

Introduction à greenDroid



par Nazim Benbourahla

Date de publication : 05/12/2011

Dernière mise à jour : 22/12/2011

Dans ce tutoriel, je vais vous présenter une bibliothèque que j'ai souvent utilisé et qui est très utile pour vos développements Android, il s'agit de <http://android.cyilmottier.com/> créé par <http://www.cyilmottier.com/>.

1 - GreenDroid : à quoi ça sert ?.....	3
2 - GreenDroid : Les fonctionnalités disponibles.....	3
3 - GreenDroid : Installation.....	3
4 - GreenDroid : Mise en place du projet GreenDroid.....	4
5 - GreenDroid : Action Bar.....	7
6 - GreenDroid : QuickActions.....	11
7 - GreenDroid : Surcharger le thème.....	13
8 - Conclusion.....	16
9 - Remerciements.....	16
10 - Liens.....	16

1 - GreenDroid : à quoi ça sert ?

Cette bibliothèque facilitera la création de vos applications Android, grâce à plusieurs mécanismes, elle vous permettra :

- d'optimiser vos applications ;
- de gagner du temps sur la création d'interface et la gestion de plusieurs fonctionnalités de votre application ;
- de factoriser votre code.

2 - GreenDroid : Les fonctionnalités disponibles

Voici une introduction aux fonctionnalités de **GreenDroid**, la liste est beaucoup plus grande que ce que je présente dans cet article.

