

# Table of Contents

## SUGAR Unity Client

### Tutorials

[Quick Start](#)

[Seeding](#)

[Client Prefabs](#)

[Build Instructions](#)

### Features

[Saving Data](#)

[SUGARManager](#)

### Development

[Documentation](#)

### Api

[PlayGen.SUGAR.Unity](#)

[AccountUnityClient](#)

[ActorResponseAllowableActions](#)

[BaseAccountInterface](#)

[BaseEvaluationListInterface](#)

[BaseEvaluationPopupInterface](#)

[BaseGroupMemberInterface](#)

[BaseInterface](#)

[BaseLeaderboardInterface](#)

[BaseLeaderboardListInterface](#)

[BaseUnityClient<T>](#)

[BaseUserFriendInterface](#)

[BaseUserGroupInterface](#)

[CommandLineOptions](#)

[CommandLineUtility](#)

[Config](#)

[CustomInterface](#)

[EvaluationUnityClient](#)

[GameDataUnityClient](#)

[GroupMemberUnityClient](#)

[LeaderboardListUnityClient](#)

LeaderboardUnityClient  
ResourceUnityClient  
ResponseHandler  
SavedPrefsHandler  
SUGARManager  
SUGARUnityManager  
UserFriendUnityClient  
UserGroupUnityClient  
PlayGen.SUGAR.Unity.Editor  
AutoLogin  
EditGameSeed  
EditGameSeedWindow  
SeedGame  
SeedGameWindow  
SetEditorAutoLogin  
SetEditorAutoLogin.AutoLoginOption  
SetEditorAutoLogin.BoolValue  
SetEditorAutoLogin.StringValue  
PlayGen.SUGAR.Unity.WebGL  
UnityWebGLHttpHandler

# SUGAR-Unity Client

Welcome to the documentation for the Unity Client of the SUGAR Engine.

## Online Documentation

You can find the online version of the SUGAR Unity Client [here](#)

## SUGAR Engine

If you haven't already read the documentation for the SUGAR Engine, it is highly recommended that you do that first as that is where the core concepts are explained along with a demo and other useful information.

[SUGAR Engine Documentation](#)

# Tutorials

This section provides tutorials for the various tasks a developer may want to explore around using and customizing the SUGAR Unity Client.

# Quick Start

**Note: Please ensure you have created a SUGAR account before going through the following steps.**

If you don't already have one, you can create a SUGAR account [here](#).

## Build the SUGAR Unity Client

1. Open and build the PlayGen.SUGAR.Unity project.

## Add SUGAR

1. Add the 'SUGAR' prefab, found at SUGAR/Prefabs/SUGAR, into your starting scene. All of the interfaces referenced on the Unity Clients on this object can be found at SUGAR/Example/Prefabs/Landscape.

## Create Game Seed File

1. Open the 'Edit Game Seed' tool by clicking Tools/SUGAR/Edit Game Seed.
2. Create a new game seed by clicking the 'Create Game Seed' button.
3. Fill in the 'Name' field with the name of the game you wish to seed.
4. Save this basic Seed file by clicking the 'Save' button. If you do not change the selected file, this will overwrite the provided 'GameSeed' file.
5. If you wish to set up the achievements, leaderboards and skills for your game now, go to the guide on [Seeding](#) for further details.

## Seed Game

1. Open the 'Seed Game' tool by clicking Tools/SUGAR/Seed Game.
2. Fill in the provided Username and Password fields with your SUGAR details.
3. If you did not overwrite the provided 'GameSeed' file, change the 'Game Seed File' field to use the file you created during step 2.
4. Click the 'Sign-in and Seed' button to add the game to the platform. This step will fail if the seed file is invalid or you provide invalid user details.
5. Check that the 'Game Token' and 'Game Id' fields on the SUGAR object in your starting scene has been edited to match the details of the game you just created.

## Set Auto Log-in Values (optional)

1. Open the 'Set Auto Log-in Values' tool by clicking Tools/SUGAR/Set Auto Log-in Values.
2. Fill in the details you want to use to automatically sign in when testing in Unity.
3. This feature is disabled if either 'Auto Log-in' within the tool or 'Allow Auto Login' in 'Account Unity Client' on the 'SUGAR' prefab is not checked.

## Testing SUGAR features in Unity Project

In the unity project, open *Unity/Assets/SUGAR/Example/Scene.unity*, after rebuilding the PlayGen.SUGAR.Unity project you can test the functionality with the following commands:

SHORTCUT	TEST
T + L	GameLeaderboard.DisplayGameList
T + K	Evaluation.DisplayAchievementList
T + A	Evaluation.ForceNotification
T + S	Unity.StartSpinner
T + H	Unity.StopSpinner

SHORTCUT	TEST
T + F	UserFriend.Display
T + G	UserGroup.Display
Escape	Application.Quit
Delete	Account.Logout

Shortcuts can be seen and changed in *Unity/Assets/SUGAR/Example/Scripts/TestImplementation.cs*.

# Seeding

- **Name:** The name of the game you are adding to the SUGAR system. Please note that this name must be unique to the host.

## Achievements/Skills

- **Name:** The name of the achievement/skill you are creating. Please note that this name must be unique to the game.
- **Description:** The description of the achievement/skill you are creating.
- **Token:** The unique identification token for the achievement/skill you are creating. Please note that this name must be unique to the game.
- **ActorType:** The type of Actor which this achievement/skill applies to.

