

DU Ad Platform_SDK for Android 接入手册（天气通）

DUAd_SDK_WeatherWizard v1.0.2

百度在线网络技术（北京）有限公司

目 录

- 1. 获取身份 1
- 2. 加载与配置 1
 - 2.1 加载 SDK 文件 1
 - 2.2 配置 Manifest 2
 - 2.3 混淆 2
- 3. Application 中初始化环境 3
 - 3.1 Json 配置 3
 - 3.2 初始化函数 3
- 4. 天气功能使用 3
 - 4.1 应用内天气卡片 3
 - 4.2 桌面天气悬浮框 4
 - 4.3 通知栏天气 5
 - 4.4 直接使用天气结果页 6

前提: WeatherWizard SDK 需要依赖 DU Ad Platform_SDK HW1.0.9.8 或 CW1.0.9.7 (含)以上版本。

在接入 WeatherWizard 之前 需要完成 HW 或 CW 初始化, 加载, 代码混淆, 三个部分。

1. 获取身份

请参照 HW 或 CW 版 DUADplatform SDK 文档 第 3 章; 申请 WeatherWizard 广告位 ID 时, 需申请的广告位类型为「天气通」。



创建广告位

* 广告位名称:

* 广告触发步骤: ⓘ

* 广告形式: ☐ 原生 ☐ 插屏 ☐ 广告墙 ☐ 横幅 ☐ 视频
☐ 触发式 ☐ 来电提醒 ☒ 天气通

☐ 第三方平台广告位ID配置 (当选择的广告来源为第三方平台时填写)

保存

2. 加载与配置

2.1 加载 SDK 文件

- 1) 将 DAPSDK_Weather-release-xxx.aar 加入工程 libs 文件夹下:
- 2) 配置工程 build.gradle:

```
repositories {  
    flatDir {  
        dirs 'libs'  
    }  
}  
  
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile(name: 'DuappsAd-xW-xxx-release', ext: 'aar')  
    // WeatherSDK 必须依赖 DAP 广告 SDK  
    compile(name: 'DAPSDK_Weather-release-xxx', ext: 'aar')  
}
```

*注: flatDir 指定的位置 即为 aar 存放的位置

2.2 配置 Manifest

```
<!--定位-->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<!--悬浮框-->
<uses-permission android:name="android.permission.GET_TASKS"/>

<!--dapweather begin-->
<activity
    android:name="com.daps.weather.DapWeatherActivity"
    android:screenOrientation="portrait"
    />
<receiver android:name="com.daps.weather.reciver.DapWeatherBroadcastReceiver">
    <intent-filter android:priority="90000">
        <action android:name="android.intent.action.USER_PRESENT"/>
    </intent-filter>
</receiver>
<service android:name="com.daps.weather.service.DapWeatherMsgService"></service>
<service android:name="com.daps.weather.location.DapWeatherLocationsService">
</service>
<!--dapweather end-->
```

2.3 混淆

```
-keep public class com.daps.weather.notification.DapWeatherNotification {
    public <methods>;
}
-keep public class com.daps.weather.weathercard.DapWeatherView {
    public <methods>;
}
-keep public class com.daps.weather.DapWeatherActivity {
    public <methods>;
}
-keep public class com.daps.weather.service.DapWeatherMsgService {
    *;
}
-keep public class com.daps.weather.location.DapWeatherLocationsService {
    *;
}
-keep public class com.daps.weather.DapWeather {
    public <methods>;
}
-keep public class com.daps.weather.weathercard.DapWeatherEnterImageView {
```

```
public <methods>;
}
-keep public class com.daps.weather.floatdisplay.FloatDisplayController {
    *;
}
-keep public class com.daps.weather.base.SharedPrefsUtils {
    public static boolean isSuspensionOn(android.content.Context);
    public static void setSuspensionOn(android.content.Context,boolean);
}
-keep class com.daps.weather.bean.**{*};
```

3. Application 中初始化环境

3.1 Json 配置

```
{
  "weather": [
    {
      "pid": "YOUR_DAP_PLACEMENT_ID"
    }
  ]
}
```

注：用来初始化 DuAdNetwork.init()，json 数组部分，具体参见 HW CW 版本初始化章节。

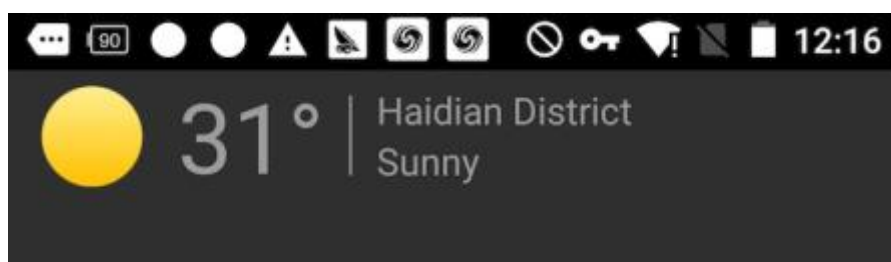
3.2 初始化函数

```
DapWeather.init((Application) getApplicationContext(),pid);
```