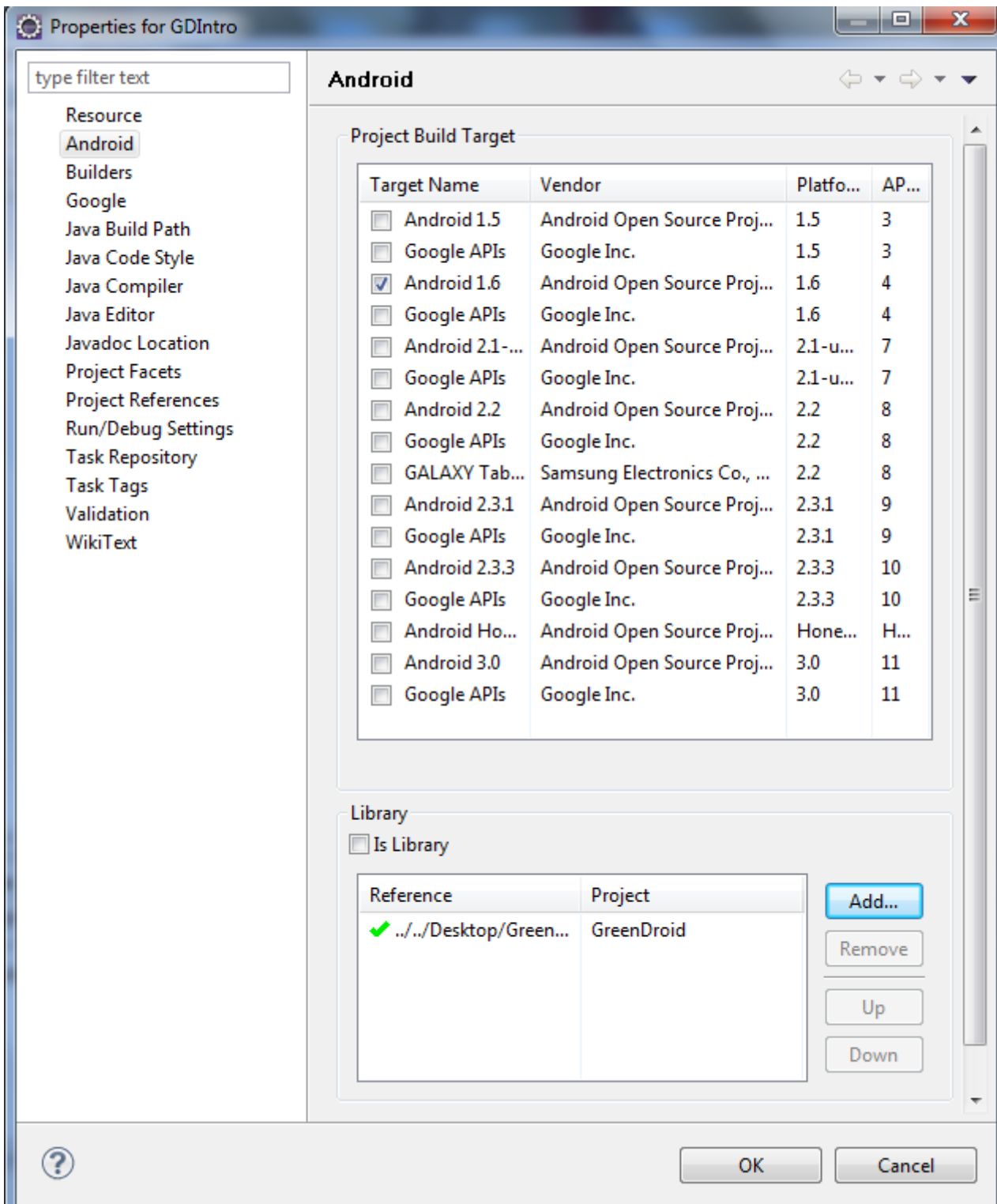
3 - GreenDroid : Installation

Pour commencer, vous devez cloner (**git clone <https://github.com/cyrimottier/GreenDroid.git>**) ou télécharger GreenDroid sur : **<https://github.com/cyrimottier/GreenDroid>**, puis importer le projet dans votre eclipse (File -> Import). Une fois cette étape effectuée, cliquez droit sur le projet GreenDroid puis allez dans Properties puis dans Android et vérifiez que **“Is Library”** est bien coché sinon faites-le.

Nous allons créer un projet qui va utiliser GreenDroid.

- Nom du projet : GDIntro ;
- Android SDK : 1.6 ;
- Nom de l'application : GD Intro ;
- Package : com.nazim.tuto.gd ;
- Activité de départ : GDIntroActivity ;

Une fois le projet créé, faites un clic droit sur votre projet, allez dans Properties puis Android et dans Library. Cliquez sur Add puis sélectionnez **GreenDroid** (si GreenDroid n'apparaît pas, c'est que l'étape “Is Library” n'est pas correctement faite).



4 - GreenDroid : Mise en place du projet GreenDroid

Afin de mettre en place votre projet, il faut faire quelques modifications pour intégrer **GreenDroid**.

- Premièrement, nous allons créer une classe qui présentera l'application et qui s'appellera **GDIntroApp**. Elle contiendra une seule méthode « **getHomeActivityClass** » qui retournera votre vue de base

```
packagecom.nazim.tuto.gd;
```

```
import greendroid.app.GDApplication;

public class GDIntroApp extends GDApplication {

    @Override
    public Class getHomeActivityClass() {
        return GDIntroApp.class;
    }
}
```

et finalement, il ne reste plus qu'à modifier le manifest, pour rajouter :

- notre classe GDIntroApp : android:name= »GDIntroApp » ;
- rajouter le thème GreenDroid : android:theme= »@style/Theme.GreenDroid ».

Ce qui donnera

```
<?xmlversion="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.nazim.tuto.gd"
    android:versionCode="1"
    android:versionName="1.0">

    <application android:icon="@drawable/icon" android:label="@string/app_name"
        android:theme="@style/Theme.GreenDroid" android:name="GDIntroApp">
        <activity android:name=".GDIntroActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Ce qui vous donnera :



5 - GreenDroid : Action Bar

L'une des fonctionnalités majeures de **GreenDroid** est l'**ActionBar**, qui correspond à la barre verte qui est sur le screen au dessus. Nous allons l'enrichir avec plusieurs fonctionnalités.

Nous allons rajouter à notre Action Bar les boutons suivants :

- un bouton pour rafraîchir ;
- un bouton pour partager ;
- un bouton pour localiser.

Pour cela nous allons créer une petite fonction pour initialiser les différents éléments.

```
private void initActionBar() {  
    addActionBarItem(Type.Locate, LOCATE);  
    addActionBarItem(Type.Refresh, REFRESH);  
    addActionBarItem(Type.Share, SHARE);  
}
```

et les entiers utilisés

```
private final int LOCATE = 0;  
private final int REFRESH = 1;  
private final int SHARE = 2;
```

Sans oublier d'appeler la méthode dans le onCreate

