互动白板

快速入门

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 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】

腾讯云

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
  - Web
  - 小程序
  - HarmonyOS
- 集成 SDK
  - Android
  - iOS
  - Mac
  - Web
  - 小程序
  - HarmonyOS

版权所有：腾讯云计算（北京）有限责任公司

快速入门
跑通 Demo
Android

最近更新时间：2024-12-13 22:13:02

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

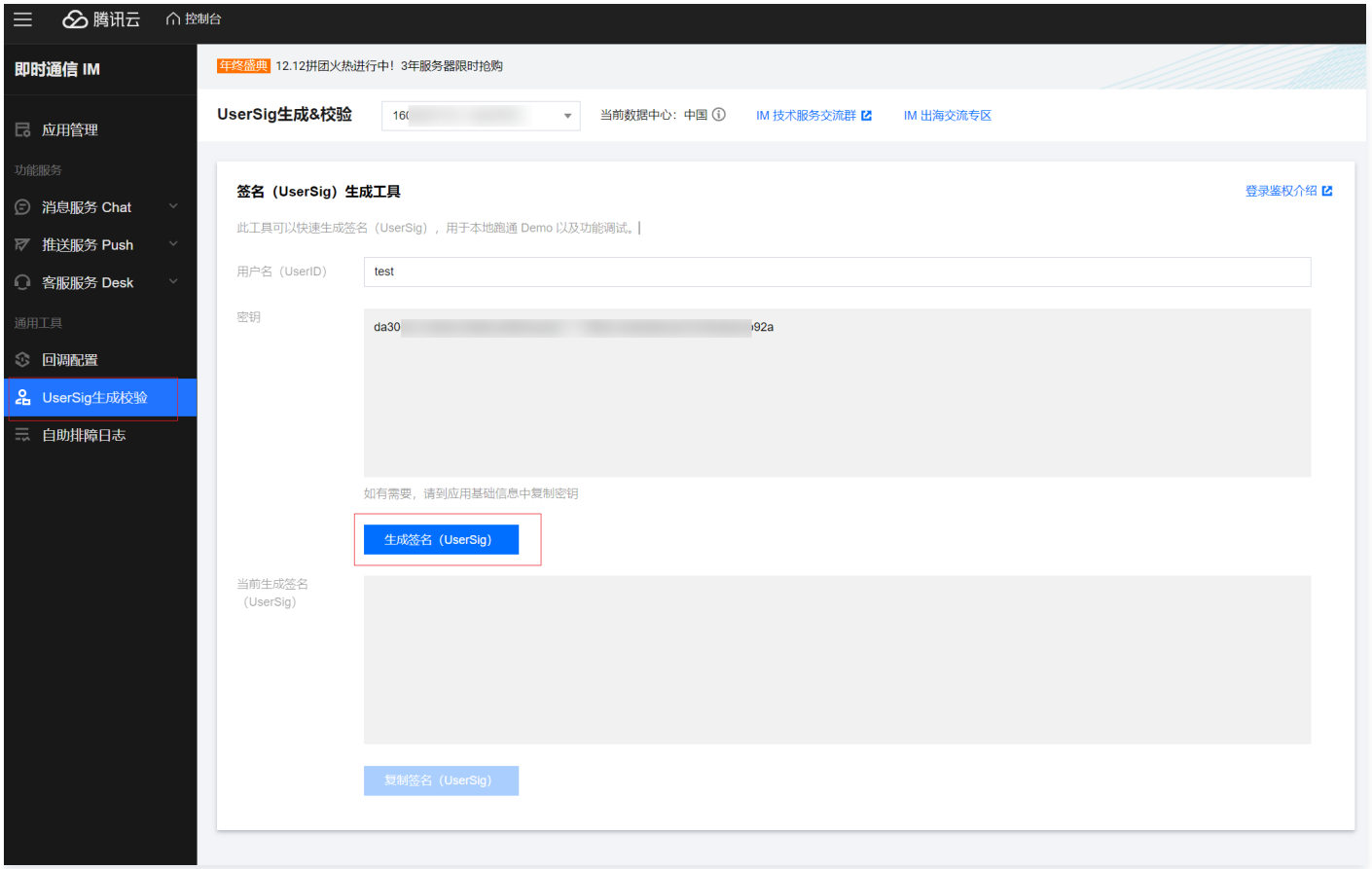
Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

1. 创建白板应用

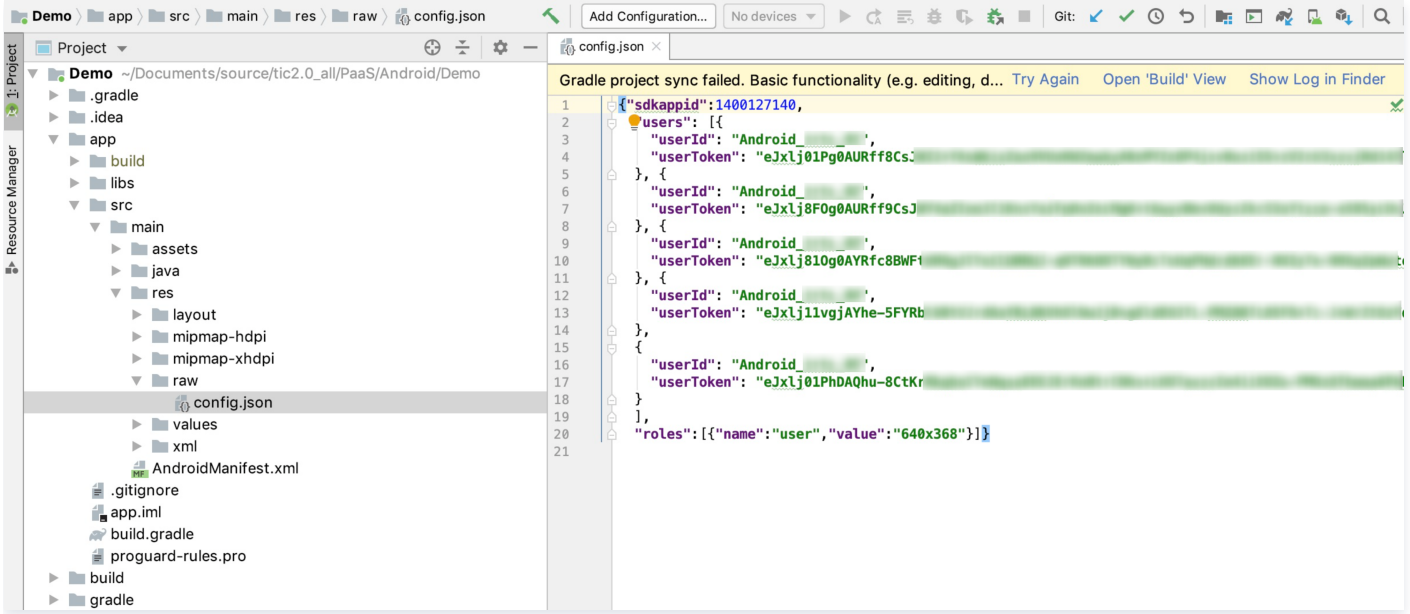
步骤一：登录 互动白板控制台，在左侧导航栏单击**应用管理**，进入互动白板应用列表。
步骤二：单击**创建应用**，进入创建新应用对话框。
步骤三：您可以选择创建一个新应用或从已有的 IM 应用中导入，文档转码和实时录制资源存储桶配置请参考 存储桶配置，回调配置请参考 回调配置，如果您暂时不需要使用文档转码和实时录制功能，可以在创建应用对话框中关闭相应功能，您可以在**应用配置**里随时配置资源存储桶打开该文档转码或实时录制功能。

2. 配置测试账号

白板 Demo 需要通过 IM 服务实时拉取白板操作信令进行白板渲染，因此需要提供 IM 应用的密钥用于生成 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：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）**。



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。



iOS

最近更新时间：2024-12-13 22:13:02

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

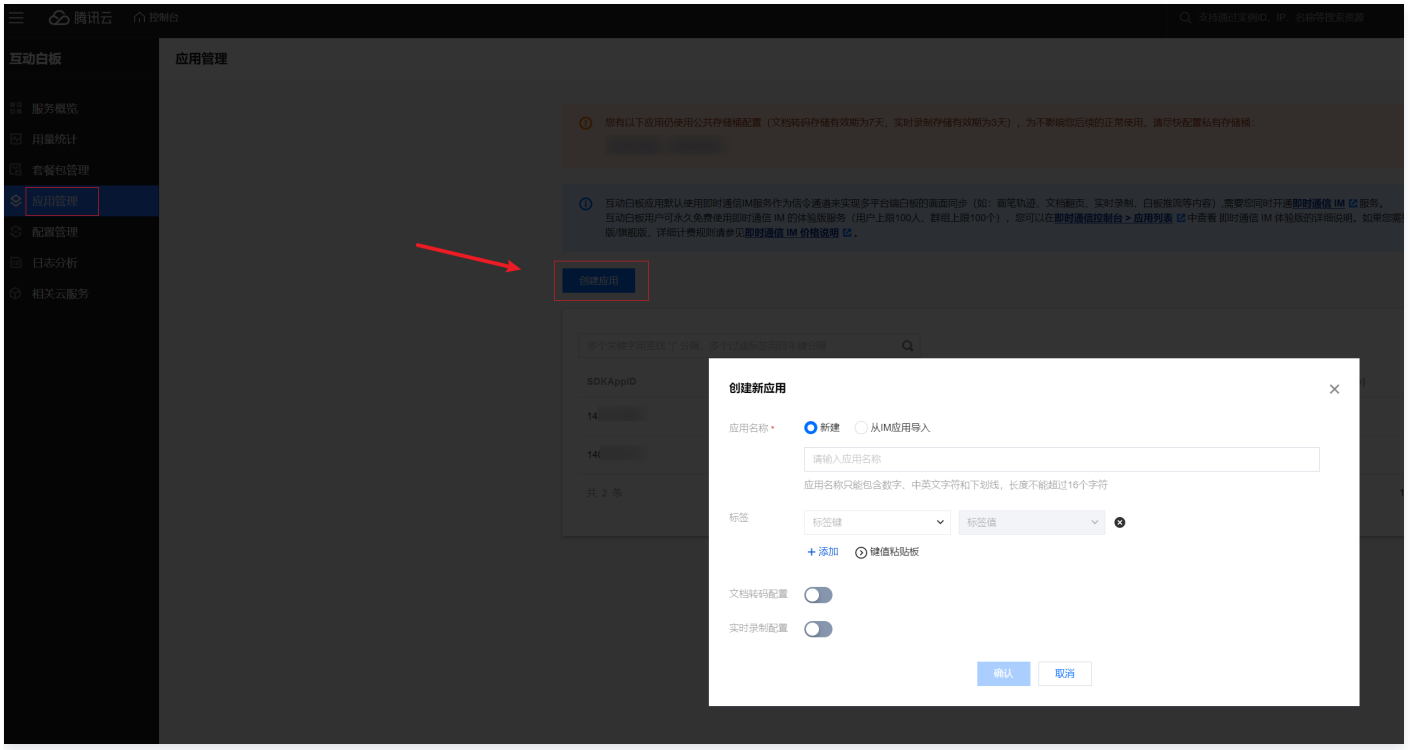
1. 在终端窗口进入 Demo Podfile 文件所在目录。
2. 执行 `pod install` 命令安装 TEduBoard SDK、TIMSDK、TRTC SDK，或执行 `pod update` 命令，更新本地库版本。
3. 使用 XCode（9.0及以上的版本）打开源码目录下的 .xcworkspace 工程，编译并运行即可。

