Article

Authenticating a player

Confirm player credentials and device capabilities and check for account restrictions.

Overview

To identify themselves and access their game data from all their devices, players create a Game Center account that includes a profile with an avatar and nickname. A single account allows players to sign in to Game Center once and use the same credentials for all Game Center games on their device.

In the GameKit framework, you use player objects to post scores, award achievements, build leaderboards, and start multiplayer games. The *local player* (`GKLocalPlayer`) represents the user playing your game and other *players* (`GKPlayer`) may be friends, recent matches, or global players of your and other games.

Before you access any Game Center data, you must verify that the local player signed in to Game Center on the device. Check if the player has any account restrictions and adjust your game accordingly. On Apple TV, your game can also support the ability to switch between user accounts.

Initialize the local player

In your game, you need to initialize the local player before you can use any GameKit APIs and Game Center services. Game Center verifies the credentials of the player and ensures their account is ready for use. Game Center also checks whether you configured your game for Game Center.

To initialize the user, set the handler (`authenticateHandler`) on the shared instance of `GKLocalPlayer` that represents the player of your game as in:

```
GKLocalPlayer.local.authenticateHandler = { viewController, error in
   // Handle the initialization callbacks.
}
```

Note

You use the handler to initialize Game Center. If you need to authenticate a person with your own server, continue the flow with `GKLocalPlayer/fetchitems(foridentityverificationsignature:)`.

GameKit calls the handler, possibly several times, for the following cases:

- If the local player needs to perform some action, GameKit passes a view controller that you must present to the player to complete initialization.
- If the player successfully signs in, GameKit sets the local player's `isAuthenticated` property to `true` and calls the handler again, this time passing `nil` for both the view controller and error parameters. You can then start the game.
- If the player decides not to sign in or create a Game Center account, GameKit sets the local player's `isAuthenticated` property to `false` and calls the handler again by passing an error that indicates the reason the player isn't available. In this case, disable Game Center in your game.
- If the local player previously signed in on the device when you set the handler, GameKit sets the local player's `isAuthenticated` property to `true` and passes `nil` for both the view controller and error parameters, and you can start the game.

Check for restrictions

Before starting a game, check the local player's account for restrictions and disable or hide features and content accordingly.

Check if there are any restrictions in the handler when the player signs in to Game Center. In addition to checking whether the player is underage (`isUnderage`) or not allowed to play multiplayer games (`isMultiplayerGamingRestricted`), check for any communication restrictions (`isPersonalizedCommunicationRestricted`).

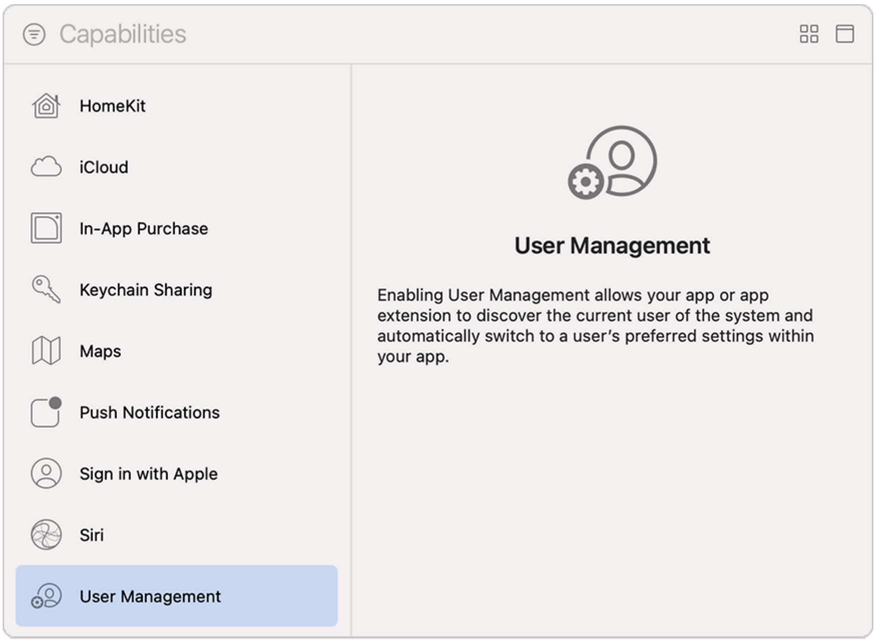
```
GKLocalPlayer.local.authenticateHandler = { viewController, error in
    if let viewController = viewController {
        // Present the view controller so the player can sign in.
        return
    }
    if error != nil {
        // Player is not available
        // Disable Game Center in the game.
        return
    }

    // Player is available.
    // Check if there are any player restrictions before starting the game.

    if GKLocalPlayer.local.isUnderage {
        // Hide explicit game content.
    }

    if GKLocalPlayer.local.isMultiplayerGamingRestricted {
        // Disable multiplayer game features.
    }

    if GKLocalPlayer.local.isPersonalizedCommunicationRestricted {
        // Disable in game communication UI.
    }

    // Perform any other configurations as needed (for example, access point).
}
```

If the `isPersonalizedCommunicationRestricted` property is `true`, then the player isn't allowed to use voice or messaging features during a multiplayer game. If your game includes any custom communication features, you should disable them. Note that if the player is underage, this property is always `true`.

Support user switching

Multiple people can sign in to their accounts on a single Apple TV and you can allow players to run your game on Apple TV using their individual Game Center account. After you configure your app, add code to support user switching; otherwise, the system runs your game as the default user.

To support user switching, add the User Management capability to your app in Xcode. In the project editor, select the target, click the Signing & Capabilities tab, click the Library button (+), and then double-click the User Management capability or drag it to the Signing & Capabilities pane. Under User Management in the Signing & Capabilities pane, select Run as Current User. This entitlement grants your game access to the current user's Game Center data.



Enabling User Management allows your app or app extension to discover the current user of the system and automatically switch to a user's preferred settings within your app.

When the user switches on Apple TV, the system relaunches your game. To save game data if the users switch while your game is in the foreground, implement the `applicationWillTerminate(_:)` method. To save data when the user switches to another app, implement the `applicationWillResignActive(_:)` method.

When the system relaunches your game, GameKit passes the new user to the handler you use to initialize Game Center. For more information on user switching, see Personalizing Your App for Each User on Apple TV.

See Also

Essentials

Initializing and configuring Game Center

Enable Game Center, configure features, and test them locally in your Xcode project.

Improving the player experience for games with large downloads

Provide ample content in your base installation and then use on-demand resources and the Background Assets API to handle additional content.

`Game Center Entitlement`

A Boolean value that indicates whether users of the app may see and compare achievements on a leaderboard, invite friends, and start multiplayer games.