

互动白板 快速入门



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

快速入门

跑通 Demo

Android

iOS

Mac

Windows

Web

小程序

集成 SDK

Android

iOS

Mac

Windows

Web

小程序

快速入门

跑通 Demo

Android

最近更新时间：2023-07-20 10:45:01

跑通 Demo

本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

- Android Studio 2.0+
- Android 4.2 (SDK API 17) 及以上系统

编译运行

使用 Android Studio (2.0 以上的版本) 打开源码工程，单击运行即可。

账号配置

Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

1. 创建白板应用

步骤一：登录 [互动白板控制台](#)，在左侧导航栏单击[应用管理](#)，进入互动白板应用列表。

步骤二：单击[创建应用](#)，进入创建新应用对话框。

步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 [存储桶配置](#)，回调配置请参考 [回调配置](#)，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在[应用配置](#)里随时配置资源存储桶打开该文档转码或实时录制功能。

创建新应用

应用名称 * 新建 从IM应用导入

请输入应用名称

应用名称只能包含数字、中英文字符和下划线，长度不能超过16个字符

标签 ×

[+ 添加](#)

文档转码配置

文档转码目标存储桶 *

请确保目标存储桶具备“公有读”权限，建议配置“公有读私有写”权限，[前往配置](#)

目标文件前缀

资源域名

文档转码回调地址

回调签名密钥

实时录制配置

实时录制目标存储桶 *

请确保目标存储桶具备“公有读”权限，建议配置“公有读私有写”权限，[前往配置](#)

目标文件前缀

资源域名

实时录制回调地址

回调签名密钥

2. 配置测试账号

步骤一：登录 [即时通信 IM 控制台](#)，单击 SDKAppID 对应的应用。

步骤二：单击 [辅助工具 > UserSig生成&校验](#)，输入测试用户名，单击 [生成签名 \(UserSig\)](#)。

即时通信 IM

- 基本配置
- 功能配置
- 帐号管理
- 群组管理
- 回调配置
- 内容审核
- 插件市场
- 监控仪表盘
- 辅助工具
- UserSig生成&校验
- 离线推送自查
- 自助排障日志
- 客服插件

← UserSig生成&校验
1222
当前站点: 中国 [IM 技术服务交流群](#)

签名 (UserSig) 生成工具 [登录鉴权介绍](#)

此工具可以快速生成签名 (UserSig), 用于本地跑通 Demo 以及功能调试。

用户名 (UserID)

密钥

20

如有需要, 请到应用基础信息中复制密钥

生成签名 (UserSig)

当前生成签名 (UserSig)

```
EgoLMFCU
rQxMC
TxUzhQ_
```

步骤三: 打开 Demo 的配置文件 config.json, 替换新的 SDKAppID 以及新创建的 UserId 和 UserSig (UserToken) 即可。

The screenshot shows an IDE window with the following content:

```

1  "sdkappid":1400127140,
2  "users": [{
3    "userId": "Android_trtc_01",
4    "userToken": "eJxlj01Pg0AURff8CsJWI2*YAsWkixZao9VUoKW2mkyA0xMYZx0P4jxv0uxiSS+zV2ck3vzvRd143
5  }, {
6    "userId": "Android_trtc_02",
7    "userToken": "eJxlj8F0g0AURff9CsJWY4aZIomJC1KnxYaiFpDoZoLMgK+tAwyvBmv8dys2kcS3uYtzc-e58SyLDs
8  }, {
9    "userId": "Android_trtc_03",
10   "userToken": "eJxlj810g0AYRfc8BWFtdH6gJCYuCCURRUJ-qNTNhDDTYWyBcToUqPHrdhEEr-NXZyTe-N9GqZpMut
11  }, {
12   "userId": "Android_trtc_04",
13   "userToken": "eJxlj11vgjAYhe-5FYRbL60tVJrd6aYRLRkVh0LNw2jB*gGLdE637L-PMZ0R7L05F8+Tc-J*WrZt0zFc
14  },
15  }, {
16   "userId": "Android_trtc_05",
17   "userToken": "eJxlj01PhDAQhu-8CtKrRkqhoCYeNqsuEEEJErKnBtrCNhs*LK07avzvIm4iix0Zw-PM0z0fhmma6Pk
18  }
19  ],
20  "roles": [{"name": "user", "value": "640x368"}]}
21
    
```

iOS

最近更新时间：2023-07-20 10:37:21

跑通 Demo

本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

- Xcode 9.0+
- OS X 10.10+ 的 Mac 真机
- 项目已配置有效的开发者签名

编译运行

1. 在终端窗口进入 Demo Podfile 文件所在目录。
2. 执行 `pod install` 命令安装 TEduBoard SDK、TIMSDK、TRTC SDK，或执行 `pod update` 命令，更新本地库版本。
3. 使用 XCode（9.0及以上的版本）打开源码目录下的 `.xcworkspace` 工程，编译并运行即可。

账号配置

Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

1. 创建白板应用

步骤一：登录 [互动白板控制台](#)，在左侧导航栏单击**应用管理**，进入互动白板应用列表。

步骤二：单击**创建应用**，进入创建新应用对话框。

步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 [存储桶配置](#)，回调配置请参考 [回调配置](#)，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在**应用配置**里随时配置资源存储桶打开该文档转码或实时录制功能。



2. 配置测试账号

步骤一：登录 [即时通信 IM 控制台](#)，单击 SDKAppID 对应的应用。

步骤二：单击 [辅助工具 > UserSig 生成&校验](#)，输入测试用户名，单击 [生成签名 \(UserSig\)](#)。

即时通信 IM

- 基本配置
- 功能配置
- 帐号管理
- 群组管理
- 回调配置
- 内容审核
- 插件市场
- 监控仪表盘
- 辅助工具
- 离线推送自查
- UserSig生成&校验
- 自助排障日志
- 客服插件

← UserSig生成&校验
1222
当前站点: 中国 [IM 技术服务交流群](#)

签名 (UserSig) 生成工具 [登录鉴权介绍](#)

此工具可以快速生成签名 (UserSig), 用于本地跑通 Demo 以及功能调试。

用户名 (UserID)

密钥

如有需要, 请到应用基础信息中复制密钥

生成签名 (UserSig)

当前生成签名 (UserSig)

EgoLMFCU
rQxMC
TxUzhQ__

步骤三: 打开 Demo Config 目录下的配置文件 config.json, 替换新的 SDKAppID 以及新创建的 UserID 和 UserSig (UserToken) 即可。

```

1  {"sdkappid": "1400127140", "users":
   [
     {"userId": "tic_ios_01",
      "userToken": "eJwtjNEKgjAYRt91t4X829SZ0EUgYaghJHQp4Vb8hHPqECT694Z59fGdA*dDqvziTwogMW
      Eek03yUSpt8Y4LttjU2I010NW08nkzBiWJQ9AmXDzN2o2DCjHgyBgACu12C5M0Cja8ZCVfXy4*HXWade-S9
      Q8SXQ7nossn-CVFrrP7DE8HSpemg0ykfh78v0BVbsz0A__"},
     {"userId": "tic_ios_02",
      "userToken": "eJyrVgrxCdYrSy1Ss1Iy0jNQ0gHzM1NS80oy0zLBwiwZyFGZ*cXxBkZQ2eKU7MSCgswUJS
      tDEwMDQyNzIAWRsa0oyCxBYqbmppoaGRhARUysyc8Fi5oYWFpbGfQZQUzLTgYYHG-r6*2a7GBVqexf5eUV1lw
      Zm*hSVGnrF6GdnmOd1WBSWenjm02m7RoWHGtgq1QIAZf0zAQ__"},
     {"userId": "tic_ios_03",
      "userToken": "eJyrVgrxCdYrSy1Ss1Iy0jNQ0gHzM1NS80oy0zLBwiwZyFGZ*cXxBkZQ2eKU7MSCgswUJS
      tDEwMDQyNzIAWRsa0oyCxBYqbmppoaGRhARUysyc8Fi5oYWFpbGfQZQUzLTgYbnWhh7m-v6GRmXFHp60DrH6D
      smep1llpUGVVUk0hcVBTim1wr4R3klhwcVeNoq1QIAXfzRg__"},
     {"userId": "tic_ios_04",
      "userToken": "eJyrVgrxCdYrSy1Ss1Iy0jNQ0gHzM1NS80oy0zLBwiwZyFGZ*cXxBkZQ2eKU7MSCgswUJS
      tDEwMDQyNzIAWRsa0oyCxBYqbmppoaGRhARUysyc8Fi5oYWFpYmZ1DR4sx0oEuBb1BZpnMf05BvkB7ikFmY
      lmSa1FvuaugS1*YRF1EYyW4ZGeJaH5xs616bZkQB-kDPZ"},
     {"userId": "tic_ios_05",
      "userToken": "eJyrVgrxCdYrSy1Ss1Iy0jNQ0gHzM1NS80oy0zLBwiwZyFGZ*cXxBkZQ2eKU7MSCgswUJS
      tDEwMDQyNzIAWRsa0oyCxBYqbmppoaGRhARUysyc8Fi5oYWFpYmRqZQUzLTgYa7Gd1W1ZtYJoe1aWeWm1VkrE
      anG1aUh0Z6FxcY*3p5mpgk55gUZfk717t52irVAgA-1jMu"}]]
2
    
```


Mac

最近更新时间：2023-07-20 10:58:51

跑通 Demo

本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

- Xcode 9.0+
- OS X 10.10+ 的 Mac 真机
- 项目已配置有效的开发者签名

编译运行

1. 在终端窗口进入 Demo Podfile 文件所在目录。
2. 执行 `pod install` 命令安装 TEduBoard SDK、TIMSDK、TRTC SDK，或执行 `pod update` 命令，更新本地库版本。
3. 使用 XCode（9.0及以上的版本）打开源码目录下的 `.xcworkspace` 工程，编译并运行即可。

账号配置

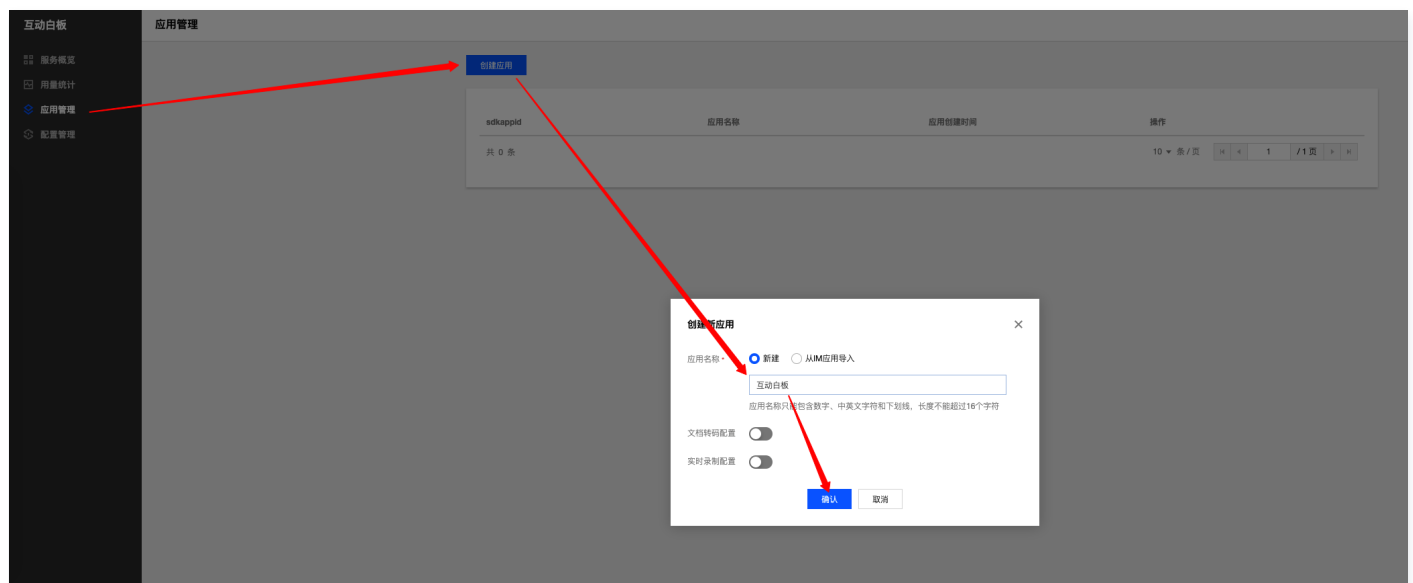
Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

