

# ActionBarCompat

A T.A.U.G. live coding

by



**Android<sup>2EE</sup>**  
la programmation sous Android

2014 - Android2EE



## Android2EE trainings

Your guide for mobility



Lieu : Paris, Toulouse, Lyon,  
Rennes, Nantes, Lille  
Durée : 3 jours  
Prix : 1680 €



Lieu : Paris, Toulouse, Lyon,  
Rennes, Nantes, Lille  
Durée : 5 jours  
Prix : 2980 €



Lieu : Paris, Toulouse, Lyon,  
Rennes, Nantes, Lille  
Durée : 3 jours  
Prix : 1780 €

Android2EE est référencé en tant qu'organisme de formation, vous pouvez faire prendre en charge tout ou partie du montant de cette formation par votre OPCA. Cette formation Initiation avancée à Android est éligible au titre du **DIF et CIF**.

2014 - Android2EE



# Action Bar Sherlock ActionBarCompat



SupportLib v18

2014 - Android2EE



## Installation en 2 étapes

0 Downloads:

Mettez à jour votre Android SDK (surtout la support library)

 **Attention**

**SupportLib V18**

Soyez sûr d'avoir votre environnement à jour et d'utiliser la version 18 (au moins).

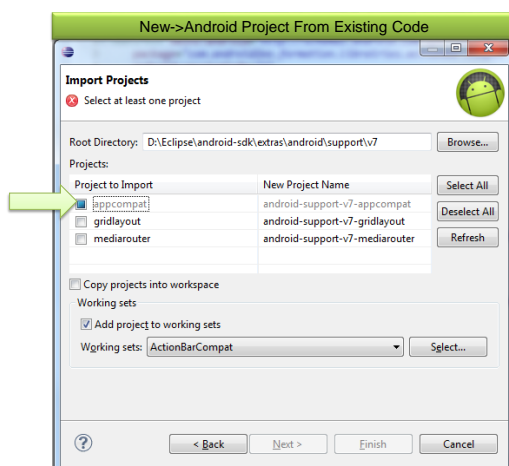
2014 - Android2EE



## Ajout du projet v7-appcompat dans Eclipse

### 1 Install the project android-support-v7-appcompat in workspace as an Eclipse project:

1. Dans Eclipse **New->Android Project From Existing Code**. Choisir android-support-v7-appcompat (dans le dossier de android-sdk\extras\support\v7).



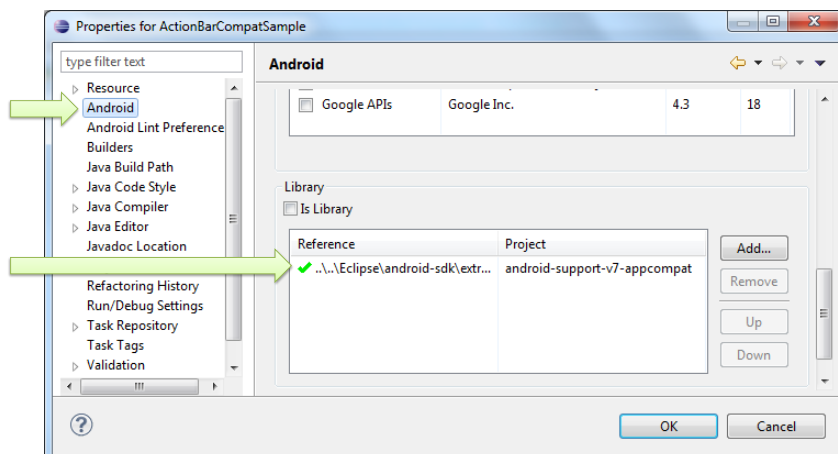
2014 - Android2EE



## Mise à jour de votre projet

### 2 Bind it to your project

1. Dans votre projet, toujours dans la fenêtre properties(ou BuildPath)->Android et ajoutez la lib support-v7



2014 - Android2EE



## Utilisation

Il en vous reste plus qu'à modifier le projet ainsi:  
 Vous utiliser toujours les ActionBarActivity (au lieu des FragmentActivity)  
 Votre Manifest doit avoir pour thème le thème Sherlock