注：在初始化 DapWeather 的同时，也要初始化 DuAdNetwork.init(),才能正常展示广告（参见 HW，CW 版本初始化章节）；

4. 天气功能使用

4.1 应用内天气卡片



- 天气卡片控件，可以添加到需要显示的布局中。

DapWeatherView

- 加载天气数据接口

DapWeatherView.load()

```
//卡片入口
RelativeLayout rl = (RelativeLayout) findViewById(R.id.demo_weather_view_rl);
mDuWeatherView = new DapWeatherView(this);
rl.addView(mDuWeatherView);
findViewById(R.id.btn_weather_view).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        mDuWeatherView.load();//天气卡片加载天气数据，有数据自动填充DapWeatherView卡片
    }
});
```

4.2 桌面天气悬浮框

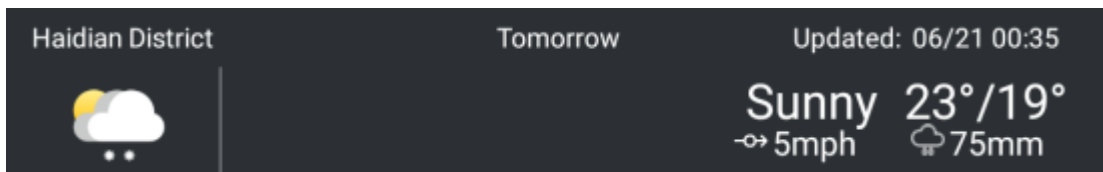
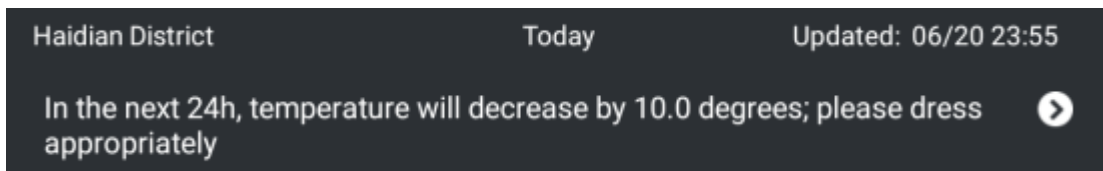
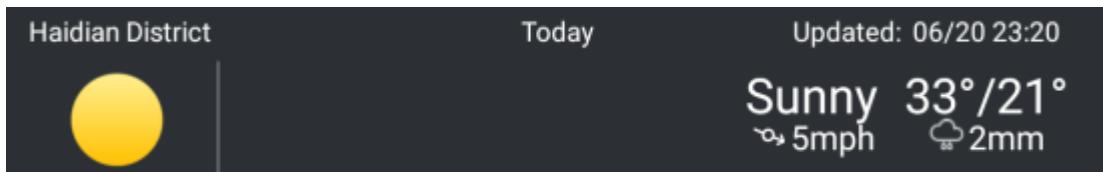


- 悬浮框开关
`FloatDisplayController.setFloatSerachWindowIsShow(Context context, Boolean isShown);`

参数	说明
Context context	Application context
Boolean isShown	True 表示打开悬浮框

```
//悬浮框入口
findViewById(R.id.btn_weather_suspension).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //悬浮窗入口，传入true开启，false关闭。
        FloatDisplayController.setFloatSerachWindowIsShow(getApplicationContext(), isShow);
        isShow = !isShow;
    }
});
```

4.3 通知栏天气



注：通知栏是定时推送服务，由百度服务器控制；

7:30-12:00 推送当日天气；

13:00-20:30 推送升降温天气；

20:30-23:00 推送次日天气；

每个时间段只会推送一次天气通知。

- 初始化
`mNotification = DapWeatherNotification.getInstance(getApplicationContext());`
- 设置通知栏常驻
`mNotification.setOngoing(Boolean bool)`
true 为常驻通知栏，false 为临时通知栏
- 设置升降温推送功能
`mNotification.setPushElevatingTemperature(Boolean bool);`
是否开启升降温推送功能，默认为 true
- 设置今天和明日天气推送功能
`mNotification.setPushTodayTomorrowWeather(Boolean bool);`

是否开启今天和明日天气推送功能，默认是 true

4.4 直接使用天气结果页

```
startActivity(new Intent(MainActivity.this, DemoWeatherViewActivity.class));
```