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。

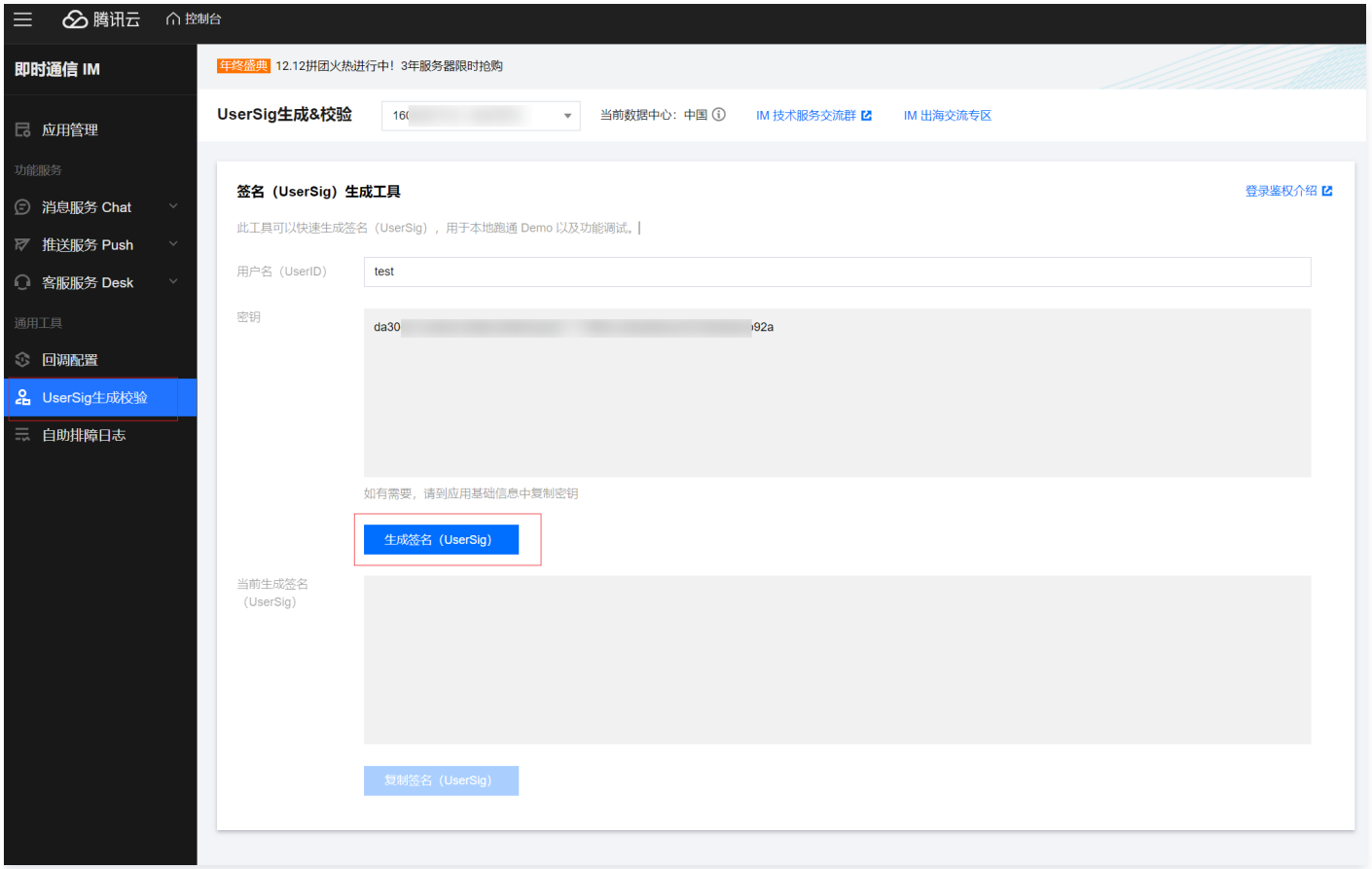
1. 创建白板应用

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。

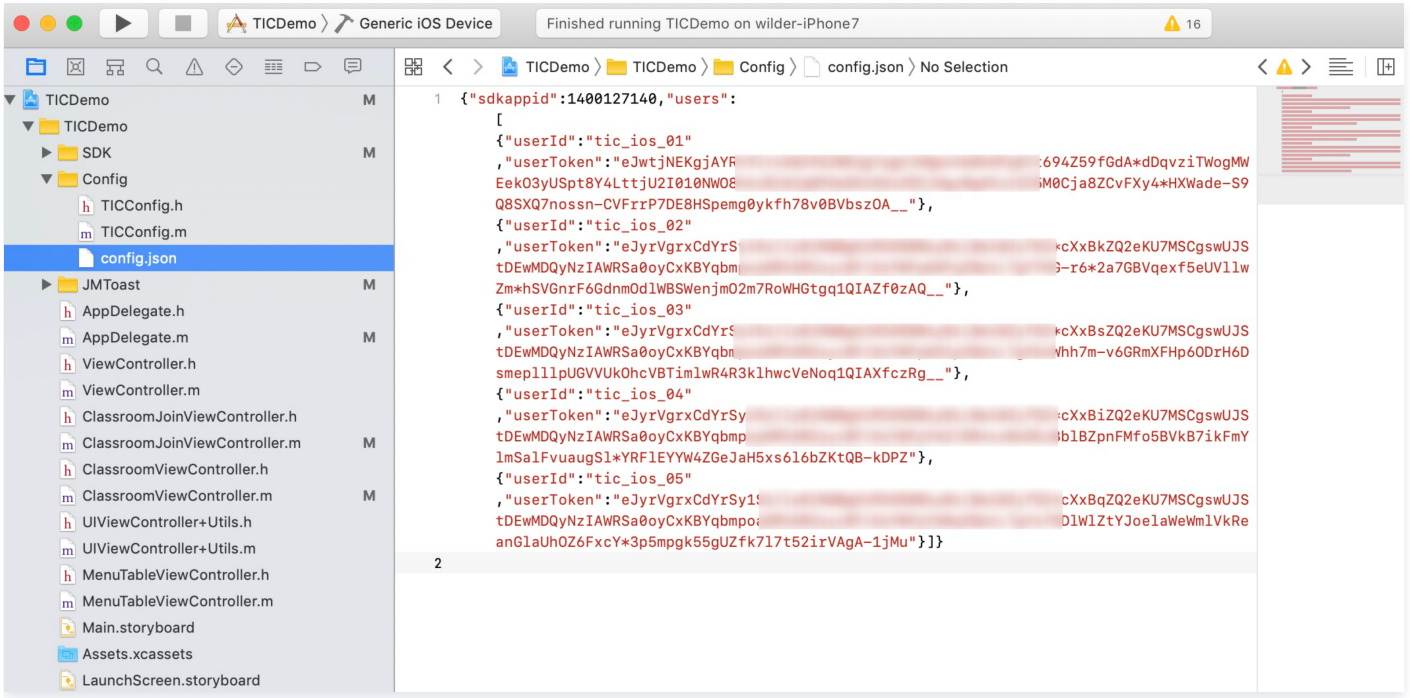


2. 配置测试账号

白板 Demo 需要通过 IM 服务实时拉取白板操作信令进行白板渲染，因此需要提供 IM 应用的密钥用于生成 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：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）**。



