Toolbar ActionBar Menu Spinner

Toolbar в качестве ActionBar, во-первых, отключите обычный ActionBar. (Самый простой способ сделать это — унаследовать вашу тему от Theme.AppCompat.NoActionBar)

Во-вторых, создайте экземпляр панели инструментов. Например, включив её в ваш xml-файл разметки

<android.support.v7.widget.Toolbar

android:id="@+id/my\_awesome\_toolbar"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:minHeight="?attr/actionBarSize"

android:background="?attr/colorPrimary" />

Toolbar toolbar = (Toolbar) findViewById(R.id.my\_awesome\_toolbar);

setSupportActionBar(toolbar);

Начиная с этого момента все элементы меню будут отображаться в вашей панели инструментов, наполняясь с помощью стандартных вызовов настройки меню.

**Автономное использование** вы не задаете Toolbar в качестве ActionBar. (в таком случае вы можете использовать любую тему AppCompat и вам не нужно отключать обычный ActionBar).

[Иконка бокового меню из Material Design (как в Google Play)](http://habrahabr.ru/post/242033/)

toggle.setDrawerIndicatorEnabled(true);

**Toolbar Styles**

<android.support.v7.widget.Toolbar

...

//основной стиль, который вы должны использовать, когда вы используете Toolbar в качестве ActionBar

app:theme="@style/ThemeOverlay.AppCompat.ActionBar"

Чтобы Toolbar выглядела как «DarkActionBar» (тёмное содержимое, светлое меню переполнения), укажите атрибуты theme и popupTheme:

app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"

app:popupTheme="@style/ThemeOverlay.AppCompat.Light"

To have a shadow under action bar use AppBarLayout

<android.support.design.widget.AppBarLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary"

app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>

</android.support.design.widget.AppBarLayout>

**внедрение menu в автономный toolbar**

// Set an OnMenuItemClickListener to handle menu item clicks

toolbar.setOnMenuItemClickListener(new Toolbar.OnMenuItemClickListener() {

@Override

public boolean onMenuItemClick(MenuItem item) {

// Handle the menu item

return true;

}

});

// Inflate a menu to be displayed in the toolbar

toolbar.inflateMenu(R.menu.your\_toolbar\_menu);

**Collapsing Toolbar**

* <http://antonioleiva.com/collapsing-toolbar-layout/>
* <https://github.com/safakesberk/CollapsingToolbarExample>
* [Hiding the ActionBar on RecyclerView/ListView onScroll](http://stackoverflow.com/questions/13559275/hiding-the-actionbar-on-recyclerview-listview-onscroll)

**For simple hiding simple action bar on scroll** (CoordinatorLayout is root layout):

<android.support.v7.widget.Toolbar

app:layout\_scrollFlags="scroll|enterAlways"

<android.support.v4.widget.NestedScrollView

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

Restriction: It works ugly if statusBarColor set to transparent

**Toolbar Notes**

* **Note:** Когда внедряете что-то для отображения в панели (например, SpinnerAdater для навигации), удостоверьтесь, что используете контекст с темой панели, полученный с помощью getSupportActionBar().getThemedContext().
* **Note:** используйте новый класс android.support.v7.widget.SwitchCompat
* **Note:** На Lollipop тень под панелью создается с помощью нового API подъема. Чтобы убрать её, вам нужно вызвать getSupportActionBar().setElevation(0) или задать значение атрибута elevation в описании стиля Actionbar.
* **Note:** Что бы вы не помещали в Toolbar, если вы собираетесь управлять отображением этого элемента через visibility. Нужно учесть тот факт, что если где-то в toolbar'e используется searchView, то при колапсе searchView, система видимо проходится по всем элементам бара, и устанавливает им visibility VISIBLE

Варианты решения:

1) мб программно добавлять view в toolbar, и удалять его (вместо управления через visibility) (не пробовал)

2) повешать слушатель на MenuItem, и при колапсе делать visibility gone для элемента

/\*\* Видимо после закрывания serchView система проходится по всем элементам actionBar'a и устанавливает visibility VISIBLE, а нам Spinner нужен только в SPEECHES and MY\_SHEDULE \*/

private void fixBugWithShowingSpinnerWhenSearchViewCollapsed(MenuItem item) {

MenuItemCompat.setOnActionExpandListener(item, new MenuItemCompat.OnActionExpandListener() {

