

ADok

راهنمای استفاده در نرم افزار Unity3D

فهرست

[شروع کار](#) ▶

[Sign in / Sign Up](#) ▶

[کاربر مهمان](#) ▶

[Sign In](#) ▶

[Sign Up](#) ▶

[Forget Password](#) ▶

[Change Password](#) ▶

[Change Name](#) ▶

[کد تایید](#) ▶

[کد معرف](#) ▶

[خروج از حساب کاربری](#) ▶

[تصویر آواتار](#) ▶

[کار با داده ها](#) ▶

[ذخیره اطلاعات](#) ▶

[دریافت اطلاعات](#) ▶

[بروزرسانی، اطلاعات کاربر از سرور](#) ▶

[بروز رسانی، اطلاعات کاربر در سرور](#) ▶

[دریافت لیدرورد کلی](#) ▶

[دریافت لیدرورد من](#) ▶

[لیگ ها](#) ▶

[دریافت لیگ ها](#) ▶

[عضویت](#) ▶

[ذخیره اطلاعات](#) ▶

[دریافت اطلاعات](#) ▶

[دریافت لیدرورد](#) ▶

[دریافت حوایز](#) ▶

[نمایش، تبلیغات ویدئویی](#) ▶

[توابع کاربردی](#) ▶

[دریافت تاریخ](#) ▶

[دریافت ساعت](#) ▶

[غیره](#) ▶

شروع کار

1. نرم افزار یا بازی خود را در سایت ثبت کنید و کد API را دریافت کنید
2. پکیج یونیتی را از سایت دانلود کنید و در یونیتی Import کنید
3. از مسیر Adok->Prefab ، Adok ، را به داخل Scene خود درگ کنید.
4. از Hierarchy ، Adok را انتخاب کنید و در قسمت ApiKey در Inspector ، کد دریافتی از سایت را قرار دهید. توجه کنید که کد بر اساس نام پکیج شما ایجاد می شود. در صورتی که نام پکیج خود را تغییر دادید حتما از پنل خود کد جدید دریافت کنید.
5. حال شما می توانید از تمام قابلیت های Adok استفاده کنید.
6. برای چک کردن دسترسی ها می توانید از توابع isReadStoragePermissionGranted و isWriteStoragePermissionGranted و isPhoneStatePermissionGranted و isLocationPermissionGranted و isCameraPermissionGranted استفاده کنید
7. برای درخواست دسترسی ها از توابع RequestReadStoragePermission و RequestWriteStoragePermission و RequestPhoneStatePermission و RequestLocationPermission و RequestCameraPermission استفاده کنید.

Sign In / Sign Up

▶ سامانه ادوک تمام امکانات مورد نیاز برای ذخیره کاربران به صورت ثبت نامی و مهمان را دارا می باشد.

1. کارربر مهمان
2. ثبت نام
3. ورود
4. فراموشی رمز عبور
5. تغییر رمز عبور
6. تغییر نام مستعار
7. کد معرف
8. خروج از حساب کاربری
9. تنظیمات تصویر آواتار

کاربر مهمان

- ▶ بعد از اولین ورود به بازی در صورتی که کاربر ثبت نام انجام نداده باشد به صورت کاربر مهمان ذخیره می شود و نام آن به صورت `Player_123456` ذخیره میشود. برای اینکه متوجه بشویم کاربر ثبت نام شده یا نه از دستور زیر استفاده می کنیم:
- ▶ `adok.CheckLogin()`
- ▶ خروجی تابع از نوع `boolean` می باشد و در صورتی که مقدار آن `true` باشد نیازی به ثبت نام و یا ورود ندارد
- ▶ توجه داشته باشید که اطلاعات بازیکن بین بازی های مختلف یکسان است.