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。



Mac

最近更新时间：2024-12-13 22:13:02

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
- 项目已配置有效的开发者签名。

编译运行

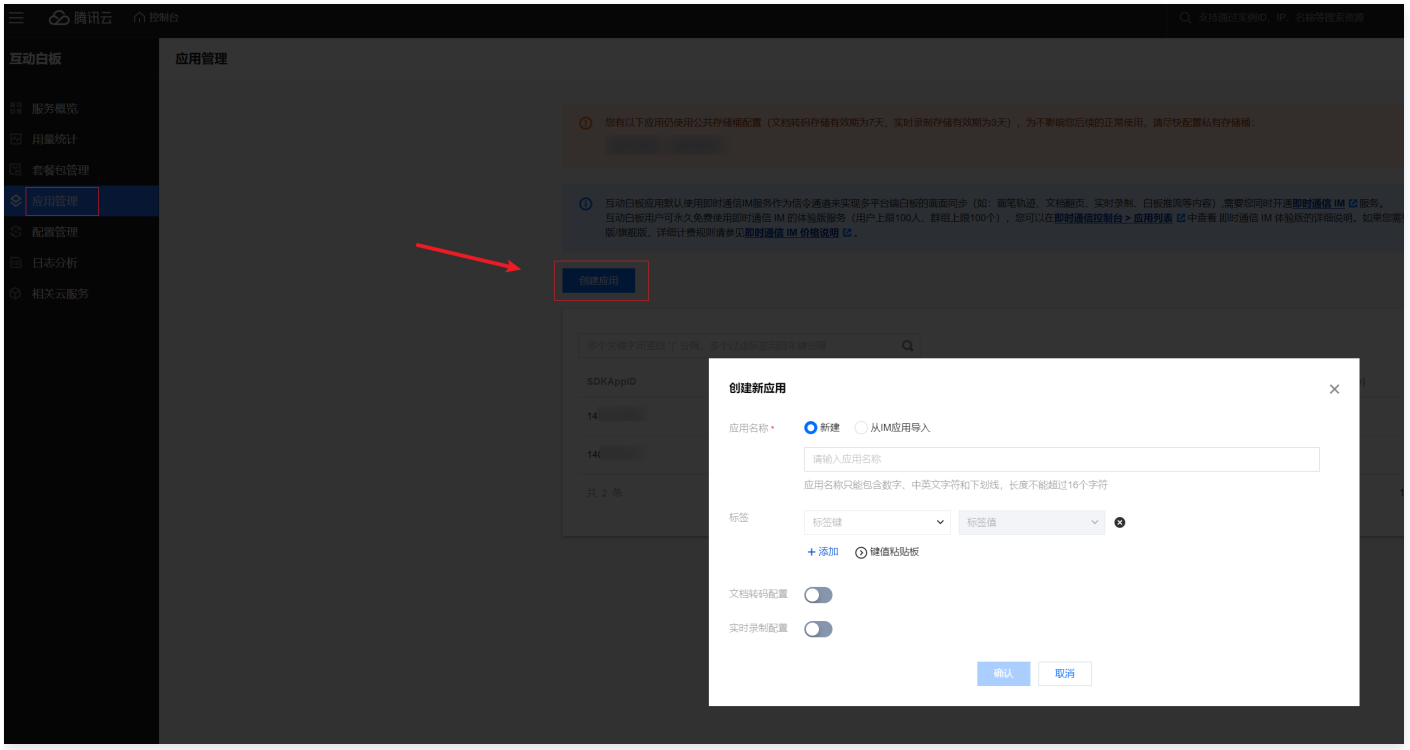
1. 在终端窗口进入 Demo Podfile 文件所在目录。
2. 执行 `pod install` 命令安装 TEduBoard SDK、TIMSDK、TRTC SDK，或执行 `pod update` 命令，更新本地库版本。
3. 使用 XCode（9.0及以上的版本）打开源码目录下的 .xcworkspace 工程，编译并运行即可。

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。

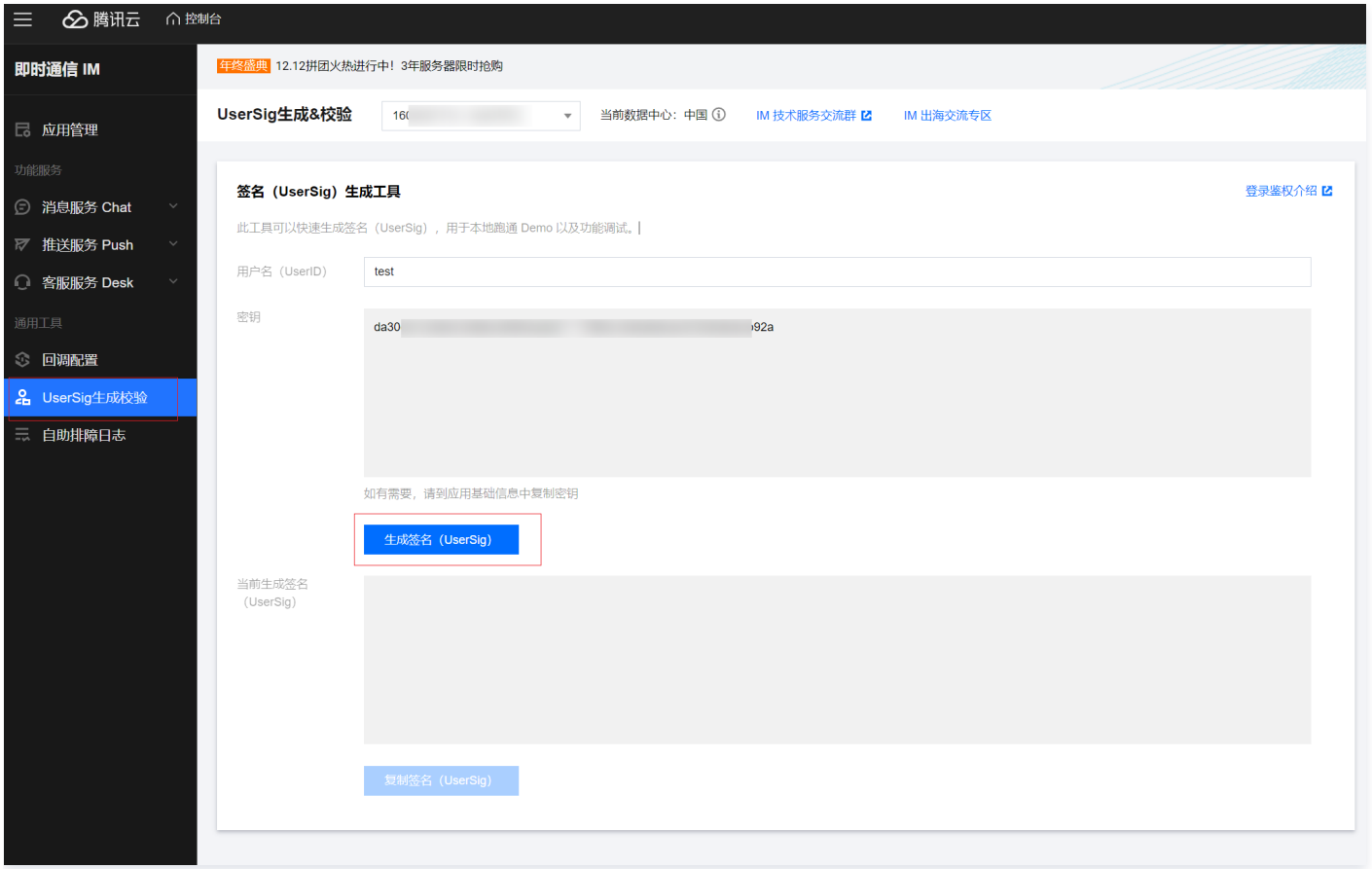
1. 创建白板应用

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。

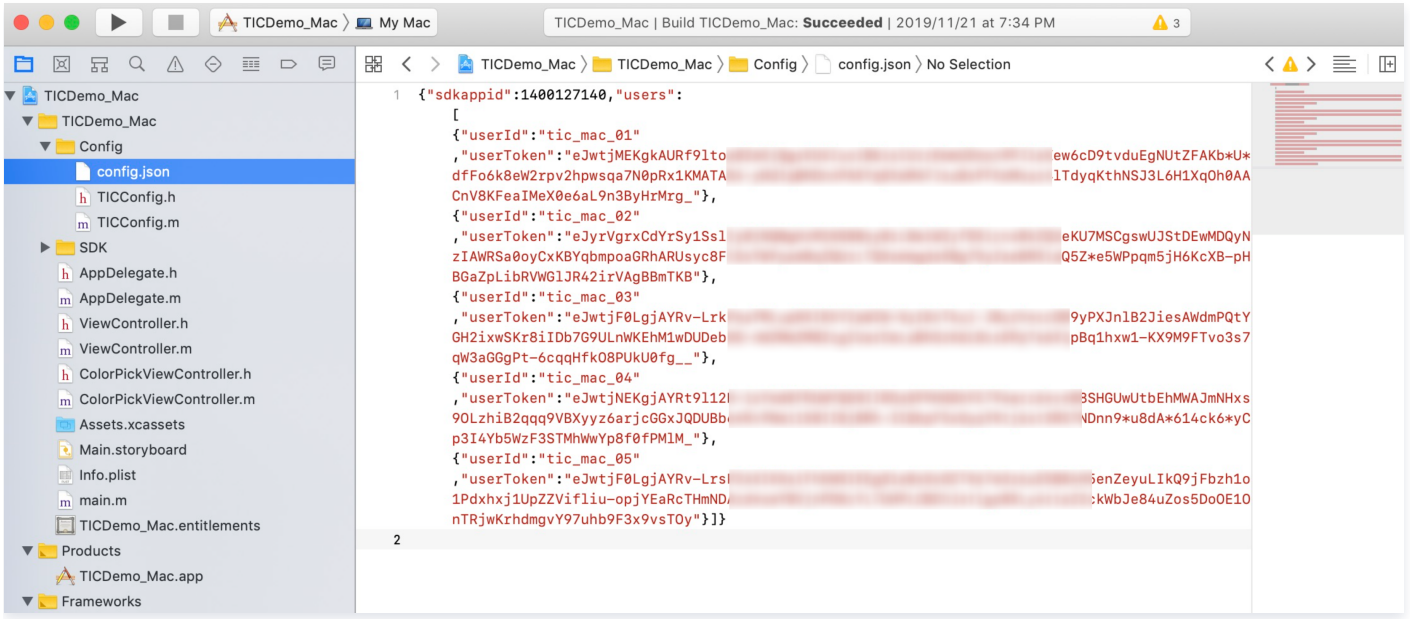


2. 配置测试账号

白板 Demo 需要通过 IM 服务实时拉取白板操作信令进行白板渲染，因此需要提供 IM 应用的密钥用于生成 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：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）**。