@Override

public boolean onMenuItemActionExpand(MenuItem item) {

return true;

}

@Override

public boolean onMenuItemActionCollapse(MenuItem item) {

if (mActivity != null) {

new Handler().post(new Runnable() {

@Override

public void run() {

((MainActivity) mActivity).setSpinnerHalsVisibility(View.GONE);

}

});

}

return true;

}

});

}

note: OnCloseListener не отрабатывает

**Spinner in Toolbar**

* [StackOverFlow Android Lollipop, add popup menu from title in toolbar](http://stackoverflow.com/questions/26510930/android-lollipop-add-popup-menu-from-title-in-toolbar/26511653#26511653)
* [Spinner guide](http://developer.android.com/guide/topics/ui/controls/spinner.html#Populate)

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_height="?attr/actionBarSize"

android:layout\_width="match\_parent"

android:background="?attr/colorPrimary">

<Spinner

android:id="@+id/spinner\_nav"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

</android.support.v7.widget.Toolbar>

чтобы убрать default bckg

android:background="@null"

**custom spinner item, that use icon like in google I/O app**

<TextView xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@android:id/text1"

style="?attr/spinnerDropDownItemStyle"

android:singleLine="true"

android:layout\_width="match\_parent"

android:layout\_height="?attr/dropdownListPreferredItemHeight"

android:ellipsize="marquee"

android:textColor="@color/white"

android:gravity="center\_vertical"

android:paddingRight="20dp"

android:background="@drawable/abc\_spinner\_mtrl\_am\_alpha"

android:textSize="@dimen/drawer\_item\_text\_size"/>

**ActionBar**

// remove title

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

--->При использовании support library v7 use:

import android.support.v7.app.ActionBarActivity;

Use (or extend) one of the Theme.AppCompat themes for your activity. For example:

<activity android:theme="@style/Theme.AppCompat.Light" ... >

<----- Theme.Holo.NoActionBar - without actionbar

getActionBar().hide(); //отключить ActionBar

actionBar.setDisplayShowHomeEnabled (boolean showHome) //показывать или нет home button

//установить action home icon (чтобы сработало, нужно вызывать после setDisplayShowHomeEnabled(true))

actionBar.setHomeAsUpIndicator(R.drawable.icon\_service\_drawyer\_normal);

actionBar.setHomeButtonEnabled (boolean enabled) // кликабельна или нет home button (?вместе с title)

actionBar.setDisplayHomeAsUpEnabled(true); //показать со стрелкой, символизирующей переход на один уровень к корню (не назад)

actionBar.setBackgroundDrawable(getResources().getDrawable(R.color.white)); //set color backgraund

actionBar.setTitle (CharSequence title)

actionBar.setSubtitle (CharSequence subtitle)

actionBar.setDisplayShowTitleEnabled (boolean showTitle) //скрыть/показать title

actionBar.setIcon

//обработка нажатий

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case android.R.id.home: //button home

finish();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

Title для активити можно так же установить в xml (!!! но если это main activity, то это идет в имя приложения, баг видимо)

<activity

android:name=".MainActivity"

android:label="SomeTitle">

**Menu**

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.accept\_menu, menu);

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.itemAccept:

commitChangesAndGoBack();

return true;

default:

return super.onOptionsItemSelected(item);

}

}

xml ресурс с menu

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:yourapp="http://schemas.android.com/apk/res-auto"> //add for support v7

<item

android:id="@+id/item1" //id

android:title="@string/item1" //title

android:icon="@android:drawable/ic\_menu\_call" //icon

//режим показа элемента

android:showAsAction="always" //всегда показывать

android:showAsAction="ifRoom" //показывать, если есть место

android:showAsAction="ifRoom|withText"

android:showAsAction="never" //не показывать элемент

yourapp:showAsAction="" //for support v7 необходимо использовать атрибут с yourapp prefix

//withText. Актуально для элементов с указанной иконкой. В этом случае для элемента будет показана не только иконка, но и текст из title.