1. 创建白板应用

步骤一：登录 [互动白板控制台](#)，在左侧导航栏单击应用管理，进入互动白板应用列表。

步骤二：单击创建应用，进入创建新应用对话框。

步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 [存储桶配置](#)，回调配置请参考 [回调配置](#)，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在 [应用配置](#) 里随时配置资源存储桶打开该文档转码或实时录制功能。



2. 配置测试账号

步骤一：登录 [即时通信 IM 控制台](#)，单击 SDKAppID 对应的应用。

步骤二：单击 [辅助工具](#) > [UserSig生成&校验](#)，输入测试用户名，单击 [生成签名 \(UserSig\)](#)。

即时通信 IM

- 基本配置
- 功能配置
- 帐号管理
- 群组管理
- 回调配置
- 内容审核
- 插件市场
- 监控仪表盘
- 辅助工具
- UserSig生成&校验
- 离线推送自查
- 自助排障日志
- 客服插件

← UserSig生成&校验
1222
当前站点: 中国 [IM 技术服务交流群](#)

签名 (UserSig) 生成工具 [登录鉴权介绍](#)

此工具可以快速生成签名 (UserSig)，用于本地跑通 Demo 以及功能调试。

用户名 (UserID)

密码

如有需要，请到应用基础信息中复制密码

生成签名 (UserSig)

当前生成签名 (UserSig)

rQxMC
EgoLMFCU

TxUzhQ__

步骤三：打开 Demo Config 目录下的配置文件 config.json，替换新的 SDKAppID 以及新创建的 UserID 和 UserSig (UserToken) 即可。

The screenshot shows a code editor with the following JSON content in config.json:

```

1  {"sdkappid": "1400127140", "users":
   [
   {"userid": "tic_mac_01",
   "usertoken": "eJwtjMEKgAURf91toa8Zw5jQgsXUKIuoiBbie1UrzKmmUEmon9PjleXew6cD9tvduEgNUTZFAKb*U*dfFo6k8eW2rpv2hpsqqa7N0pRx1KMATAS4-yNdIq0HDnnPAKYqKXeM4FJsuBxPFXoMsaz41TdyqKthNSJ3L6H1Xq0h0AAcNv8KFeaImeX0e6aL9n3ByHzMrG"},
   {"userid": "tic_mac_02",
   "usertoken": "eJyrVgrxCdYrSy1Ss1Iy0jNQ0gHzM1NS80oy0zLBwiWZyfG5icnx8kZQ2eKU7MSCGswJStDEwMDQyNzIAWRSa0oyCxBYqbmppoaGRhARUysc8F15oYWFpamBqZQzLTQXammgda5Bg70yZaeBR51aQ5Z*e5WPpqm5jH6KcXB-phBGaZpLibRWG1JR42irVAgBBmTKB"},
   {"userid": "tic_mac_03",
   "usertoken": "eJwtjF0LgjAYRv-LrkPeaFmLupKKIKhY2aIW-Ky1Kntkui-J8urh*cc0B9yPXJn1B2JiesAWdmPQtYGH2ixwSKr8iIdb7G9ULnWKEhM1wDUDeb5G-nW2MmZM8ZcgIUarCwLaBhGzKdLBCs5PpTe6X1pBq1xw1-KX9M9FTvo3s7qW3aG0gPt-6cqqHfk08PukU0fg_"},
   {"userid": "tic_mac_04",
   "usertoken": "eJwtjNEKgjAYRt9112H-1oYmdKFRUWYQE0IIRoasP9HGhFE794wrz6*c*B8SHGUWtbEhMWAJmNHxs90LzhiB2qqq9VBXyyz6arjcGGxJQDUBb6*Rv9Nmi150IIBjBRh-3IQhpFSxGyqYKtj6st3RS7Ndn9*u8dA*614c6*kycp3I4Yb5WzF3STMhWwYp8f0fPMLM_"},
   {"userid": "tic_mac_05",
   "usertoken": "eJwtjF0LgjAYRv-LrsP24XIKXeIFhEWSIEEgS1e8xGxOCTX67w3z6ue5B84H5enZeyuLIkQ9jFbz1o1Pdxhxj1UpZZVifliu-opjYeArcTHmNDAzd*owYBVjnPOKcYL7UHPLCBChJxtlgo8Xly4rieZ2ckWbJen84uZos5Do0E10nTRjwKrhdmgyY97uhb9F3x9vsTOy"}]}
2

```

Windows

最近更新时间：2024-02-19 17:22:51

跑通 Demo

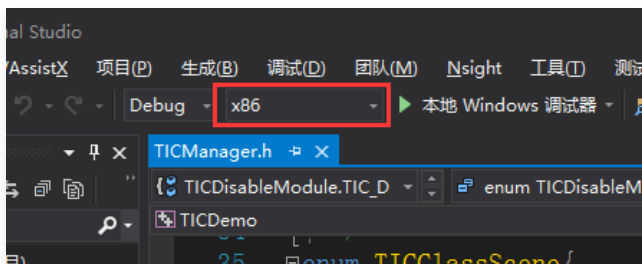
本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

- Microsoft Visual Studio 2015及以上版本，推荐使用 Microsoft Visual Studio 2015。
- Windows SDK 8.0及以上版本，推荐使用 Windows SDK 8.1。

编译运行

1. 进入 Demo 目录下,使用 Visual Studio 打开解决方案文件 TICDemo.sln 。
2. 在 Visual Studio 中选择“x86”配置(如下图)，然后编译运行即可。



账号配置

Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

1. 创建白板应用

步骤一：登录 [互动白板控制台](#)，在左侧导航栏单击应用管理，进入互动白板应用列表。

步骤二：单击创建应用，进入创建新应用对话框。

步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 [存储桶配置](#)，回调配置请参考 [回调配置](#)，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在[应用配置](#)里随时配置资源存储桶打开该文档转码或实时录制功能。

创建新应用 ✕

应用名称 ● 新建 ○ 从IM应用导入

请输入应用名称

应用名称只能包含数字、中英文字符和下划线，长度不能超过16个字符

标签 标签键 标签值 ✕

[+ 添加](#) [📌 键值粘贴板](#)

文档转码配置 🔵

文档转码目标存储桶 请选择

请确保目标存储桶具备“公有读”权限，建议配置“公有读私有写”权限，[前往配置](#)

目标文件前缀 请输入目标文件前缀

资源域名 请选择资源域名

文档转码回调地址 请输入文档转码回调地址

回调签名密钥 请输入回调签名密钥

实时录制配置 🔵

实时录制目标存储桶 请选择

请确保目标存储桶具备“公有读”权限，建议配置“公有读私有写”权限，[前往配置](#)

目标文件前缀 请输入目标文件前缀

资源域名 请选择资源域名

实时录制回调地址 请输入实时录制回调地址

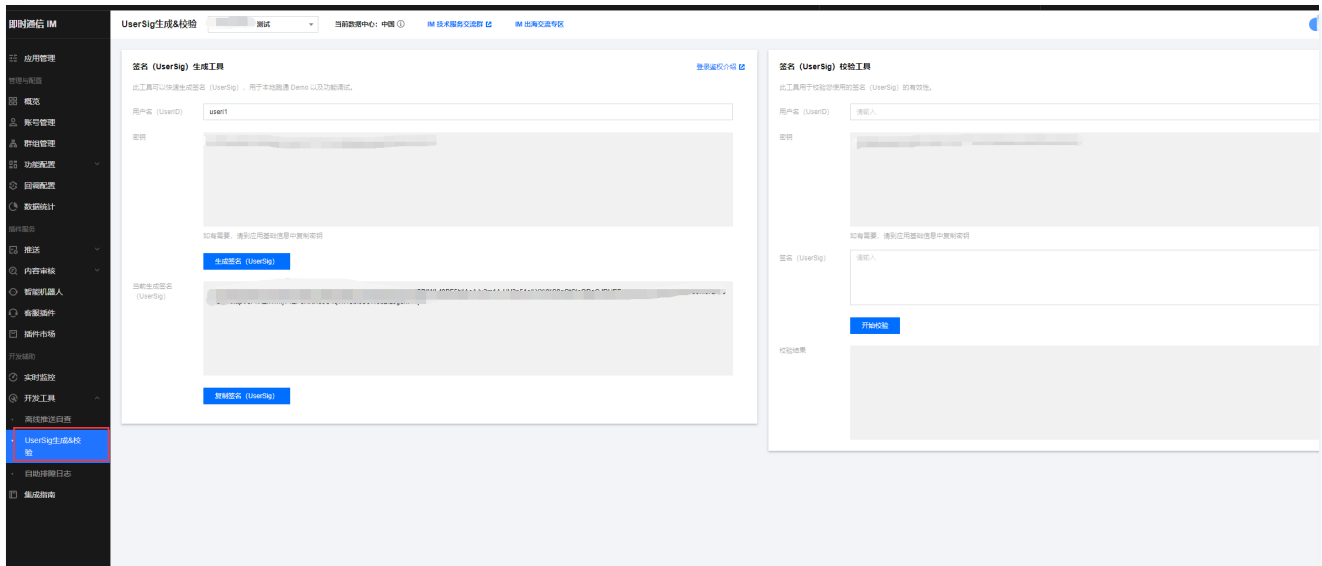
回调签名密钥 请输入回调签名密钥

[确认](#) [取消](#)

2. 配置测试账号

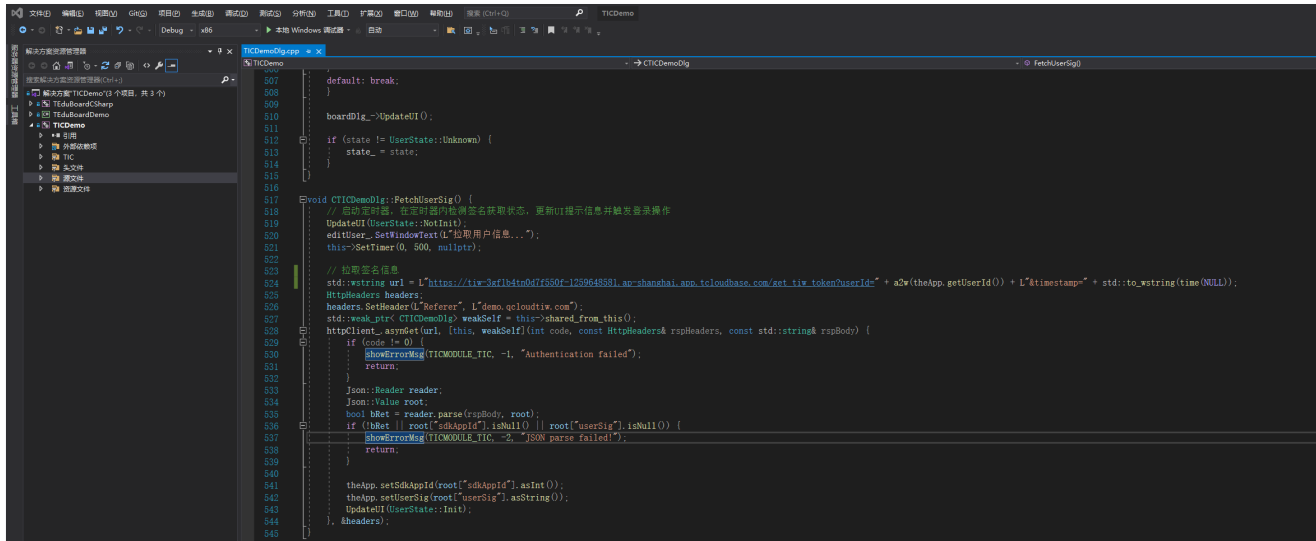
步骤一：登录 [即时通信 IM 控制台](#)，单击 SDKAppID 对应的应用。

