

91 桌面皮肤主题插件平台参考手册

更新日期：2013-2-28

Version 1.00 last saved by: 陈福斌

91 桌面皮肤主题插件平台参考手册	1
一、 91 桌面皮肤插件 SDK 构成	1
二、 支持 Android 版本	1
三、 91 桌面皮肤插件 SDK 环境搭建	1
1. 导入 91 桌面 Lib 工程	1
2. 为您的工程引入 91 桌面 Lib 工程	1
3. 配置 AndroidManifest.xml	2
四、 集成 91 桌面皮肤插件 SDK 功能	2
1. 设置 AppId、AppKey 和 AppSkinPath	2
2. 创建 View 并添加到需要展示的地方	2
3. 自定义对话框风格	3
4. 下载统计接口	3
5. 接收并处理换肤广播	4

一、 91 桌面皮肤插件 SDK 构成

(略)

二、 支持 Android 版本

支持 Android SDK2.1 及以上版本。

三、 91 桌面皮肤插件 SDK 环境搭建

1. 导入 91 桌面 Lib 工程

(略)

2. 为您的工程引入 91 桌面 Lib 工程

(略)

3. 配置 AndroidManifest.xml

您需要将 NdHiExAndroidManifest.xml 里面的权限，如图：

```
<uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
```

Activity,service,receiver 如图：

```
<!-- 下载任务管理 -->
<activity android:name="com.nd.hilauncherdev.lib.theme.HiLauncherExDownTaskManagerActivity"/>
<!-- 下载服务 -->
<service android:name="com.nd.hilauncherdev.lib.theme.down.DownloadService"></service>
<!-- 通知栏打开确认对话框 -->
<service android:name="com.nd.hilauncherdev.lib.theme.service.CreateDialogService"></service>
<!--桌面主题安装结果接收器-->
<receiver android:name="com.nd.hilauncherdev.lib.theme.api.ThemeInstallResultReceiver">
    <intent-filter>
        <action android:name="nd.pandahome.external.response.theme.apk.install" />
    </intent-filter>
</receiver>
```

拷贝到您的工程的 AndroidManifest.xml 文件里面。

四、集成 91 桌面皮肤插件 SDK 功能

1. 设置 AppId、AppKey 和 AppSkinPath

首先您需要在程序开始的地方，优先设置 AppId、AppKey 和 AppSkinPath(皮肤下载后的解压路径)。所以在接入 91 桌面皮肤插件之前，您需要向 91 桌面团队申请可接入的 AppId 和 AppKey。所有的 SDK 操作都需要设置这两个参数才能正常工作。AppId 和 AppKey 通过 NdLauncherExThemeApi 里的 init 方法设置。

举例如下：

```
//1. 设置工作 略
NdLauncherExAppSkinSetting appSkinSetting = new NdLauncherExAppSkinSetting();
appSkinSetting.setAppId("1000");
appSkinSetting.setAppKey("3");
appSkinSetting.setAppSkinPath("/sdcard/appSkinTest/");
//appSkinSetting.setThemeExDialog(themeExDialog) //自定义对话框时设置
//appSkinSetting.setThemeExDownAction(themeExDownAction) //下载动作回调设置

//2. 执行初始化
NdLauncherExThemeApi.init(this, appSkinSetting);
```

2. 创建 View 并添加到需要展示的地方

Sdk 提供两个一个是浏览下载的 View，一个是下载任务管理的 View。

```
//动态创建浏览皮肤的view
HiLauncherExThemeShinView themeShinView = new HiLauncherExThemeShinView(this);
themeShinView.initView();

//在不使用的时候销毁view,一般在Activity的onDestroy执行以下操作
if (themeShinView!=null){
    themeShinView.destroyView();
}
```

```
//动态创建下载任务管理View
DownTaskManageView downTaskManageView = new DownTaskManageView(this);
downTaskManageView.initView();

//在不使用的时候销毁view,一般在Activity的onDestroy执行以下操作
if (downTaskManageView!=null){
    downTaskManageView.destroyView();
}
```

3. 自定义对话框风格

如果要实现自定义的对话框风格,请实现接口 `NdLauncherExDialogCallback` 并在初始化时,将实例设置到 `NdLauncherExAppSkinSetting` 对象中。

```
public interface NdLauncherExDialogCallback {
    /**
     * 通用对话框
     * @param ctx
     * @param icon 图标
     * @param title 标题
     * @param view 视图
     * @param positive 确定按钮文字
     * @param negative 取消按钮文字
     * @param ok 确定回调
     * @param cancel 取消回调
     */
    public Dialog createThemeDialog(Context ctx, int icon, CharSequence title, CharSequence message,
    }

appSkinSetting.setThemeExDialog(themeExDialog); //自定义对话框时设置
```

4. 下载统计接口

如果要统计下载动作,请实现接口 `NdLauncherExDownActionCallback` 并在初始化时,将实例设置到 `NdLauncherExAppSkinSetting` 对象中。

```

public interface NdLauncherExDownActionCallback{

    /**
     * 资源首次下载回调(断点续传不调用)
     * -----实现此方法时,为防止ANR请在内部开启 Thread处理统计。
     * @param ctx
     * @param itemType
     *      ThemeItem.ITEM_TYPE_SKIN 表示皮肤
     *      ThemeItem.ITEM_TYPE_THEME 表示主题
     *      ThemeItem.ITEM_TYPE_LAUNCHER 表示桌面
     * @param resID 服务端资源id统一为主题id+itemType
     */
    public void firstDown(Context ctx, int itemType, String resID);
}

```

```
appSkinSetting.setThemeExDownAction(themeExDownAction); //下载动作回调设置
```

5. 接收并处理换肤广播

以下是接入应用接收自己能处理的换肤广播,并做相应响应的例子:

```

public class SkinAppReceiver extends BroadcastReceiver{
    public void onReceive(Context context, Intent intent) {

        if (intent==null)
            return ;
        if (intent.getAction().equals(NdLauncherExThemeApi.ND_HILAUNCHER_THEME_SKIN_APPLY_ACTION)) {

            String appID = intent.getStringExtra( NdLauncherExThemeApi.ND_HILAUNCHER_THEME_APP_ID_KEY );

            if (appID!=null && appID.equals(NdLauncherExThemeApi.getAppId()) ){
                //Do Something here
                //获取本次皮肤应用的皮肤路径
                String skinPath = intent.getStringExtra( NdLauncherExThemeApi.ND_HILAUNCHER_THEME_APP_SKIN_PATH_KEY );
            }
        }
    }
}

/**皮肤应用广播*/
public static final String ND_HILAUNCHER_THEME_SKIN_APPLY_ACTION = "nd.pandahome.external.request.skin.apply";

/**接入应用ID key*/
public static final String ND_HILAUNCHER_THEME_APP_ID_KEY = "skinAppID";

/**皮肤路径 key*/
public static final String ND_HILAUNCHER_THEME_APP_SKIN_PATH_KEY = "skinPath";

```