ANDROID – CRIANDO UMA NOTIFICAÇÃO

[](https://zarelli.files.wordpress.com/2011/10/notificacao.png)

A notificação no android é muito utilizada, quando temos uma aplicação rodando em segundo plano por exemplo e essa aplicação necessita informar algo para o usuário a melhor maneira é exibindo uma notificação, nós não podemos simplismente abrir uma tela na frente, pois pode atrapalhar o que o usuário está fazendo no momento.

Abaixo segue um exemplo prático de como implementar uma aplicação que execute uma notificação.

Criei essa classe para controlar a criação das minhas notificações.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74 | package br.gui;    import android.app.Activity;  import android.app.Notification;  import android.app.NotificationManager;  import android.app.PendingIntent;  import android.content.Intent;    /\*\*   \*   \* @author guilherme   \*/  public class Notificacao {        public static final int TIPO\_ACTIVITY = 1;      public static final int TIPO\_BROADCAST = 2;      public static final int TIPO\_SERVICE = 3;      public static final String INTENT\_STRING\_MSG\_STATUS = "MSGs";      public static final String INTENT\_STRING\_MSG = "MSG";      public static final String INTENT\_STRING\_TITULO = "titu";      public static final String INTENT\_LONG\_QUANDO = "WHEN";      public static final String INTENT\_INT\_ICONE = "icone";      public static final String INTENT\_INT\_FLAGS = "FLS";        /\*\*       \*       \* @param activity Atividade       \* @param param Parametros da notificacao       \* @param acao Ação quando é selecionado a notificacao       \* @param tipo Tipo da ação /broadcast/atividade/serviço/       \* @return Notification       \*/      public static Notification criarNotificacao(Activity activity, Intent param, Intent acao, int tipo) {          //cria a notificacao          Notification n = new Notification(param.getIntExtra(INTENT\_INT\_ICONE, 0),                  param.getStringExtra(INTENT\_STRING\_MSG\_STATUS),                  param.getLongExtra(INTENT\_LONG\_QUANDO, System.currentTimeMillis()));            //Responsavel pela acao caso o usuario selecione a notificacao;          PendingIntent p;          if (tipo == TIPO\_ACTIVITY) {              p = PendingIntent.getActivity(activity, 0, acao, param.getIntExtra(INTENT\_INT\_FLAGS, 0));          } else if (tipo == TIPO\_BROADCAST) {              p = PendingIntent.getBroadcast(activity, 0, acao, param.getIntExtra(INTENT\_INT\_FLAGS, 0));          } else if (tipo == TIPO\_SERVICE) {              p = PendingIntent.getService(activity, 0, acao, param.getIntExtra(INTENT\_INT\_FLAGS, 0));          } else {              throw new IllegalArgumentException("tipo indefinido");          }            //Vincula o PendingIntent com a notificacao          n.setLatestEventInfo(activity, param.getStringExtra(INTENT\_STRING\_TITULO), param.getStringExtra(INTENT\_STRING\_MSG), p);            //Define valores default para a notificacao ex. som,vibra.          n.defaults = Notification.DEFAULT\_ALL;            return n;      }        /\*\*       \*       \* @param activity Atividade       \* @param notification Notificação a ser exibida       \* @param id Identificador unico da notificacao       \* @return  NotificationManager       \*/      public static NotificationManager notificar(Activity activity, Notification notification, int id) {          //pega serviço de notificacao          NotificationManager nm = (NotificationManager) activity.getSystemService(Activity.NOTIFICATION\_SERVICE);          //exibe a notificacao          nm.notify(id, notification);          return nm;      }  } |

No código acima, após a criação do notificação definimos um PedingIntent de acordo com o tipo passado no método. O PedingIntent contém as ações que devem ser tomadas caso o usuário selecione a notificação, os métodos diferente de chamada são para distintas decisões, por exemplo, o getActivty é responsável em abrir uma atividade, o getBroadcast é responsável por transmitir a intent pelo S.O (mensagem de broadcast), e o getService é responsável por chamar um serviço.

Veja o exemplo que utiliza esta classe para realizar a notificação:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | package br.gui;    import android.app.Activity;  import android.app.Notification;  import android.content.Intent;  import android.net.Uri;  import android.os.Bundle;    public class MainActivity extends Activity {        /\*\* Called when the activity is first created. \*/      @Override      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.main);            //define a intent responsavel por passar os parametros necessarios para          //criar uma notificacao.          Intent it = new Intent();          it.putExtra(Notificacao.INTENT\_STRING\_MSG, "Meu teste de notificacao.");          it.putExtra(Notificacao.INTENT\_STRING\_MSG\_STATUS, "Olá");          it.putExtra(Notificacao.INTENT\_STRING\_TITULO, "Titulo da notificação");          it.putExtra(Notificacao.INTENT\_LONG\_QUANDO, System.currentTimeMillis());          it.putExtra(Notificacao.INTENT\_INT\_ICONE, android.R.drawable.ic\_menu\_week);          //Intente de ação/ou seja/ao clicar na notificação, o sistema irá abrir          //uma pagina com o seguinte endereco          String url = "[http://www.example.com](http://www.example.com/)";          Intent i = new Intent(Intent.ACTION\_VIEW);          i.setData(Uri.parse(url));          //Cria a notificação.          Notification notf = Notificacao.criarNotificacao(this, it, i, Notificacao.TIPO\_ACTIVITY);          //Manda notificar.          Notificacao.notificar(this, notf, 550);        }  } |

É uma aplicação bem simples, executa uma notificação e quando você a seleciona é aberto uma pagina em seu navegador. Nota-se que na linha que eu notifico eu passo um id, esse id é o identificador da notificação, caso eu queira cancelar a mesma eu utilizo o método .cancel(550) da classe NotificationManager.

ex:

|  |  |
| --- | --- |
| 1  2 | NotificationManager nm = Notificacao.notificar(this, notf, 550);  nm.cancel(550); |

Pronto. Não é necessário mais nada para estar exibindo uma notificação na tela. Caso queira personalizar ainda mais sua notificação verifique a documentação no site do android:

<http://developer.android.com/reference/android/app/Notification.html>

Ainda é possível definir um áudio diferente, vibrar, acender a luz de fundo etc.