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。



Web

最近更新时间：2025-02-07 16:02:02

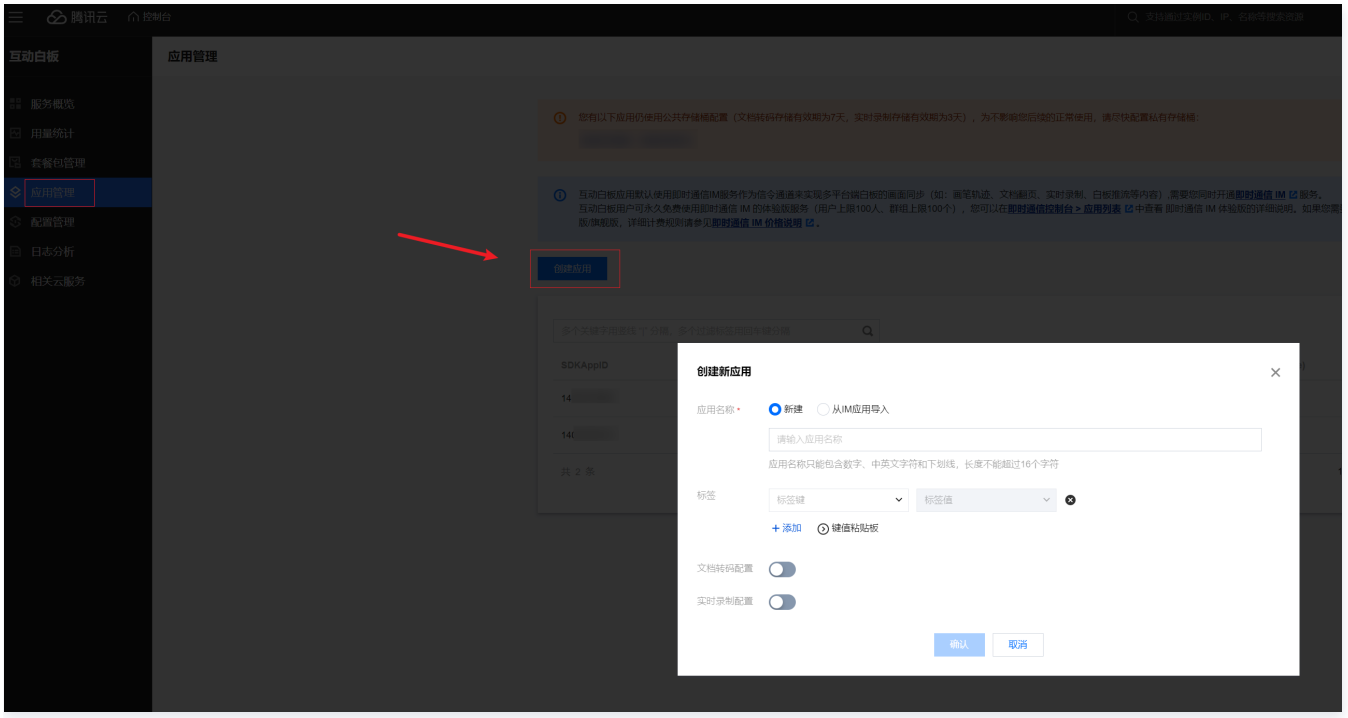
跑通 Demo

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。

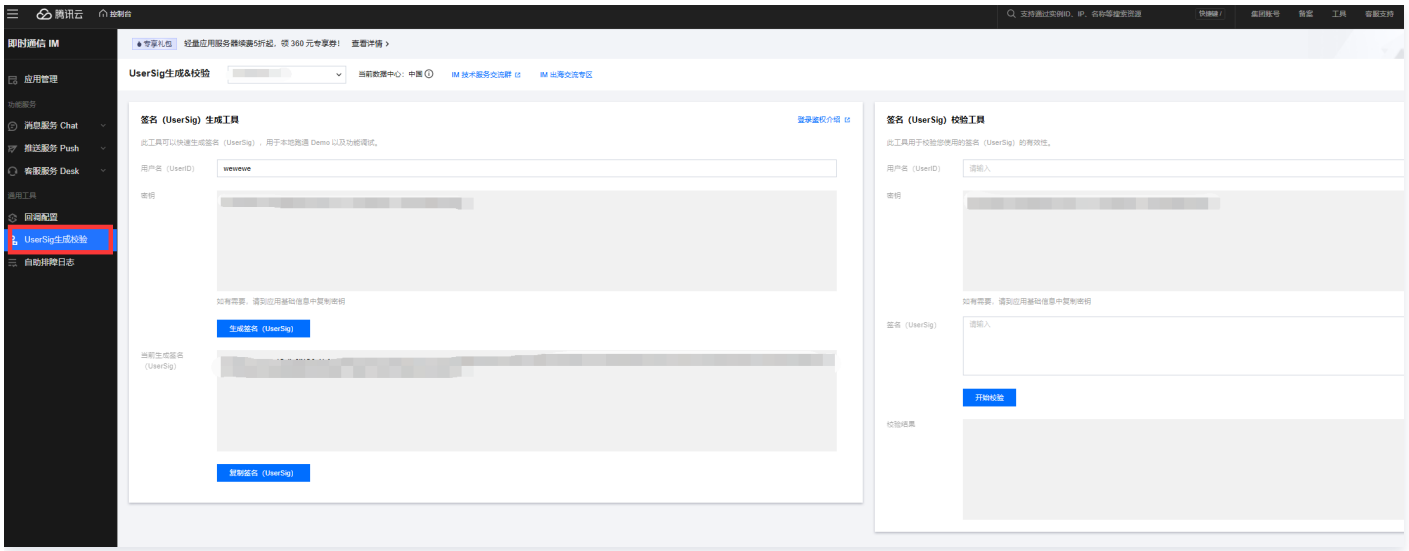


2. 配置测试账号

白板 Demo 需要通过 IM 服务实时拉取白板操作信令进行白板渲染，因此需要提供 IM 应用的密钥用于生成 UserSig。

> **注意**
> 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：单击通用**工具 > UserSig 生成&校验**，输入**用户名（UserID）**，单击**生成签名（UserSig）**。（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 ×
src > config > JS index.js > ...
You, a minute ago | 2 authors (zhutaohuang and others)
export default {
  // 白板应用SdkAppId
  sdkAppId: 0,
  // 白板应用对应的密钥 https://cloud.tencent.com/document/product/1137/39907
  secretKey: '',
  // 白板应用签名过期时间（单位：秒）
  expireTime: 84600,
  // cos存储桶配置
  cosBucket: '',
  cosRegion: '',
  // 腾讯云账号API密钥（https://console.cloud.tencent.com/cam/capi）
  tencentSecretId: '',
  tencentSectetKey: '',
  // 发起转码请求服务端地址，参考 https://cloud.tencent.com/document/product/1137/40060
  createTranscodeUrl: '',
  // 查询转码进度服务端地址，参考 https://cloud.tencent.com/document/product/1137/40059
  describeTranscodeUrl: '',
};
```

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：2025-02-14 14:59:22

跑通 Demo

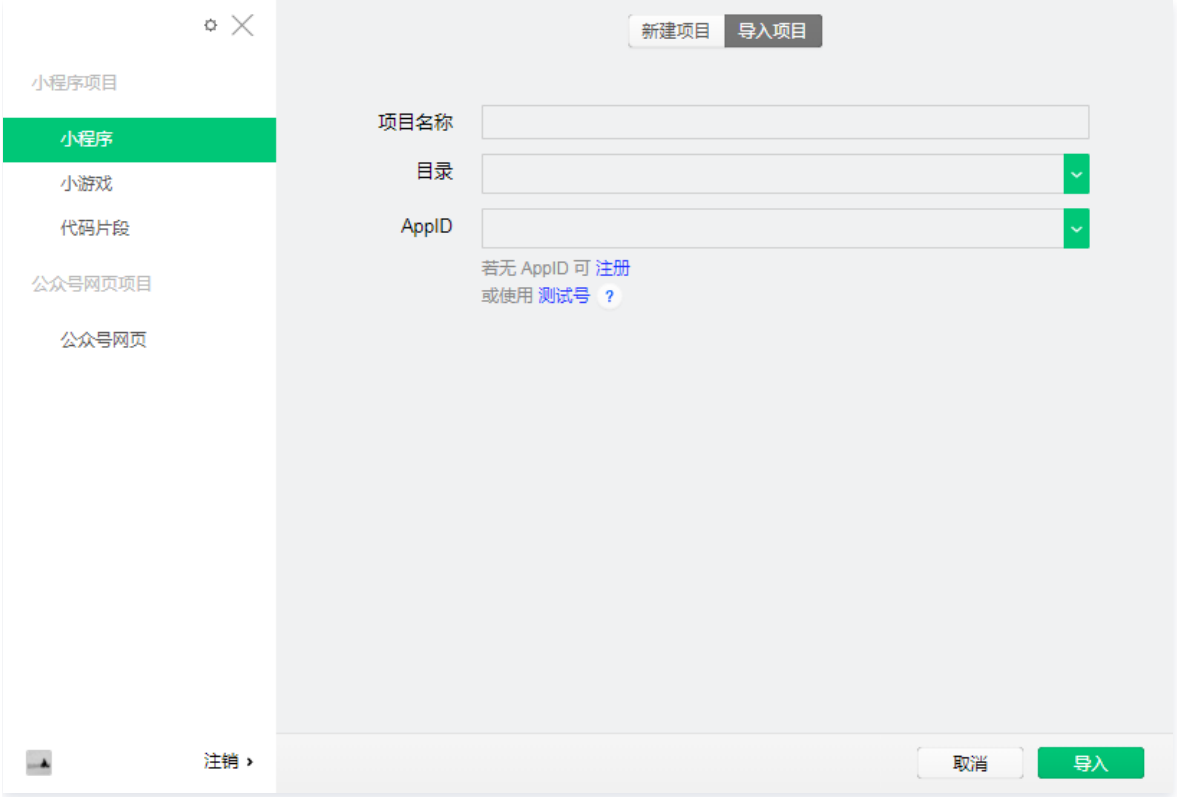
本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