步骤二：单击 [开发工具 > UserSig 生成&校验](#)，输入测试用户名，单击生成签名 (UserSig)。



步骤三：打开 Demo 项目下的，直接在代码里面设置申请的 SDKAppID 以及新创建的 UserID 和 UserSig (UserToken) 即可；

第四步：另外，客户可以直接使用互动白板 demo 里面的测试账号，但是 demo 通过云函数获取的 UserSig (UserToken)，每十分钟会使用过期，需要重新登录。



```
507         default: break;
508     }
509 }
510 boardDlg->UpdateUI();
511
512 if (state != UserState::Unknown) {
513     state_ = state;
514 }
515 }
516
517 void CTICDemoDlg::FetchUserSig() {
518     // 获取用户Sig：通过接口获取登录状态，更新UI提示信息并触发登录操作
519     UpdateUI(UserState::NotInit);
520     editUser_->SetWindowText(L"获取用户信息...");
521     this->SetTimer(0, 500, nullptr);
522
523     // 获取签名信息
524     std::wstring url = L"https://tiw-3af1b41n0474350f-1299648881.ap-shanghai.app-logs.tencentcloud.com/get_tiw_token?userId=" + a2w(theApp.getUserId()) + L"&timestamp=" + std::to_wstring(time(NULL));
525     HttpHeaders headers;
526     headers.SetHeader(L"Referer", L"demo.qqcloudtiw.com");
527     std::weak_ptr<CTICDemoDlg> weakSelf = this->shared_from_this();
528     httpClient_.asyncGet(url, [this, weakSelf](int code, const HttpHeaders& rspHeaders, const std::string& rspBody) {
529         if (code != 0) {
530             ShowErrorMessage(TICMODULE_TIC, -1, "Authentication failed");
531             return;
532         }
533         Json::Reader reader;
534         Json::Value root;
535         bool bRet = reader.parse(rspBody, root);
536         if (!bRet || root["sdkAppId"].isNull() || root["userSig"].isNull()) {
537             ShowErrorMessage(TICMODULE_TIC, -2, "JSON parse failed!");
538             return;
539         }
540
541         theApp.setSdkAppId(root["sdkAppId"].asInt());
542         theApp.setUserSig(root["userSig"].asString());
543         UpdateUI(UserState::Init);
544     }, &headers);
545 }
```

Web

最近更新时间：2023-07-20 15:26:51

跑通 Demo

本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

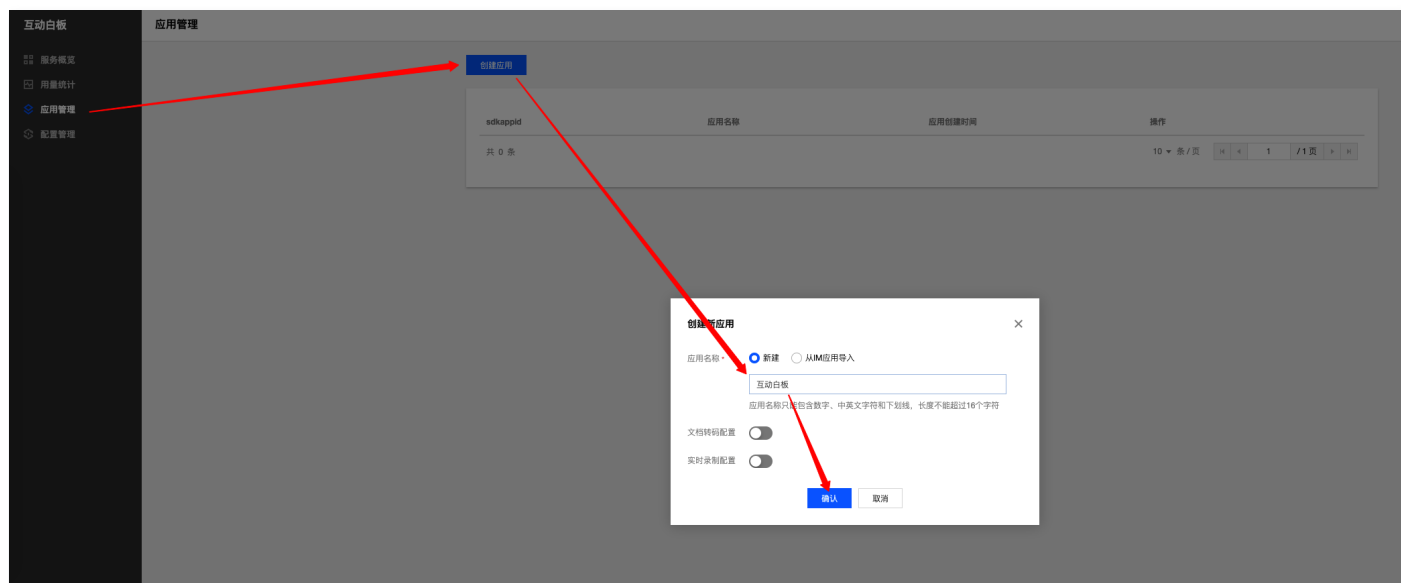
账号配置

1. 创建白板应用

步骤一：登录 [互动白板控制台](#)，在左侧导航栏单击应用管理，进入互动白板应用列表。

步骤二：单击创建应用，进入创建新应用对话框。

步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 [存储桶配置](#)，回调配置请参考 [回调配置](#)，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在[应用配置](#)里随时配置资源存储桶打开该文档转码或实时录制功能。



2. 配置测试账号



该方式生成 userSig 的方式只建议在开发阶段使用，生产环境建议使用服务端生成 userSig。

步骤一：登录 [即时通信 IM 控制台](#)，单击 SDKAppID 对应的应用。

步骤二：在基本配置 > 应用资料中获取 SDKAppID 和基本配置 > 基础信息中获取密钥。（注意保密您的密钥）



步骤三：打开 `src/config/index.js` 文件，将该文件的 `sdkAppId` 和 `secretKey` 替换为您自己的信息即可。

```
JS index.js M X
src > config > JS index.js > ...
You, a minute ago | 2 authors (zhutaohuang and others)
1  export default {
2    // 白板应用SdkAppId
3    sdkAppId: 0,
4    // 白板应用对应的密钥 https://cloud.tencent.com/document/product/1137/39907
5    secretKey: '',
6    // 白板应用签名过期时间（单位：秒）
7    expireTime: 84600,
8    // cos存储桶配置
9    cosBucket: '',
10   cosRegion: '',
11   // 腾讯云账号API密钥 (https://console.cloud.tencent.com/cam/capi)
12   tencentSecretId: '',
13   tencentSecretKey: '',
14   // 发起转码请求服务端地址, 参考 https://cloud.tencent.com/document/product/1137/40060
15   createTranscodeUrl: '',
16   // 查询转码进度服务端地址, 参考 https://cloud.tencent.com/document/product/1137/40059
17   describeTranscodeUrl: '',
18 };
19
```

