

ADok

راهنمای استفاده در نرم افزار B4A

فهرست

- ▶ [شروع کار](#)
- ▶ [Sign in / Sign Up](#)
- ▶ [کاربر مهمان](#)
- ▶ [Sign In](#)
- ▶ [Sign Up](#)
- ▶ [Forget Password](#)
- ▶ [Change Password](#)
- ▶ [Change Name](#)
- ▶ [کد تایید](#)
- ▶ **کار با داده ها**
- ▶ [ذخیره اطلاعات](#)
- ▶ [دریافت اطلاعات](#)
- ▶ [بروزرسانی اطلاعات کاربر از سرور](#)
- ▶ [بروز رسانی اطلاعات کاربر در سرور](#)
- ▶ [دریافت لیدریورد کلی](#)
- ▶ [دریافت لیدریورد من](#)
- ▶ **لیگ ها**
- ▶ [دریافت لیگ ها](#)
- ▶ [عضویت](#)
- ▶ [ذخیره اطلاعات](#)
- ▶ [دریافت اطلاعات](#)
- ▶ [دریافت لیدر یورد](#)
- ▶ [دریافت جوایز](#)
- ▶ [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. فایل کتابخانه B4A را دانلود کنید و از حالت فشرده خارج کنید. سپس محتویات پوشه را در محل کتابخانه های خود قرار دهید.
 3. بعد از بروزرسانی کتابخانه ها، کتابخانه Adok و OkHttp را انتخاب کنید.
 4. #AdditionalJar: ADok.aar را به Main اضافه کنید.
 5. Dim ADok As Adok را در Process_Globals خود تعریف کنید.
 6. با استفاده از (ADok.Initialize(APIKEY,Oriantation,canSkipAdvertisement,Tag,Current Activity) ، Adok را آماده استفاده کنید.
- Oriantation: مقادیر برای حالت عمودی و In برای حالت افقی.
- canSkipAdvertisement: اگر مقدار true را بدهیم، کاربر می تواند ویدئو های تبلیغاتی را اسکیپ کند.
- Tag: برای استفاده در نوتیفیکیشن.

شروع کار

▶ برای استفاده از گزارش خطا باید قطعه کد زیر را در starter اضافه کنید.

```
Sub Application_Error (Error As Exception, StackTrace As String) As Boolean
```

```
    Main.ADok.Crash(Error)
```

```
    Return False
```

```
End Sub
```

▶ موارد مربوط به منیفست را از سایت دریافت در محل خود قرار دهید.

1. برای چک کردن دسترسی ها می توانید از توابع `isReadStoragePermissionGranted` و `isWriteStoragePermissionGranted` و `isPhoneStatePermissionGranted` و `isLocationPermissionGranted` و `isCameraPermissionGranted` استفاده کنید

2. برای درخواست دسترسی ها از توابع `RequestWriteStoragePermission` و `RequestReadStoragePermission` 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

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 کاربر بین برنامه های مختلف یکسان است.

Sign in / Register

- ▶ برای ثبت نام کاربر از تابع زیر استفاده می کنیم
- ▶ `adok.RegisterPlayer` (نام رویداد, فعال سازی با پیامک, شماره تماس, رمز عبور, نام کاربری, نام مستعار);
- ▶ در صورت قرار دادن مقدار `true` در فعال سازی با پیامک، در صورت موفقیت آمیز بودن ثبت نام، کاربر با دریافت کد تایید از طریق پیامک حساب کاربری خود را فعال می کند.
- ▶ 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:

▶ `EventName_onRegisterPlayerEnd(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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 عبور 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 up / Login

▶ برای ورود کاربر از تابع زیر استفاده می کنیم

▶ `adok.LoginUser(EventName, فعال سازی با پیامک, رمز عبور, نام کاربری)`;

▶ در صورت قرار دادن مقدار `true` در فعال سازی با پیامک، در صورت موفقیت آمیز بودن ورود، کاربر با دریافت کد تایید از طریق پیامک حساب کاربری خود را فعال می کند.

▶ 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:

▶ `EventName_onLoginEnd(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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.