### Evaluation Criteria

- **EvaluationDataKey:** The key of the data which is being used for this criteria.
- **EvaluationDataCategory:** The category of data which the criteria is being checked against.
- **ComparisonType:** What the current value has to be compared to the target value in order for this criteria to be completed.
- **CriteriaQueryType:** The data set from which the current value will be gathered.
- **EvaluationDataType:** The type of data which the criteria is being checked for.
- **Scope:** The range of actors from which the data will be collected for comparison.
- **Value:** The target value of this criteria.

### Rewards

- **EvaluationDataKey:** The key of the data which will be provided to the actor upon completion of all criteria.
- **EvaluationDataCategory:** The category of data which will be provided to the actor upon completion of all criteria.
- **EvaluationDataType:** The type of data which will be provided to the actor upon completion of all criteria.
- **Value:** The value which will be provided to the actor upon completion of all criteria.

## Leaderboards

- **Token:** The unique identification token for the leaderboard you are creating. Please note that this name must be unique to the game.
- **Name:** The name of the leaderboard you are creating. Please note that this name must be unique to the game.
- **Key:** The key of the data which will be used to form the leaderboard.
- **ActorType:** The type of Actor which this leaderboard applies to.
- **EvaluationDataCategory:** The category of data which will be used for this leaderboard.
- **EvaluationDataType:** The type of data which will be used for this leaderboard.
- **CriteriaScope:** The range of actors from which the data will be collected.
- **LeaderboardType:** How collected data will be sorted for this leaderboard.

## Enums

### ActorType

- **Undefined:** Intended use not defined and as such means it applies to both Users and Groups.
- **User:** Intended for use for Users only.
- **Group:** Intended for use for Groups only.

### ComparisonType

- **Equals:** The current value and target value must exactly match.
- **Not Equal:** The current value and target value do not exactly match.
- **Greater:** The current value is greater in value than the target value (Long and Float only).
- **Greater or Equal:** The current value is greater or exactly equal in value to the target value (Long and Float only).
- **Lesser:** The current value is lower in value than the target value (Long and Float only).
- **Lesser or Equal:** The current value is lower or exactly equal in value to the target value (Long and Float only).

## CriteriaScope

- **Actor:** Data collected will have been submitted by the actor themselves.
- **Related Users:** Data collected will have been submitted by the actor and their friends (if the actor is a User) or their members (if the actor is a Group). ActorType cannot also be Undefined.
- **Related Groups:** Data collected will have been submitted by the actor and their alliances. ActorType must be Group.
- **Related Group Users:** Data collected will have been submitted by their members and the members of their alliances. ActorType must be Group.

## CriteriaQueryType

- **Any:** Any collected data will be compared against the value set in the criteria. Can only be used when Scope is set to Actor.
- **Sum:** The sum of all collected data will be compared against the value set in the criteria. Can only be used when EvaluationDataType is set to Long or Float.
- **Latest:** The latest piece of data will be compared against the value set in the criteria. Can only be used when Scope is set to Actor.

## EvaluationDataCategory

- **Game Data:** The data has been stored as a piece of Game Data, which allows for multiple values for the same key and all EvaluationDataTypes.
- **Resource:** The data has been stored as a Resource, which allows for only one value per key and has its value stored as a Long.
- **Skill:** The data has been stored as a Skill, marking the completion of all criteria for a Skill.
- **Achievement:** The data has been stored as a Achievement, marking the completion of all criteria for a Achievement.
- **Match Data:** The data has been stored as a piece of Match Data, which allows for multiple values for the same key in relation to a Match and all EvaluationDataTypes.

## EvaluationDataType

- **String:** The data has been stored and can be parsed as a String.
- **Long:** The data has been stored and can be parsed as a Long.
- **Float:** The data has been stored and can be parsed as a Float.
- **Boolean:** The data has been stored and can be parsed as a Boolean.

## LeaderboardType

- **Highest:** The leaderboard will be sorted from highest single value to lowest (Long and Float only).
- **Lowest:** The leaderboard will be sorted from lowest single value to highest (Long and Float only).
- **Cumulative:** The leaderboard will be sorted from highest combined value to lowest (Long and Float only).
- **Count:** The leaderboard will be sorted from the highest amount which the key has been recorded to the lowest (String and Boolean only).
- **Earliest:** The leaderboard will be sorted from the earliest time the key was recorded to the latest (String and Boolean only).
- **Latest:** The leaderboard will be sorted from the latest time the key was recorded to the earliest (String and Boolean only).



# Unity Client Prefabs

There are a number of prefabs included within the Unity package. Example usage can be viewed in the demo scene which can be found at: Assets/SUGAR/Example/Scene.unity

After setting up a game using the [quick start guide](#), all prefabs can be seen working by running the demo scene and entering the shortcuts to display each.