如果您需要体验完整功能，请根据 Demo 源码配置文件的提示进行配置。

运行

项目安装

```
npm install
```

项目运行

```
npm run serve
```

小程序

最近更新时间：2023-12-21 17:52:11

跑通 Demo

本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

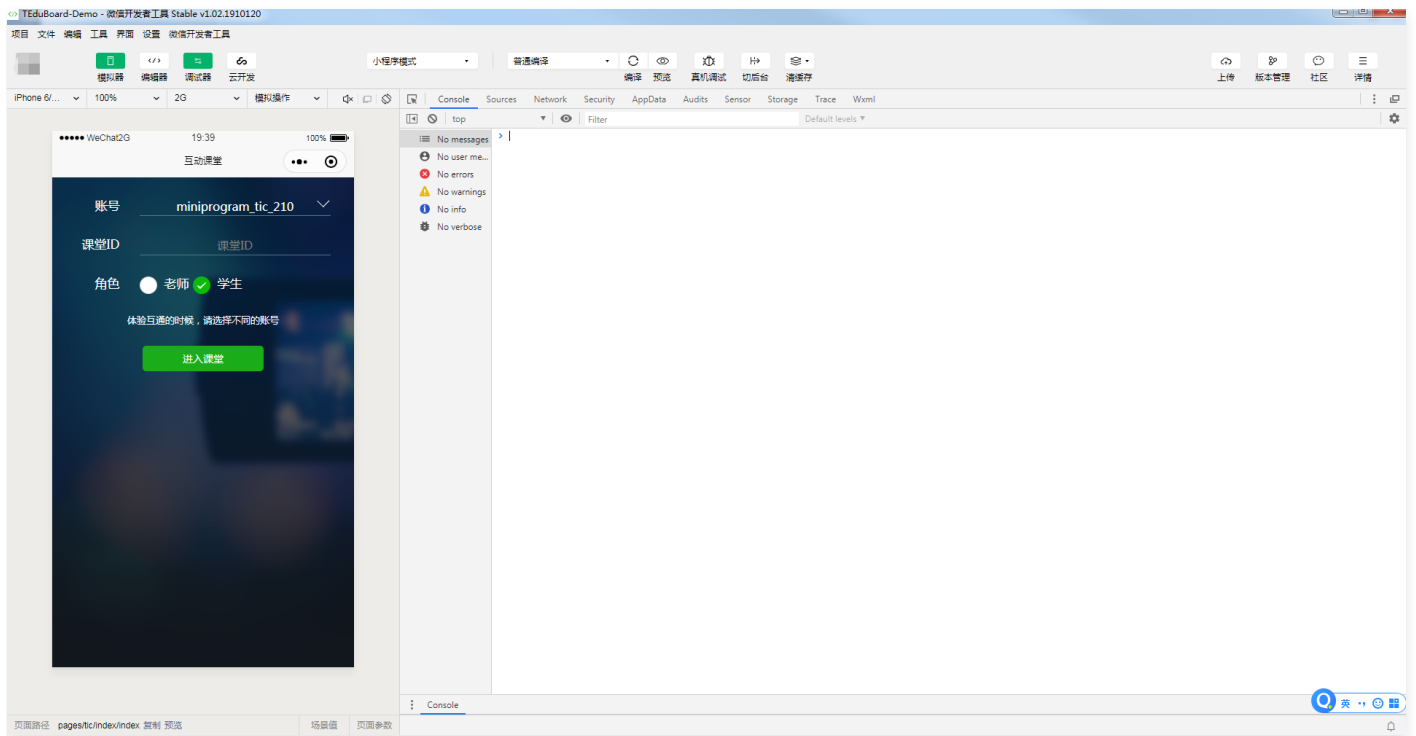
- 微信版本6.6.6+
- 微信开发者工具最新版本

运行

1. 导入 Demo 项目



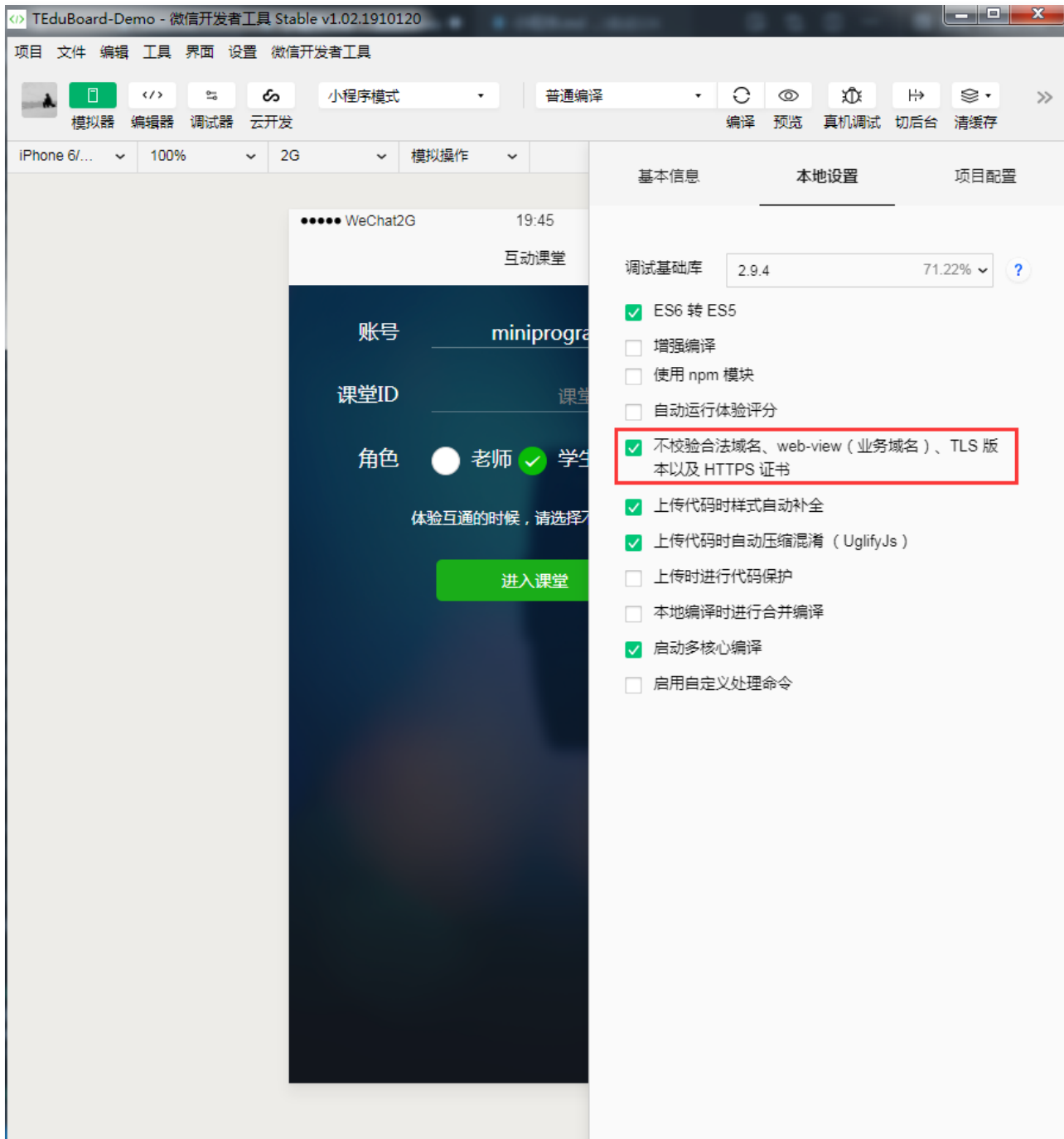
2. 编译运行



3. 配置业务域名

注意

1. 配置业务域名，同 [集成 SDK 文档的配置业务域名](#)。
2. 可以在调试工具中，打开 [开发者工具 > 详情 > 不校验合法域名](#)。



注意:

在真机打开开发调试，也可以不校验合法域名。



账号配置

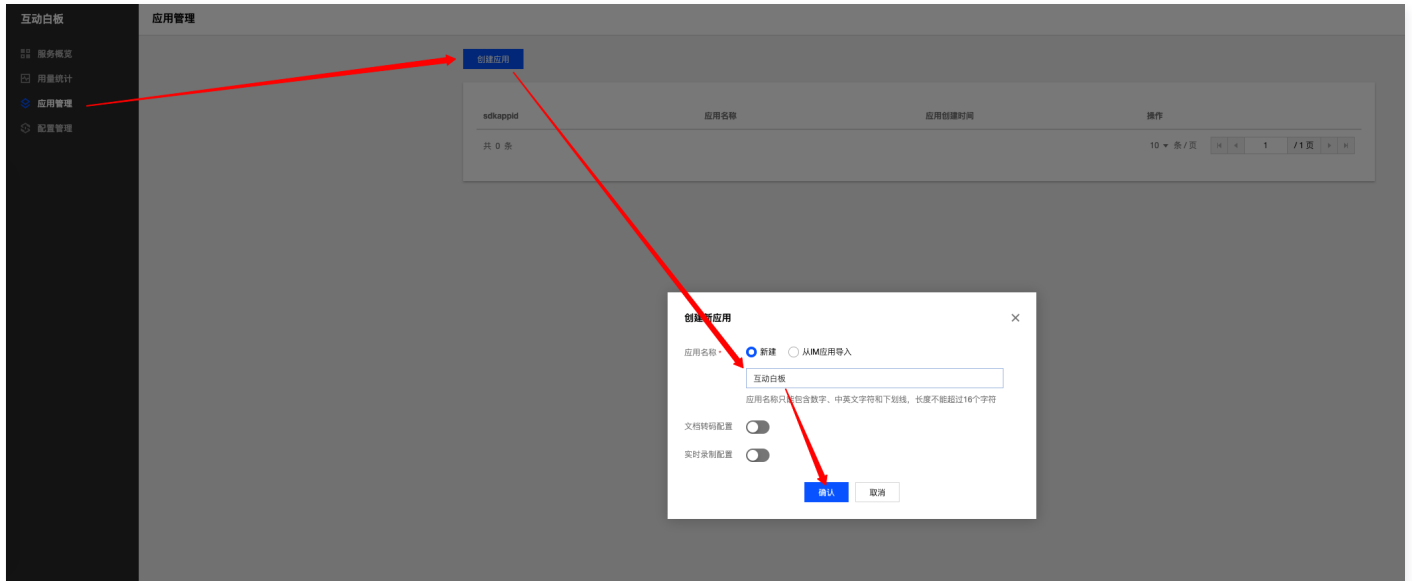
Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

1. 创建白板应用

步骤一：登录 [互动白板控制台](#)，在左侧导航栏单击[应用管理](#)，进入互动白板应用列表。

步骤二：单击[创建应用](#)，进入创建新应用对话框。

步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 [存储桶配置](#)，回调配置请参考 [回调配置](#)，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在[应用配置](#)里随时配置资源存储桶打开该文档转码或实时录制功能。



2. 开通音视频服务

Demo 集成了实时音视频 TRTC SDK，为保证音视频功能的正常使用，请开通音视频服务。

步骤一：登录 [即时通信 IM 控制台](#)，单击目标应用卡片，进入应用的基础配置页面。

步骤二：单击 [开通腾讯实时音视频服务](#) 区域的 **立即开通**。

步骤三：在弹出的开通实时音视频 TRTC 服务对话框中，单击 **确定**。

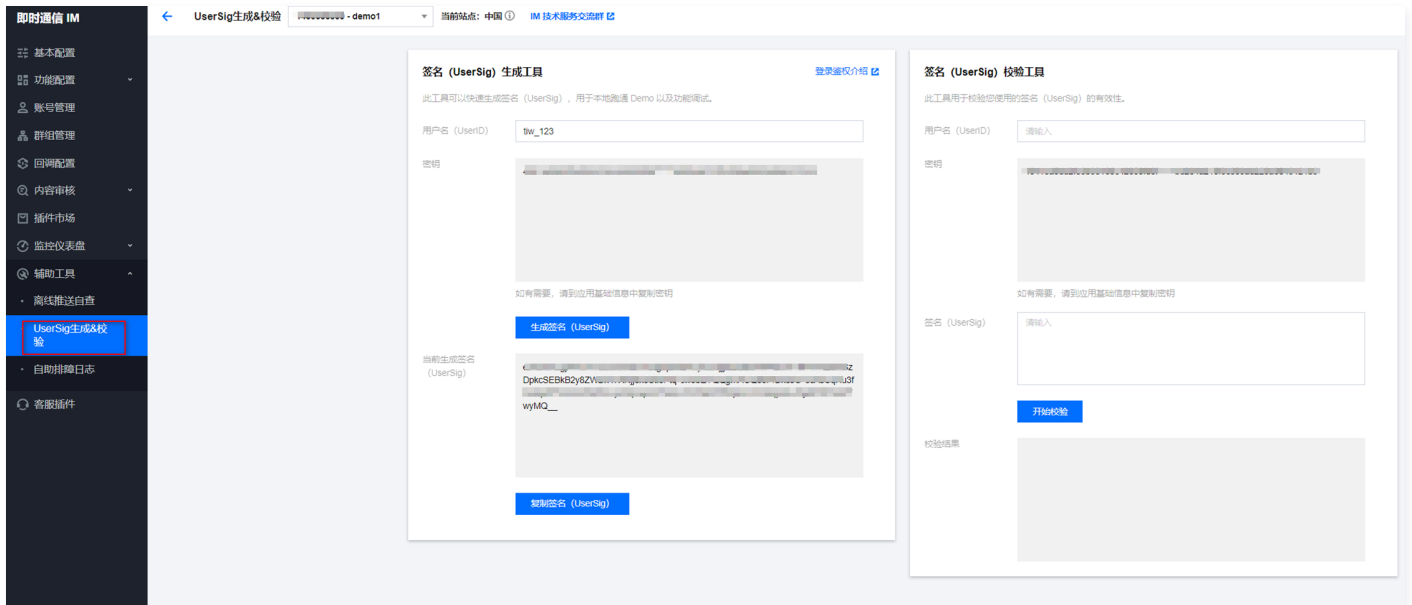
系统将为您在 [实时音视频控制台](#) 创建一个与相同 SDKAppID 的实时音视频应用，互动白板，IM，实时音视频三者的账号与鉴权可复用。



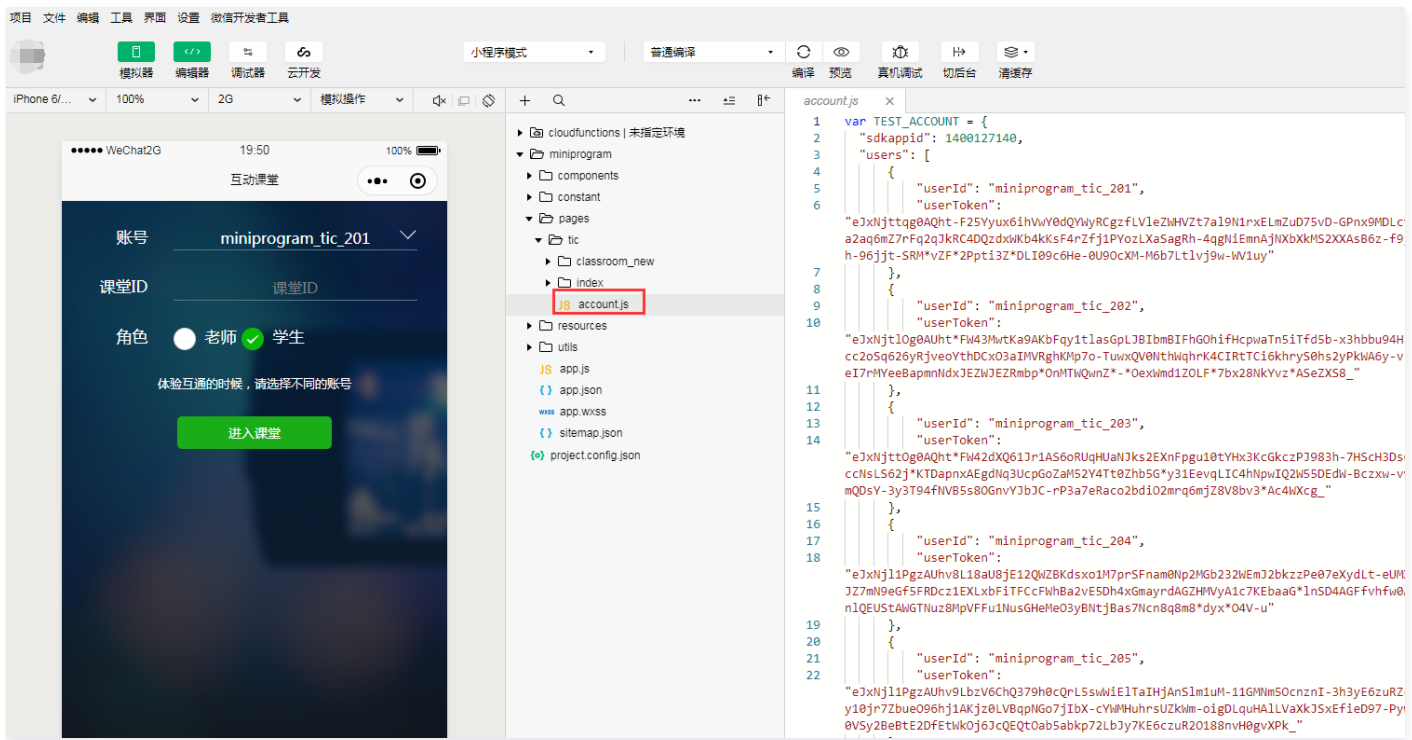
3. 配置测试账号

步骤一：登录 [即时通信 IM 控制台](#)，单击 SDKAppID 对应的应用。

步骤二：单击 [辅助工具](#) > [UserSig生成&校验](#)，输入用户名（UserID），单击生成签名（UserSig）。



步骤三：打开 Demo/miniprogram/pages/tic 目录下的配置文件 account.js，替换新的 SDKAppId 以及新创建的 UserId 和 UserSig (UserToken) 即可。