```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
  
    setContentView(R.layout.main);  
    initActionBar();  
}
```

Donc pour rajouter un élément, il suffit juste d'appeler `addActionBarItem`, avec comme arguments :

- le type de l'élément souhaité. Une vingtaine d'éléments sont disponibles ;
- un entier qui sera l'identifiant de votre élément quand on clique dessus.

Ce qui vous donnera :



Maintenant passons à la gestion du clic sur un élément de votre Action Bar. Pour cela il suffit de surcharger la méthode **onOptionsItemSelected**.

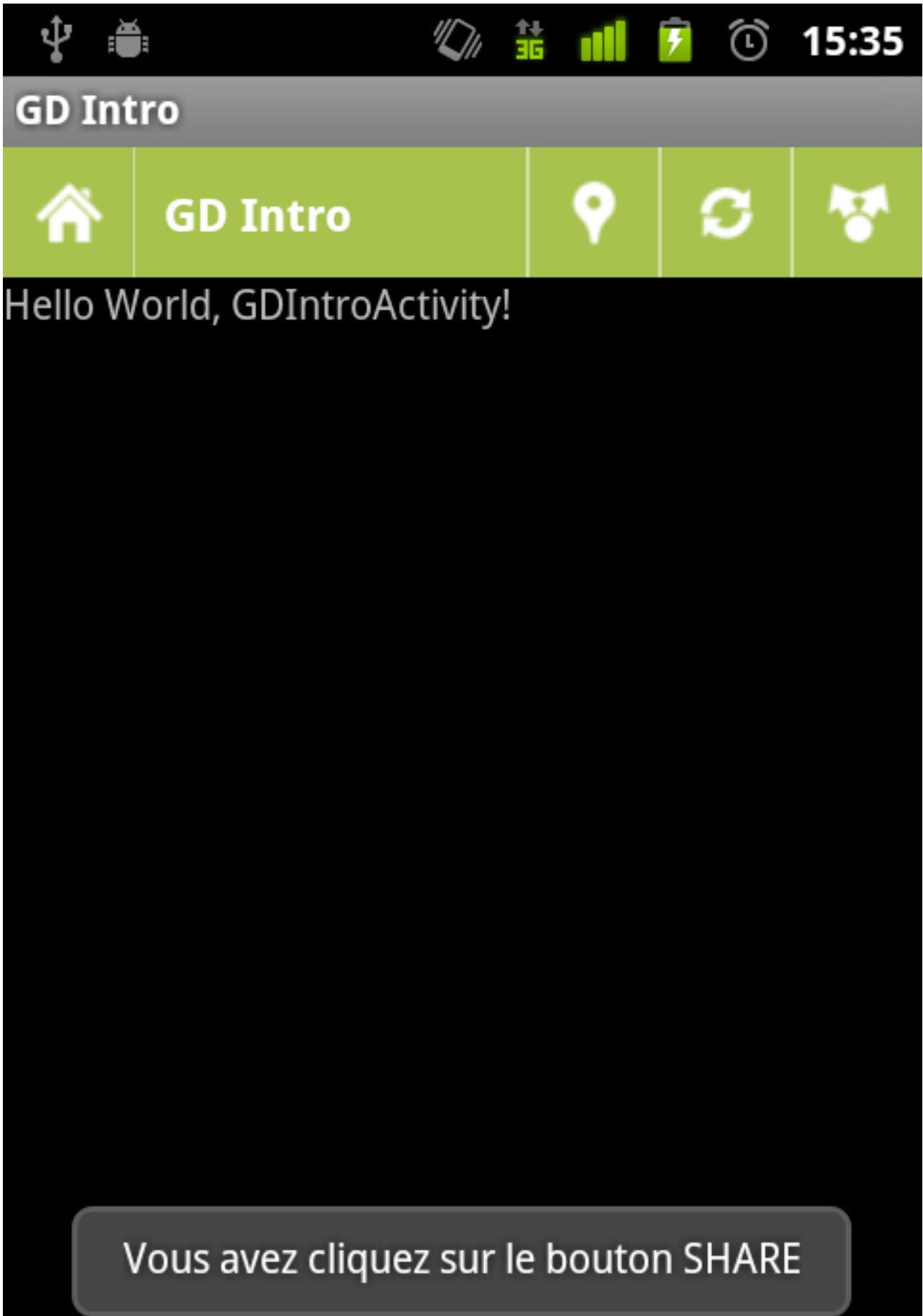
```
@Override
public boolean onOptionsItemSelected (ActionBarItem item, int position) {
    switch (item.getItemId()) {
        case LOCATE:
            Toast.makeText (getApplicationContext (),
                "Vous avez cliqué sur le bouton LOCATE",
                Toast.LENGTH_SHORT).show ();
            break;

        case REFRESH:
            Toast.makeText (getApplicationContext (),
                "Vous avez cliqué sur le bouton REFRESH",
                Toast.LENGTH_SHORT).show ();
            break;

        case SHARE:
            Toast.makeText (getApplicationContext (),
                "Vous avez cliqué sur le bouton SHARE",
                Toast.LENGTH_SHORT).show ();
            break;
        default:
            return super.onOptionsItemSelected (item, position);
    }

    return true;
}
```

Ce qui donnera :



Voilà pour la première fonctionnalité Je ne pourrai pas aborder les différents cas d'utilisation. Si votre cas est spécifique, n'hésitez pas à me poser une question.

6 - GreenDroid : QuickActions

Nous allons passer au **QuickActions**, vous allez comprendre à quoi cela correspond. Nous allons faire en sorte que lorsque l'utilisateur clique sur le bouton Share, cela affiche des quick actions.

Nous allons commencer par créer une variable qui représentera notre quick action.