Sign in / Register

- ▶ برای ثبت نام کاربر از توابع زیر استفاده می کنیم
- ▶ `adok.RegisterPlayer`(سوال امنیتی , تکرار رمز عبور , رمز عبور , نام کاربری);
 - ▶ `adok.RegisterPlayer`(شماره تماس);
 - ▶ برای دریافت سوال امنیتی از تابع زیر استفاده می کنیم.
- ▶ `adok.getSecurityQuestion()`
- ▶ سیستم ادوک شامل 5 سوال امنیتی فارسی می باشد.
- ▶ در صفحه بعد با نحوه مدیریت روند ثبت نام آشنا می شوید.

Sign in / Register

- ▶ روند انجام ثبت نام را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:
- ▶ `adok.OnRegisterChange += Adok_OnRegisterChange;`
- ▶ تابع رویداد دارای ورودی از نوع `ADokManager.RegisterResult` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد
2. `Sucsses`: عملیات با موفقیت انجام شد
3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد
4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.
5. `Exsist`: کاربر با این شماره تماس و یا نام کاربری در سیستم وجود دارد
6. `UserNameProblem`: مشکل در نام کاربری وارد شده
7. `PassProblem`: مشکل در رمز عبود وارد شده
8. `PhoneNoProblem`: مشکل در شماره تماس وارد شده
9. `NickNameProblem`: مشکل در نام مستعار وارد شده
10. `alreadyRegistered`: کاربر در گذشته ثبت نام کرده و نیازی به ثبت نام نیست.

Sign in / Register

- ▶ در صورت ثبت نام با شماره تماس باید مرحله چک کردن کد تایید نیز انجام شود که به صورت زیر می باشد.
- ▶ `adok.CheckCodeTaied(کد تایید);`
- ▶ با استفاده از رویداد زیر می توانیم روند انجام را کنترل کنیم
- ▶ `adok.OnCodeTaiedChange += Adok_OnCodeTaiedChange;`
- ▶ در صورت نیاز به ارسال مجدد کد تایید از تابع زیر استفاده می کنیم
- ▶ `adok.ResendCodeTaied();`
- ▶ توجه داشته باشید که هر 60 ثانیه 1 بار می توان کد تایید مجدد دریافت کرد

Sign up / Login

- ▶ برای ورود کاربر از توابع زیر استفاده می کنیم
- ▶ `adok.LoginUser(نام کاربری, رمز عبور)`;
- ▶ `adok.LoginUser(شماره تماس)`;
- ▶ در صفحه بعد با نحوه مدیریت روند ورود آشنا می شوید.

Sign up / Login

▶ روند انجام ورود را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnLoginChange += Adok_OnLoginChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.LoginResult` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Succses`: عملیات با موفقیت انجام شد

.3 `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.

.5 `notExist`: کاربر با این نام کاربری و رمز عبور وجود ندارد

.6 `UserNameProblem`: مشکل در نام کاربری وارد شده

.7 `PassProblem`: مشکل در رمز عبور وارد شده

.8 `alreadyRegistered`: کاربر در گذشته ورود کرده و نیازی به ثبت نام نیست.

Sign up / Login

▶ در صورت ورود با شماره تماس باید مرحله چک کردن کد تایید نیز انجام شود که به صورت زیر می باشد.

▶ `adok. CheckCodeTaiedLogin(کد تایید);`

▶ با استفاده از رویداد زیر می توانیم روند انجام را کنترل کنیم

▶ `adok.OnCodeTaiedChange += Adok_OnCodeTaiedChange;`

▶ در صورت نیاز به ارسال مجدد کد تایید از تابع زیر استفاده می کنیم

▶ `adok.ResendCodeTaied();`

▶ توجه داشته باشید که هر 60 ثانیه 1 بار می توان کد تایید مجدد دریافت کرد

Forget Password

- ▶ برای فراموشی رمز عبور از توابع زیر استفاده می کنیم
- ▶ `adok.ForgetPassAuth` (پاسخ سوال امنیتی , نام کاربری);
- ▶ `adok.GetPassWithPhone` (شماره تماس);
- ▶ در حالت دوم در صورت وجود داشتن کاربر رمز عبور به شماره پیامک او ارسال خواهد شد
- ▶ در صفحه بعد با نحوه مدیریت روند ورود آشنا می شوید.