Each of the prefabs can be activated using the SUGAR prefab provided (which must be added to the scene to use SUGAR). The SUGAR prefab has its own canvas which is rendered above the main Canvas which contains each of the panel prefabs. Each panel can be activated using their display functions, examples of which can be seen in:

Assets/SUGAR/Example/Scripts/TestImplementation.cs

## Panel Prefabs

Panel Prefabs provided

- AccountPanel
- EvaluationPanel
- EvaluationPopup
- FriendsListPanel
- UserGroupsPanel
- GroupMembersPanel
- LeaderboardListPanel
- LeaderboardPanel

There are 2 variations provided for each of the panels, one for landscape and one for portrait. To switch the orientation used, change the Interface references in the SUGAR prefab to the required orientation prefabs.

## Panel Overview

- **AccountPanel**

Handles logging in to SUGAR with username and password entered, does not display if auto login is enabled and the login details are correct.

- **EvaluationPanel**

Demo Shortcut: Hold T & Press K.

Displays the achievements/skills for the current game and which ones have been completed.

- **EvaluationPopup**

Demo Shortcut: Hold T & Press A.

Displays a pop-up to notify users that an achievement or skill has been completed.

- **FriendsListPanel**

Demo Shortcut: Hold T & Press F.

Displays the users that the logged in account is friends with, allowing them to make new friendships, remove existing ones and review pending requests.

- **UserGroupsPanel**

Demo Shortcut: Hold T & Press G.

Displays the groups that the logged in account is a member of, allowing them to leave current groups, join new groups and see pending group requests.

- **GroupMembersPanel**

Demo Shortcut: Hold T & Press G -> select a group.

Displays the members of a specific group, allowing accounts with permissions to manage other accounts within that group.

- **LeaderboardListPanel**

Demo Shortcut: Hold T & Press L.

Displays a list of leaderboards for the current game, allowing users to select one and see the standings.

- **LeaderboardPanel**

Demo Shortcut: Hold T & Press L -> select a leaderboard.

Displays the current standings for the selected leaderboard, allows users to filter results by:

- Top (the best scores for the current leaderboard),
- Nearby (the scores near to the current users score),
- Friends (the current users' friends scores),
- Group Members (the scores for all members of your primary group),
- Alliances (the scores of all groups in an alliance with your primary group).

# Build Instructions

1. Open and build the PlayGen.SUGAR.Unity project.
2. Open the Unity project.
3. Click Menu/Tools/Build SUGAR Package.
4. Copy the built package from Build/SUGAR.unitypackage.
5. Import it into your project and let the magic begin.

## Making Changes to SUGAR Client

1. Build SUGAR project
2. Open SUGAR/PlayGen.SUGAR.Client.Development/bin/Debug/net46
3. Copy new .dll files to sugar-unity/lib/SUGAR/Client
4. Open and Build PlayGen.SUGAR.Unity project

# Development

This section covers various aspects to do with developing the SUGAR Unity Client.

# Documentation

SUGAR Unity's documentation is generated using [DocFX](#) using tripple slash code comments and DicFX .md and .toc files located in docs/.

## Requirements

- [DocFX](#)
- "docfx" as a command needs to be available via the command console for the scripts to work.
- PDF documentation requires [wkhtmltopdf](#).

## Process

There are various build scripts in docs/tools to build, copy and serve the docs.

TOOL	FUNCTION
all.bat	Build the docs site and pdf.
copy_to_unity.bat	Copy the built pdf into the unity project.
all_and_copy.bat	all.bat and copy_to_unity.bat
metadata_build_and_serve.bat	Build the site and serve. Use this to test the generated docs.
metadata_pdf.bat	Build the pdf.

Note: The PDF docfx config was created by following [this guide](#).



# Namespace PlayGen.SUGAR.Unity

## Classes

### [AccountUnityClient](#)

Unity client for functionality related to signing in, registering and signing out.

### [ActorResponseAllowableActions](#)

ActorResponse with additional information on if the current user can add and remove them.

### [BaseAccountInterface](#)

Base abstract class for controlling the UI object related to user accounts.

### [BaseEvaluationListInterface](#)

Base abstract class for controlling the interface related to evaluation lists.

### [BaseEvaluationPopupInterface](#)

Base abstract class for controlling the interface related to evaluation notifications.

### [BaseGroupMemberInterface](#)

Base abstract class for controlling the UI object related to group member lists.

### [BaseInterface](#)

Base abstract class for controlling UI objects

### [BaseLeaderboardInterface](#)

Base abstract class for controlling the UI object related to leaderboards

### [BaseLeaderboardListInterface](#)

Base abstract class for controlling the UI object related to getting all leaderboards for a game

### [BaseUnityClient<T>](#)

Base abstract class for UnityClient classes

### [BaseUserFriendInterface](#)

Base abstract class for controlling the UI object related to friends lists

### [BaseUserGroupInterface](#)

Base abstract class for controlling the UI object related to user groups.

### [CommandLineOptions](#)

### [CommandLineUtility](#)

### [Config](#)

### [CustomInterface](#)

### [EvaluationUnityClient](#)

Unity client for calls related to evaluations.

### [GameDataUnityClient](#)

Get and send GameData for the currently signed in user.

### [GroupMemberUnityClient](#)

Unity client for calls related to group member lists.

### [LeaderboardListUnityClient](#)

Unity client for calls related to leaderboards for an application.

### [LeaderboardUnityClient](#)

Unity client for calls related to leaderboard standings.

### [ResourceUnityClient](#)

Unity client for calls related to resources.

### [ResponseHandler](#)

### [SavedPrefsHandler](#)

### [SUGARManager](#)

Access point for SUGAR related classes.