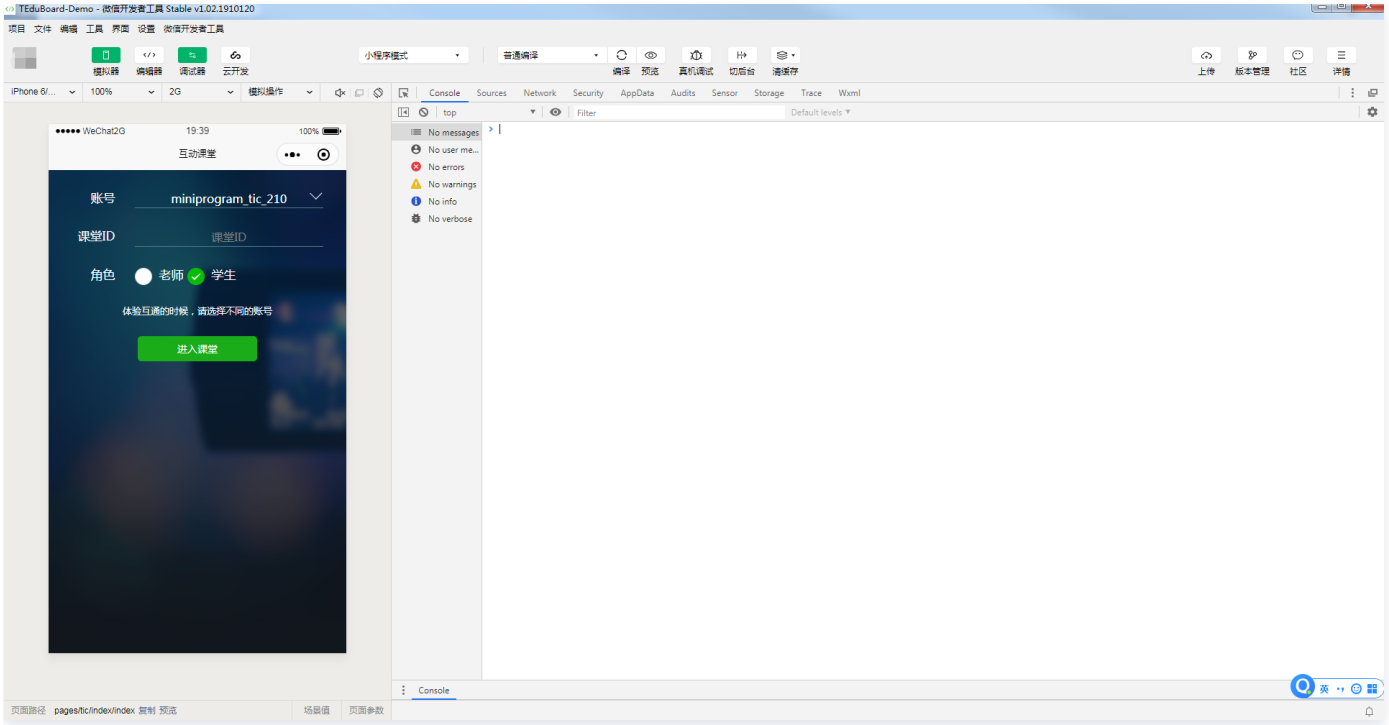
- 微信版本6.6.6+。
- 微信开发者工具最新版本。

运行

1. 导入 Demo 项目

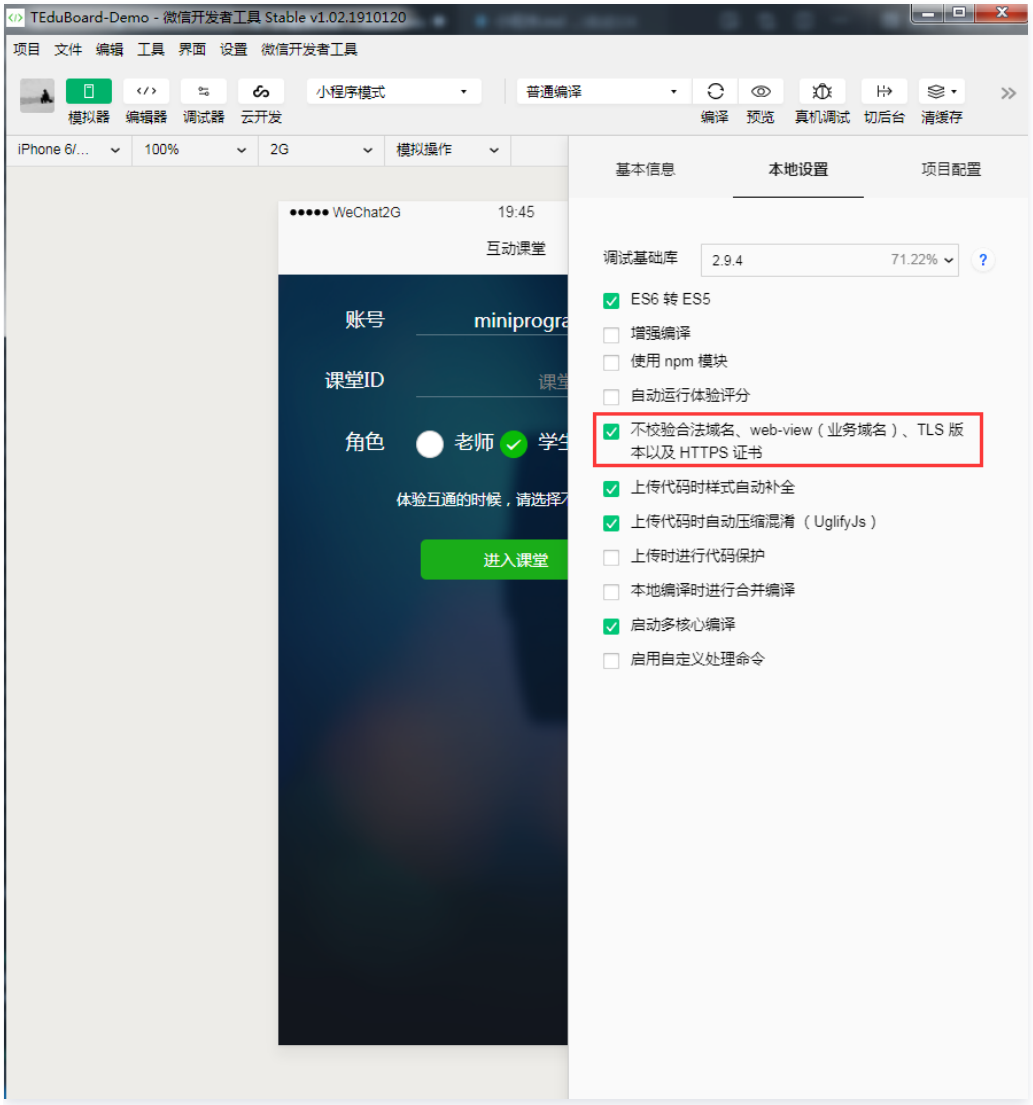


2. 编译运行



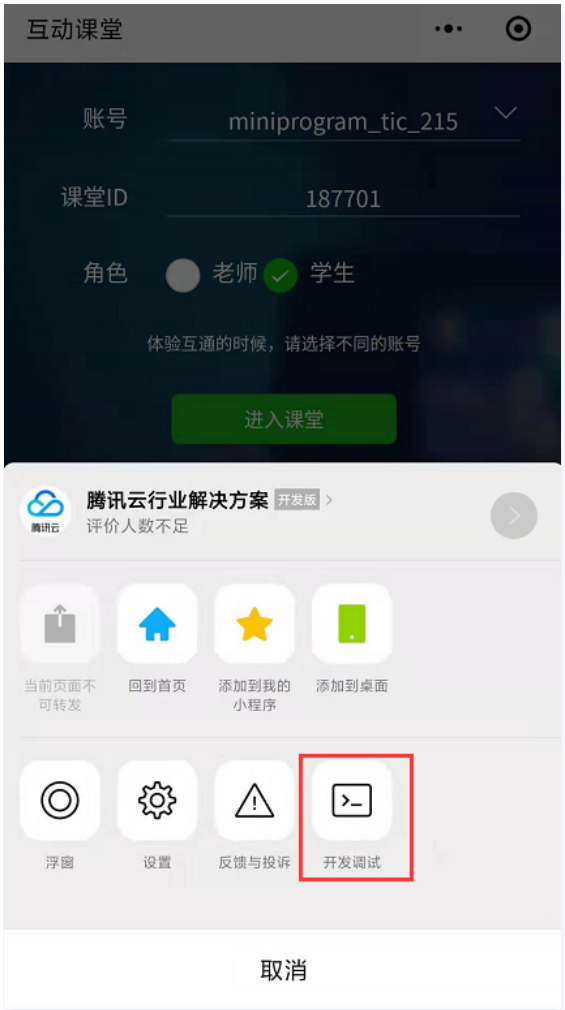
3. 配置业务域名

> **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。

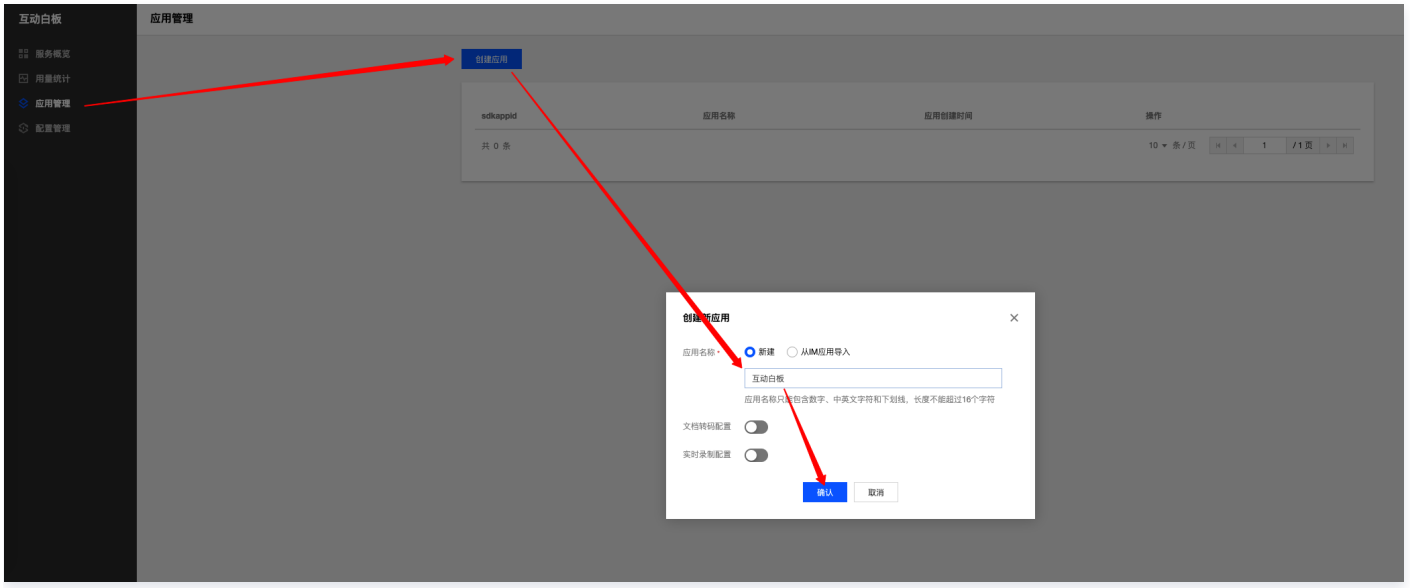


账号配置

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

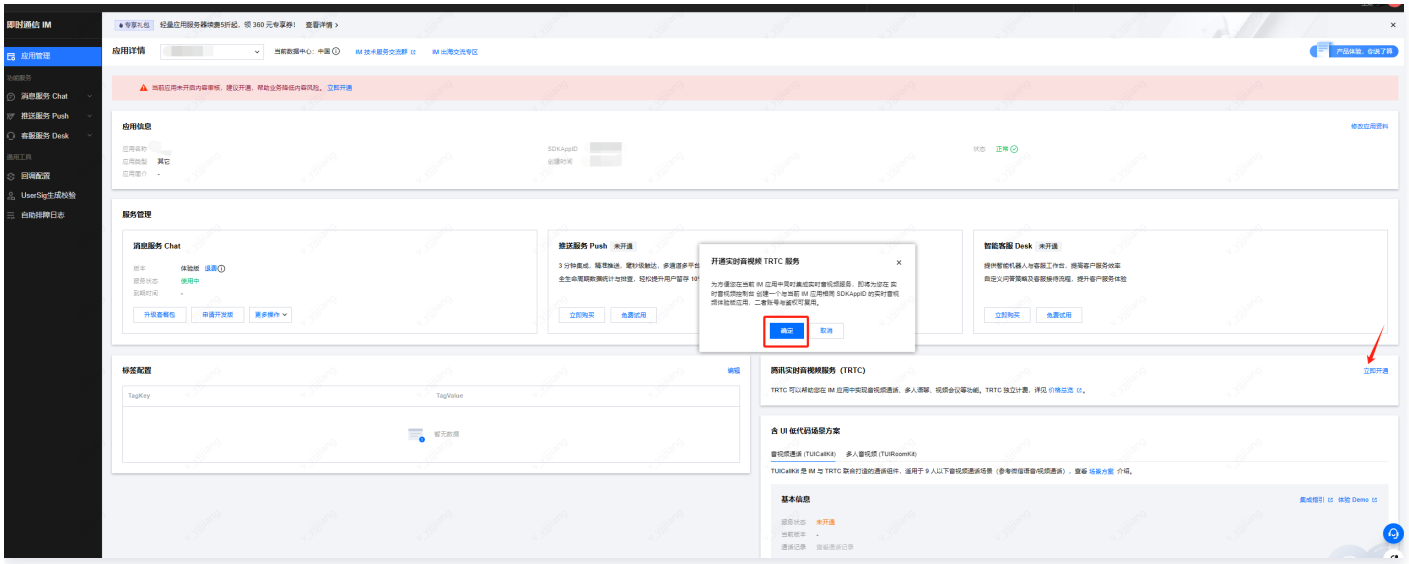
Demo 集成了实时音视频 TRTC SDK，为保证音视频功能的正常使用，请开通音视频服务。

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。

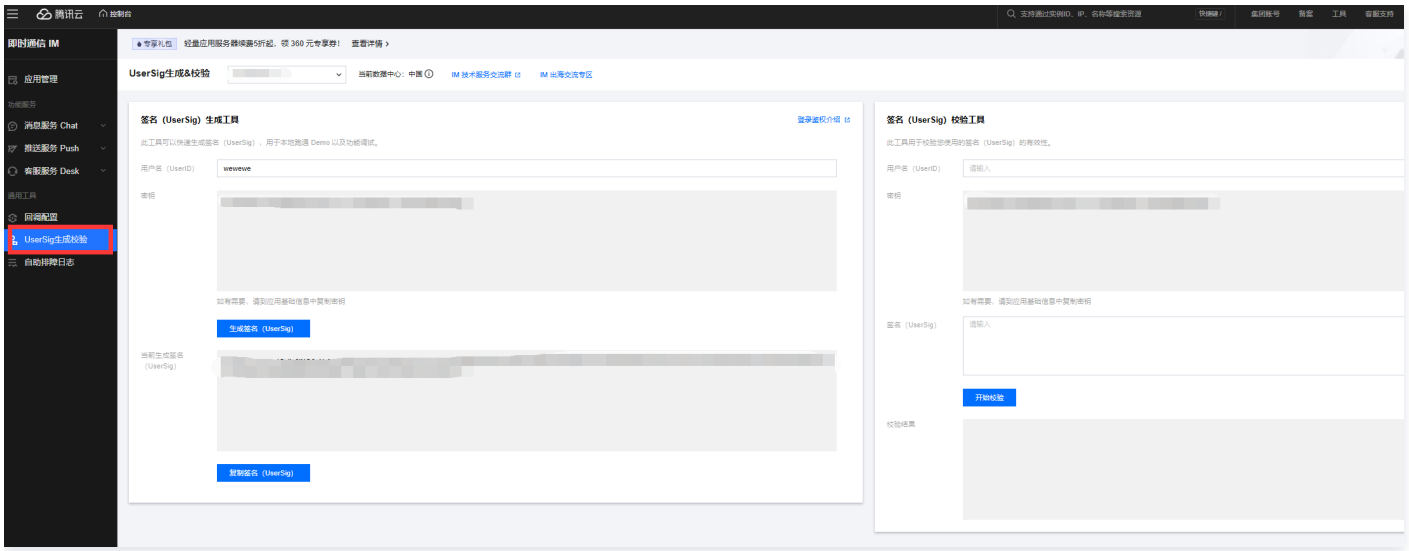


3. 配置测试账号

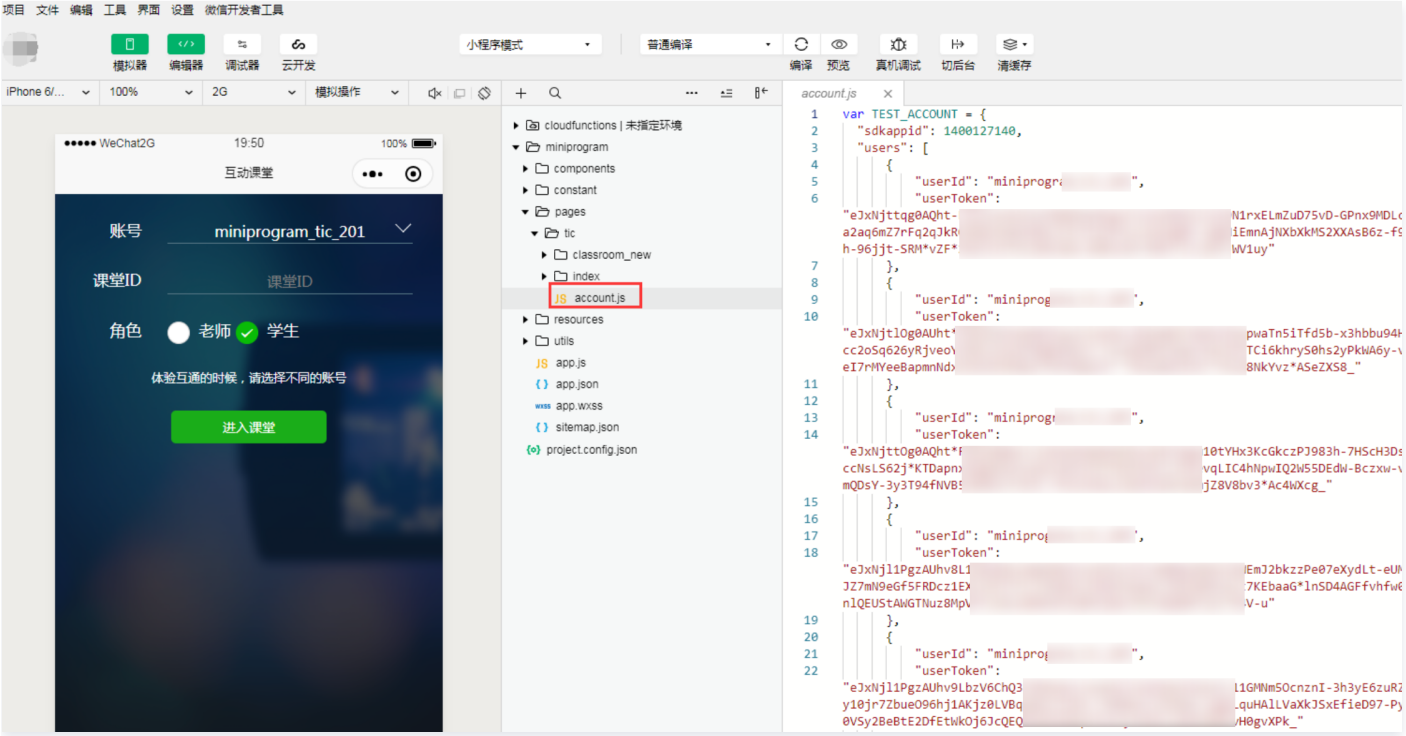
白板 Demo 需要通过 IM 服务实时拉取白板操作信令进行白板渲染，因此需要提供 IM 应用的密钥用于生成 UserSig。

步骤一：登录 即时通信 IM 控制台，单击 SDKAppID 对应的应用。

步骤二：单击通用**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。



HarmonyOS

最近更新时间：2025-04-07 16:22:11

跑通 Demo

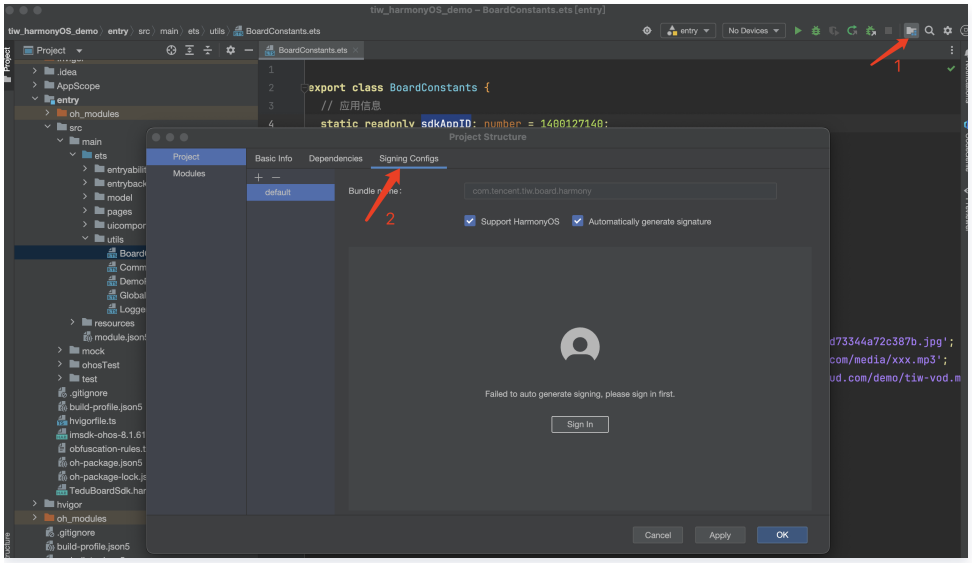
本文主要介绍如何快速运行腾讯云 TEduBoard Demo。

环境要求

- DevEco Studio 5.0 Release 及以上。
- HarmonyOS SDK API 12 及以上。

编译运行

1. 使用 DevEco Studio（5.0 以上的版本）打开源码工程，配置签名。



2. 选择设备，编译运行。

账号配置

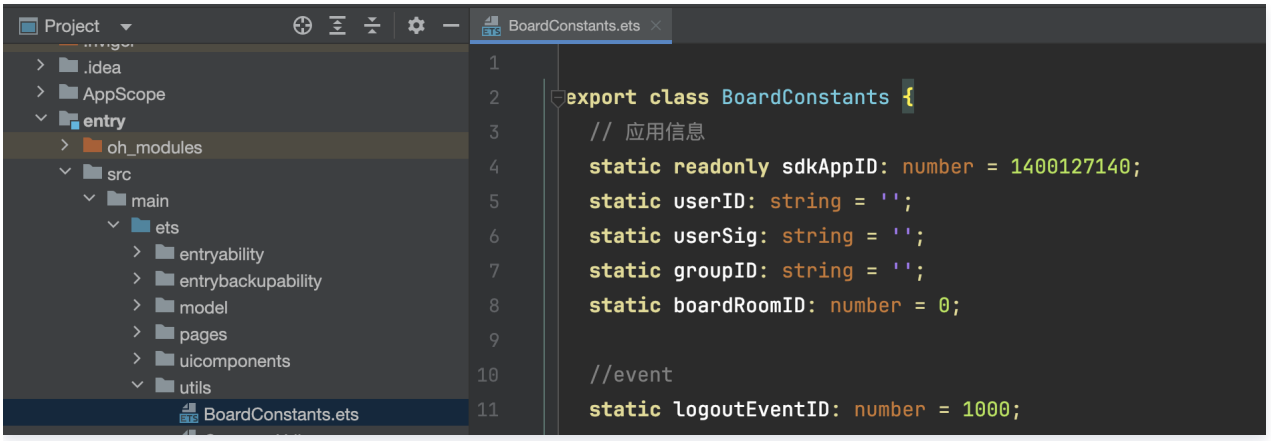
Demo 的测试账号为公共账号，多人同时登录会被强制退出，为保证您的顺利体验，请重新配置账号。

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

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



集成 SDK
Android

最近更新时间：2024-09-11 09:08:11

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

