

# iOS 接入说明文档

# iOS 接入说明文档

接入YXT SDK 首先遵循以下步骤：

- 配置 SDK 集成环境
- 按照以下接入文档集成代码

## SDK 集成环境配置

### 开发与运行环境

- Xcode
- iOS 8.0 及以上版本

## Cocoapods 快速集成

### 1、使用 CocoaPods 导入YXT SDK

1、请参见Cocoapods接入方式 (<https://cocoapods.org/>)

```
1. pod 'YXTSDK' , '~> 1.2'
```

2、开发人员请点击[SDK下载](#)，下载对应的Demo进行体验。

### 2、声明SDK需要的权限

Example for Supporting Files > Y> K-Info.plist > No Selection		
Key	Type	Value
▼ Information Property List	Dictionary	(24 items)
Localization native development region	String	China
Bundle display name	String	
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.8
Bundle creator OS Type code	String	????
▶ URL types	Array	(2 items)
Bundle version	String	1.0
Application Category	String	
▶ LSApplicationQueriesSchemes	Array	(1 item)
Application has localized display name	Boolean	YES
Application requires iPhone environment	Boolean	YES
▶ App Transport Security Settings	Dictionary	(2 items)
Privacy - Camera Usage Description	String	亲，可以访问你的照相机吗？
Privacy - Microphone Usage Description	String	亲，可以访问你的麦克风吗？
Privacy - Photo Library Usage Description	String	亲，可以访问你的图片库吗？
Launch screen interface file base name	String	LaunchScreen
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (iPad)	Array	(4 items)
View controller-based status bar appearance	Boolean	NO

# SDK集成步骤

## 1、AppDelegate中配置代码

```
1. [YXTHelper getInstance].environment = YXTEnvironmentDevelopment;  
2. [[YXTHelper getInstance] appLaunchWithOptions:launchOptions];
```

## 2、打开播放器

```
1. [YXTHelper getInstance].closePreview = NO; // 是否关闭预览界面（YES：关闭，NO：打开。默认NO）  
2. [YXTHelper getInstance].hideNavBar = YES; // 是否显示导航栏（YES：隐藏，NO：显示。默认NO）  
3. [[YXTHelper getInstance] setBeforeViewController:当前界面ViewController];  
4. YXTModel *model = [[YXTModel alloc] init];  
5. model.id = @"文件id";  
6. model.type = @"文件类型";  
7. model.cid = @"课程ID";  
8. model.pid = @"学习计划ID";  
9. model.cid = @"课程ID";  
10. model.genscore = @"是否产生学分记录";  
11. model.allowdown = @"是否允许下载";  
12. model.url = @"优先使用这个url播放 为空，由api获取播放地址";  
13. model.workshopId = @"专题ID";  
14. model.token = @"用户凭据";  
15. model.userId = @"用户唯一标识";  
16. [[YXTHelper getInstance] openPlayWithModel:model];
```

YXTModel 格式参考

## 3、下载配置

```
1. [[YXTHelper getInstance] setBeforeViewController:当前界面ViewController];  
2. [[YXTHelper getInstance] openDownloadList];
```

## 4、回调协议

```
1. [YXTHelper getInstance].delegate = self; // 实现代理，并添加代理 YXTDelegate  
2. // 相关协议 查看 YXTHelper 里 YXTDelegate
```

# Android 接入说明文档

## ANDROID 接入说明文档

接入YXT SDK 首先遵循以下步骤：

- 配置 SDK 集成环境
- 按照以下接入文档集成代码

## SDK 集成环境配置

### 开发与运行环境

- Android Studio
- Android 4.0 及以上版本

### 集成步骤

#### 1. 项目主build中加入

```
1. maven { url 'https://devinner.yunxuetang.com.cn/nexus/content/repositories/releases/' }
```

#### 2. 使用 Maven 导入YXT SDK

```
1. compile "com.yxt.opensdk:yxtsdk:1.2"
```

开发人员请点击[SDK下载](#)，下载对应的Demo进行体验。

#### 3. manifest声明SDK需要的权限与初始化

```
1. <!-- 网络访问 -->
2. <uses-permission android:name="android.permission.INTERNET" />
3. <!-- 获取网络状态 -->
4. <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
5. <!-- 往SDCard写入数据权限 -->
6. <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
7. <!-- 读取手机信息 如串号 序列号等 -->
8. <uses-permission android:name="android.permission.READ_PHONE_STATE" />
9. <!-- 允许应用程序监听、控制、取消呼出电话 -->
10. <uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
11. <!-- WiFi 状态 -->
12. <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
13. <!-- Wifi 改变状态 -->
14. <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
15. <!-- 允许应用访问联系人通讯录信息 -->
16. <uses-permission android:name="android.permission.READ_CONTACTS" />
17. <!-- 写入联系人 -->
18. <uses-permission android:name="android.permission.WRITE_CONTACTS" />
19. <!-- 录音 -->
20. <uses-permission android:name="android.permission.RECORD_AUDIO" />
21. <!-- 允许访问闪光灯 -->
22. <uses-permission android:name="android.permission.FLASHLIGHT" />
23. <!-- application标签内中设置app适配标准设计图尺寸 -->
24. <meta-data
25.     android:name="design_width"
26.     android:value="360dp" />
```

```
25.         android:name="design_width"
26.         android:value="1080" />
27.     <meta-data
28.         android:name="design_height"
29.         android:value="1920" />
```