### [SUGARUnityManager](#)

Class for managing Unity elements of the asset

### [UserFriendUnityClient](#)

Unity client for calls related to user friends.

### [UserGroupUnityClient](#)

Unity client for calls related to group lists.



# Class AccountUnityClient

Unity client for functionality related to signing in, registering and signing out.

Inheritance

System.Object

AccountUnityClient

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class AccountUnityClient : MonoBehaviour
```

## Properties

### HasInterface

Has an interface been provided for this Unity Client?

Declaration

```
public bool HasInterface { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

### IsActive

Is there an interface and if so is it currently active?

Declaration

```
public bool IsActive { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

## Methods

### DisplayLogInPanel(Action<Boolean>)

Displays interface if provided and allowAutoLogin is false. Attempts automatic sign in using provided details if allowAutoLogin is true. Note: allowAutoLogin is made false after automatic sign in is first attempted.

Declaration

```
public virtual void DisplayLogInPanel(Action<bool> success)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	success	Callback which will result whether the user successfully signed in.

Hide()

Hide the UI object if it is currently active.

Declaration

```
public virtual void Hide()
```

Logout(Action<Boolean>)

Sign out the currently signed in user

Declaration

```
public virtual void Logout(Action<bool> success)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	success	Callback which will result whether the user successfully signed out.

# Class ActorResponseAllowableActions

ActorResponse with additional information on if the current user can add and remove them.

Inheritance

System.Object

ActorResponseAllowableActions

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class ActorResponseAllowableActions : object
```

## Constructors

ActorResponseAllowableActions(ActorResponse, Boolean, Boolean)

Declaration

```
public ActorResponseAllowableActions(ActorResponse actor, bool add, bool remove)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	actor	
System.Boolean	add	
System.Boolean	remove	

## Properties

### Actor

ActorResponse contains the actor ID and Name.

Declaration

```
public ActorResponse Actor { get; set; }
```

Property Value

TYPE	DESCRIPTION
ActorResponse	

### CanAdd

Can the currently signed in user add or accept a request from this actor?

Declaration

```
public bool CanAdd { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

CanRemove

Can the currently signed in user remove or reject a request from this actor?

Declaration

```
public bool CanRemove { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

# Class BaseAccountInterface

Base abstract class for controlling the UI object related to user accounts.

Inheritance

System.Object

BaseAccountInterface

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseAccountInterface : MonoBehaviour
```

## Fields

### \_closeButton

Button used to disable this object. Can be left null.

Declaration

```
protected Button _closeButton
```

Field Value

TYPE	DESCRIPTION
Button	

### \_errorText

Text object which displays errors if/when they occur. Can be left null.

Declaration

```
protected Text _errorText
```

Field Value

TYPE	DESCRIPTION
Text	

### \_loginButton

Button used to trigger attempting to sign in. Can be left null.

Declaration

```
protected Button _loginButton
```

Field Value

TYPE	DESCRIPTION
Button	

### \_name

Input field used for providing usernames. Required.

#### Declaration

```
protected InputField _name
```

#### Field Value

TYPE	DESCRIPTION
InputField	

#### \_password

Input field used for providing passwords. Required.

#### Declaration

```
protected InputField _password
```

#### Field Value

TYPE	DESCRIPTION
InputField	

#### \_registerButton

Button used to trigger attempting to register a new account. Can be left null.

#### Declaration

```
protected Button _registerButton
```

#### Field Value

TYPE	DESCRIPTION
Button	

#### \_rememberMeToggle

Toggle used to enable/disable remembering the next signed in user's details for future sessions. Can be left null.

#### Declaration

```
protected Toggle _rememberMeToggle
```

#### Field Value

TYPE	DESCRIPTION
Toggle	

#### Methods

##### Awake()

Base Awake method adds onClick listeners for the login, register and close buttons.

#### Declaration

```
protected virtual void Awake()
```

# Class BaseEvaluationListInterface

Base abstract class for controlling the interface related to evaluation lists.

Inheritance

System.Object

[BaseInterface](#)

BaseEvaluationListInterface

Inherited Members

[BaseInterface.\\_errorText](#)

[BaseInterface.\\_closeButton](#)

[BaseInterface.\\_signinButton](#)

[BaseInterface.Awake\(\)](#)

[BaseInterface.PreDisplay\(\)](#)

[BaseInterface.Show\(Boolean\)](#)

[BaseInterface.Draw\(\)](#)

[BaseInterface.OnSignIn\(\)](#)

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: [PlayGen.SUGAR.Unity.dll](#)

Syntax

```
public abstract class BaseEvaluationListInterface : BaseInterface
```

Methods

ErrorDraw(Boolean)

Used to set error text in case of no user being signed in, loading issues or if no results are available.