集成 SDK Android

最近更新時間：2023-07-18 14:55:22

本文主要介紹如何快速的將騰訊云 TEduBoard SDK 集成到您的項目中。如果您使用互動課堂方案，請前往 [互動課堂集成](#)。

開發環境

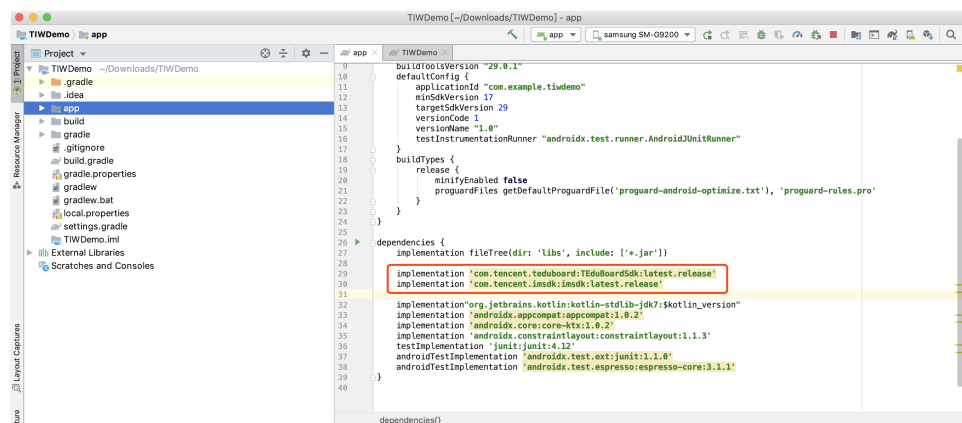
- Android Studio 2.0+
- Android 4.2 (SDK API 17) 及以上系統

集成 TEduBoard SDK

您可以選擇使用 Gradle 自動加載的方式，或者手動下載 aar 再將其導入到您當前的工程項目中。由於 TEduBoard SDK 內部使用 TIMSDK 作為內部信令通道，您還需自動或手動添加 TIMSDK 依賴項。

自動加載 (aar)

TEduBoard SDK 和 TIMSDK 已經發布到 Maven Central 庫，您可以通過配置 gradle 自動下載更新。



1. 添加 SDK 依賴

```
dependencies {  
    implementation 'com.tencent.edu:TEduBoardSdk:latest.release'  
    implementation 'com.tencent.imsdk:imsdk:latest.release'  
}
```

2. 同步 SDK

單擊 Sync Now，如果您的網絡連接 Maven Central 沒有問題，SDK 會自動下載集成到工程里。

手動下載 (aar)

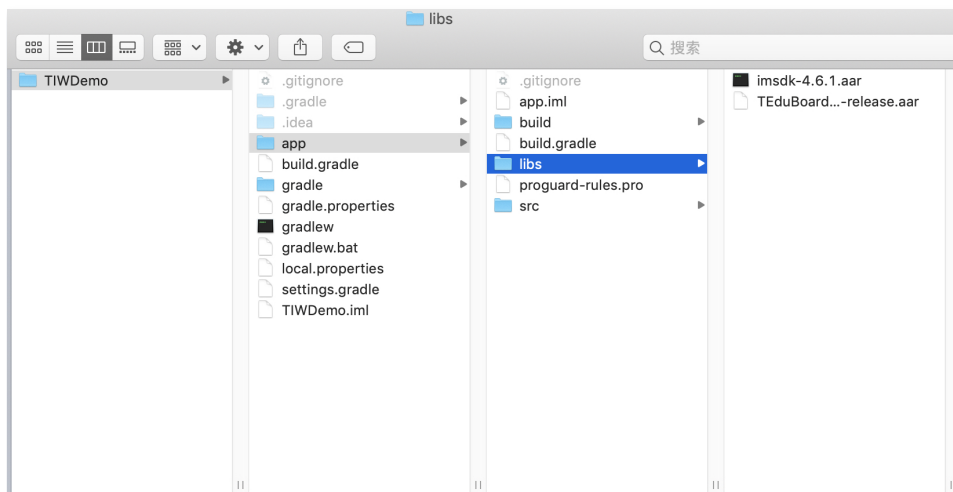
如果您的網絡連接 Maven Central 有問題，也可以手動下載 SDK 集成到工程里。

1. 下載 SDK

單擊下載最新版 [TEduBaord SDK](#)。前往 [即時通信 IM](#) 下載 TIMSDK。

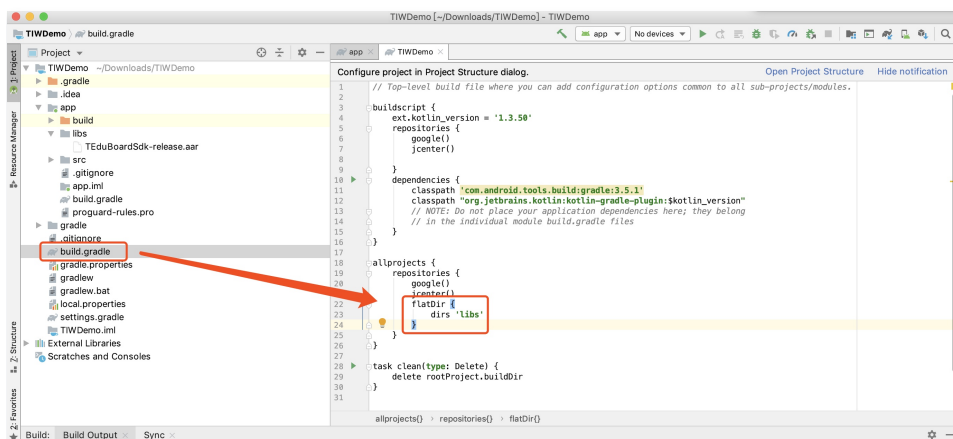
2. 導入 SDK

将下载到的 aar 文件拷贝到工程的 app/libs 目录下。



3. 指定本地仓库路径

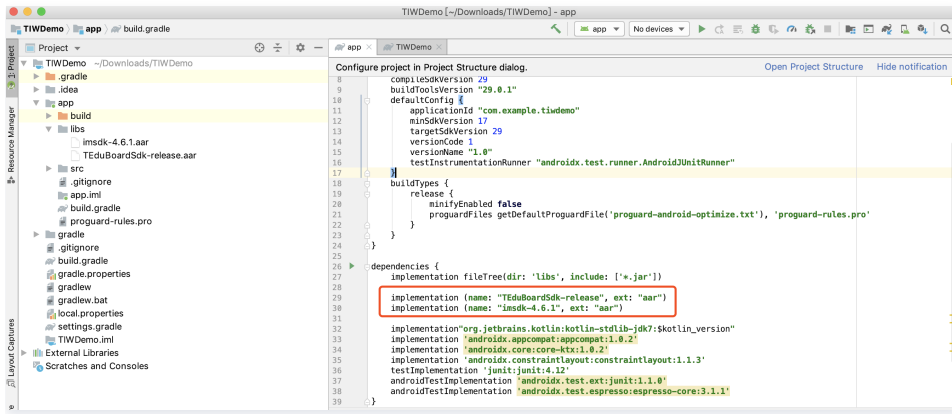
在工程根目录下的 build.gradle 中，添加 flatDir，指定本地仓库路径。



4. 添加 SDK 依赖

在 app/build.gradle 中，添加引用 aar 包的代码。

```
dependencies {
    implementation (name: "TEduBoardSdk-release", ext: "aar")
    implementation (name: "imsdk-plus-6.6.3002", ext: "aar") // IM的版本请尽量用新版本，具体请查阅IM的更新日志
    implementation 'com.tencent.edu:TIWLogger:1.0.1.76'
    implementation 'com.tencent.edu:TIWCache:1.0.0.91'
}
```



5. 同步 SDK

单击 Sync Now，完成 TEduBoard SDK 集成。

配置 App 权限

在 AndroidManifest.xml 中配置 App 的权限，TEduBoard SDK 需要以下权限：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

使用 TEduBoard SDK

1. 创建白板控制器

使用如下代码创建并初始化白板控制器：

```
// 创建并初始化白板控制器
// (1) 鉴权配置
TEduBoardController.TEduBoardAuthParam authParam = new TEduBoardController.TEduBoardAuthParam(sdkAppId, userId,
userSig);

// (2) 白板默认配置
TEduBoardController.TEduBoardInitParam initParam = new TEduBoardController.TEduBoardInitParam();
mBoard = new TEduBoardController(context);

// (3) 添加白板事件回调 实现TEduBoardCallback接口
TEduBoardCallback callback = new TEduBoardController.TEduBoardCallback();
mBoard.addCallback(callback);

// (4) 进行初始化
mBoard.init(authParam, classId, initParam);
```

其中 `sdkAppId`、`userId`、`userSig`、`classId` 为需要您自己填写的参数。

⚠ 注意

- 请在主进程中执行初始化操作，如果您的 App 使用了多进程，请注意避免重复初始化。
- 由于 HarmonyOS 的 `evaluatejavascript` 的内部调用改成异步主线程调用，白板 `get` 接口在主线程等待并同步返回结果将会造成死锁，在沿用 `get` 接口同步返回的使用方式的前提下，SDK 内部已无法规避，请更新到 2.8.2 及以上版本后在外部使用子线程调用白板 `get` 相关接口（如下例所示），后续我们将 `get` 相关接口进行异步优化。

```
new Thread(new Runnable() {
    @Override
    public void run() {
```



```
TEduBoardFileInfo info = board.getFileInfo(fileId);
}
}).start();
```

2. 监听白板关键事件

在白板事件回调接口 `TEduBoardCallback` 的 `onTEBError` 和 `onTEBWarning` 回调方法内监听白板事件

- [onTEBError 错误详情](#)
- [onTEBWarning 警告详情](#)

```
/**
 * 白板错误回调
 * 必须要监听的事件
 *
 * @param code 错误码
 * @param msg 错误信息，编码格式为 UTF8
 */
void onTEBError(int code, String msg);

/**
 * 白板警告回调
 * @param code 警告码
 * @param msg 警告信息，编码格式为 UTF8
 */
void onTEBWarning(int code, String msg);
```

3. 白板窗口获取及显示

在 `onTEBInit` 回调方法内，使用如下代码获取并显示白板视图：

```
// (1) 在 Activity 的布局 xml 文件中，用 FrameLayout 占位，将来放 board 的 View。
<FrameLayout
    android:id="@+id/board_view_container"
    android:layout_width="match_parent"
    android:layout_height="200dp"/>
```

```
// (2) 获取白板 View
View boardview = mBoard.getBoardRenderView();
// (3) 添加到父视图中
FrameLayout.LayoutParams layoutParams = new FrameLayout.LayoutParams(FrameLayout.LayoutParams.MATCH_PARENT,
    FrameLayout.LayoutParams.MATCH_PARENT);
FrameLayout container = findViewById(R.id.board_view_container);
container.addView(boardview, layoutParams);
```

SDK 所有回调都在主线程内执行，因此可以在回调里直接执行 UI 操作。

4. 白板数据同步

白板在使用过程中，需要在不同的用户之间进行数据同步（涂鸦数据等），SDK 默认使用 IMSDK 作为信令通道，您需要自行实现 IMSDK 的初始化、登录、加入群组操作，确保白板初始化时，IMSDK 已处于所指定的群组内。

