**微信小游戏接入JS-SDK**

目录

[**微信小游戏接入JS-SDK** 1](#_Toc28272328)

[JS-SDK接入 1](#_Toc28272329)

[小游戏JS-SDK接入 1](#_Toc28272330)

[配置域名 1](#_Toc28272331)

[效果检查 3](#_Toc28272332)

[sdk是否生效检查 3](#_Toc28272333)

[剪贴板格式设置 3](#_Toc28272334)

[小游戏js-sdk检查 4](#_Toc28272335)

# JS-SDK接入

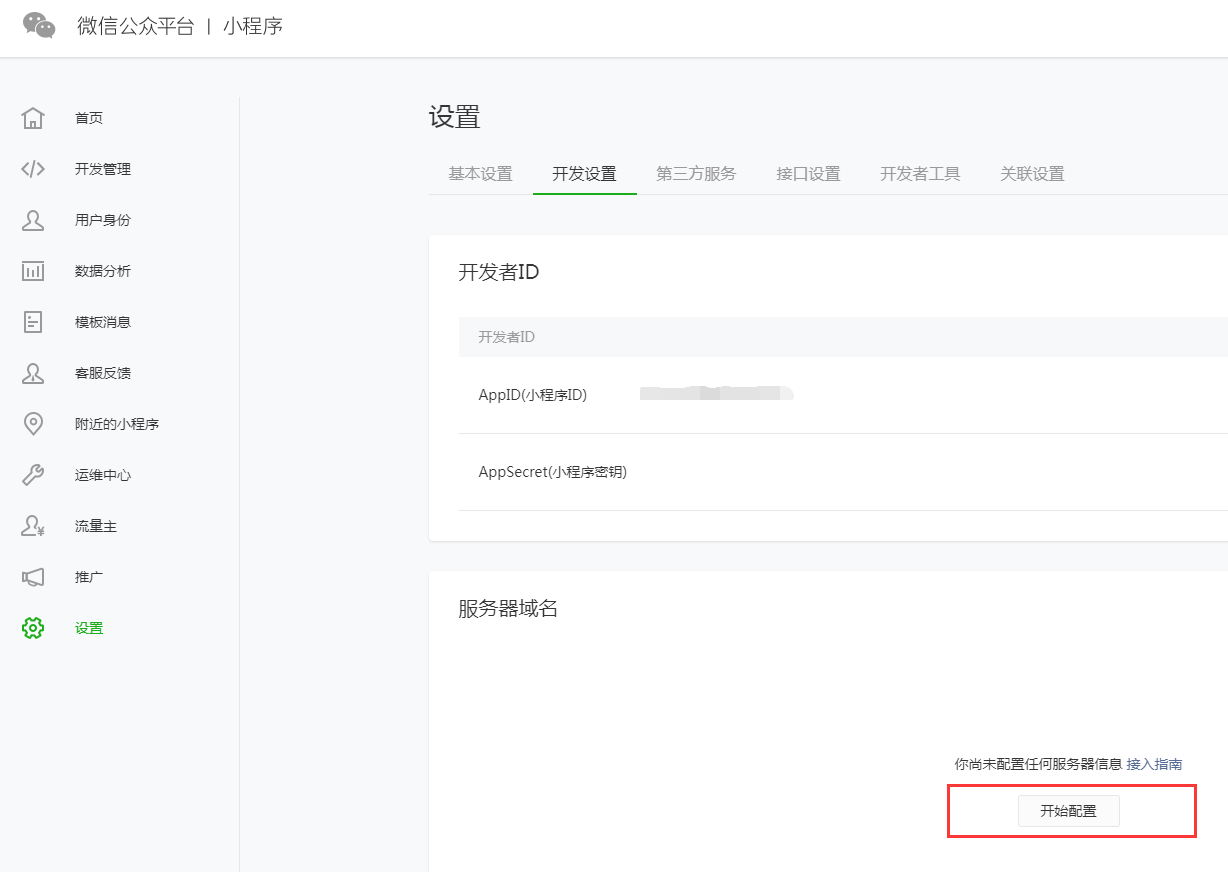
## 小游戏JS-SDK接入

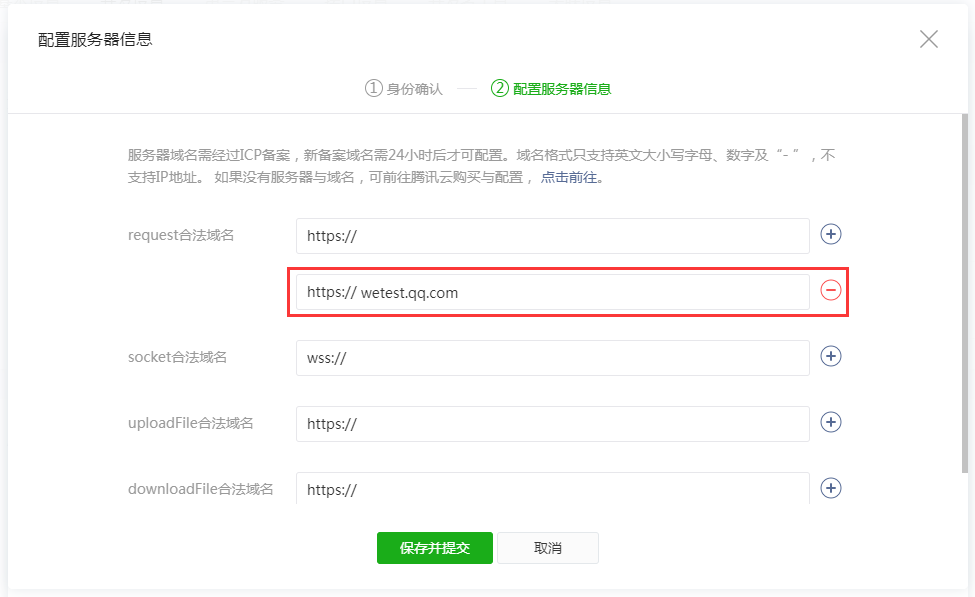
将JS文件放入工程中之后，需要在游戏引擎初始化完成之后引入，以cocos工程为例，在game.js中引入JS-SDK。见下图：