Forget Password

▶ روند انجام ورود را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnForgetPassAuthChange += Adok_OnForgetPassAuthChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.ForgetPasswordResult` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Succses`: عملیات با موفقیت انجام شد

.3 `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.

.5 `notExist`: کاربر با این نام کاربری وجود ندارد

.6 `UserNameProblem`: مشکل در نام کاربری وارد شده

Forget Password

▶ در صورت ورود تایید سوال امنیتی با استفاده از تابع زیر رمز عبور را تغییر می دهیم.

▶ `adok.SetPass(رمز عبور);`

▶ با استفاده از رویداد زیر روند انجام کار را مدیریت می کنیم

▶ `adok.OnSetPassChange += Adok_OnSetPassChange;`

▶ برای دریافت سوال امنیتی از تابع زیر استفاده می کنیم.

▶ `adok.getSecurityQuestion()`

▶ سیستم ادوک شامل 5 سوال امنیتی فارسی می باشد.

تغییر رمز عبور

- ▶ برای تغییر رمز عبور از تابع زیر استفاده می کنیم
- ▶ `adok.ChangePassword`(رمز عبور قدیمی, رمز عبور جدید, تکرار رمز عبور جدید);
- ▶ در صفحه بعد با نحوه مدیریت روند ورود آشنا می شوید.

تغییر رمز عبور

▶ روند انجام تغییر رمز عبور را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnChangePass += Adok_OnChangePass;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.ChangePassResult` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Sucsses`: عملیات با موفقیت انجام شد

.3 `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.

.5 `NotEqual`: رمز عبور وارد شده با تکرار آن برابر نیست

.6 `passIncorrect`: رمز عبور وارد شده نا معتبر است

تغییر نام مستعار

▶ برای تغییر نام مستعار از تابع زیر استفاده می کنیم

▶ `adok.ChangeNickName(نام مستعار جدید);`

▶ در صفحه بعد با نحوه مدیریت روند ورود آشنا می شوید.

تغییر نام مستعار

▶ روند انجام تغییر نام مستعار را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

```
adok.OnChangeNickName += Adok_OnChangeNickName; ▶
```

▶ تابع رویداد دارای ورودی از نوع `ADokManager.ChangeNickNameResult` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Succses`: عملیات با موفقیت انجام شد

3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.

5. `emptyNickName`: خالی بودن مقدار نام مستعار

کد تایید

▶ در ورود و یا ثبت نام کاربر شما می توانید شماره تماس شخص را با دریافت پیامک تایید کنید. در هر دو حالت از تابع زیر برای چک کردن کد تایید وارد شده استفاده می کنیم

▶ `adok.CheckCodeTaied`(کد تایید وارد شده)

▶ خروجی تابع به صورت `boolean` بوده و در صورت `true` بودن مقدار آن کد تایید صحیح وارد شده است.

▶ برای دریافت مجدد کد تایید از تابع زیر استفاده می کنیم:

▶ `adok.ResendCodeTaied();`

▶ در صفحه بعد با نحوه مدیریت روند ورود آشنا می شوید.