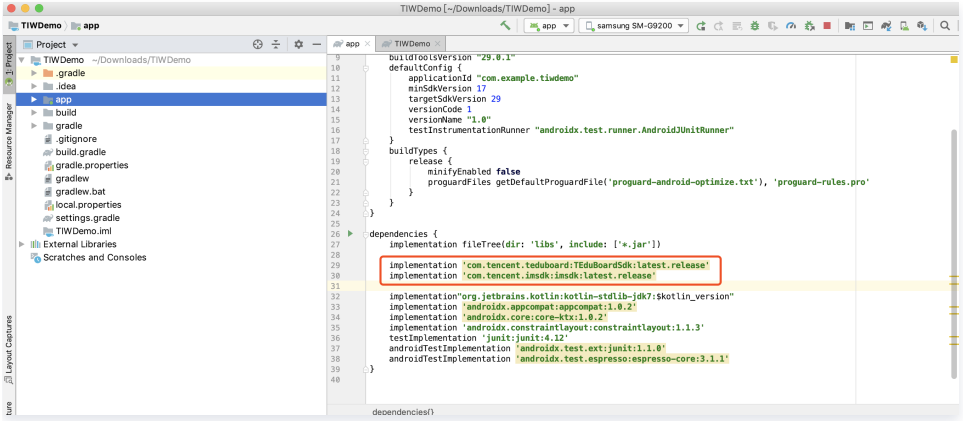
- Android Studio 2.0+
- Android 4.2（SDK API 17）及以上系统

集成 TEduBoard SDK

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

自动加载（aar）

TEduBoard SDK 和 TIMSDK 已经发布到 Maven Central 库，您可以通过配置 gradle 自动下载更新。



1. 添加 SDK 依赖

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

2. 同步 SDK

单击 Sync Now，如果您的网络连接 Maven Central 没有问题，SDK 会自动下载集成到工程里。

手动下载（aar）

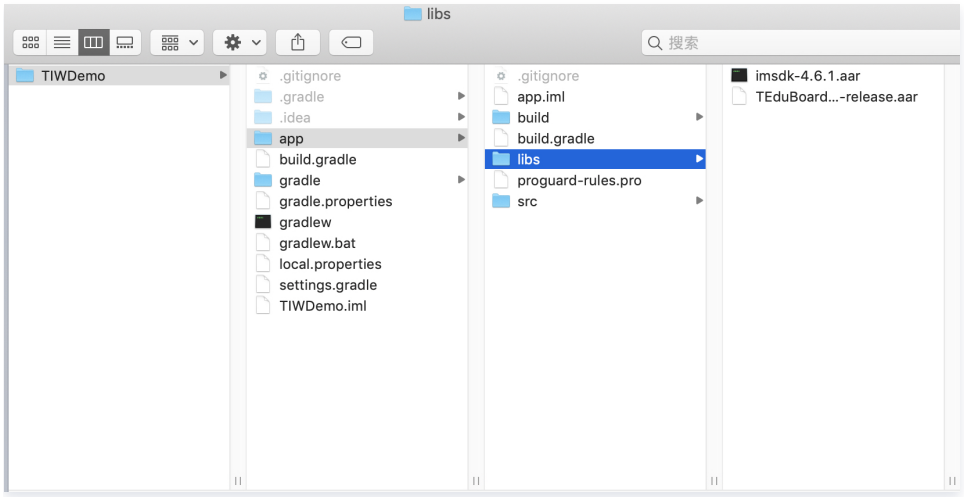
如果您的网络连接 Maven Central 有问题，也可以手动下载 SDK 集成到工程里。

1. 下载 SDK

单击下载最新版 TEduBaord SDK 。前往 即时通信 IM 下载 TIMSDK。

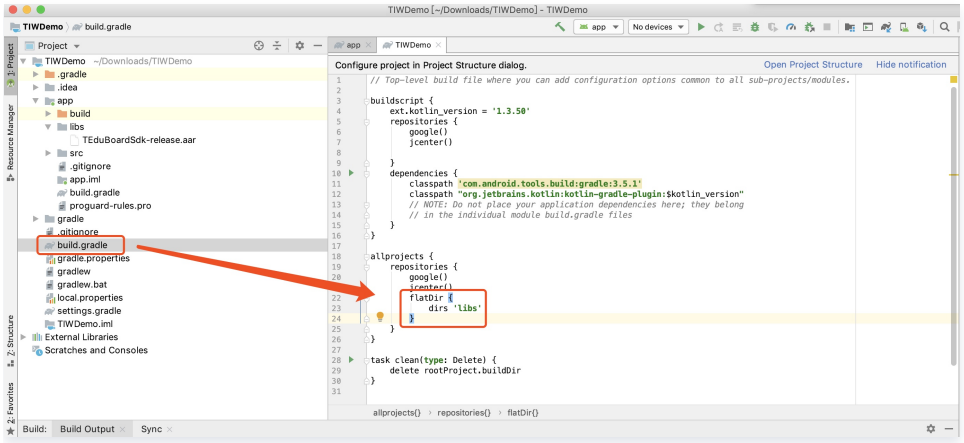
2. 导入 SDK

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

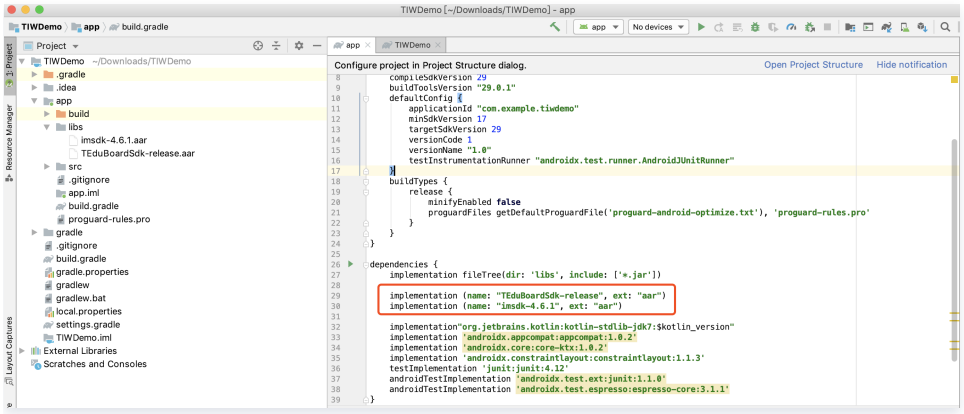
在工程根目录下的 build.gradle 中，添加 flatDir，指定本地仓库路径。



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
https://cloud.tencent.com/document/product/269/1606
    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**
> 1. 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。
> 2. 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 并期望 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，在登出后反初始化 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。

iOS

最近更新时间：2024-09-11 09:08:11

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，输入以下命令行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
```