>

</item>

<!-- ..... -->

**NOTE**: Элементы, которые не влезли, или для которых было указано showAsAction="never", по умолчанию будут показаны при нажатии кнопки меню как стандартные меню элементы.  
**NOTE**: Если для элемента показана только иконка, то долгим нажатием на нее можно добиться показа текста

**Adding custom view in actionbar as menu item**

onOptionsItemSelected не будет отрабатывать, поэтому назначаем случатель для view

@Override

public boolean onCreateOptionsMenu(Menu menu) {

menuStar = getLayoutInflater().inflate(R.layout.star\_menu\_item, null);

tvStars = (TextView) menuStar.findViewById(R.id.tvStars);

tvStars.setText("15");

getMenuInflater().inflate(R.menu.menu\_star, menu);

MenuItem menuItem = menu.findItem(R.id.itemStar);

menuItem.setActionView(menuStar);

menuStar.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Log.d("myLogs", "menuStars click");

}

});

return super.onCreateOptionsMenu(menu);

}

Начиная с четвертой версии Андроид, можно использовать атрибут uiOptions для Activity или Application в манифесте. Если присвоить ему значение splitActionBarWhenNarrow, то при нехватке места actio item'ы уйдут в нижний actionbar (Правда, при этом почему-то перестает работать withText в showAsAction. - startandroid)

[**Overlay actionbar**](http://developer.android.com/training/basics/actionbar/overlaying.html)

ActionBar прорисовывается поверх основного лэйаута (по умолч если тема Dark, то прозрачный но с линией внизу, а если Light, то не прозрачный)

<style name="CustomActionBarTheme"

parent="@android:style/Theme.AppCompat"> //@style/ instead if support library using

<item name="android:windowActionBarOverlay">true</item>

<!-- Support library compatibility -->

<item name="windowActionBarOverlay">true</item>

</style>

The one with the android: prefix is for versions of Android that include the style in the platform and the one without the prefix is for older versions that read the style from the Support Library.

android:paddingTop="?android:attr/actionBarSize"

<!-- using support library, use code below instead -->

android:paddingTop="?attr/actionBarSize"

[Хабр transparent actionbar](http://habrahabr.ru/post/190102/)

**ActionMode**

<http://developer.android.com/reference/android/view/ActionMode.html><http://developer.android.com/reference/android/view/ActionMode.Callback.html><http://startandroid.ru/ru/uroki/vse-uroki-spiskom/183-urok-113-android-3-actionmode-kak-alternativa-kontekstnomu-menju.html>

Чтобы вызвать ActionMode, используется метод startActionMode. На вход он берет объект callback, который будет обрабатывать все события, связанные с ActionMode. На выходе получаем объект ActionMode.

public ActionMode startActionMode (ActionMode.Callback callback)

actionMode.finish(); - выключаем action mode

private ActionMode.Callback callback = new ActionMode.Callback() {

public boolean onCreateActionMode(ActionMode mode, Menu menu) {

mode.getMenuInflater().inflate(R.menu.context, menu);

return true;

}

public boolean onPrepareActionMode(ActionMode mode, Menu menu) {

return false;

}

public boolean onActionItemClicked(ActionMode mode, MenuItem item) {

Log.d(LOG\_TAG, "item " + item.getTitle());

return false;

}

public void onDestroyActionMode(ActionMode mode) {

Log.d(LOG\_TAG, "destroy");

actionMode = null;

}

};

onCreateActionMode– вызывается при создании ActionMode. Возвращаем true, если ActionMode можно создавать. Здесь мы наполняем ActionMode пунктами меню (через объект Menu).

onPrepareActionMode– вызывается при обновлении ActionMode. Например, в случае вызова метода invalidate. Возвращаем true, если ActionMode можно обновить.

onActionItemClicked– обработка нажатия на какой-либо пункт ActionMode. Будем выводить в лог текст нажатого пункта.

onDestroyActionMode– вызывается при закрытии ActionMode. Пишем лог и обнуляем переменную actionMode, чтобы в onClick (см.выше) у нас работала проверка (actionMode == null).