کد تایید

- ▶ روند انجام ورود را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:
 - ▶ `adok.OnResendCodeTaiedChange += Adok_OnResendCodeTaiedChange;`
 - ▶ تابع رویداد دارای ورودی از نوع `ADokManager.CodeTaiedResult` می باشد که شامل مقادیر زیر است
1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد
 2. `Sucsses`: عملیات با موفقیت انجام شد
 3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد
 4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لو دینگ وجود دارد.
 5. `notExist`: کاربر با این نام کاربری وجود ندارد

کد معرف

▶ برای استفاده از قابلیت های کد معرف از توابع زیر استفاده می کنیم:

▶ دریافت کد معرفی خود برای ارسال به دیگران

▶ `adok.GetCodeMoaref()`

▶ تعداد اشخاصی که از کد معرفی کاربر استفاده کرده اند و کاربر هنوز بابت آنها جایزه دریافت نکرده است

▶ `adok.GetMoarefiCount()`

▶ تعداد کل اشخاصی که از کد معرفی کاربر استفاده کرده اند

▶ `adok.GetMoarefiCountAll()`

▶ دریافت جایزه معرفی دیگران

▶ `adok.GetMoarefiSetReward();`

▶ ثبت کردن کد معرف

▶ `adok.SetCodeMoaref(کد معرف)`

کد معرف

▶ برای مدیریت روند انجام توابع از رویداد های زیر کمک می گیریم

- ▶ `adok.OnCodeMoarefChange += Adok_OnCodeMoarefChange;`
- ▶ `adok.OnMoarefiCountChange += Adok_OnMoarefiCountChange;`
- ▶ `adok.OnMoarefiCountAllChange += Adok_OnMoarefiCountAllChange;`
- ▶ `adok.OnMoarefiSetRewardChange += Adok_OnMoarefiSetRewardChange;`

▶ از سه تابع زیر برای دریافت تعداد معرفی استفاده میکنیم. توجه داشته باشید که در صورت `sucessess` بودن رویداد های بالا از این توابع استفاده کنید.

- ▶ `adok.GetMoreafiCountVal()`
- ▶ `adok.GetMoreafiCountValAll()`
- ▶ `adok.GetMoreafiSetRewardCount()`

خروج از حساب کاربری

- ▶ برای خروج از حساب کاربری از تابع زیر استفاده می کنیم
- ▶ از رویداد زیر برای مدیریت روند انجام آن استفاده می کنیم
- ▶ `adok.OnLogOutChange += Adok_OnLogOutChange;`
- ▶ در صورت `success` شدن رویداد بالا کاربر به عنوان کاربر مهمان در نظر گرفته می شود.

تصویر آواتار

- ▶ برای استفاده از تصویر آواتار از سه تابع زیر استفاده می کنیم
 - ▶ دریافت تصویر آواتار از طریق دوربین
 - ▶ دریافت تصویر آواتار از طریق گالری
 - ▶ نمایش تصویر آواتار
- ▶ `adok.AvatarCaptureImage()`
 - ▶ `adok.AvatarChooseImage()`
 - ▶ `Texture2D avatar = adok.GetAvatarImage();`

ذخیره اطلاعات

- ▶ برای ذخیره اطلاعات نیازی به ورود و ثبت نام نیست و کاربر به صورت مهمان نیز می تواند اطلاعات خود را ذخیره کند، اما نمی تواند آن را بازیابی کند.
- ▶ برای ذخیره اطلاعات از تابع زیر استفاده می کنیم:
- ▶ `adok.SetData(نوع فیلد, مقدار فیلد, نام فیلد)`;
- ▶ نوع فیلد شامل موارد زیر است:
- ▶ `ADokManager.SyncType.Local`: برای فیلدهایی که نیازی به ذخیره در سرور ندارند و قابل بازیابی نیستند
- ▶ `ADokManager.SyncType.Sync`: برای فیلدهایی که در پنل تعریف شده اند و قابل بازیابی هستند

ذخیره اطلاعات

- ▶ برای ایجاد و مدیریت فیلد ها از پنل خود در قسمت ذخیره اطلاعات وارد صفحه فیلد ها بشوید
- ▶ بازی خود را انتخاب کنید
- ▶ تعدادی فیلد به صورت پیش فرض ایجاد شده است که می توانید از آنها استفاده کنید و یا با زدن دکمه اضافه فیلد جدید ایجاد کنید.
- ▶ توجه داشته باشید که می توانید فیلد های خود را برای امنیت بیشتر رمزگذاری کنید، اما فیلد های رمزگذاری شده قابلیت مرتب سازی ندارند.

