



Android - Cards de Referência

Intents e Intents Filters

Bem vindo ao nosso sétimo card sobre Android! :-)) Desta vez, vamos conversar sobre as Intents e os Intents Filters. Intents são importantíssimos no desenvolvimento Android, pois são a forma padrão de comunicação entre os componentes de uma aplicação Android e também entre sua aplicação e os serviços do sistema.

INTENTS

Os Intents são a forma padrão de comunicação entre os diversos componentes que formam uma aplicação Android. Quais são esses componentes? Activities, Content Providers, Broadcast Receivers, Services entre outros. A forma como você envia uma mensagem de uma Activity para outra é com Intents. A forma como você se comunica de uma Activity com um Service é com Intents.

Enfim, Intents são importantíssimas no desenvolvimento Android e você precisa entender elas mais do que apenas para iniciar outras Activities. Uma Intent é uma mensagem assíncrona, ou seja, quem lança ela não precisa “travar” aguardando uma resposta. Um objeto do tipo Intent carrega consigo todas as informações necessárias para que o “tratador/receptor” realize algo baseado nesses dados.

Existem dois tipos de Intents: implícitas e explícitas. Na explícita, você diz na Intent quem é o “cara” responsável por tratar ela. É como uma mensagem direcionada, já tem um destino.

A Intent implícita, ao contrário, você não especifica exatamente quem vai tratar ela, mas define filtros que vão delimitar quem pode tratar ela. Por exemplo, você pode lançar uma Intent informando que quer enviar um e-mail. O “enviar um e-mail” é o filtro. Apenas programas que enviam e-mail serão capazes de responder por esta Intent.

Já aconteceu com você de tentar abrir um tipo de arquivo, um link, um e-mail, um PDF e se deparar com uma lista de aplicativos que podem fazer esta operação? Adivinha? Isto foi uma Intent que foi lançada e todos os programas que você vê na lista responderam: “Ei, eu sei tratar essa Intent, deixa ela comigo!”.

ANATOMIA DE UMA INTENT

Conforme já falamos, um objeto do tipo **android.content.Intent** é uma mensagem assíncrona. E como toda mensagem, ela carrega informações consigo. Vamos ver quais são essas informações.

Component Name - O nome do componente que deve tratar essa Intent. Você define esse campo apenas no caso de Intents Explícitas: você informa quem, exatamente, vai tratar ela.

Você define o Component Name através de três métodos da classe **android.content.Intent**. São eles: **setComponent(ComponentName)**, **setClass(Class)** e **setClassName(String)**.

O método **setComponent()** recebe como parâmetro um objeto do tipo **ComponentName**. Neste objeto, você deve informar o pacote e a classe do componente que deve responder a essa Intent. O pacote você encontra no arquivo de Manifest do projeto e a classe pode ser uma Activity, Service e por aí vai.

O método **setClass()** recebe dois parâmetros: o contexto da aplicação e a classe que receberá o Intent. **setClassName()** funciona de forma similar. A única diferença é o segundo parâmetro, que é uma **String** e não uma instância de **Class**.

Action - Aqui você define uma ação para ser executada. Você pode definir suas próprias ações ou usar ações pré-determinadas pelo Android. Que, aliás, são muitas. Não vamos listar todas elas aqui. O melhor local para conhecer elas é no endereço: <http://developer.android.com/reference/android/content/Intent.html>.

Para exemplificar, vamos supor que você queira compartilhar um texto ou imagem. Você usaria a ACTION_SEND. Caso você queira criar suas próprias Actions, lembre-se de definir o nome dela usando o pacote da sua aplicação, como, por exemplo, com.meupacote.projeto.ACAO_EXIBIR_TUDO. Você define a ação usando o método **setAction()**.

Uma vez que você define qual a ação, acaba determinando como serão os campos EXTRA e DATA, que comentaremos logo a seguir.

Data - Você informa, neste campo, uma URI que contém um conjunto de dados com o qual o componente que recebe a Intent trabalhará. Por exemplo, se você quer enviar uma ação ACTION_EDIT, deve informar sobre qual documento a ação de edição ocorrerá.

Observe que, como já comentamos, os campos Action e Data estão sintonizados. Você define a URI usando o método **setData()**. Tem também o método **setDataAndType()**, para o caso de haver necessidade de definir qual o data e o tipo dele.

O tipo pode ser inferido pela própria URI, mas há casos em que é interessante informar explicitamente qual o tipo.

Extras - São dados extras que você pode enviar para o componente que tratará a Intent. Você define extras usando uma série de métodos que começam com **putXXX()**.

Category - Ao definir uma categoria, você está informando os tipos de componente que podem responder a essa Intent. Por exemplo, você pode informar uma categoria como **CATEGORY_APP_BROWSER**, que informa que quem deve responder é um aplicativo de browser.

Já existe uma lista de categorias pré-definidas na classe Intent e que você pode ver no endereço <http://developer.android.com/reference/android/content/Intent.html>.

Flags - Finalizando, temos as flags. Definem como a Intent deve ser tratada. Muitas são usadas para definir com uma Activity deve ser iniciada e como deve ser tratada após sua inicialização.