Declaration

```
protected override void ErrorDraw(bool loadingSuccess)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	loadingSuccess	

Overrides

[BaseInterface.ErrorDraw\(Boolean\)](#)

HideInterfaces()

Hides Account, GameLeaderboard, Leaderboard, UserFriend, GroupMember and UserGroup interfaces.

Declaration

```
protected override void HideInterfaces()
```

Overrides

[BaseInterface.HideInterfaces\(\)](#)

LoadErrorText()

Get error string from Localization with key "EVALUATION\_LOAD\_ERROR" if there were issues loading the evaluation list.

Declaration

```
protected override string LoadErrorText()
```

#### Returns

TYPE	DESCRIPTION
System.String	

#### Overrides

[BaseInterface.LoadErrorText\(\)](#)

#### NoResultsErrorText()

Get error string from Localization with key "NO\_EVALUATION\_ERROR" if there were no evaluations to display.

#### Declaration

```
protected override string NoResultsErrorText()
```

#### Returns

TYPE	DESCRIPTION
System.String	

#### Overrides

[BaseInterface.NoResultsErrorText\(\)](#)



# Class BaseEvaluationPopupInterface

Base abstract class for controlling the interface related to evaluation notifications.

Inheritance

System.Object

BaseEvaluationPopupInterface

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseEvaluationPopupInterface : MonoBehaviour
```

## Fields

`_evaluationQueue`

Queue of notifications to be displayed.

Declaration

```
protected readonly List<EvaluationNotification> _evaluationQueue
```

Field Value

TYPE	DESCRIPTION
List<EvaluationNotification>	

`_image`

Image displayed alongside notification. Can be left null.

Declaration

```
protected Image _image
```

Field Value

TYPE	DESCRIPTION
Image	

`_name`

Text used for display notification string (usually evaluation name). Can be left null.

Declaration

```
protected Text _name
```

Field Value

TYPE	DESCRIPTION
Text	

## Methods

Display(EvaluationNotification)

Functionality to be triggered when a notification is received.

Declaration

```
protected abstract void Display(EvaluationNotification notification)
```

Parameters

TYPE	NAME	DESCRIPTION
EvaluationNotification	notification	

# Class BaseGroupMemberInterface

Base abstract class for controlling the UI object related to group member lists.

Inheritance

System.Object  
BaseInterface  
BaseGroupMemberInterface

Inherited Members

BaseInterface.\_errorText  
BaseInterface.\_closeButton  
BaseInterface.\_signinButton  
BaseInterface.Awake()  
BaseInterface.PreDisplay()  
BaseInterface.Show(Boolean)  
BaseInterface.Draw()  
BaseInterface.OnSignIn()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseGroupMemberInterface : BaseInterface
```

Fields

\_groupName

Text used for providing the group name for this list. Can be left null.

Declaration

```
protected Text _groupName
```

Field Value

TYPE	DESCRIPTION
Text	

Methods

ErrorDraw(Boolean)

Used to set error text in case of no user being signed in, loading issues or if no results are available.

Declaration

```
protected override void ErrorDraw(bool loadingSuccess)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	loadingSuccess	

Overrides

BaseInterface.ErrorDraw(Boolean)

## HideInterfaces()

Hides Account, Evaluation, Leaderboard, GameLeaderboard and UserFriend UI objects. Set groupName text to match name of CurrentGroup.

Declaration

```
protected override void HideInterfaces()
```

Overrides

[BaseInterface.HideInterfaces\(\)](#)

## LoadErrorText()

Get error string from Localization with key "GROUPS\_LOAD\_ERROR" if there were issues loading the evaluation list.

Declaration

```
protected override string LoadErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.LoadErrorText\(\)](#)

## NoResultsErrorText()

Get error string from Localization with key "NO\_RESULTS\_ERROR" if there were no group members to display.

Declaration

```
protected override string NoResultsErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.NoResultsErrorText\(\)](#)

# Class BaseInterface

Base abstract class for controlling UI objects

Inheritance

System.Object

BaseInterface

[BaseEvaluationListInterface](#)

[BaseGroupMemberInterface](#)

[BaseLeaderboardInterface](#)

[BaseLeaderboardListInterface](#)

[BaseUserFriendlyInterface](#)

[BaseUserGroupInterface](#)

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseInterface : MonoBehaviour
```

## Fields

**\_closeButton**

Button used to disable this object. Can be left null.

Declaration

```
protected Button _closeButton
```

Field Value

TYPE	DESCRIPTION
Button	

**\_errorText**

Text object which displays errors if/when they occur. Can be left null.

Declaration

```
protected Text _errorText
```

Field Value

TYPE	DESCRIPTION
Text	

**\_signinButton**

Button used to display account interface (if available) if no user is signed in. Can be left null.

Declaration

```
protected Button _signinButton
```

Field Value

TYPE	DESCRIPTION
Button	

## Methods

### Awake()

Base Awake method adds onClick listeners for the close and signin buttons.

Declaration

```
protected virtual void Awake()
```

### Draw()

Should be used to set, create and place UI on this object.

Declaration

```
protected abstract void Draw()
```

### ErrorDraw(Boolean)

Should be used to set error text and disable UI objects due to errors, if required. By default sets error text in case of no user being signed in or loading issues.

Declaration

```
protected virtual void ErrorDraw(bool loadingSuccess)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	loadingSuccess	

### HideInterfaces()

Should be used to enable/disable UI on this object and hide other UI objects.

Declaration

```
protected abstract void HideInterfaces()
```

### LoadErrorText()

Get error string if there were issues loading what was required.