دریافت اطلاعات

► برای دریافت اطلاعات از تابع زیر استفاده می کنیم:

► `adok.GetData(نام فیلد);`

بروز رسانی اطلاعات کاربر از سرور

▶ برای دریافت اطلاعات کاربر از سرور ذخیره آن از تابع زیر استفاده می کنیم:

▶ `adok.SynLastDataFromServer();`

▶ روند انجام دریافت اطلاعات را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnSyncGetChange += Adok_OnSyncGetChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.SyncResults` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Sucsses`: عملیات با موفقیت انجام شد

3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

5. `NotRegistered`: کاربر چه به صورت مهمان و یا کاربر نام دار ذخیره نشده است.

ذخیره اطلاعات کاربر در سرور

▶ برای ذخیره اطلاعات کاربر در سرور از تابع زیر استفاده می کنیم:

▶ `adok.SyncNow();`

▶ روند انجام دریافت اطلاعات را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnSyncChange += Adok_OnSyncChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.SyncResults` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Succses`: عملیات با موفقیت انجام شد

.3 `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

.5 `NotRegistered`: کاربر چه به صورت مهمان و یا کاربر نام دار ذخیره نشده است.

توجه داشته باشید اطلاعات کاربر را بعد از ذخیره بروزرسانی کنید. به عنوان مثال بعد از تغییر دادن مقدار سکه و جم و ... تابع فوق را فراخوانی کنید

دریافت لیدربرد کلی از سرور

- ▶ برای دریافت لیدربرد کلی از سرور از تابع زیر استفاده می کنیم:
- ▶ `adok.GetLeaderBoard`(نام فیلد برای مرتب سازی ,محدوده زمانی);
- ▶ نام فیلد باید از فیلدهایی باشد که در پنل شما در قسمت اطلاعات وجود داشته باشد و رمز گذاری نشده باشد
- ▶ محدوده زمانی شامل موارد زیر است:
- ▶ .1 `ADokManager.TimePeriod.all`: همیشه- از ابتدای ایجاد بازی بر روی سرور
- ▶ .2 `ADokManager.TimePeriod.day`: امروز
- ▶ .3 `ADokManager.TimePeriod.week`: هفته جاری
- ▶ .4 `ADokManager.TimePeriod.mounth`: ماه جاری

دریافت لیدربرد کلی از سرور

▶ روند انجام دریافت لیدربرد کلی را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnLeaderChange += Adok_OnLeaderChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.LeaderResults` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Sucsses`: عملیات با موفقیت انجام شد

3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لو دینگ وجود دارد.

در صورت موفقیت آمیز بودن عملیات می تواند نتیجه لیدربرد که به صورت `json` می باشد را از متغیر `adok.leaderText` دریافت کنید.

فرمت `json` به صورت زیر می باشد:

```
{[rank:# , playerName:# , [مقادیر:فیلدهای ایجاد شده توسط کاربر , [....],[....]]]}
```

دریافت لیدربرد کلی از سرور

- ▶ بعد از اولین دریافت لیدربرد از توابع زیر جهت پیمایش لیدر برد می توان استفاده کرد:
- ▶ `adok.GetLeaderBoardNextPage()`: رفتن به 10 نفر بعدی (هر صفحه شامل 10 نفر می باشد)
- ▶ `GetDataLeaderboardPre()`: رفتن به 10 نفر قبلی(هر صفحه شامل 10 نفر می باشد)
- ▶ `adok. GetDataLeaderboardExact()`: رفتن به صفحه مورد نظر (هر صفحه شامل 10 نفر می باشد)
- ▶ همانند دریافت لیدربرد هر سه تابع بالا از رویداد زیر استفاده می کنند.
- ▶ `adok.OnLeaderChange += Adok_OnLeaderChange;`