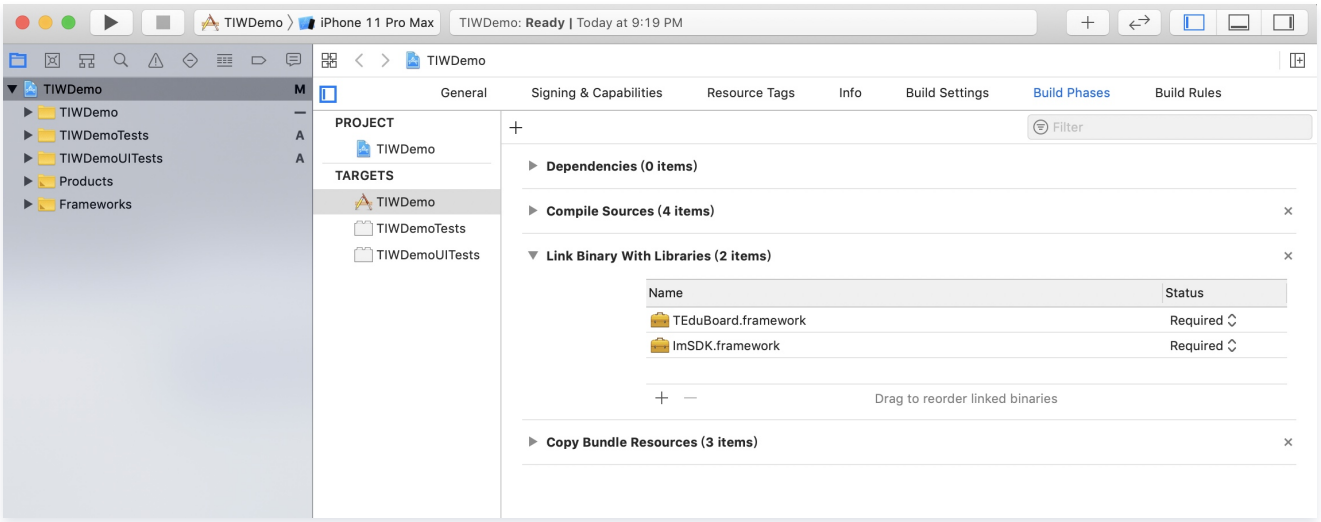
或使用以下命令更新本地库版本：

```
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 onTEBHistroyDataSyncCompleted 回调方法内，使用如下代码获取并显示白板视图。