Declaration

```
protected abstract string LoadErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

### NoResultsErrorText()

Get error string if there were no results to display.

#### Declaration

```
protected abstract string NoResultsErrorText()
```

#### Returns

TYPE	DESCRIPTION
System.String	

### OnSignIn()

Triggered by successful sign-in via this UI object.

#### Declaration

```
protected abstract void OnSignIn()
```

### PreDisplay()

Functionality triggered before displaying the UI object.

#### Declaration

```
protected abstract void PreDisplay()
```

### Show(Boolean)

Used to display/redraw the UI on this object. Triggers methods in this order: HideInterfaces - abstract method used to enable/disable UI on this object and hide other UI objects. PreDraw - private method. Activates object using SUGARManager.Unity.EnableObject, resets error text and hides signin button. Draw - abstract method where creation and placement of the UI should be performed. ErrorDraw - where error text is determined and set, if required.

#### Declaration

```
protected void Show(bool loadingSuccess)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	loadingSuccess	

# Class BaseLeaderboardInterface

Base abstract class for controlling the UI object related to leaderboards

Inheritance

System.Object

BaseInterface

BaseLeaderboardInterface

Inherited Members

BaseInterface.\_errorText

BaseInterface.\_closeButton

BaseInterface.\_signinButton

BaseInterface.PreDisplay()

BaseInterface.Show(Boolean)

BaseInterface.Draw()

BaseInterface.OnSignIn()

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseLeaderboardInterface : BaseInterface
```

## Fields

\_alliancesButton

Button used to change the current leaderboard filter to 'Alliances'. Can be left null.

Declaration

```
protected Button _alliancesButton
```

Field Value

TYPE	DESCRIPTION
Button	

\_friendsButton

Button used to change the current leaderboard filter to 'Friends'. Can be left null.

Declaration

```
protected Button _friendsButton
```

Field Value

TYPE	DESCRIPTION
Button	

\_leaderboardName

Text used for displaying leaderboard name. Can be left null.

Declaration



**protected** Text \_leaderboardName

Field Value

TYPE	DESCRIPTION
Text	

### \_leaderboardType

Text used for displaying current leaderboard filter. Can be left null.

Declaration

**protected** Text \_leaderboardType

Field Value

TYPE	DESCRIPTION
Text	

### \_membersButton

Button used to change the current leaderboard filter to 'Group Members'. Can be left null.

Declaration

**protected** Button \_membersButton

Field Value

TYPE	DESCRIPTION
Button	

### \_nearButton

Button used to change the current leaderboard filter to 'Near'. Can be left null.

Declaration

**protected** Button \_nearButton

Field Value

TYPE	DESCRIPTION
Button	

### \_topButton

Button used to change the current leaderboard filter to 'Top'. Can be left null.

Declaration

**protected** Button \_topButton

Field Value

TYPE	DESCRIPTION
Button	

## Methods

### Awake()

Base Awake method adds onClick listeners for the close, signin, top, near and friends filter buttons.

Declaration

```
protected override void Awake()
```

Overrides

[BaseInterface.Awake\(\)](#)

### ErrorDraw(Boolean)

Used to set error text in case of no user being signed in, loading issues or if no results are available. Filter button interactable set to false if no user is signed in or loading issues occur.

Declaration

```
protected override void ErrorDraw(bool loadingSuccess)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	loadingSuccess	

Overrides

[BaseInterface.ErrorDraw\(Boolean\)](#)

### HideInterfaces()

Hides Account, Evaluation, UserFriend, GroupMember and UserGroup UI objects. Makes filter buttons interactable.

Declaration

```
protected override void HideInterfaces()
```

Overrides

[BaseInterface.HideInterfaces\(\)](#)

### LoadErrorText()

Get error string from Localization with key "LEADERBOARD\_LOAD\_ERROR" if there were issues loading the leaderboard standings list.

Declaration

```
protected override string LoadErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.LoadErrorText\(\)](#)

NoResultsErrorText()

Get error string from Localization with key "NO\_LEADERBOARD\_ERROR" if there were no leaderboard standings to display.

Declaration

```
protected override string NoResultsErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.NoResultsErrorText\(\)](#)

# Class BaseLeaderboardListInterface

Base abstract class for controlling the UI object related to getting all leaderboards for a game

Inheritance

System.Object

[BaseInterface](#)

BaseLeaderboardListInterface

Inherited Members

[BaseInterface.\\_errorText](#)

[BaseInterface.\\_closeButton](#)

[BaseInterface.\\_signinButton](#)

[BaseInterface.PreDisplay\(\)](#)

[BaseInterface.Show\(Boolean\)](#)

[BaseInterface.Draw\(\)](#)

[BaseInterface.OnSignIn\(\)](#)

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseLeaderboardListInterface : BaseInterface
```

## Fields

### \_combinedButton

Button used to change the current actor type filter to 'Combined'. Can be left null.

Declaration

```
protected Button _combinedButton
```

Field Value

TYPE	DESCRIPTION
Button	

### \_groupButton

Button used to change the current actor type filter to 'Group'. Can be left null.

Declaration

```
protected Button _groupButton
```

Field Value

TYPE	DESCRIPTION
Button	

### \_leaderboardType

Text used for displaying current leaderboard type. Can be left null.

Declaration

```
protected Text _leaderboardType
```

Field Value

TYPE	DESCRIPTION
Text	

## \_userButton

Button used to change the current actor type filter to 'User'. Can be left null.

Declaration