دریافت لیدربرد من از سرور

- ▶ برای دریافت لیدربرد من از سرور از تابع زیر استفاده می کنیم:
- ▶ `adok.GetMyLeaderBoard`(نام فیلد برای مرتب سازی ,محدوده زمانی);
- ▶ لیدربرد من شامل 5 نفر بالاتر از من و من و 5 نفر پایین تر از من می باشد.
- ▶ نام فیلد باید از فیلدهایی باشد که در پنل شما در قسمت اطلاعات وجود داشته باشد و رمز گذاری نشده باشد
- ▶ محدوده زمانی شامل موارد زیر است:
- ▶ .1 `ADokManager.TimePeriod.all`: همیشه- از ابتدای ایجاد بازی بر روی سرور
- ▶ .2 `ADokManager.TimePeriod.day`: امروز
- ▶ .3 `ADokManager.TimePeriod.week`: هفته جاری
- ▶ .4 `ADokManager.TimePeriod.mounth`: ماه جاری

دریافت لیدربرد من از سرور

▶ روند انجام دریافت لیدربرد من را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnMyLeaderChange += Adok_OnMyLeaderChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.LeaderResults` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Sucsses`: عملیات با موفقیت انجام شد

3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لو دینگ وجود دارد.

در صورت موفقیت آمیز بودن عملیات می تواند نتیجه لیدربرد که به صورت `json` می باشد را از متغیر `adok.myleaderText` دریافت کنید.

فرمت `json` به صورت زیر می باشد:

```
{[rank:# , playerName:# , [مقادیر:فیلدهای ایجاد شده توسط کاربر , [....],[....]]}
```

دریافت لیگ های بازی از سرور

Adok. GetLeague(type) ►

Type مقادیر 0 و 1 و 2 و 3 برای دریافت لیگ های فعال و بزودی و غیر فعال و همه استفاده می شود. ►

روند انجام دریافت لیگ های بازی از سرور را با تعریف رویداد به شکل زیر می توانید مشاهده کنید: ►

```
adok.OnGetLeagues += Adok_OnGetLeagues; ►
```

تابع رویداد دارای ورودی از نوع ADokManager.GetLeagueResult می باشد که شامل مقادیر زیر است ►

1. NotInitialized: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن Apikey باشد

2. Succses: عملیات با موفقیت انجام شد

3. Fail: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. Loading: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

5. noLeagueFound: لیگی وجود ندارد

6. noAppFound: چنین بازی وجود ندارد

7. FailedToLoad: خطا در لود اطلاعات

در صورت موفقیت آمیز بودن عملیات می تواند نتیجه را در adok.MyLeagues مشاهده و استفاده کنید که نوع آن json می باشد و برای سهولت کار آماده و نیازی به parse ندارد.

دریافت لیگ های بازی از سرور

- ▶ adok.MyLeagues شامل سه نوع داده می باشد:
- ▶ Active: آرایه که در آن لیگ های فعال قرار دارد
- ▶ comingSoon: آرایه ای که در آن لیگ های آینده قرار دارند. (هنوز شروع نشده اند)
- ▶ Ended: آرایه ای که لیگ های به پایان رسیده در آن قرار دارند.

- ▶ بدوم در نظر گرفتن نوع لیگ هر لیگ شامل اطلاعات زیر می باشد:
- ▶ Name: نام لیگ
- ▶ Des: توضیحات لیگ
- ▶ Id: شناسه لیگ
- ▶ myCount: تعداد انجام شده لیگ (در لیگ های اتوماتیک)
- ▶ Join: آرایه ای از عضویت در لیگ که شامل تاریخ و ساعت عضویت می باشد
- ▶ Rewards: آرایه ای از جوایز تعیین شده در پنل

عضویت در لیگ

adok.JoinLeagues(leagueId); ▶

با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد: ▶

adok.OnGetJoinLeagues += Adok_OnGetJoinLeagues; ▶

تابع رویداد دارای ورودی از نوع ADokManager.GetJoinLeagueResult می باشد که شامل مقادیر زیر است ▶

.1:NotInitialized: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن Apikey باشد

.2:Sucsses: عملیات با موفقیت انجام شد

.3:Fail: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4:Loading: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.