Fondamental

Utiliser toujours les ActionBarActivity

Manifest

```
<application
  android:allowBackup="true"
  android:icon="@drawable/ic_launcher"
  android:label="@string/app_name"
  android:theme="@style/Theme.AppCompat" >
  <activity
    android:name="com.android2ee.formation.librairies.actionbar.compat.MainActivity"
    android:label="@style/Theme.AppCompat" >
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />

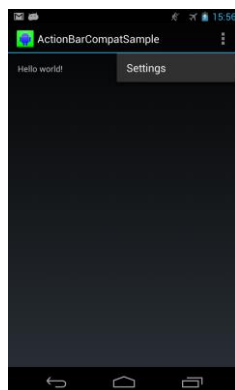
      <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
  </activity>
</application>
```

2014 - Android2EE

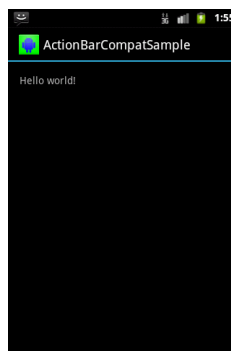


## ActionBar every where

Il en vous reste plus qu'à lancer votre projet



SDK 4.2.2 Nexus4



SDK 2.3.3 Emulator

2014 - Android2EE



# ActionBar

## Main Features



2014 - Android2EE

## Ajouter les MenuItems d'actions

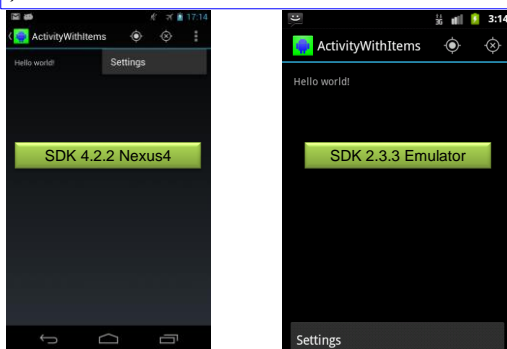
La principale fonctionnalité est simple à mettre en place, à un détail, le name space pour le showAsAction.

### Java

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.activity_with_items, menu);
    return super.onCreateOptionsMenu(menu);
}
```

### menu/activity\_with\_items.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:actionbarcompat_mse="http://schemas.android.com/apk/res-auto" >
    <item
        android:id="@+id/action_one"
        android:icon="@drawable/ic_action_one_inverse"
        android:orderInCategory="0"
        actionbarcompat_mse:showAsAction="always"
        android:title="One"/>
    <item
        android:id="@+id/action_two"
        android:icon="@drawable/ic_action_two_inverse"
        android:orderInCategory="0"
        actionbarcompat_mse:showAsAction="ifRoom"
        android:title="Two"/>
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        actionbarcompat_mse:showAsAction="never"
        android:title="@string/action_settings"/>
</menu>
```



2014 - Android2EE



## Ajouter les MenuItem d'actions

### ! Attention

#### SupportLib V18

Pas de construction dynamique pour l'instant.

#### Using XML attributes from the support library

Notice that the `showAsAction` attribute above uses a custom namespace defined in the `<menu>` tag. This is necessary when using any XML attributes defined by the support library, because these attributes do not exist in the Android framework on older devices. So you must use your own namespace as a prefix for all attributes defined by the support library.

<https://github.com/exciys/androidannotations/wiki/WorkingWithThreads>

2014 - Android2EE



## Utiliser les ActionsView

Les ActionView permettent d'afficher une vue dans l'ActionBar quand l'utilisateur clique sur le MenuItem associé.



