

版本更新记录

1.0版本

- 1) SDK的基础框架
- 2) 用户注册、登录登出、获取并验证用户信息
- 3) 支持手机注册、游客模式、游客升级
- 4) 自动检测第一弹APP登录用户、下次自动登录
- 5) 上传用户信息 (enterServer, createRole, levelUp)

接入流程

1. 将所有文件放入项目内，进入 target 的 Build Phases 标签，在 Link Binary With Libraries 下添加依赖库 libsqlite3.0
2. 在 info.plist 文件添加 LSApplicationQueriesSchemes 字段（数组类型），在 LSApplicationQueriesSchemes 目录下添加 diyidanOpenApiV1 字段（字符串）和 diyidanApp 字段（字符串）
3. 在 info.plist 里的 URL Types 项目下，添加一个 URL Schemes 为 dyd76367281，76367281 是第一弹发放的 AppId
4. 在 AppDelegate 里引入头文件 DYDLoginHeader.h
5. 使用 [DYDLoginHeader dyd_initializeWithAppId:@"76367281" appKey:@"SLNsJD9DcgKhZ3y8"]; 开启第一弹登录的工作
使用 [DYDLoginHeader dyd_setCallBackDelegate:self]; 监听第一弹账号的登录和退出，实现 DYDSDKCallBackDelegate 的回调方法
6. 登录部分接口展示，根据需求来实现接入。

```
+ (void)dyd_login;           //调用第一弹登录/注册界面，回调DYDSDKCallBackDelegate
+ (void)dyd_logout;         //注销第一弹账号，回调DYDSDKCallBackDelegate的代理方
+ (DYDSDKUserEntity *)dyd_getCurrentUser;      // 获取玩家的信息
```

关于登录账号结果回调的说明：

一般的登录错误，例如密码或者账号错误，SDK会自行判断。

result 为YES表示登录成功，为NO时是用户取消登录，当登录失败登录界面不会消失，不回调代理方法 类型是单独针对在注册登录界面，按返回键导致登录失败的回调。

这个是必须要实现的接口。

在没有登录成功的情况，不能去获取 token ，如果登录失败的情况下，去获取 token ，会报空指针的异常。

如果需要实现切换账号功能，需要先调用 logout 再去调用 login 。

任何调用注销接口 logout 的地方都要做提示，来告知用户账号即将被注销。

7. 用户验证

```
String getToken(long duration) //获取当前用户的Token 字符串
```

如果需要验证用户的有效性，请在客户端请求此接口，然后使用得到的 token 字符串去服务端验证。

Token的服务器端验证

接口：<https://gameapi.diyidan.net/api/uid/check>
(<https://gameapi.diyidan.net/api/uid/check>)

方法：GET，大小写敏感如果 token 有效，则返回字符串 true ，无效则返回 false
编码：UTF-8
注意：对于 token 长度的限制不能低于800字符

参数名	内容
userid	登录账户 uid
token	登录账户验证信息
app_id	游戏app_id,数字类型

通过 dyd_login 的回调 dyd_loginFinishWithResult:userE: 中的 DYDSDKUserEntity 来获取登录成功用户的 userId 和 token 。

8. 信息上传接口

```

/**
 * 上传角色信息
 * @param roleId 当前登录的玩家角色ID, 若无, 可传入userid
 * @param roleName 当前登录的玩家角色名, 不能为空, 不能为null, 若无, 传入"游戏名称+use
 * @param roleLevel 当前登录的玩家角色等级, 且不能为0, 若无, 传入1
 * @param roleZoneId 当前登录的游戏区服ID, 且不能为0, 若无, 传入1
 * @param roleZoneName 当前登录的游戏区服名称, 不能为空, 不能为null, 若无, 传入游戏名称
 * @param roleBalance 当前用户游戏币余额, 若无, 传入0
 * @param roleVipLevel 当前用户VIP等级, 若无, 传入0
 * @param rolePartyName 当前用户所属帮派, 不能为空, 不能为null, 若无, 传入"无帮派"
 * @param roleScene 当前场景: enterServer, levelUp, createRole
 * @param roleExp 当前登录的玩家角色经验值, 最小值为0
 */
+ (void)dyd_uploadRoleDataWithRoleId:(NSString *)roleId
                      roleName:(NSString *)roleName
                      roleLevel:(NSString *)roleLevel
                      roleZoneId:(NSString *)roleZoneId
                      roleZoneName:(NSString *)roleZoneName
                      roleBalance:(NSString *)roleBalance
                      roleVipLevel:(NSString *)roleVipLevel
                      rolePartyName:(NSString *)rolePartyName
                      roleScene:(NSString *)roleScene
                      roleExp:(NSString *)roleExp
                      success:(void(^)( ))success
                      failure:(void(^)( ))failure;

```

根据当前情景, 来选择 `scene` 这个字段对应的参数, 分别支持 **enterServer** , **levelUp** , **createRole** 这三种状态, 它们分别表示进入服务区、等级提升、创建角色事件。

以上上参数均不能填写 ``null``, 如果实在没有, 可以填写虚拟数据,

/ **

* 上传用户闯关/对战信息

* @param roleId 当前登录的玩家角色ID, 若无, 可传入userid

* @param zoneId 当前登录的游戏区服ID, 必须为数字, 且不能为0, 若无, 传入1

* @param zoneName 当前登录的游戏区服名称, 不能为空, 不能为null, 若无, 传入游戏名称+"1"

* @param mode 闯关模式，直接传模式名。常见的有"普通模式"、"精英模式"。若无不同模式的区分

* @param stage 关卡名, 以 "章节名+关卡数字" 表示。如, "黑暗森林5-3"。 如果为对战, 则传

* @param battleNo 唯一标识此次闯关/对战的编号，不能为空

* @param starNumber 所获星级，标识此次闯关表现，以"star*数字"表示。如 "star*0" 表

* @param isSuccess 是否闯关成功, true表示成功, false表示失败

* @param exp 此次闯关所获经验值，无经验则为0，扣经验则为负

* @param gains 此次闯关掉落/损失的物品, "物品名+数字,物品名-数字"来表示, 中间用英文逗号

*/

```
+ (void)dyd_uploadBattleDataWithRoleId:(NSString *)roleId  
    stage:(NSString *)stage  
    zoneId:(NSString *)zoneId  
    zoneName:(NSString *)zoneName  
    battleNo:(NSString *)battleNo  
    starNumber:(NSString *)starNumber  
    isSuccess:(NSString *)isSuccess  
    mode:(NSString *)mode  
    exp:(NSString *)exp  
    gains:(NSString *)gains  
    success:(void(^)( ))success  
    failure:(void(^)( ))failure;
```