.5:noLeagueFound: لیگی وجود ندارد

.6:AlreadyJoined: قبلا عضو شده است

.7:LeagueExpired: مدت زمان لیگ به پایان رسیده است و یا سقف آن پر شده است

دریافت اطلاعات بازیکن در لیگ

`:adok.GetLeagueDatas(leagueld, myCount);` ▶

▶ با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد:

`adok.OnGetSavedLeagueDatas += Adok_OnGetSavedLeagueDatas;` ▶

▶ تابع رویداد دارای ورودی از نوع `ADokManager.GetSavedLeagueDataResult` می باشد که شامل مقادیر زیر است

.1 `:NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `:Sucsses`: عملیات با موفقیت انجام شد

.3 `:Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `:Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لو دینگ وجود دارد.

.5 `:noLeagueFound`: لیگی وجود ندارد

.6 `:NoData`: اطلاعاتی وجود ندارد

ذخیره اطلاعات بازیکن در لیگ

▶ `adok.SetDataLeague(مقدار, نام فیلد, leagueId, true);`

▶ با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد:

▶ `adok.OnSyncLeagueChange += Adok_OnSyncLeagueChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.SyncLeagueResults` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Sucsses`: عملیات با موفقیت انجام شد

.3 `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

.5 `LeagueExpired`: مدت زمان لیگ به پایان رسیده است

دریافت لیدربرد لیگ

`adok.GetLeagueLeaderBoard(leagueld, myCount);` ▶

▶ با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد:

`adok.OnLeaderLeagueChange += Adok_OnLeaderLeagueChange;` ▶

▶ تابع رویداد دارای ورودی از نوع `ADokManager.LeaderResults` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Succses`: عملیات با موفقیت انجام شد

3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

در صورت موفقیت آمیز بودن عملیات می تواند نتیجه لیدربرد که به صورت `json` می باشد را از متغیر `adok.LeaderLeagueJsonText` دریافت کنید.

فرمت `json` به صورت زیر می باشد:

```
{[rank:# , playerName:# , [مقادیر:فیلدهای ایجاد شده توسط کاربر , [....],[....]]]}
```


دریافت جوایز لیگ

▶ `adok.GetMyRewards(leagueld);`

▶ با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد:

▶ `adok.OnRewardChange += Adok_OnRewardChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.RewardResults` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Sucsses`: عملیات با موفقیت انجام شد

.3 `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لوودینگ وجود دارد.

در صورت موفقیت آمیز بودن عملیات می تواند نتیجه را در `adok.rewardsJson.result` مشاهده و استفاده کنید که نوع آن `json` می باشد و برای سهولت کار آماده و نیازی به `parse` ندارد.

دریافت جوایز لیگ

- ▶ نتیجه دریافتی آرایه ای از اطلاعات زیر است
- ▶ Id: شناسه جایزه
- ▶ Name: نام جایزه
- ▶ Des: توضیحات
- ▶ logoAdd: آدرس تصویر لوگو
- ▶ rewardName: نام جایزه جهت نمایش
- ▶ myCount: تعداد
- ▶ Kind: نوع جایزه - داخل بازی یا فیزیکی
- ▶ FName: اسم فیلد در صورتی که جایزه داخلی باشد
- ▶ leagueCount: شمارنده لیگ برای لیگ ها بی پایان مثل هفتگی یا ماهانه یا ...

دریافت جوایز لیگ

▶ برای دریافت جایزه از تابع `myLeague.adok.SetReward(RewardId, myLeague.leagueld, leagueCount)` استفاده می کنیم.