```
protected Button _userButton
```

Field Value

TYPE	DESCRIPTION
Button	

## Methods

### Awake()

Base Awake method adds onClick listeners for the close, signin, user, group and combined filter buttons.

Declaration

```
protected override void Awake()
```

Overrides

[BaseInterface.Awake\(\)](#)

### ErrorDraw(Boolean)

Used to set error text in case of no user being signed in, loading issues or if no leaderboards are available. Filter button interactable set to false if no user is signed in or loading issues occur.

Declaration

```
protected override void ErrorDraw(bool loadingSuccess)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	loadingSuccess	

Overrides

[BaseInterface.ErrorDraw\(Boolean\)](#)

### HideInterfaces()

Hides Account, Evaluation, UserFriend, GroupMember and UserGroup UI objects. Makes filter buttons interactable.

Declaration

```
protected override void HideInterfaces()
```

Overrides

[BaseInterface.HideInterfaces\(\)](#)

LoadErrorText()

Get error string from Localization with key "LEADERBOARD\_LIST\_LOAD\_ERROR" if there were issues loading the leaderboard list.

Declaration

```
protected override string LoadErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.LoadErrorText\(\)](#)

NoResultsErrorText()

Get error string from Localization with key "NO\_LEADERBOARD\_LIST\_ERROR" if there were no leaderboards to display.

Declaration

```
protected override string NoResultsErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.NoResultsErrorText\(\)](#)

# Class BaseUnityClient<T>

Base abstract class for UnityClient classes

Inheritance

- System.Object
- BaseUnityClient<T>
- EvaluationUnityClient
- GroupMemberUnityClient
- LeaderboardListUnityClient
- LeaderboardUnityClient
- UserFriendUnityClient
- UserGroupUnityClient

Namespace: [PlayGen.SUGAR.Unity](#)  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseUnityClient<T> : MonoBehaviour where T : BaseInterface
```

Type Parameters

NAME	DESCRIPTION
T	

Fields

\_landscapeInterface

Landscape interface for this area of functionality. Can be left null if not required.

Declaration

```
protected T _landscapeInterface
```

Field Value

TYPE	DESCRIPTION
T	

\_portraitInterface

Portrait interface for this area of functionality. Can be left null if not required.

Declaration

```
protected T _portraitInterface
```

Field Value

TYPE	DESCRIPTION
T	

Properties

\_interface

The interface that is used for the current aspect ratio.

Declaration

```
protected T _interface { get; }
```

Property Value

TYPE	DESCRIPTION
T	

## HasInterface

Has an interface been provided for this Unity Client?

Declaration

```
public bool HasInterface { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

## IsActive

Is there an interface and if so is it currently active?

Declaration

```
public bool IsActive { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

## Methods

### Hide()

Hide the UI object if it is currently active.

Declaration

```
public void Hide()
```

### Update()

Change the used interface if the aspect ratio changes.

Declaration

```
protected virtual void Update()
```



# Class BaseUserFriendInterface

Base abstract class for controlling the UI object related to friends lists

Inheritance

System.Object

[BaseInterface](#)

BaseUserFriendInterface

Inherited Members

[BaseInterface.\\_errorText](#)

[BaseInterface.\\_closeButton](#)

[BaseInterface.\\_signinButton](#)

[BaseInterface.Awake\(\)](#)

[BaseInterface.PreDisplay\(\)](#)

[BaseInterface.Show\(Boolean\)](#)

[BaseInterface.Draw\(\)](#)

[BaseInterface.ErrorDraw\(Boolean\)](#)

[BaseInterface.OnSignIn\(\)](#)

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseUserFriendInterface : BaseInterface
```

## Methods

### GetFriends()

Get and display the friends list for the currently signed in user.

Declaration

```
protected void GetFriends()
```

### GetPendingReceived()

Get and display the list of pending received friend requests for the currently signed in user.

Declaration

```
protected void GetPendingReceived()
```

### GetPendingSent()

Get and display the list of pending sent friend requests for the currently signed in user.

Declaration

```
protected void GetPendingSent()
```

### GetSearchResults(String)

Get and display the search results for the provided string.

Declaration

```
protected void GetSearchResults(string search)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	search	

## HideInterfaces()

Hides Account, GameLeaderboard, Leaderboard, Evaluation, GroupMember and UserGroup UI objects.

Declaration

```
protected override void HideInterfaces()
```

Overrides

[BaseInterface.HideInterfaces\(\)](#)

## LoadErrorText()

Get error string from Localization with key "FRIENDS\_LOAD\_ERROR" if there were issues loading the friends list.

Declaration

```
protected override string LoadErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.LoadErrorText\(\)](#)

## NoResultsErrorText()

Get error string from Localization with key "NO\_RESULTS\_ERROR" if there were no friends to display.

Declaration

```
protected override string NoResultsErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.NoResultsErrorText\(\)](#)

# Class BaseUserGroupInterface

Base abstract class for controlling the UI object related to user groups.

Inheritance

System.Object  
BaseInterface  
BaseUserGroupInterface

Inherited Members

BaseInterface.\_errorText  
BaseInterface.\_closeButton  
BaseInterface.\_signinButton  
BaseInterface.Awake()  
BaseInterface.PreDisplay()  
BaseInterface.Show(Boolean)  
BaseInterface.Draw()  
BaseInterface.ErrorDraw(Boolean)  
BaseInterface.OnSignIn()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class BaseUserGroupInterface : BaseInterface
```

Methods

GetGroups()

Get and display the list of groups the currently signed in user is in.