```
privateQuickActionWidget quickActions;
```

Puis, nous allons créer une méthode qui initialise notre Quick Actions Widget et initialise le clic listener dessus

```
privatevoidinitQuickActionBar() {
    quickActions = newQuickActionBar(this);
    quickActions.addAction(newQuickAction(this, R.drawable.facebook,
        "Facebook"));
    quickActions.addAction(newQuickAction(this, R.drawable.twitter, "Twitter"));
    quickActions.addAction(newQuickAction(this, R.drawable.mail, "Email"));
    quickActions.setOnQuickActionClickListener(newOnQuickActionClickListener() {
        publicvoidonQuickActionClicked(QuickActionWidget widget,
            intposition) {
            Toast.makeText(GDIntroActivity.this,
                "Item "+ position + " clicked", Toast.LENGTH_SHORT)
                .show();
        }
    });
}
```

Il vous faut les trois images **suivantes**, à mettre dans le dossier Drawable de res.

Nous allons modifier notre méthode **onHandleActionBarItemClick**, en cliquant sur le bouton Share

```
caseSHARE:
    quickActions.show(item.getItemView());
    Toast.makeText(getApplicationContext(),
        "Vous avez cliqué sur le bouton SHARE",
        Toast.LENGTH_SHORT).show();
    break;
```



Sans oublier d'appeler la méthode d'initialisation dans le onCreate

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);
    initActionBar();
    initQuickActionBar();
}
```

Ce qui vous donnera

7 - GreenDroid : Surcharger le thème

Maintenant vous allez me dire **“GreenDroid c'est génial”**, mais je ne veux pas d'une barre verte dans mon application. Ne vous inquiétez pas, vous pouvez changer ça, en surchargeant le thème. Je vais vous expliquer comment faire.

Alors la première étape est de créer votre style. Dans values, créez un fichier themes.xml. Dans ce fichier vous allez créer votre thème comme ci-dessous :

```
<?xmlversion="1.0"encoding="utf-8"?>
<resources>
    <style name="Theme.Mytheme" parent="@style/Theme.GreenDroid.NoTitleBar">
        <item name="gdActionBarTitleColor">#fff</item>
        <item name="gdActionBarDividerWidth">2px</item>
        <item name="gdActionBarBackground">@color/background</item>
    </style>
</resources>
```

Il existe plusieurs propriétés que vous pouvez surcharger de GreenDroid, ici on surcharge :

- la couleur du texte dans l'action bar ;
- la taille des séparateurs dans l'action bar ;
- la couleur de fond de l'action bar.

Vous allez sûrement remarquer le **@color**, pour celui-ci créez un fichier colors dans values qui contiendra :

```
<?xmlversion="1.0"encoding="utf-8"?>
<resources>
    <color name="background">#f00</color>
</resources>
```

Vous pouvez à partir de maintenant prendre la bonne habitude de mettre toutes les couleurs que vous utilisez dans votre application dans ce fichier là.

Dernière étape, il faut modifier le fichier AndroidManifest pour rajouter notre thème (**android:theme= »@style/Theme.Mytheme**), ce qui donnera