▶ با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد:

▶ `myLeague.adok.OnSetRewardChange += Adok_OnSetRewardChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.SetRewardResults` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Sucsses`: عملیات با موفقیت انجام شد

3. `Fail`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

5. `GetBefore`: قبلا دریافت شده است

در صورت موفقیت آمیز بودن و داخلی بودن جایزه تابع

`adok.SetData(flname, مقدار جایزه, ADokManager.SyncType.Sync);`

را استفاده می کنیم.

نمایش تبلیغات ویدئویی

▶ `:adok.ShowAdd()`

▶ با استفاده از رویداد زیر می توان نتیجه را مشاهده کرد:

▶ `adok.OnVideoAdvShowEnd += Adok_OnVideoAdvShowEnd;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.AdvsResults` می باشد که شامل مقادیر زیر است

.1 `:NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `:ShowComplete`: عملیات با موفقیت انجام شد

.3 `:Error`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `:process`: در حال لود اطلاعات. در این مرحله امکان نمایش لو دینگ وجود دارد.

.5 `:ShowCompleteAndClicked`: نمایش کامل شد و بر روی دکمه تبلیغ کلیک شد

.6 `:Skipped`: تبلیغ اسکیپ شد و کامل نمایش داده نشد

دریافت تاریخ از سرور

▶ برای دریافت تاریخ از سرور از تابع زیر استفاده می کنیم:

▶ `adok. GetDate();`

▶ روند انجام دریافت اطلاعات را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnGetDateChange += Adok_OnGetDateChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.GetDateResults` می باشد که شامل مقادیر زیر است

.1 `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

.2 `Sucsses`: عملیات با موفقیت انجام شد

.3 `Error`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

.4 `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

در صورت موفقیت آمیز بودن عملیات می توانید مقدار تاریخ امروز را از متغیر `adok.curDate` بخوانید. فرمت تاریخ به صورت `yyyymmdd` می باشد.

دریافت ساعت از سرور

▶ برای دریافت ساعت از سرور از تابع زیر استفاده می کنیم:

▶ `adok. GetTime();`

▶ روند انجام دریافت اطلاعات را با تعریف رویداد به شکل زیر می توانید مشاهده کنید:

▶ `adok.OnGetTimeChange += Adok_OnGetTimeChange;`

▶ تابع رویداد دارای ورودی از نوع `ADokManager.GetTimeResults` می باشد که شامل مقادیر زیر است

1. `NotInitialized`: پلاگین ادوک آماده استفاده نمی باشد. ممکن است به دلیل صحیح نبودن `Apikey` باشد

2. `Sucsses`: عملیات با موفقیت انجام شد

3. `Error`: عملیات انجام نشد. اشکال در اتصال به سرور بیشترین دلیل برای این خروجی می باشد

4. `Loading`: در حال لود اطلاعات. در این مرحله امکان نمایش لودینگ وجود دارد.

در صورت موفقیت آمیز بودن عملیات می توانید مقدار تاریخ امروز را از متغیر `adok.curTime` بخوانید. فرمت ساعت به صورت `hhmm` و به صورت `24` ساعته می باشد.

توابع کاربردی

- ▶ `adok.GetUserId()`: دریافت شناسه کاربر
- ▶ `adok.GetNickName()`: دریافت نام مستعار کاربر
- ▶ `adok.GetUserName()`: دریافت نام کاربری
- ▶ `adok.GetPhoneNo()`: دریافت شماره تماس
- ▶ `adok.GetDeviceUniqueld()`: دریافت کد منحصر به فرد دستگاه

پایان

- ▶ برای سهولت در استفاده حتما از مثال موجود در مسیر ادوک استفاده کنید
- ▶ نظرات و پیشنهادات و سوالات خود را از طریق سایت [Adok](#) با ما در میان بگذارید
- ▶ از اعتماد شما سپاس گزاریم.