# 配置域名



登录公众平台，将Wetest 域名<https://wetest.qq.com/> 添加到request合法域名当中。

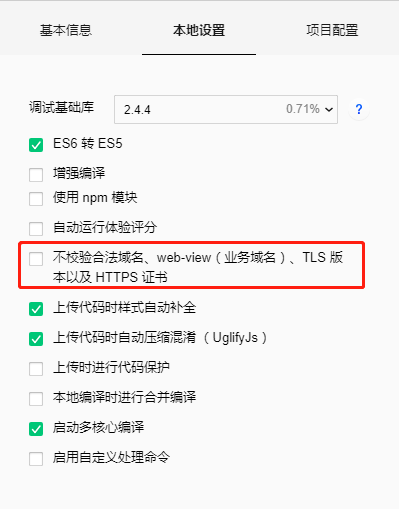




# 效果检查

## sdk是否生效检查

取消本地项目中“不校验合法域名、web-view（业务域名）、TLS版本以及HTTPS证书”的选项。



sdk接入成功后， 会检测后台是否配置wetest.qq.com白名单，并弹框提示检测结果。如下图：

* 已配置白名单：



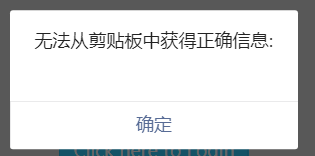
* 未配置白名单：



出现上述任何一个弹框，都说明jssdk接入成功。**如果没有弹框，说明jssdk未接入。**

## 剪贴板格式设置

sdk接入成功后，会弹框提示剪贴板解析错误。如下图：