步骤1：初始化 IMSDK

```
V2TIMSDKConfig timSdkConfig = new V2TIMSDKConfig();
boolean result = V2TIMManager.getInstance().initSDK(context, sdkAppID, timSdkConfig);
```

如果您有其他业务使用了 IMSDK 并希望 IMSDK 的生命周期与 App 的生命周期保持一致，请在 `Application` 的 `onCreate` 方法中初始化 IMSDK，否则请在登录前初始化 IMSDK，在登录后反初始化 IMSDK。

步骤2：登录 IMSDK

```

TIMGroupManager.getInstance().login(userId, userSig, new TIMCallback() {
    @Override
    public void onSuccess(String s) {

    }

    @Override
    public void onError(int errCode, String errMsg) {
        // 创建 IM 群组失败
    }
});

V2TIMManager.getInstance().login(userId, userSig, new V2TIMCallback() {
    @Override
    public void onSuccess() {
        // 登录成功
    }

    @Override
    public void onError(int errCode, String errMsg) {
        // 登录失败
    }
});
    
```

步骤3: 加入群组

登录 IMSDK 成功后加入白板所在的群组。

```

V2TIMManager.getInstance().joinGroup(classId, "board group" + classId, new V2TIMCallback() {
    @Override
    public void onSuccess() {
        // 加群成功
    }

    @Override
    public void onError(int i, String s) {
        // 加群失败
    }
});
    
```

如果 IM 群组不存在, 请先创建群组。

```

V2TIMManager.getGroupManager().createGroup(groupInfo, null, new V2TIMValueCallback<String>() {
    @Override
    public void onError(int errCode, String errMsg) {

    }

    @Override
    public void onSuccess(String s) {

    }
});
    
```

⚠ 注意

1. 推荐业务后台使用 IM REST API 提前创建群组。
2. 不同的群组类型, 群组功能以及成员数量有所区别, 具体请查看 IM 群组系统。

5. 销毁白板

调用 unInit 方法后, 内部将彻底销毁白板并停止计费, 请您确保此接口的调用。

```
mBoard.uninit();
```

如果您使用 IMSDK 作为信令通道，请根据业务的需要决定是否退出群组、退出登录并反初始化。

步骤1：退出群组

```
V2TIMManager.getInstance().quitGroup(classId, new V2TIMCallback() {
    @Override
    public void onSuccess() {

    }

    @Override
    public void onError(int i, String s) {

    }
});
```

步骤2：登出 IMSDK

```
// IM登出
public void logout(final IMCallBack callBack) {
    V2TIMManager.getInstance().logout(new V2TIMCallback(){
        @Override
        public void onSuccess() {

        }

        @Override
        public void onError(int errCode, String errMsg) {

        }
    });
}
```

步骤3：反初始化 IMSDK

```
V2TIMManager.getInstance().unInitSDK();
```

如果您有其他业务使用了 IMSDK 并希望 IMSDK 的生命周期与 App 的生命周期保持一致，无需调用此接口。

iOS

最近更新时间：2024-03-07 17:56:31

集成 SDK

本文主要介绍如何快速的将腾讯云 TEduBoard SDK 集成到您的项目中。如果您使用互动课堂方案，请前往 [互动课堂集成](#)。

开发环境

- Xcode 9.0+
- OS X 10.10+ 的 Mac 真机
- 项目已配置有效的开发者签名

集成 TEduBoard SDK

您可以选择使用 CocoaPods 自动加载的方式，或者先下载 SDK 再将其导入到您当前的工程项目中。由于 TEduBoard SDK 内部使用 IMSDK 作为内部信令通道，您还需 CocoaPods 或手动添加 IMSDK 依赖项。

CocoaPods

1. 安装 CocoaPods

在终端窗口中输入如下命令（需要提前在 Mac 中安装 Ruby 环境）：

```
sudo gem install cocoapods
```

2. 创建 Podfile 文件

进入项目所在路径，输入以下命令之后项目路径下会出现一个 Podfile 文件。

```
pod init
```

3. 编辑 Podfile 文件

编辑 Podfile 文件，支持选择版本号。

```
platform :ios, '8.0'

target 'TICDemo' do
  #pod 'TEduBoard_iOS', :podspec=>'https://sdk.qcloudtiw.com/ios/prod/2.8.9.204/TEduBoard_iOS.podspec'
  pod 'TEduBoard_iOS', '>=2.8.9.204'
  pod 'TXIMSDK_Plus_iOS', '6.2.2363'
  pod 'YYModel'
end
```

互动白板默认使用 IMSDK 作为信令通道，如果您有独立的信令通道，无需集成 IMSDK。

4. 更新并安装 SDK

在终端窗口中输入如下命令以更新本地库文件，并安装 TEduBoard SDK：

```
pod install
```

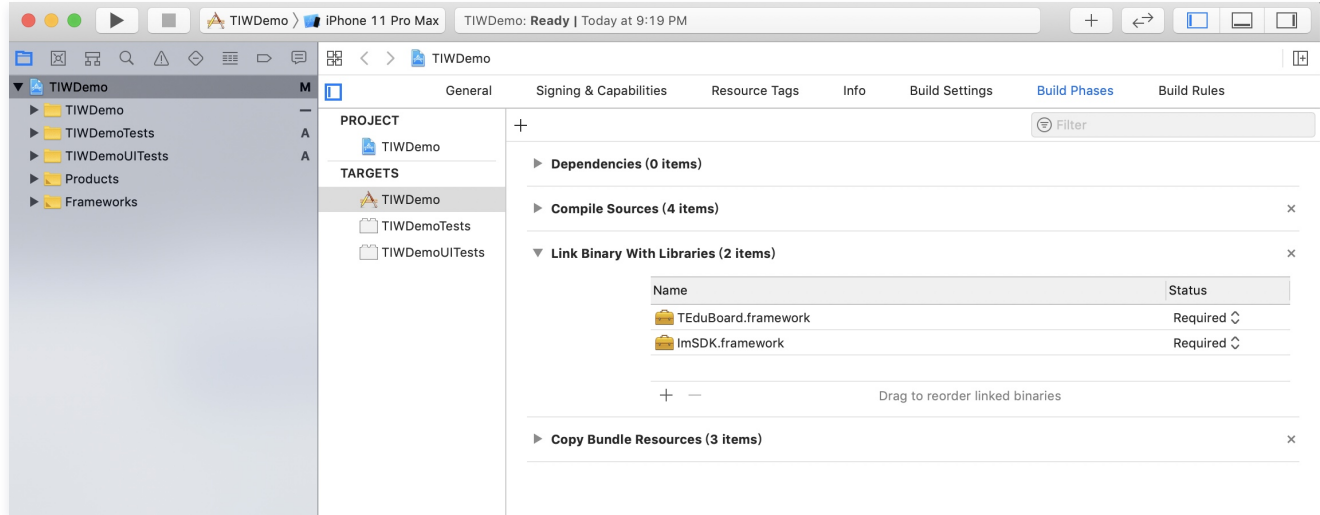
或使用以下命令更新本地库版本：

```
pod update
```

pod 命令执行完后，会生成集成了 SDK 的 .xcworkspace 后缀的工程文件，双击打开即可。

手动集成

1. 前往 [版本信息](#)，下载 SDK。
2. 前往 [即时通信 IM](#) 下载 IMSDK。互动白板默认使用 IMSDK 作为信令通道，如果您有独立的信令通道，请跳过此步。
3. 打开您的 Xcode 工程项目，选择要运行的 target，选中 Build Phases 项。
4. 单击 Link Binary with Libraries 项展开，单击底下的“+”号图标去添加依赖库。



5. Other Linker Flags添加-ObjC



使用 TEduBoard SDK

1. #import SDK

在项目需要使用 SDK API 的文件里，引入具体的头文件。

```
#import <TEduBoard/TEduBoard.h>
```

2. 创建白板控制器

使用如下代码创建并初始化白板控制器：

```
// 创建并初始化白板控制器
// (1) 鉴权配置
TEduBoardAuthParam *authParam = [[TEduBoardAuthParam alloc] init];
authParam.sdkAppId = _sdkAppId;
authParam.userId = _userId;
authParam.userSig = _userSig;
// (2) 白板默认配置
TEduBoardInitParam *initParam = [[TEduBoardInitParam alloc] init];
_boardController = [[TEduBoardController alloc] initWithAuthParam:authParam roomId:_classId initParam:initParam];
// (3) 添加白板事件回调
[_boardController addDelegate:self];
```

其中 _sdkAppId、_userId、_userSig、_classId 为需要您自己填写的参数。

3. 白板窗口获取及显示

在 onTEBInit 或 onTEBHistoryDataSyncCompleted 回调方法内，使用如下代码获取并显示白板视图。

```
-(void)onTEBHistoryDataSyncCompleted
```

```

{
// (1) 获取白板 UIView
UIView *boardView = [_boardController getBoardRenderView];
// (2) 设置显示位置和大小
boardView.frame = CGRectMake(0, 0, width, height);
// (3) 添加到父视图中
[self.view addSubview:boardView];
}
    
```

⚠ 注意

1. onTEBInit 表示白板创建并鉴权完成。
2. onTEBHistoryDataSyncCompleted 表示历史数据加载完成，此时可调用白板的相关接口。
3. SDK 所有回调都在主线程内执行，因此可以在回调里直接执行 UI 操作。

4. 监听白板关键事件

在 onTEBError 和 onTEBWarning 回调方法内监听白板事件

- [onTEBError 错误详情](#)
- [onTEBWarning 警告详情](#)

```

// 监听白板错误事件
- (void)onTEBError:(TEduBoardErrorCode)code msg:(NSString *)msg
{
}

// 监听白板告警事件
- (void)onTEBWarning:(TEduBoardWarningCode)code msg:(NSString *)msg
{
}
    
```

5. 白板数据同步

白板在使用过程中，需要在不同的用户之间进行数据同步（涂鸦数据等），SDK 默认使用 IMSDK 作为信令通道，您需要自行实现 IMSDK 的初始化、登录、加入群组操作，确保白板初始化时，IMSDK 已处于所指定的群组内。

步骤1：初始化 IMSDK

```

V2TIMSDKConfig *config = [[V2TIMSDKConfig alloc] init];
config.logLevel = V2TIM_LOG_WARN;
[[V2TIMManager sharedInstance] initWithSDKAppID:config.sdkAppID config:config];
[[V2TIMManager sharedInstance] addIMSDKListener:listener];
[[V2TIMManager sharedInstance] addAdvancedMsgListener:self];
    
```

ⓘ 说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

如果您有其他业务使用了 IMSDK 并期望 IMSDK 的生命周期与 App 的生命周期保持一致，请在 AppDelegate 的 application:didFinishLaunchingWithOptions 方法中初始化 IMSDK，否则请在登录前初始化 IMSDK，在登出后反初始化 IMSDK。

步骤2：登录 IMSDK

```

[[V2TIMManager sharedInstance] login:userID userSig:userSig succ:^(
// 登录 IMSDK 成功
) fail:^(int code, NSString *desc) {
// 登录 IMSDK 失败
}];
    
```

说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

步骤3：加入群组

登录 IMSDK 成功后加入白板所在的群组。

```
[[V2TIMManager sharedInstance] joinGroup:groupId succ:^(int code, NSString *desc) {
    // 加入 IM 群组成功
} fail:^(int code, NSString *desc) {
    // 加入 IM 群组失败
}];
```

说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

如果 IM 群组不存在，请先创建群组。

```
V2TIMGroupInfo *groupInfo = [[V2TIMGroupInfo alloc] init];
groupInfo.groupID = groupId;
groupInfo.groupName = groupId;
groupInfo.groupType = @"Public";
groupInfo.groupAddOpt = V2TIM_GROUP_ADD_ANY;
__weak typeof(self) ws = self;
[[V2TIMManager sharedInstance] createGroup:groupInfo memberList:nil succ:^(NSString *groupID) {
    // 创建 IM 群组成功
} fail:^(int code, NSString *desc) {
    // 创建 IM 群组失败
}];
```