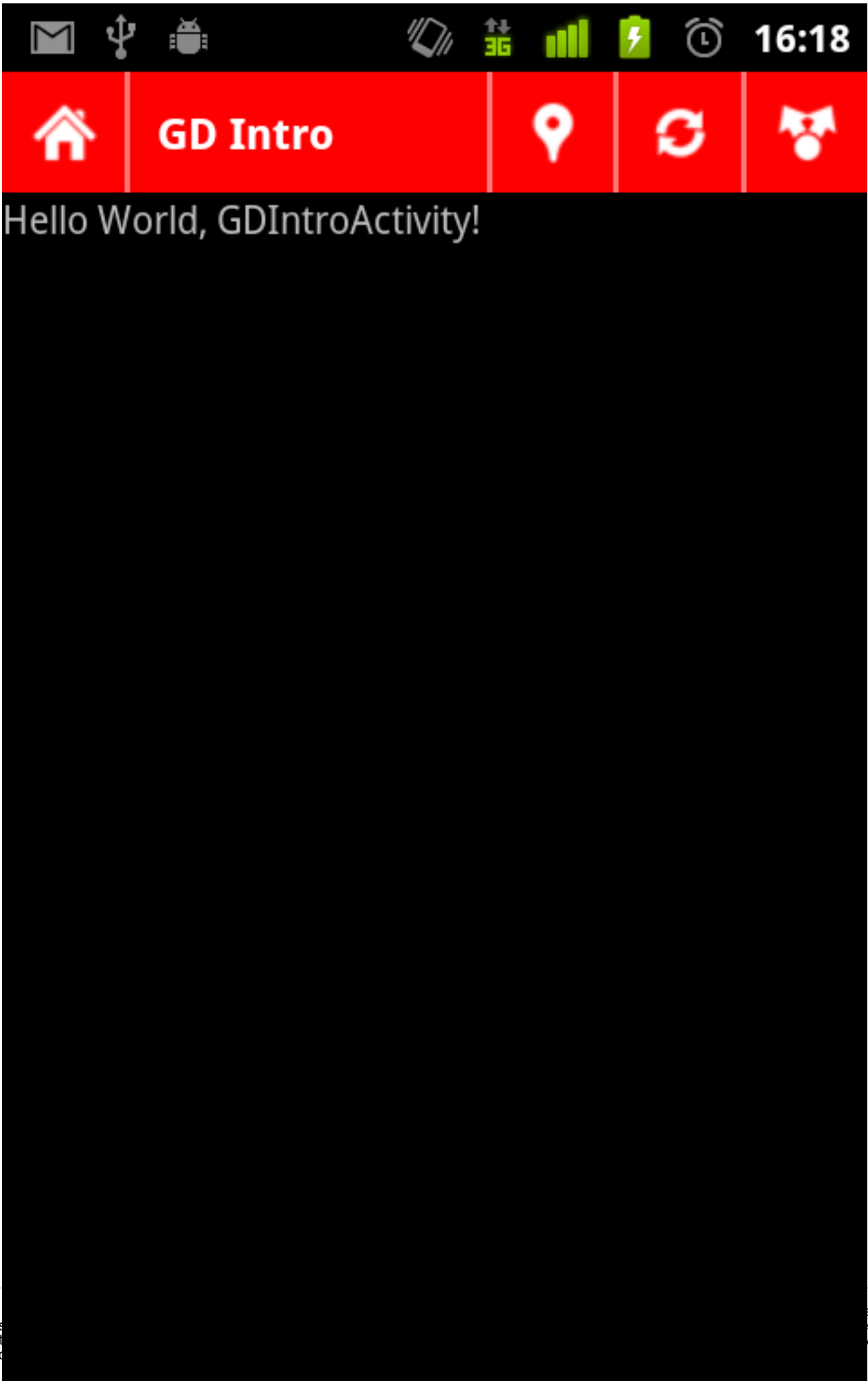
```
<?xmlversion="1.0"encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.nazim.tuto.gd"
    android:versionCode="1"
    android:versionName="1.0">

    <application android:icon="@drawable/icon" android:label="@string/app_name"
        android:theme="@style/Theme.Mytheme" android:name="GDIntroApp">
        <activity android:name=".GDIntroActivity"
            android:label="@string/app_name">
```

```
<intent-filter>
  <actionandroid:name="android.intent.action.MAIN"/>
  <categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>

</application>
</manifest>
```

Voilà vous devriez obtenir le résultat suivant :



8 - Conclusion

Voilà ce tutoriel s'arrête ici, vous pouvez télécharger le projet [ici](#).

Ce tutoriel fait un bref tour des fonctionnalités de **GreenDroid**. D'autres fonctionnalités sont disponibles dans cette bibliothèque. Vous trouverez plus de détails sur le [site officiel](#).

Parmi ces fonctionnalités :

- **SegmentedBar** ;
- **AsyncImageView** ;
- et plein d'autres fonctionnalités. N'hésitez pas à aller sur le site pour chercher si une fonctionnalité que vous souhaitez existe.

N'hésitez pas à me poser des questions et s'il y a de la demande, je ferai une suite à ce tutoriel avec les autres fonctionnalités. En attendant, j'espère que ce tutoriel vous a permis de connaître cette bibliothèque et vous a donné envie de l'utiliser. Bonne continuation à tous !

9 - Remerciements

Je tiens à remercier tout particulièrement [djibril](#) qui a mis ce tutoriel au format Developpez.com.

Merci également à [djibril](#) d'avoir pris le temps de le relire et de le corriger.

10 - Liens

- [Tutoriel origine sur Tutos-Android.](#)