```
- (void)onTEBHistroyDataSyncCompleted
{
  //（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. onTEBHistroyDataSyncCompleted 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] initSDK: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:classId 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**
> - 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: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。

Mac

最近更新时间：2024-09-11 09:08:11

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，输入以下命令行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
```

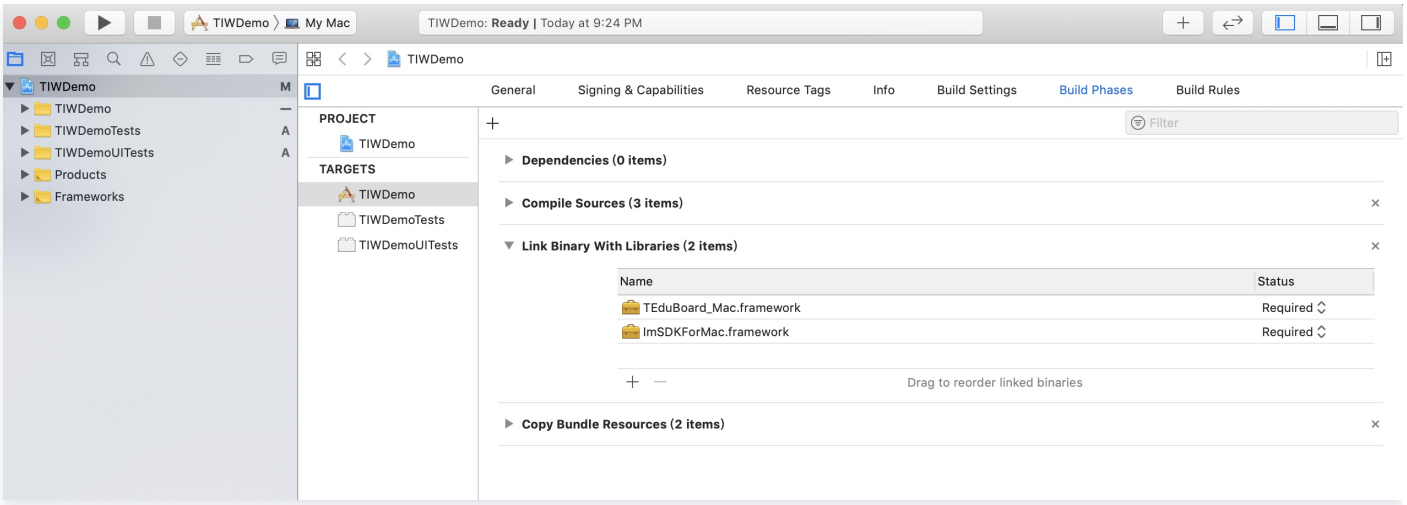
或使用以下命令更新本地库版本：

```
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 `onTEBHistroyDataSyncCompleted` 回调方法内，使用如下代码获取并显示白板视图。

```
- (void)onTEBHistroyDataSyncCompleted
{
  //（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. onTEBHistroyDataSyncCompleted 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] initSDK: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:classId 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。

Web

最近更新时间：2025-07-02 15:58:12

集成 SDK

本文主要介绍如何快速地将腾讯云 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，64位long型
  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：2024-12-13 21:18:52

集成 SDK

本文主要介绍如何快速地将腾讯云 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

> **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	✖	✔	–

版权所有：腾讯云计算（北京）有限责任公司

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，然后请将该桶的访问域名配置到 web-view 的业务域名中。

HarmonyOS

最近更新时间：2025-04-07 17:32:52

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

- DevEco Studio 5.0 Release 及以上
- HarmonyOS SDK API 12 及以上

集成 TEduBoard SDK

通过 har 引用

先将 下载 的 TEduBoard SDK 放在工程或模块（例如：Demo 中的 entry）目录，然后执行以下操作：
在工程或模块的 oh-package.json5 中设置三方包依赖，配置示例如下：

```
{
  dependencies: {
    "@tencentcloud/teduboardsdk": "file:TeduBoardSdk.har",
  },
}
```

依赖设置完成后，需要执行 ohpm install 命令安装依赖包，依赖包会存储在工程或模块的 oh_modules 目录下：

```
ohpm install
```

设置权限

在模块的 module.json5 中，配置示例如下：

```
{
  "module": {
     "requestPermissions": [
      {
        "name": "ohos.permission.GET_NETWORK_INFO",
      },
      {
        "name": "ohos.permission.INTERNET",
      },
    ],
  },
}
```

相关链接

运行 TEduBoard Demo（HarmonyOS）