INICIANDO UMA ACTIVITY

A forma mais trivial de se usar uma Intent é através da inicialização de uma Activity. A maioria das aplicações são compostas por muitas Activities e você precisa navegar entre elas. A navegação acontece através de Intents.

```
public class MainActivity extends Activity {  
  
    public void metodo() {  
        Intent intent =  
            new Intent(this, MainActivity2.class);  
        startActivity(intent);  
    }  
}
```

O trecho de código acima demonstra essa situação. Inicialmente, criamos uma Intent passando dois parâmetros para seu construtor. Esses dois parâmetros definem esta Intent como do tipo Explícita. Estamos informando no segundo parâmetro quem vai responder por essa Intent e no primeiro parâmetro o contexto do aplicativo onde se encontra esse componente.

Informamos o contexto do nosso próprio aplicativo, no caso a própria Activity onde este código está sendo executado e depois a classe da Activity que vai responder. Feito isto, basta chamar o método **startActivity(Intent)** da classe Activity.

RESOLUÇÃO DE INTENTS

Conforme já falamos, as Intents podem ser implícitas e explícitas. Nas explícitas, nós definimos o componente que responde por ela. Para isto, definimos o parâmetro Component Name da Intent.

No caso das Intents implícitas, a situação é diferente. Você não informa exatamente quem vai tratar. Neste caso, o Android lança-se do mecanismo denominado Intent Resolution. É através dele que o Android vai “olhar” sua Intent e, através dos campos dela, encontrar quem pode responder por esta Intent.

E isto é feito comparando o conteúdo dos campos Category, Data e Action com os Intent Filters.

INTENT FILTERS

Os Intent Filters definem quais tipos de Intents um componente pode receber. Um filtro para um componente é definido usando os campos Category, Data e Action. Caso um componente não defina seus Intent Filters, isto significa que ele pode receber apenas Intents Explícitas.

Um exemplo de Intent Filters está logo quando criamos nosso projeto. Observe no arquivo AndroidManifest.xml a definição de nossa única Activity.

```
<activity  
    android:name=".MainActivity"  
    android:label="@string/title_activity_main" >  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category  
            android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>
```

Definimos um filtro que informa que esta Activity responde por Intents com a ação **android.intent.action.MAIN** e com a categoria **android.intent.category.LAUNCHER**. Na prática, o que estamos informando com esses filtros é que esta Activity é a porta de entrada de nossa aplicação.

Você pode definir quantos conjuntos de Intent Filters quiser. E isto é feito colocando quantos **<intent-filter>** forem necessários.

Para que um componente possa responder por um Intent, todos estes três campos devem coincidir. Observe que os campos Category, Action e Data do Intent Filter refletem os campos com o mesmo nome na Intent. Se um deles for diferente, significa que a Intent não será entregue para este componente.

PENDING INTENT

Existe mais um tipo de Intent que deve ser vista. São as Pending Intents. Elas tem duas características interessantes. Primeiro, elas não são necessariamente executadas imediatamente. Podem ser enviadas para uma execução no futuro.

A segunda característica, e mais importante, é que as Pending Intents são executadas com as mesmas permissões que o aplicativo que a lançou. Muito profundo? :-) Lembre-se que as Intents, e também as Pending Intents, não estão, necessariamente, restritas à sua aplicação. Outras aplicações podem pegar e tratar as Intents que você lança. Como no caso em que você quer chamar um aplicativo de e-mail ou o browser.

Neste caso, uma Pending Intent é uma tarefa, ou um token, que é passado para outra aplicação fazer, mas usando as mesmas permissões que sua aplicação possui.

Um uso muito comum das Pending Intents é com o serviço chamado AlarmManager. Com ele, você pode agendar tarefas para serem executadas em algum momento futuro. Quando chegar o momento, o AlarmManager deve executar algo e usando as mesmas permissões do seu aplicativo.

Ainda não veremos um exemplo de Pending Intents neste card, pois o exploraremos com detalhes em um card que trate de Alarm Managers.



Sobre o Autor

Marlon Silva Carvalho

É um programador de longa data. Já peregrinou por diversas linguagens e hoje se considera um programador agnóstico. Atualmente, está fascinado pelo mundo mobile e suas consequências para a humanidade. Está mais focado no desenvolvimento para Android, embora também goste de criar aplicativos para iOS.

Trabalha em projetos sobre mobilidade no SERPRO, tendo atuado no projeto Pessoa Física, para a Receita Federal do Brasil. Você pode encontrar ele no Twitter, no perfil @marlonscarvalho, em seu blog, no endereço <http://marlon.silvacarvalho.net> e através do e-mail marlon.carvalho@gmail.com.

Sobre os Cards

Caso você já tenha visto os excelentes RefCards da DZone, deve ter percebido a semelhança. E é proposital. A ideia surgiu após acompanhar estes RefCards por um bom tempo. Contudo, eles são mais gerais. O objetivo destes Cards é tratar assuntos mais específicos. Trazendo informações relevantes sobre eles.

Caso tenha gostado, continue nos acompanhando! Tentaremos trazer sempre novos cards sobre os mais variados assuntos. Inicialmente, focaremos no Android.

O ícone principal deste card é de autoria de Wallec e foi obtido em seu profile no DeviantArt.

<http://wwalczyszyn.deviantart.com/>

Este trabalho usa a licença **Creative Commons Attribution-NonCommercial-ShareAlike 3.0**

<http://creativecommons.org/licenses/by-nc-sa/3.0/>



Basicamente, você pode copiar e modificar, desde que redistribua suas modificações. Também não é permitido usar este material ou suas derivações de forma comercial.