Вызов ActionMode по долгому нажатию уже реализован для наследников класса AbsListView, например GridView и ListView.

lvActionMode.setChoiceMode(ListView.CHOICE\_MODE\_MULTIPLE\_MODAL);

lvActionMode.setMultiChoiceModeListener(new MultiChoiceModeListener() {

public void onItemCheckedStateChanged(ActionMode mode,

int position, long id, boolean checked) {

Log.d(LOG\_TAG, "position = " + position + ", checked = "

+ checked);

}

//далее все те же методы как в callback actionmode выше

};

Для списка в режиме CHOICE\_MODE\_MULTIPLE\_MODAL есть еще один способ, кроме долгого нажатия на пункт, вызвать ActionMode. Это метод setItemChecked. Если выделить этим методом какой-либо пункт, то ActionMode появится.

У ActionMode, как и у ActionBar есть методы:

setTitle- установить свой заголовок

setSubtitle- установить свой подзаголовок

**Для sdk = 23+** есть возможность сделать плавающий actionMode ( [см. здесь (Новое, Пятое. Еще одно контекстное меню)](http://habrahabr.ru/post/271737/) )

**Transparent Status bar**

<http://www.androprogrammer.com/2015/01/navigation-drawer-with-transparent.html>

<http://stackoverflow.com/questions/26440879/how-do-i-use-drawerlayout-to-display-over-the-actionbar-toolbar-and-under-the-st>

<http://stackoverflow.com/questions/26745300/navigation-drawer-semi-transparent-over-status-bar-not-working>

**Depricated**

**УСТАР програмное добавление меню с помощью MenuCompat**

@Override

public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {

MenuItem itemTag = menu.add(Menu.NONE, MENU\_TAG, Menu.NONE, "Tag");

itemTag.setIcon(R.drawable.ic\_launcher);

MenuCompat.setShowAsAction(itemTag, 2); //deprecated (использовать, если нужна поддержка api < 11)

super.onCreateOptionsMenu(menu, inflater);

}

**УСТАР табы**

actionBar.setNavigationMode(ActionBar.NAVIGATION\_MODE\_TABS);

ActionBar.Tab tab1 = actionBar.newTab();

tab1.setText("tab1");

tab1.setTabListener(tabListener);

ActionBar.TabListener tabListener = new ActionBar.TabListener() {

@Override

public void onTabSelected(ActionBar.Tab tab, FragmentTransaction fragmentTransaction) {

Log.d(LOG\_TAG, "reselected tab: " + tab.getText());

}

@Override

public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction fragmentTransaction) {

}

@Override

public void onTabReselected(ActionBar.Tab tab, FragmentTransaction fragmentTransaction) {

}

};

**УСТАР Navigation mode**

actionBar.setNavigationMode(ActionBar.NAVIGATION\_MODE\_LIST);

//Для выпадающего списка необходимо создать адаптер, реализующий SpinnerAdapter.

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_spinner\_item, data);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

actionBar.setListNavigationCallbacks(adapter, new ActionBar.OnNavigationListener() {

@Override

public boolean onNavigationItemSelected(int i, long l) {

return false;

}

});

Для обоих видов навигации мы всегда можем получить текущий выбранный элемент с помощью метода getSelectedNavigationIndex

для навигации с табами есть также метод getSelectedTab, возвращающий текущий выбранный таб.

**Solution c unfo, чтобы не расплывались item's menu в action bar'e**

mActivity.getMenuInflater().inflate(R.menu.filter\_menu, menu);

LayoutInflater inflater = LayoutInflater.from(getActivity());

final MenuItem filterItem = menu.findItem(R.id.filter);

View actionView = inflater.inflate(R.layout.custom\_action\_button, null);

ImageButton menuItemButton = (ImageButton) actionView.findViewById(R.id.custom\_action\_bar\_button);

menuItemButton.setImageResource(R.drawable.filtr);

filterItem.setActionView(actionView);

custom\_action\_button

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:clickable="true">

<ImageButton

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:id="@+id/custom\_action\_bar\_button"

android:background="@null"

android:clickable="false"

android:focusable="true"

android:scaleType="centerInside"

android:paddingTop="8dp"

android:paddingBottom="8dp"/>

</FrameLayout>

filter\_menu

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item android:id="@+id/filter" android:title="Фильтр"

android:icon="@drawable/filtr" android:showAsAction="ifRoom|withText"/>

</menu>