2014 - Android2EE



## Utiliser les ActionsView

Trois fichiers sont nécessaires pour mettre en place une ActionView: du code Java, le fichier xml des MenuItem et le fichier des layouts de la vue affichant l'ActionView.

menu/actionview.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:actionbarcompat_mse="http://schemas.android.com/apk/res-auto" >

    <item
        android:id="@+id/menu_item_actionview"
        actionbarcompat_mse:actionLayout="@layout/actionview_view"
        actionbarcompat_mse:showAsAction="always|collapseActionView"
        android:icon="@drawable/ic_action_provider_extends"
        android:title="ActionView"/>

    <item
        android:id="@+id/action_one"
        actionbarcompat_mse:showAsAction="always"
        android:icon="@drawable/ic_action_one_inverse"
        android:orderInCategory="0"
        android:title="One"/>

</menu>
```

Le fichier des menus déclarent l'ActionView en tant qu'icone à afficher sur la barre d'action.

Le point crucial est le name space pour l'**actionLayout** qui est le même que pour la balise **showAsAction**.

La balise **actionLayout** pointe vers le fichier de layout à utiliser quand la vue de l'ActionView s'affiche.

Le mot clef **collapseActionView** permet de réduire automatiquement l'ActionView à la création.

2014 - Android2EE



## Utiliser les ActionsView

Le fichier des layouts est un fichier de layout normal.

layout/actionview\_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"    android:layout_height="match_parent"
    android:layout_gravity="center"    android:focusable="true" >

    <EditText
        android:id="@+id/edtActionView"    android:layout_width="wrap_content"
        android:layout_height="wrap_content"    android:ems="12"
        android:hint="I am an EditText but I could be whatever"
        android:textSize="12sp" />

    <ImageButton
        android:id="@+id/btnActionView"
        android:layout_width="32dip"    android:layout_height="32dip"
        android:layout_gravity="center"    android:adjustViewBounds="true"
        android:background="@drawable/ic_action_actionview_ok"
        android:scaleType="fitCenter" />

</LinearLayout>
```

Hello ActionView



2014 - Android2EE



## Utiliser les ActionsView

Enfin, en Java, il suffit de mettre en place la classe qui étend `ActionProvider` et qui décrit son comportement.

```
MenuItem menuItemActionView;
LinearLayout lilActionView;
EditText edtActionView;
ImageButton btnActionView;
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.actionview, menu);
    menuItemActionView = menu.findItem(R.id.menu_item_actionview);
    MenuItemCompat.getActionView(menuItemActionView);
    edtActionView = (EditText)
        lilActionView.findViewById(R.id.edtActionView);
    btnActionView = (ImageButton)
        lilActionView.findViewById(R.id.btnActionView);

    btnActionView.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {actionOfTheActionView();}
    });

    return super.onCreateOptionsMenu(menu);
}
```

**ActionViewActivity.class**

```
/**
 * Handling Action from the btnActionView contained by the
 * ActionView
 */
private void actionOfTheActionView() {
    Toast.makeText(this,
        ActionView edt = "+edtActionView.getText().toString(),
        Toast.LENGTH_SHORT)
        .show();

    MenuItemCompat.collapseActionView(menuItemActionView);
}
```

2014 - Android2EE



## Utiliser les ActionsView

Ajouter un Listener pour le changement d'état (collapse/expand)

```
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.actionview, menu);
    menuItemActionView = menu.findItem(R.id.menu_item_actionview);

    // When using the support library, the setOnActionExpandListener()
    // method is
    // static and accepts the MenuItem object as an argument
    MenuItemCompat.setOnActionExpandListener(menuItemActionView,
        new OnActionExpandListener() {
            @Override
            public boolean onOptionsItemSelected(MenuItem item) {
                // Do something when collapsed
                return true; // Return true to collapse action view
            }
            @Override
            public boolean onOptionsItemSelected(MenuItem item) {
                // Do something when expanded
                return true; // Return true to expand action view
            }
        });

    return super.onCreateOptionsMenu(menu);
}
```

