

[GameKit](#) / [GKLocalPlayer](#)

Class

GKLocalPlayer

The local player who signs in to Game Center on the device running the game.

iOS 4.1+ | iPadOS 4.1+ | Mac Catalyst 13.1+ | macOS 10.8+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 3.0+

```
class GKLocalPlayer
```

Mentioned in

- 📄 [Authenticating a player](#)
- 📄 [Initializing and configuring Game Center](#)
- 📄 [Protecting the player's privacy using scoped identifiers](#)

Overview

Only one player can sign in to Game Center on a device at a time and that player is the *local player*. Before you can start a game that uses GameKit features, verify that the local player signs in to their Game Center account.

You set the handler of the local player shared instance using the [authenticateHandler](#) property. Then implement this method to handle the multiple times GameKit invokes it during the sign-in process. If the local player needs to create an account or sign in, GameKit provides a view controller that you present to the local player. If the local player successfully signs in, determine whether they have any account restrictions and adjust your game accordingly. For more information about the initialization of the local player, see [Authenticating a player](#).

After the local player signs in, their account data and GameKit features are available. You can display the local player's nickname and avatar, access their recent players and friends, and load their leaderboards and achievements. You can also register a listener object that GameKit calls when the local player sends or accepts invitations to play with others.

Topics

Accessing the Local Player

```
class var local: GKLocalPlayer
```

The shared instance of the local player.

Authenticating the Local Player

```
var authenticateHandler: ((UIViewController?, (any Error)?) -> Void)?
```

A handler that GameKit calls while initializing the local player.

```
var isAuthenticated: Bool
```

A Boolean value that indicates whether a local player has signed in to Game Center.

```
func fetchItems(forIdentityVerificationSignature: ((URL?, Data?, Data?,  
UInt64, (any Error)?) -> Void)?)
```

Generates a signature that you can use to authenticate the local player on your own server.

```
static let GKPlayerAuthenticationDidChangeNotificationName:  
NSNotification.Name
```

A notification that posts after GameKit authenticates the local player.

Determining Whether the Player Is Underage or Restricted

```
var isUnderage: Bool
```

A Boolean value that indicates whether the local player is underage.

```
var isMultiplayerGamingRestricted: Bool
```

A Boolean value that indicates whether the player can join multiplayer games.

```
var isPersonalizedCommunicationRestricted: Bool
```

A Boolean value that indicates whether the player can use personalized communication on the device.

Accessing Friends and Recents

```
func loadFriendsAuthorizationStatus((GKFriendsAuthorizationStatus, (any Error)?) -> Void)
```

Returns whether the player authorizes your game to access their friends list.

```
enum GKFriendsAuthorizationStatus
```

Constants that indicate if the local player grants access to their friends list.

```
func loadFriends(([GKPlayer]?, (any Error)?) -> Void)
```

Loads the local player's friends list if the local player and their friends grant access.

```
func loadFriends(identifiedBy: [String], completionHandler: ([GKPlayer]?, (any Error)?) -> Void)
```

Loads the player's friends list, scoped by the identifiers, if the player and their friends grant access.

NSGKFriendListUsageDescription

A message that tells people why the app needs access to their Game Center friends list.

```
func loadChallengableFriends(completionHandler:(([GKPlayer]?, (any Error)?)) -> Void)?
```

Loads players to whom the local player can issue a challenge.

```
func loadRecentPlayers(completionHandler:(([GKPlayer]?, (any Error)?)) -> Void)?
```

Loads players from the friends list or players that recently participated in a game with the local player.

Adding Friends

```
var isPresentingFriendRequestViewController: Bool
```

A Boolean value that indicates whether your game presents the friends request view controller.

```
func presentFriendRequestCreator(from: UIViewController) throws
```

Presents a view controller with a Messages sheet for the player to request friends.

```
func presentFriendRequestCreator(from: NSWindow?) throws
```

Opens the Messages app with a sheet for the player to request friends.

Working with Leaderboards

```
func loadDefaultLeaderboardIdentifier(completionHandler: ((String?,  
(any Error)?)) -> Void)?)
```

Loads the identifier for the local player's default leaderboard.

Deprecated

```
func setDefaultLeaderboardIdentifier(String, completionHandler: (((any  
Error)?)) -> Void)?)
```

Sets the local player's default leaderboard.

Deprecated

Registering Listeners

```
func register(any GKLocalPlayerListener)
```

Registers a listener for a particular event.

```
func unregisterAllListeners()
```

Unregisters all listeners in your game.

```
func unregisterListener(any GKLocalPlayerListener)
```

Unregisters a listener object.

Saving Game Data

 Saving the player's game data to an iCloud account

Save game data during play or after a game in the player's iCloud account that's accessible from any device.

```
func saveGameData(Data, withName: String, completionHandler: ((GKSaved  
Game?, (any Error)?)) -> Void)?)
```

Saves game data with the specified name.

```
func fetchSavedGames(completionHandler: (([GKSavedGame]?, (any Error)?)  
-> Void)?)
```

Retrieves all available saved games.

```
func resolveConflictingSavedGames([GKSavedGame], with: Data, completion  
Handler: (([GKSavedGame]?, (any Error)?)) -> Void)?)
```

Replaces duplicate saved games that use the same filename with one file containing the specified game data.

```
func deleteSavedGames(withName: String, completionHandler: (((any Error?)?) -> Void)?)
```

Deletes saved games with the specified filename.

```
class GKSavedGame
```

An object that represents a file containing saved game data.

```
protocol GKSavedGameListener
```

A protocol that handles events related to saving game data.

Deprecated

☰ Deprecated symbols

Review unsupported symbols and their replacements.

Relationships

Inherits From

GKPlayer

Conforms To

CVarArg

Copyable

CustomDebugStringConvertible

CustomStringConvertible

Equatable

GKSavedGameListener

Hashable

NSCopying

NSObjectProtocol

See Also

Players

- Connecting players with their friends in your game

Give players the ability to connect and interact with friends in your game.

- Saving the player's game data to an iCloud account

Save game data during play or after a game in the player's iCloud account that's accessible from any device.

- Protecting the player's privacy using scoped identifiers

Use the scoped identifiers that GameKit provides you as player IDs when transmitting or saving player data.

`class GKPlayer`

A remote player who the local player running your game can invite and communicate with through Game Center.

`class GKBasePlayer`

A class that provides common data and methods for the different player objects.

`protocol GKLocalPlayerListener`

A protocol that handles events for Game Center players.

`static let GKPlayerAuthenticationDidChangeNotificationName: NSNotification.Name`

A notification that posts after GameKit authenticates the local player.

`static let GKPlayerDidChangeNotificationName: NSNotification.Name`

A notification that posts when a player object's data changes.