说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

注意

- 推荐业务后台使用 [IM REST API](#) 提前创建群组。
- 不同的群组类型，群组功能以及成员数量有所区别，具体请查看 [IM 群组系统](#)。

6. 销毁白板

调用 unInit 方法后，内部将彻底销毁白板并停止计费，请您确保此接口的调用。

```
[_boardController unInit];
```

如果您使用 IMSDK 作为信令通道，请根据业务的需要决定是否退出群组、退出登录并反初始化。

步骤1：退出群组

```
[[V2TIMManager sharedInstance] quitGroup:groupId succ:^(int code, NSString *desc) {
    // 登出 IMSDK 成功
} fail:^(int code, NSString *desc) {
    // 登出 IMSDK 失败
}];
```

说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

步骤2：登出 IMSDK

```
[[V2TIMManager sharedInstance] logout:^(  
    // 登出 IMSDK 成功  
} fail:^(int code, NSString *desc) {  
    // 登出 IMSDK 失败  
}];
```

说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

步骤3：反初始化 IMSDK

```
[[V2TIMManager sharedInstance] removeIMSDKListener:self];  
[[V2TIMManager sharedInstance] removeAdvancedMsgListener:self];  
[[V2TIMManager sharedInstance] unInitSDK];
```

说明

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

如果您有其他业务使用了 IMSDK 并期望 IMSDK 的生命周期与 App 的生命周期保持一致，无需调用此接口。

Mac

最近更新时间：2024-03-28 17:40:01

集成 SDK

本文主要介绍如何快速的将腾讯云 TEduBoard SDK 集成到您的项目中。如果您使用互动课堂方案，请前往 [互动课堂集成](#)。

开发环境

- Xcode 9.0+
- OS X 10.10+ 的 Mac 真机
- 项目已配置有效的开发者签名

集成 TEduBoard SDK

您可以选择使用 CocoaPods 自动加载的方式，或者先下载 SDK 再将其导入到您当前的工程项目中。由于TEduBoard SDK 内部使用 IMSDK 作为内部信令通道，您还需 CocoaPods 或手动添加 IMSDK 依赖项。

CocoaPods

1. 安装 CocoaPods

在终端窗口中输入如下命令（需要提前在 Mac 中安装 Ruby 环境）：

```
sudo gem install cocoapods
```

2. 创建 Podfile 文件

进入项目所在路径，输入以下命令之后项目路径下会出现一个 Podfile 文件。

```
pod init
```

3. 编辑 Podfile 文件

编辑 Podfile 文件，支持选择版本号。

```
platform :macos, '10.11'

target 'TICDemo_Mac' do
  # pod 'TEduBoard_Mac',:podspec=>'https://sdk.qcloudtiw.com/mac/prod/2.8.9.204/TEduBoard_Mac.podspec'
  pod 'TEduBoard_Mac', '>=2.8.9.204'
  pod 'TXIMSDK_Plus_Mac', '6.2.2363'
  pod 'YYModel'
end
```

互动白板默认使用 IMSDK 作为信令通道，如果您有独立的信令通道，无需集成 IMSDK。

4. 更新并安装 SDK

在终端窗口中输入如下命令以更新本地库文件，并安装 TEduBoard SDK：

```
pod install
```

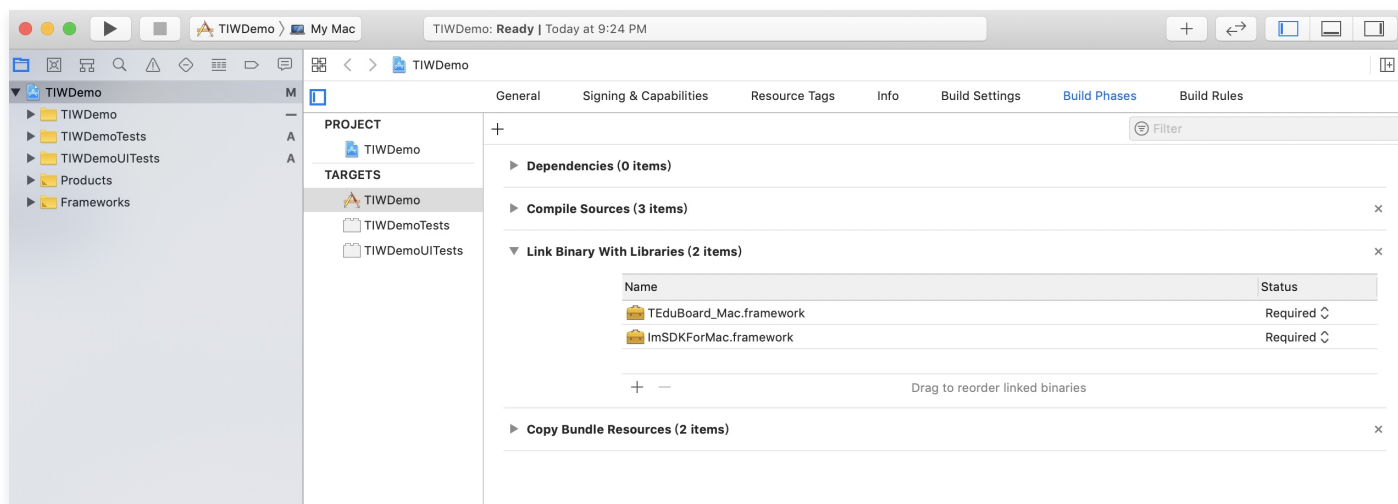
或使用以下命令更新本地库版本：

```
pod update
```

pod 命令执行完后，会生成集成了 SDK 的 .xcworkspace 后缀的工程文件，双击打开即可。

手动集成

1. 下载 [TEduBoard SDK](#)。
2. 前往 [即时通信 IM](#) 下载 IMSDK。互动白板默认使用 IMSDK 作为信令通道，如果您有独立的信令通道，请跳过此步。
3. 单击 Link Binary with Libraries 项展开，单击底下的“+”号图标去添加依赖库。



使用 TEduBoard SDK

1. #import SDK

在项目需要使用 SDK API 的文件里，引入具体的头文件：

```
#import <TEduBoard_Mac/TEduBoard.h>
```

2. 创建白板控制器

使用如下代码创建并初始化白板控制器：

```
// 创建并初始化白板控制器
// (1) 鉴权配置
TEduBoardAuthParam *authParam = [[TEduBoardAuthParam alloc] init];
authParam.sdkAppId = _sdkAppId;
authParam.userId = _userId;
authParam.userSig = _userSig;
// (2) 白板默认配置
TEduBoardInitParam *initParam = [[TEduBoardInitParam alloc] init];
_boardController = [[TEduBoardController alloc] initWithAuthParam:authParam roomId:_classId initParam:initParam];
// (3) 添加白板事件回调
[_boardController addDelegate:self];
```

其中 _sdkAppId、_userId、_userSig、_classId 为需要您自己填写的参数。

3. 白板窗口获取及显示

在 onTEBInit 或 onTEBHistoryDataSyncCompleted 回调方法内，使用如下代码获取并显示白板视图。

```
-(void)onTEBHistoryDataSyncCompleted
{
    // (1) 获取白板 NSView
    NSView *boardView = [_boardController getBoardRenderView];
    // (2) 设置显示位置和大小
    boardView.frame = CGRectMake(0, 0, width, height);
    // (3) 添加到父视图中
    [self.view addSubview:boardView];
}
```

注意

1. onTEBInit 表示白板创建并鉴权完成。
2. onTEBHistoryDataSyncCompleted 表示历史数据加载完成，此时可调用白板的相关接口。
3. SDK 所有回调都在主线程内执行，因此可以在回调里直接执行 UI 操作。

4. 监听白板关键事件

在 onTEBError 和 onTEBWarning 回调方法内监听白板事件。

- [onTEBError 错误详情](#)
- [onTEBWarning 警告详情](#)

```
// 监听白板错误事件
- (void)onTEBError:(TEduBoardErrorCode)code msg:(NSString *)msg
{

}

// 监听白板告警事件
- (void)onTEBWarning:(TEduBoardWarningCode)code msg:(NSString *)msg
{

}
```

5. 白板数据同步

白板在使用过程中，需要在不同的用户之间进行数据同步（涂鸦数据等），SDK 默认使用 IMSDK 作为信令通道，您需要自行实现 IMSDK 的初始化、登录、加入群组操作，确保白板初始化时，IMSDK 已处于所指定的群组内。

步骤1：初始化 IMSDK

```
V2TIMSDKConfig *config = [[V2TIMSDKConfig alloc] init];
config.logLevel = V2TIM_LOG_WARN;
[[V2TIMManager sharedInstance] initWithSDKAppID:config.sdkAppID config:config];
[[V2TIMManager sharedInstance] addIMSDKListener:listener];
[[V2TIMManager sharedInstance] addAdvancedMsgListener:self];
```

说明：

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

如果您有其他业务使用了 IMSDK 并希望 IMSDK 的生命周期与 App 的生命周期保持一致，请在 AppDelegate 的 application:didFinishLaunchingWithOptions 方法中初始化 IMSDK，否则请在登录前初始化 IMSDK，在登出后反初始化 IMSDK。

步骤2：登录 IMSDK

```
[[V2TIMManager sharedInstance] login:userID userSig:userSig succ:^(
// 登录 IMSDK 成功
) fail:^(int code, NSString *desc) {
// 登录 IMSDK 失败
}];
```

说明：

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

步骤3：加入群组

登录 IMSDK 成功后加入白板所在的群组。

```
[[V2TIMManager sharedInstance] joinGroup:groupId msg:nil succ:^(
// 加入 IM 群组成功
) fail:^(int code, NSString *desc) {
```

```
// 加入 IM 群组失败
}];
```

说明:

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

如果 IM 群组不存在，请先创建群组。

```
V2TIMGroupInfo *groupInfo = [[V2TIMGroupInfo alloc] init];
groupInfo.groupID = classId;
groupInfo.groupName = classId;
groupInfo.groupType = @"Public";
groupInfo.groupAddOpt = V2TIM_GROUP_ADD_ANY;
__weak typeof(self) ws = self;
[[V2TIMManager sharedInstance] createGroup:groupInfo memberList:nil succ:^(NSString *groupID) {
    // 创建 IM 群组成功
} fail:^(int code, NSString *desc) {
    // 创建 IM 群组失败
}];
```

说明:

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

注意

1. 推荐业务后台使用 IM REST API 提前创建群组。
2. 不同的群组类型，群组功能以及成员数量有所区别，具体请查看 IM 群组系统。

6. 销毁白板

调用 unInit 方法后，内部将彻底销毁白板并停止计费，请您确保此接口的调用。

```
[_boardController unInit];
```

如果您使用 IMSDK 作为信令通道，请根据业务的需要决定是否退出群组、退出登录并反初始化。

步骤1: 退出群组

```
[[V2TIMManager sharedInstance] quitGroup:classId succ:^(
    // 登出 IMSDK 成功
} fail:^(int code, NSString *desc) {
    // 登出 IMSDK 失败
}];
```

说明:

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

步骤2: 登出 IMSDK

```
[[V2TIMManager sharedInstance] logout:^(
    // 登出 IMSDK 成功
} fail:^(int code, NSString *desc) {
    // 登出 IMSDK 失败
}];
```

说明:

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

步骤3: 反初始化 IMSDK

```
[[V2TIMManager sharedInstance] removeIMSDKListener:self];  
[[V2TIMManager sharedInstance] removeAdvancedMsgListener:self];  
[[V2TIMManager sharedInstance] unInitSDK];
```

📌 说明:

以上示例为 TIM V2 版本代码，IM 的版本请尽量用新版本，具体请查阅 IM 的 [更新日志](#)。

如果您有其他业务使用了 IMSDK 并期望 IMSDK 的生命周期与 App 的生命周期保持一致，无需调用此接口。

Windows

最近更新时间：2024-04-16 09:56:31

集成 SDK

本文主要介绍如何快速的将腾讯云互动白板 SDK 集成到您的项目中。

开发环境要求

- Windows 7 及以上版本 Windows 操作系统
- Microsoft Visual Studio 2015 及以上版本，推荐使用 Microsoft Visual Studio 2015。
- Windows SDK 8.0 及以上版本，推荐使用 Windows SDK 8.1。

