

Android debug调试定位问题

- 一、明确客户端改动
- 二、debug app准备
- 三、debug调试开启（app执行操作，这里是进H5打开激励视频）

背景：app在执行业务过程中或多或少会出一些问题（比如激励视频未吊起、白屏、crash等），而开发同学因为项目排期紧张也没能及时看问题。

目标：能基本的定位问题出现哪里，跟开发同步，缩短开发解决问题的时间。

本篇拿[爱清理H5唤起激励视频-从壳工程桥接替换为本地native桥接需求](#)（检查桥接调用是否正常，H5打开播放激励视频是否正常），讲解大致debug调试流程：

一、明确客户端改动

- 改动文件：catfish_android/qcore/src/main/java/com/xiaoqiao/qclean/base/bridge/LocalBridge.java
- 新增桥接方法：**playEaVideoCallNative**（前端会去调这个方法）

```
@JavascriptApi
public void playEaVideoCallNative(Object arg, CompletionHandler handler) {
    if (ClickUtil.isFastDoubleClick(R.id.ad_bridge_id, 4000)) { // 4000
        return;
    }
    if (arg == null) {
        return;
    }
    IH5Bridge bridge = BridgeUtil.getH5Bridge(); //
    HybridContext hybridContext = this.getHybridContext();
    Activity activity = hybridContext.getActivity();
    if (hybridContext == null || activity == null || bridge == null ||
        activity.isFinishing() || activity.isDestroyed()) {
        handler.complete(EntityUtil.getResp(-1, ""));
        return;
    }

    String slotId = "";
    JSONObject adDescriptions = null;
    //
    try {
        JSONObject jsonObject = (JSONObject) arg;
        if (jsonObject.has("slotId")) {
            slotId = jsonObject.getString("slotId");
        }

        adDescriptions = jsonObject.optJSONObject("adDescriptions");
    } catch (JSONException e) {
        e.printStackTrace();
    }

    if (TextUtils.isEmpty(slotId)) {
        return;
    }

    Bundle extraBundle = new Bundle();
    extraBundle.putInt("fullscreen", 1);

    if (adDescriptions == null) {
        extraBundle.putString("descriptions", "{ 'countdown_award_des': '', 'countdown_success_des': '' }");
    } else {
        extraBundle.putString("descriptions", adDescriptions.toString());
    }

    String finalSlotId = slotId;
```

```
//
AdUtils.loadRewardVideoAd(activity, null, finalSlotId, AdUtils.getGdtAppId(ADConstants.GDT_MEDIA_ID),
extraBundle, new AdRewardSimpleCallBack() {
@Override
public void onAdLoadFailure(String msg) {
super.onAdLoadFailure(msg);
if (handler != null) {
handler.complete(EntityUtil.getResp(-1, ""));
}
}

@Override
public void onAdComplete() {
super.onAdComplete();
if (handler != null) {
handler.complete(EntityUtil.getResp(0, "{\"success\":\"true\"}"));
}
}

@Override
public void onAdError(String msg) {
super.onAdError(msg);
if (handler != null) {
handler.complete(EntityUtil.getResp(-1, ""));
}
}
});
}
}
```

二、debug app准备

- 1、数据线连接手机（开启开发者调试）和Android studio（工具栏有显示机型代表已成功连接）



- 2、Android studio本地先sync project with gradle files一下（用于同步gradle文件依赖）



- 3、打debug包



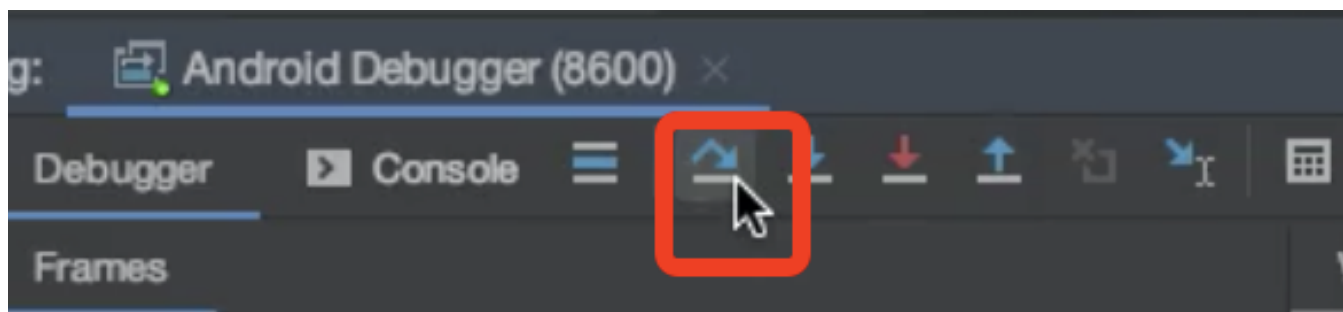
- 4、涉及改动的代码加断点（因为H5打开激励视频是调桥接，故在桥接方法下）

录屏：[加断点](#)

三、debug调试开启（app执行操作，这里是进H5打开激励视频）

可以在下方视频看到，在H5任务中心点击看视频时，桥接playEaVideoCallNative方法被执行了：

- 1、Android studio最下方的Debugger区显示当前方法调用其它文件方法的一些入参情况；
- 2、蓝色块状覆盖的行表示当前执行的代码行，点击Debugger区域的该按钮就会一步步执行代码；
- 3、方法正常执行下，所有的行、判断分支（不符合条件就会跳过该分支）都会被执行一遍；
- 4、最后方法执行完成，app内正常执行业务流程（这里是正常吊起了激励视频）；反之如果方法在执行过程中return了，我们可以通过查看debugger区的入参内容+return的条件判断初步定位问题，然后跟客户端开发说明情况；如果加断点的函数未被正常吊起，那么就是前端没有调这个桥接，找前端开发问罪即可。



录屏: [debug-app](#)