(Intent.ACTION_VIEW);
    intent.setData(Uri.parse(url));

    List<ResolveInfo> resInfo = getPackageManager().queryIntentActivities(intent, 0)
    if (!resInfo.isEmpty()){
        for (ResolveInfo info : resInfo) {
            if (info.activityInfo.packageName.toLowerCase().contains(appPackageName)
                info.activityInfo.name.toLowerCase().contains(appPackageName) )

                intent.setPackage(info.activityInfo.packageName);
                found = true;
                break;
            }
        }
        if (!found)
            return;

        startActivity(Intent.createChooser(intent, "Select"));
    }
}
```

and simply call:

```
shareOrViewUrlViaThisApp(<your package name>,<your url>);
```

This answer was inspired from [this](#).

share improve this answer

answered Jan 4 at 15:25



**Amir Hossein Ghasemi**

514 ● 1 ● 7 ● 14

<http://stackoverflow.com/a/41467367>

There is a way to directly open the intent you wants. You can get the list of intents and open only one.

See this code:

```
private void initShareIntent(String type) {
    boolean found = false;
    Intent share = new Intent(android.content.Intent.ACTION_SEND);
    share.setType("image/jpeg");

    // gets the list of intents that can be loaded.
    List<ResolveInfo> resInfo = getPackageManager().queryIntentActivities(share, 0);
    if (!resInfo.isEmpty()){
        for (ResolveInfo info : resInfo) {
            if (info.activityInfo.packageName.toLowerCase().contains(type) ||
                info.activityInfo.name.toLowerCase().contains(type) ) {
                share.putExtra(Intent.EXTRA_SUBJECT, "subject");
                share.putExtra(Intent.EXTRA_TEXT, "your text");
                share.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(new File(myPath)) );
                share.setPackage(info.activityInfo.packageName);
                found = true;
                break;
            }
        }
        if (!found)
            return;

        startActivity(Intent.createChooser(share, "Select"));
    }
}
```

If you wanna open twitter, do that:

```
initShareIntent("twi");
```

if facebook:

```
initShareIntent("face");
```

if mail:

<http://stackoverflow.com/a/9229654>

```
initShareIntent("mail"); // or "gmail"
```

# Programação para Dispositivos Móveis

**Leopoldo Teixeira**

[lm@cin.ufpe.br](mailto:lm@cin.ufpe.br) | @leopoldomt