Forget Password

- ▶ برای فراموشی رمز عبور از تابع زیر استفاده می کنیم
- ▶ `adok.ForgetPasswordPlayer(نام کاربری,EventName);`
- ▶ 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:

▶ `EventName_ onForgetPasswordEnd(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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

تغییر رمز عبور

- ▶ برای تغییر رمز عبور از تابع زیر استفاده می کنیم
- ▶ `adok.ChangePassword(EventName,تکرار رم عبور جدید,رمز عبور جدید,رمز عبور قدیمی)`
- ▶ 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:

▶ `EventName_ onChangePassword(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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 `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,EventName);`

▶ 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:

▶ `EventName_ onChangeNickName(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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(EventName);`

▶ 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:

▶ `onResendCodeTaieEnd(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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 لو دینگ وجود دارد.
5. `notExist`: کاربر با این نام کاربری وجود ندارد

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(مقدار فیلد, نام فیلد, isSync);`

▶ نوع فیلد شامل موارد زیر است:

▶ `isSync = false`: برای فیلدهایی که نیازی به ذخیره در سرور ندارند و قابل بازیابی نیستند

▶ `isSync = true`: 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. SyncDataFromServer(EventName);`

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

▶ `onSyncDataFromServer(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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(EventName);`

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

▶ `EventName_ onSyncEnd(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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.

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(EventName, 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:
 - 0: همیشه- از ابتدای ایجاد بازی بر روی سرور
 - 1: ماه جاری
 - 2: هفته جاری
 - 3: امروز

دریافت لیدربرد کلی از سرور

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

▶ `EventName_onLeaderboardGet(result as string)`

▶ تابع رویداد دارای ورودی از نوع `string` 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.
- ▶ `EventName_onLeaderboardGet(result as string)`

دریافت لیدربرد من از سرور

- ▶ برای دریافت لیدربرد من از سرور از تابع زیر استفاده می کنیم:
- ▶ `adok.GetMyLeaderBoard(EventName, 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:
 - 0: همیشه- از ابتدای ایجاد بازی بر روی سرور
 - 1: ماه جاری
 - 2: هفته جاری
 - 3: امروز

دریافت لیدربرد من از سرور

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

▶ `EventName_onMyLeaderboardGet(result as String)`

▶ تابع رویداد دارای ورودی از نوع `String` 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(EventName,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: ►

EventName _onGetLeague(result as String) ►

تابع رویداد دارای ورودی از نوع String 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. LeaguesResultJsonString مشاهده و استفاده کنید که نوع آن json می باشد.

عضویت در لیگ

`:adok.JoinLeagues(EvntName,leagueId);` ▶

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

`EventName _onJoinLeagues(result as String)` ▶

تابع رویداد دارای ورودی از نوع `String` 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(EventName,leagueId, myCount);` ▶

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

`EventName_onGetLeagueDatas(result as String)` ▶

▶ تابع رویداد دارای ورودی از نوع `String` 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 `:NoData`: اطلاعاتی وجود ندارد

ذخیره اطلاعات بازیکن در لیگ

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

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

▶ `EventName_onSetDataLeague(result as String)`

▶ تابع رویداد دارای ورودی از نوع `String` 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 `LeagueExpired`: مدت زمان لیگ به پایان رسیده است

دریافت لیدربرد لیگ

`adok.GetLeagueLeaderBoard(EventName,leagueId, myCount);` ▶

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

`EventName_onGetLeagueLeaderBoard(result as String)` ▶

▶ تابع رویداد دارای ورودی از نوع `String` 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(EventName,leagueId);`

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

► `EventName_onGetMyRewards(result as String)`

► تابع رویداد دارای ورودی از نوع `String` 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.

در صورت موفقیت آمیز بودن عملیات می تواند نتیجه را در `adok. RewardJsonText` مشاهده و استفاده کنید که نوع آن `json` می باشد.

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 `adok.SetReward(EventName,RewardId, myLeague.leagueld, leagueCount)` استفاده می کنیم.

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

▶ `EventName_onSetReward(result as String)`

▶ تابع رویداد دارای ورودی از نوع `String` 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. `GetBefore`: قبلا دریافت شده است

در صورت موفقیت آمیز بودن و داخلی بودن جایزه تابع

```
adok.SetData(flname,مقدار جایزه, true);
```

را استفاده می کنیم.

نمایش تبلیغات ویدئویی

▶ `adok.ShowAdv(EventName)`

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

▶ `EventName_onVideoViewEnd(result As String)`

▶ تابع رویداد دارای ورودی از نوع `String` 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 لوودینگ 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(EventName);`

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

▶ `EventName _onGetDate(result as String)`

▶ تابع رویداد دارای ورودی از نوع `String` 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 لو دینگ 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(EventName);`

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

▶ `EventName _onGetTime(result as String)`

▶ تابع رویداد دارای ورودی از نوع `String` 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 لوودینگ 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.