

[QbSdk](#)

[initX5Environment](#)

[setDownloadWithoutWifi](#)

[PreInitCallback](#)

[onCoreInitFinished](#)

[onViewInitFinished](#)

[WebView](#)

[getCrashExtraMessage](#)

[getWebScrollX](#)

[getWebScrollY](#)

[getView](#)

[getWidth](#)

[getHeight](#)

[常用链接: 官方论坛 接入文档 常见问题 问题反馈](#)

QbSdk

com.tencent.smtt.sdk.QbSdk

X5内核API公共接口类，主要包含一些静态方法用来设置x5内核的加载行为。

initX5Environment

public static void initX5Environment(final Context context, final [PreInitCallback](#) callback)

x5内核预加载接口，建议在进入webview场景前调用该接口提升x5内核使用率和优化冷启动耗时，可参考接入示例

-context:上下文环境
-callback: 预加载回调接口, 细节见PreInitCallback
-return: void

-代码示例:

```
public class APPApplication extends Application {

    @Override
    public void onCreate() {
        // TODO Auto-generated method stub
        super.onCreate();
        //搜集本地tbs内核信息并上报服务器, 服务器返回结果决定使用哪个内核。

        QbSdk.PreInitCallback cb = new QbSdk.PreInitCallback() {

            @Override
            public void onViewInitFinished(boolean arg0) {
                // TODO Auto-generated method stub
                Log.d("app", " onViewInitFinished is " + arg0);
            }

            @Override
            public void onCoreInitFinished() {
                // TODO Auto-generated method stub
            }
        };
        //x5内核初始化接口
        QbSdk.initX5Environment(getApplicationContext(), cb);
    }

}
```

setDownloadWithoutWifi

public static void setDownloadWithoutWifi(boolean download)

非wifi网络条件下是否允许下载内核, 默认为false。当本地无可用内核会去主动下载内核, 会产生24M左右的数据流量, 为了节省用户流量, 默认只在wifi条件下才会去下载。开发者可以通过调用该接口设置为true, 允许用户在非wifi条件下也下载内核。

-download: 是否允许非wifi场景下下载内核

PreInitCallback

com.tencent.smtt.sdk.QbSdk.PreInitCallback

X5内核初始化回调接口

onCoreInitFinished

public void onCoreInitFinished()

x5内核初始化完成回调接口，此接口回调并表示已经加载起来了x5，有可能特殊情况下x5内核加载失败，切换到系统内核。

-return:void

onViewInitFinished

public void onViewInitFinished(boolean isX5Core)

x5内核初始化完成回调接口，可通过参数判断是否加载起来了x5内核

-isX5Core:isX5Core为true表示x5内核加载成功；false表示加载失败，此时会自动切换到系统内核。如果在此回调前创建webview会导致使用系统内核
-return:void

WebView

com.tencent.smtt.sdk.WebView

public class WebView extends FrameLayout implements android.view.View.OnLongClickListener

X5 SDK 所提供的WebView类，是对WebView真正实现类的聚合包装，实际上是一个FrameLayout。

getCrashExtraMessage

public static String getCrashExtraMessage(Context context)

获取x5 Crash信息，包括x5 SDK版本号及当前使用的内核版本号。AP发生crash时务必带上该信息，便于定位分析问题。

-context: 上下文环境

-return: 返回x5内核发生crash的详细信息

-代码示例(bugly crash上报):

```
UserStrategy strategy = new UserStrategy(appContext);
strategy.setCrashHandleCallback(new CrashReport.CrashHandleCallback() {
    public Map onCrashHandleStart(int crashType, String errorType, String errorMessage, String
errorStack) {
        LinkedHashMap map = new LinkedHashMap();
        String x5CrashInfo = com.tencent.smtt.sdk.WebView.getCrashExtraMessage(appContext);
        map.put("x5crashInfo", x5CrashInfo);
        return map;
    }
});
@Override
public byte[] onCrashHandleStart2GetExtraDatas(int crashType, String errorType, String
errorMessage, String errorStack) {
    try {
        return "Extra data.".getBytes("UTF-8");
    } catch (Exception e) {
        return null;
    }
}
}});
CrashReport.initCrashReport(appContext, APPID, true, strategy);
```

getWebScrollX

```
public int getWebScrollX()
```

获取webview的scrollX。因为x5webview最外层是一个FrameLayout类型的包装层，view的getScrollX方法不可重载，所以x5webview的getScrollX方法获取的只是包装层的scrollX

-return: 返回webview在X方向上的偏移量

getWebScrollY

```
public int getWebScrollY()
```

获取webview的scrollY。因为x5webview最外层是一个FrameLayout类型的包装层，view的getScrollY方法不可重载，所以x5webview的getScrollY方法获取的只是包装层的scrollY

-return: 返回webview在Y方向上的偏移量

getView

```
public View getView()
```

SDK WebView是一个FrameLayout类型的聚合包装类，通过该接口获WebView真正webview的尺寸，如可以通过com.tencent.smtt.sdk.WebView.getView().getWidth()方式获取webview的宽度

getWidth

public int getWidth ()

SDK WebView实际上是一个包装了webview的FrameLayout，WebView.getWidth()获得的是FrameLayout的宽度，如果需要获取内置WebView真实的宽度，需要通过com.tencent.smtt.sdk.WebView.getView().getWidth()方式获取

-return: 返回webview包装层的宽度

getHeight

public int getHeight()

SDK WebView实际上是一个包装了webview的FrameLayout，WebView.getHeight()获得的是FrameLayout的宽度，如果需要获取内置WebView真实的宽度，需要通过com.tencent.smtt.sdk.WebView().getView.getHeight()方式获取

-return: 返回webview包装层的高度

更新日期：2017-04-27

常用链接: [官方论坛](#) [接入文档](#) [常见问题](#) [问题反馈](#)