## 4. Application初始化

```
1.  YXTSdk.init(Application application, boolean isTest); //isTest是否测试环境
```

## 5. 配置token和userId，打开播放器

```
1.  YXTSdk.openPlay(final Context context, final String type, final JSONObject param) // type 知识类型 例如: video
   、 doc等; param 协议JSON对象;
2.
3.  YXTConfig.isJoinPreview = true; //true 有预览页; false 无预览页
```

## 6. 分享回调

```
1.  YXTSdk.initShare(YXTShareListener s);
2.
3.  Example中提供WebViewShowActivity类，其中包含有userAgent、协议拦截使用案例。
```

## 7. 打开下载列表

```
1.  startActivity(new Intent(MainActivity.this, DownloadActivity.class));
```

# H5与APP交互

## H5与APP交互说明文档

交互业务逻辑：

- h5请求播放知识，发起协议请求
- 协议组件，解析协议，事件事件通知APP
- APP播放知识

参数说明：

名称	类型	是否必须	描述
type	string	是	知识类别 视频video/文档doc/炫页xuanye/图片photo/文章article
id	string	是	知识ID
pid	string	是	学习计划ID
cid	string	是	课程ID
type	int	是	1-计划学习 2-岗位学习3-自主学习
genscore	bool	否	是否产生学分记录
allowdown	bool	否	是否允许下载
url	string	是	优先使用这个url播放 为空，由api获取播放地址
workshopId	string	否	专题ID
userId	string	是	用户ID
token	string	是	身份

打开视频播放器示例：

方式1：

```
1. uri = yxtapp: //biz.player.open?param={
2.   "id": "84fc557e-acbc-44b8-a094-f01b6e89a024",
3.   "pid": "",
4.   "cid": "",
5.   "type": "video",
6.   "genscore": true,
7.   "path": "",
8.   "isNew": 1,
9.   "origin": 1}
10. &success=s999&fail=f598, host='biz.player.open',
11. param='{"id": "84fc557e-acbc-44b8-a094-f01b6e89a024", "pid": "", "cid": "", "type": "video", "genscore": true, "path": "", "isNew": 1, "origin": 1}', success='s999', fail='f598', mark='', protocoljson='null'}}
```

方式2：（下面各种知识打开方式2，参考该格式）

```
1. js_callShell: callYunKeTang {
2.   "id": "1389f808-dfd4-4fb6-8cef-62b189d271c4",
3.   "pid": "",
4.   "cid": "",
5.   "type": "video",
6.   "genscore": true,
```

```
7.  "workshopId": "xx",
8.  "userId": "0899ee16-870d-43a8-bd6e-6693359f28e1",
9.  "path": "",
10. "isNew": 1,
11. "origin": 1,
12. "token": "AAAAA0piNwfU1pT6yCtmGUc9tTdfGhZSPf59KZB0Th0ohJRCKyp8IU0TeC_OVW8TG4PhsVV0CAw9cGaHrRh3nWFI7D27-Z7ug3Egc113xqHaR7oLbvHtga4Kvz4vblgtLSY0g_8vaoICcZJPgUodIooJB70"}

```

打开文档示例:

```
1.  uri = yxtapp: //biz.player.open?param=
2.  {"id": "7a256522-c97d-42c1-8c90-4c4110ab3987",
3.  "pid": "",
4.  "cid": "",
5.  "type": "doc",
6.  "path": "",
7.  "isNew": 1,
8.  "origin": 1}
9.  &success=s170&fail=f413, host='biz.player.open',
10. param= {"id": "7a256522-c97d-42c1-8c90-4c4110ab3987", "pid": "", "cid": "", "type": "doc", "path": "", "isNew": 1, "origin": 1}', success='s170', fail='f413', mark='', protocoljson='null'}

```

打开图片示例:

```
1.  uri = yxtapp: //biz.player.open?param={
2.  "id": "383f1203-281d-41a1-a888-5a6830b58463",
3.  "pid": "",
4.  "cid": "",
5.  "type": "photo",
6.  "path": "",
7.  "isNew": 1,
8.  "origin": 1}
9.  &success=s174&fail=f967, host='biz.player.open',
10. param= {"id": "383f1203-281d-41a1-a888-5a6830b58463", "pid": "", "cid": "", "type": "photo", "path": "", "isNew": 1, "origin": 1}', success='s174', fail='f967', mark='', protocoljson='null'}

```

打开微课示例:

```
1.  uri = yxtapp: //biz.player.open?param={
2.  "id": "e73fc87d-d28b-44f7-8e79-9284119e428a",
3.  "pid": "",
4.  "cid": "",
5.  "type": "xuanye",
6.  "path": "",
7.  "origin": 1}
8.  &success=s763&fail=f18, host='biz.player.open',
9.  param= {"id": "e73fc87d-d28b-44f7-8e79-9284119e428a", "pid": "", "cid": "", "type": "xuanye", "path": "", "origin": 1}', success='s763', fail='f18', mark='', protocoljson='null'}

```

打开课程包示例:

```
1.  uri = yxtapp: //biz.player.open?param={
2.  "id": "878d32bb-bad5-4eaa-8dd6-72558f75fb1d",
3.  "pid": "",
4.  "cid": "",
5.  "type": "package",
6.  "path": "",
7.  "upid": "",
8.  "origin": 1}
9.  &success=s605&fail=f938, host='biz.player.open',
10. param= {"id": "878d32bb-bad5-4eaa-8dd6-72558f75fb1d", "pid": "", "cid": "", "type": "package", "path": "", "upid": "", "origin": 1}', success='s605', fail='f938', mark='', protocoljson='null'}

```