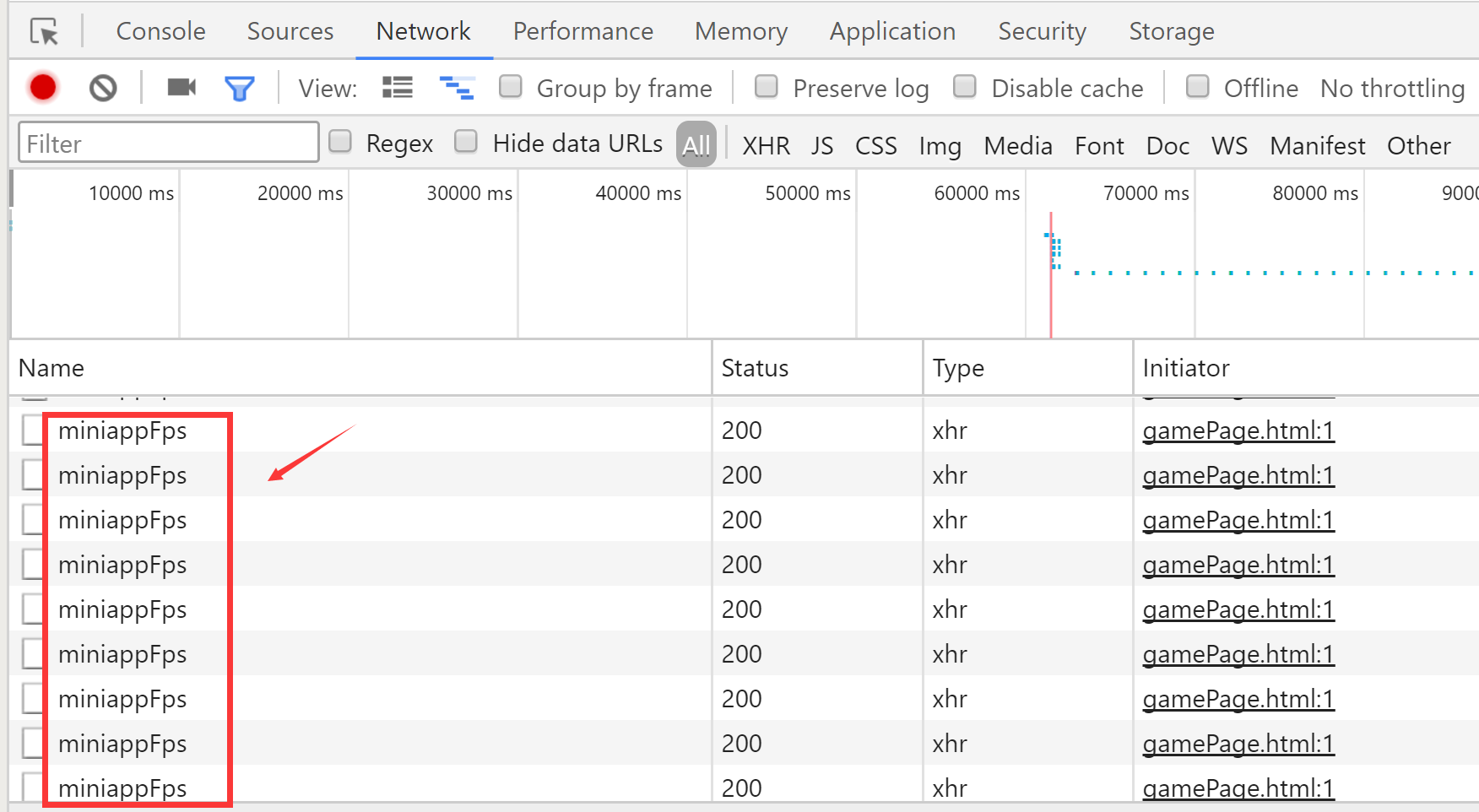
这是因为jssdk通过剪贴板与平台进行数据互通。接入sdk后可忽略此问题。

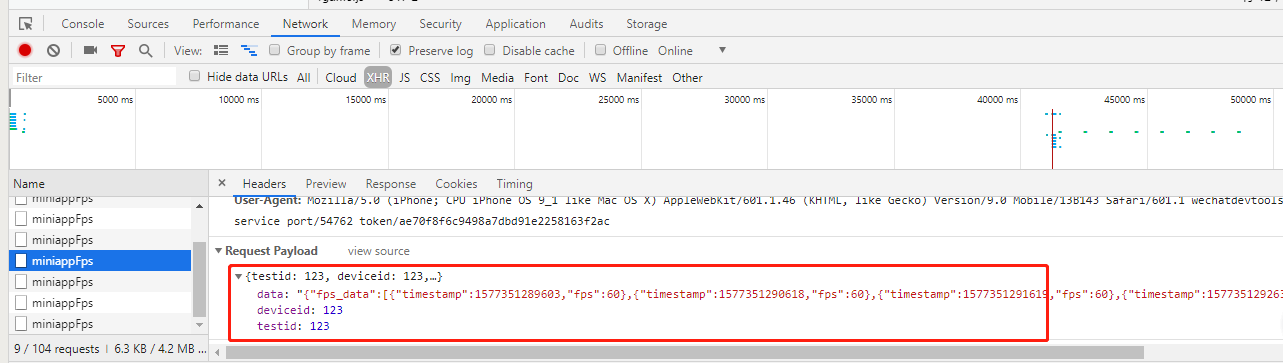
## 小游戏js-sdk检查

复制以下内容到剪贴板，

[wetest\_miniapp\_test]::{"testid":123,"sessionid":"wetest\_sessionid=sessionid;","deviceid":321}

微信开发者工具中Network栏，是否有miniappFPS的请求记录，如果有说明JS-SDK工作正常。如果没有请求记录，可联系wetest侧同学。





在Network中发送的请求发包含fps\_data数据，说明小游戏的fps上报成功。