**ActionViewActivity.class**

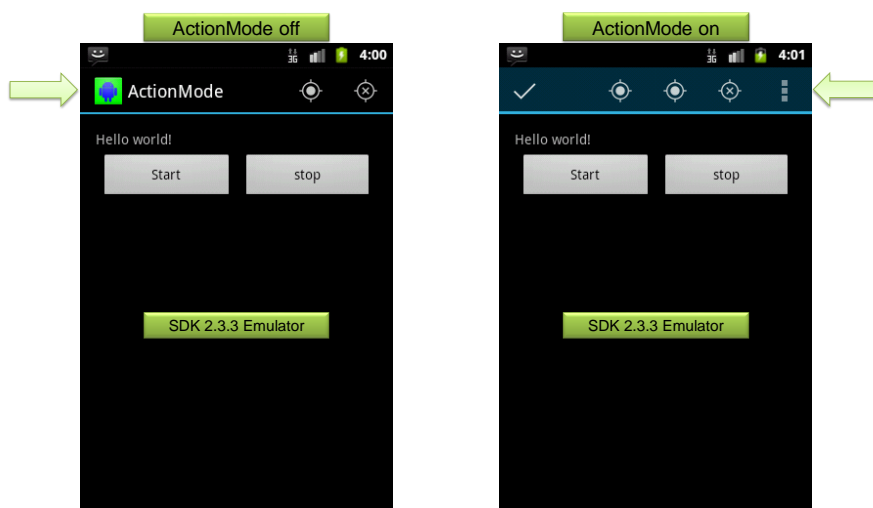
2014 - Android2EE





## Mise en place de l'ActionMode

L'ActionMode permet de changer l'ActionBar normale en ActionBar d'actions pure.



2014 - Android2EE



## Mise en place de l'ActionMode

La principale fonctionnalité est simple à mettre en place, à un détail, le name space pour le showAsAction.

### Java

```
import android.support.v4.app.NavUtils;
import android.support.v7.app.ActionBarActivity;
import android.support.v7.view.ActionMode;
import android.support.v7.view.ActionMode.Callback;
...
//To start the ActionMode
mMode = startSupportActionMode(new Callback() {
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {return false;}
    public void onDestroyActionMode(ActionMode mode) {}
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
        getMenuInflater().inflate(R.menu.action_mode, menu);
        return true;
    }
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        mode.finish();
        return true;
    }
});
//To stop the ActionMode
mMode.finish();
```

### menu\action\_mode.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:actionbarcompat_mse="http://schemas.android.com/apk/res-auto" >
    <item
        android:id="@+id/action_one"
        android:icon="@drawable/ic_action_one_inverse"
        android:orderInCategory="0"
        actionbarcompat_mse:showAsAction="always"
        android:title="One"/>
    <item
        android:id="@+id/action_two"
        android:icon="@drawable/ic_action_two_inverse"
        android:orderInCategory="0"
        actionbarcompat_mse:showAsAction="ifRoom"
        android:title="Two"/>
    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        actionbarcompat_mse:showAsAction="never"
        android:title="@string/action_settings"/>
</menu>
```

2014 - Android2EE

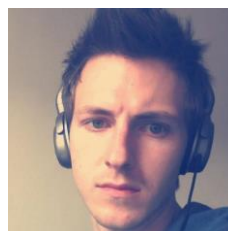


## Big Thanks to Jack Wharton & Chris Banes

Merci à Jack Wharton pour sa librairie et ses exemples d'utilisation (que j'ai utilisé à fond pour le tuto d'ActionBarActivity)



Jack Wharton



Chris Banes

Fondamental

Merci à eux.

***Vous pouvez migrer de la SherlockBar vers ActionBarActivity de manière transparente, sans modifier le code (ou presque).***

2014 - Android2EE



## Q&A

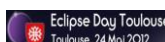
Pour plus d'informations sur l'ActionBarCompat  
un seul réflexe: les formations Android2EE

#android2ee  
[mathias.seguy@android2ee.com](mailto:mathias.seguy@android2ee.com)  
[www.android2ee.com](http://www.android2ee.com)



**Android<sup>2EE</sup>**  
la programmation sous Android

Android by passion  
Passionate by Training  
Expert by technical love



2014 - Android2EE