集成 互动白板 SDK

步骤1: 下载 Windows SDK

下载 SDK，解压打开工程文件，编译会自动下载 所需要的 SDK 文件，并保存在 SDK 目录：

目录/文件名	说明
Demo	C++ Demo 源码
SDK/TEduBoard/include/TEduBoard.h	互动白板 SDK 头文件
SDK/TEduBoard/lib/TEduBoard.lib	互动白板 SDK 导入库
SDK/TEduBoard/lib/TEduBoard.dll	互动白板 SDK 动态链接库
SDK/TEduBoard/lib/TEduBoardRender.exe	互动白板 SDK 渲染程序
SDK/TEduBoard/lib/其他文件	CEF 框架及其依赖文件
SDK/其他目录	Demo依赖的其他SDK

本文示例中，您只需要使用 SDK/TEduBoard 目录下的 互动白板 SDK 文件即可。

步骤2: 导入 SDK 到项目

在 Visual Studio 开发环境下，按如下步骤导入 SDK：

1. 从菜单中依次选择视图 > 解决方案资源管理器。
2. 在解决方案资源管理器中，右键单击要导入 SDK 的项目名称。
3. 在弹出菜单内单击属性选项，弹出项目属性对话框。
4. 从左侧配置属性列表中，选择 VC++ 目录项。
5. 将 SDK 头文件所在目录路径添加到右侧包含目录中。
6. 将 SDK 导入库所在目录路径添加到右侧库目录中。
7. 在需要使用 SDK 的源码文件内添加如下代码导入 SDK。

```
// 引入 SDK 头文件
#include "TEduBoard.h"

// 链接 SDK
#pragma comment(lib, "TEduBoard.lib")
```

步骤3: 白板创建及销毁

白板控制器创建及初始化

使用如下代码创建并初始化白板控制器：

```
// 创建白板控制器
TEduBoardController *boardCtrl = CreateTEduBoardController();
```

```
// 添加回调
boardCtrl->AddCallback(myCallback); // myCallback 为实例化 TEduBoardCallback 接口的回调接收器

// 初始化授权参数结构体
TEduBoardInitParam initParam;
initParam.sdkAppId = SDK_APP_ID; // 填写您的 SDKAppID
initParam.userId = USER_ID; // 填写用户 ID
initParam.userSig = USER_SIG; // 填写用户签名

// 初始化白板控制器（配合腾讯云 IMSDK 4.0 以上版本使用时，initParam 参数可以不填）
boardCtrl->Init(initParam, ROOM_ID);
```

监听白板关键事件

白板事件回调接口 TEduBoardCallback 的 onTEBError 和 onTEBWarning 回调方法内监听白板事件

- [onTEBError 错误详情](#)
- [onTEBWarning 警告详情](#)

```
/**
 * 白板错误回调
 *
 * @param code 错误码，参见 TEduBoardErrorCode 定义
 * @param msg 错误信息，编码格式为 UTF8
 */
virtual void onTEBError(TEduBoardErrorCode code,const char* msg)

/**
 * 白板警告回调
 *
 * @param code 警告码，参见 TEduBoardWarningCode 定义
 * @param msg 警告信息，编码格式为 UTF8
 */
virtual void onTEBWarning(TEduBoardWarningCode code,const char* msg)
```

白板窗口获取及显示

在 onTEBInit 回调方法内，使用如下代码获取并显示白板窗口：

```
// SDK所有回调都在主线程内执行，因此我们可以在回调里直接执行 UI 操作

// 获取白板窗口句柄
HWND boardHwnd = boardCtrl->GetBoardRenderView();

// 获取并计算父窗口客户区大小
RECT parentRect;
::GetClientRect(PARENT_HWND, &parentRect); // PARENT_HWND 为要将白板窗口嵌入其中显示的容器窗口句柄
int cx = parentRect.right - parentRect.left;
int cy = parentRect.bottom - parentRect.top;

// 设置白板为子窗口
::SetParent(boardHwnd, PARENT_HWND);

// 更改白板大小和位置
boardCtrl->SetBoardRenderViewPos(0, 0, cx, cy);

// 显示白板窗口
::ShowWindow(boardHwnd, SW_SHOW);
```

WM_SIZE 事件处理

在父窗口的 WM_SIZE 事件处理函数内，使用如下代码调整白板窗口大小与父窗口适配：

```
// 获取白板窗口句柄
```

```
HWND boardHwnd = boardCtrl->GetBoardRenderView();
```

```
// 获取并计算父窗口客户区大小  
RECT parentRect;  
::GetClientRect(PARENT_HWND, &parentRect);  
int cx = parentRect.right - parentRect.left;  
int cy = parentRect.bottom - parentRect.top;  
  
// 更改白板大小  
boardCtrl->SetBoardRenderViewPos(0, 0, cx, cy);
```

白板窗口关闭

需要关闭白板窗口时，使用如下代码关闭：

```
// 销毁白板控制器  
DestroyTEduBoardController(&boardCtrl);
```

步骤4：白板数据同步

白板在使用过程中，需要在不同的用户之间进行数据同步（涂鸦数据等），SDK 默认使用 IMSDK 作为信令通道，您需要自行实现 IMSDK 的初始化、登录、加入群组操作，确保白板初始化时，IMSDK 已处于所指定的群组内。

- [TIM 同步信令通道](#)
- [TIM 相关文档](#)

Web

最近更新时间：2024-03-29 10:56:21

集成 SDK

本文主要介绍如何快速的将腾讯云 TEduBoard SDK 集成到您的项目中。

支持平台

操作系统平台	浏览器/Webview	版本要求	备注
Windows	Chrome	50+	Win7+
Windows	Firefox	50+	Win7+
Mac	Chrome	50+	-
Mac	Safari	8+	-
Mac	Firefox	50+	-
iOS	-	系统版本 8.1+	-
Android	-	系统版本 4.2+	推荐使用 Chrome 浏览器，微信浏览器，手机 QQ 浏览器

集成 TEduBoard SDK

您只需要在您的 Web 页面中添加如下代码即可：

```
<!-- axios SDK -->
<script src="https://res.qcloudtiw.com/board/third/axios/axios.min.js"></script>
<!-- COS SDK -->
<script src="https://res.qcloudtiw.com/board/third/cos/5.1.0/cos.min.js"></script>
<!-- TEduBoard SDK -->
<script src="https://res.qcloudtiw.com/board/2.8.7/TEduBoard.min.js"></script>
```

如果您需要添加视频文件还需要添加以下代码：

```
<link href="https://res.qcloudtiw.com/board/third/videojs/1.0.0/video-js.min.css" rel="stylesheet">
<script src="https://res.qcloudtiw.com/board/third/videojs/1.0.0/video.min.js"></script>
```

目前互动白板中依赖 axios、cos，请使用 script:src 的方式加载，这样能够保证在全局访问到 axios 和 cos，不支持 import 的方式。

使用 TEduBoard SDK

1. 创建白板控制器

```
var initParams = {
  id: domId, // dom节点id
  classId: classId, // 课堂 ID, 32位整型, 取值范围[1, 4294967294]
  sdkAppId: sdkAppId, // 整数
  userId: userId, // 字符串
  userSig: userSig, // 字符串
};
var teduBoard = new TEduBoard(initParams);
```

2. 监听白板关键事件

使用 on 监听白板关键事件

- [onTEBError 错误详情](#)
- [onTEBWarning 警告详情](#)

```
// 监听白板错误事件
teduBoard.on(TEduBoard.EVENT.TEB_ERROR, (code, msg) => {

});
```

```
// 监听白板告警事件
teduBoard.on(TEduBoard.EVENT.TEB_WARNING, (code, msg) => {

});
```

3. 白板数据同步

白板在使用过程中，需要在不同的用户之间进行数据同步（涂鸦数据等），SDK 默认使用 IMSDK 作为信令通道，您需要自行实现 IMSDK 的初始化、登录、加入群组操作，确保白板初始化时，IMSDK 已处于所指定的群组内。

监听事件 TEduBoard.EVENT.TEB_SYNCDATA

⚠ 注意

因为 TIM 消息有限频，请将白板消息的优先级设置为最高，以保证白板信令消息不会被丢弃。

```
// 1. 监听操作白板参数的数据，并将回调的数据通过 im 发送到接收者
teduBoard.on(TEduBoard.EVENT.TEB_SYNCDATA, data => {
  let message = this.tim.createCustomMessage({
    to: '课堂号',
    conversationType: window.TIM.TYPES.CONV_GROUP,
    priority: TIM.TYPES.MSG_PRIORITY_HIGH, // 因为im消息有限频，白板消息的优先级调整为最高
    payload: {
      data: JSON.stringify(data),
      description: '',
      extension: 'TXWhiteBoardExt' // 固定写法，各端会以extension: 'TXWhiteBoardExt'为标志作为白板信令
    }
  })
  this.tim.sendMessage(message).then(() => {
    // 同步成功
  }, () => {
    // 同步失败
  });
});
```

- 将收到的数据添加到白板中 addSyncData。

```
// 2. 接收者通过监听 im 消息回调，将收到的数据添加到白板中
this.tim.on(window.TIM.EVENT.MESSAGE_RECEIVED, () => {
  let messages = event.data;
  messages.forEach((message) => {
    // 群组消息
    if (message.conversationType === window.TIM.TYPES.CONV_GROUP) {
      if (message.to === '课堂号') { // 如果是当前课堂，这里需要注意一定只能接受当前课堂群组的信令。如果用户加入多个im群组，恰巧这几个群组也在上课，白板信令则会多个课堂错乱，引发互动白板不能同步的异常行为。
        let elements = message.getElements();
        if (elements.length) {
          elements.forEach((element) => {
            if (element.type === 'TIMCustomElem') {
              if (element.content.extension === 'TXWhiteBoardExt') {
                if (message.from !== this.userId) { // 过滤自己本人的操作信令
                  teduBoard.addSyncData(JSON.parse(element.content.data));
                }
              }
            }
          });
        }
      }
    }
  });
});
```

```
}  
} else {  
  // 其他群组消息忽略  
}  
}  
});  
, this)
```

小程序

最近更新时间：2024-03-29 10:56:21

集成 SDK

本文主要介绍如何快速的将腾讯云 TEduBoard SDK 集成到您的项目中。

环境要求

- 微信版本6.6.6+。
- 微信开发者工具最新版本。

集成 TEduBoard SDK

① 说明

推荐使用 web-view 方案集成，即基于 Web 端 SDK 集成，参考 web 端文档将集成好的网页，在小程序中使用 web-view 加载即可。目前受限于小程序组件规则，使用原生的小程序组件实现的白板（canvas 方案）SDK，缺少对 PPT 的能力，缩放，移动等能力，故不推荐原生组件方案（canvas 方案）。

⚠ 注意：

原生小程序组件 canvas 方案目前不接新需求。

白板功能点	canvas 方案	webview 方式	说明
画笔	✓	✓	
橡皮	✓	✓	
激光笔	✓	✓	
直线	✓	✓	
空心椭圆	✓	✓	
空心矩形	✓	✓	
实心椭圆	✓	✓	
实心矩形	✓	✓	
背景色	✓	✓	
背景图	✓	✓	
H5背景	✗	✓	
点选	✓	✓	
框选	✓	✓	
撤销	✓	✓	
重做	✓	✓	
课件转静态（转图片）	✓	✓	支持 PPT、PDF 转静态图片
课件转动态（转h5）	✗	✓	
放大	✗	✓	
缩小	✗	✓	
拖动	✗	✓	

初始化设置白板比例	✓	✓	
文字输入/展示	✗	✓	
涂鸦平滑级别	✓	✓	
动态设置白板比例	✗	✓	

配置业务域名

在 [微信公众平台](#) 登录您小程序的账号和密码，并在 **开发管理 > 开发设置** 下的服务器域名，配置 web-view 业务域名。

! 说明

配置 web-view 业务域名主要是为了加载转码后的动态 PPT，请先参考 [存储桶配置](#)，在互动白板控制台中配置好存储桶，新的转码文件会存储到该桶下，然后将该桶的访问域名配置到 web-view 的业务域名中。