Declaration

```
protected void GetGroups()
```

GetPendingSent()

Get and display the list of groups the currently signed in user has applied to join.

Declaration

```
protected void GetPendingSent()
```

GetSearchResults(String)

Get and display the list of groups that match the search string provided.

Declaration

```
protected void GetSearchResults(string search)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	search	

HideInterfaces()

Hides Account, Evaluation, Leaderboard, GameLeaderboard and UserFriend UI objects.

Declaration

```
protected override void HideInterfaces()
```

Overrides

[BaseInterface.HideInterfaces\(\)](#)

LoadErrorText()

Get error string from Localization with key "GROUPS\_LOAD\_ERROR" if there were issues loading the group list.

Declaration

```
protected override string LoadErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.LoadErrorText\(\)](#)

NoResultsErrorText()

Get error string from Localization with key "NO\_RESULTS\_ERROR" if there were no groups to display.

Declaration

```
protected override string NoResultsErrorText()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

[BaseInterface.NoResultsErrorText\(\)](#)

# Class CommandLineOptions

Inheritance

System.Object

CommandLineOptions

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class CommandLineOptions : object
```

## Properties

### AuthenticationSource

Declaration

```
public string AuthenticationSource { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

### AutoLogin

Declaration

```
public bool AutoLogin { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

### ClassId

Declaration

```
public string ClassId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

### Custom

Declaration

```
public string Custom { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Password

Declaration

```
public string Password { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

UserId

Declaration

```
public string UserId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

# Class CommandLineUtility

Inheritance

System.Object

CommandLineUtility

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public static class CommandLineUtility : object
```

## Fields

## CustomArgs

Declaration

```
public static Dictionary<string, string> CustomArgs
```

Field Value

TYPE	DESCRIPTION
Dictionary<System.String, System.String>	

## Methods

## ParseArgs(String[])

Declaration

```
public static CommandLineOptions ParseArgs(string[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	args	

Returns

TYPE	DESCRIPTION
<a href="#">CommandLineOptions</a>	

# Class Config

Inheritance

System.Object  
Config

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class Config : object
```

## Properties

BaseUri

Declaration

```
public string BaseUri { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	



# Class CustomInterface

Inheritance

System.Object

CustomInterface

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class CustomInterface : object
```

## Fields

### GameObject

Declaration

```
public GameObject GameObject
```

Field Value

TYPE	DESCRIPTION
GameObject	

## Name

Declaration

```
public string Name
```

Field Value

TYPE	DESCRIPTION
System.String	

# Class EvaluationUnityClient

Unity client for calls related to evaluations.

Inheritance

System.Object  
BaseUnityClient<BaseEvaluationListInterface>  
EvaluationUnityClient

Inherited Members

BaseUnityClient<BaseEvaluationListInterface>.\_landscapeInterface  
BaseUnityClient<BaseEvaluationListInterface>.\_portraitInterface  
BaseUnityClient<BaseEvaluationListInterface>.\_interface  
BaseUnityClient<BaseEvaluationListInterface>.HasInterface  
BaseUnityClient<BaseEvaluationListInterface>.IsActive  
BaseUnityClient<BaseEvaluationListInterface>.Hide()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class EvaluationUnityClient : BaseUnityClient<BaseEvaluationListInterface>
```

## Properties

### Progress

Current completion status for evaluations in this application for this user.

Declaration

```
public List<EvaluationProgressResponse> Progress { get; }
```

Property Value

TYPE	DESCRIPTION
List<EvaluationProgressResponse>	

## Methods

### DisplayAchievementList()

Gathers current user achievement completion status and displays interface if provided.

Declaration

```
public void DisplayAchievementList()
```

### DisplayGroupAchievementList()

Gathers current group achievement completion status and displays interface if provided.

Declaration

```
public void DisplayGroupAchievementList()
```

### DisplayGroupSkillList()

Gathers current group skill completion status and displays interface if provided.

#### Declaration

```
public void DisplayGroupSkillList()
```

#### DisplaySkillList()

Gathers current user skill completion status and displays interface if provided.

#### Declaration

```
public void DisplaySkillList()
```

#### ForceNotification(String)

Force an evaluation notification to be displayed with the provided text.

#### Declaration

```
public void ForceNotification(string notification = "Test Notification")
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	notification	String which will be used in the notification.

#### Update()

Change the used interfaces if the aspect ratio changes.

#### Declaration

```
protected override void Update()
```

#### Overrides

PlayGen.SUGAR.Unity.BaseUnityClient<PlayGen.SUGAR.Unity.BaseEvaluationListInterface>.Update()

# Class GameDataUnityClient

Get and send GameData for the currently signed in user.

Inheritance

System.Object

GameDataUnityClient

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class GameDataUnityClient : object
```

## Methods

Get(Action<IEnumerable<EvaluationDataResponse>>, String[])

Get GameData for the currently signed in user for this game.

Declaration

```
public void Get(Action<IEnumerable<EvaluationDataResponse>> success, string[] keys = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<IEnumerable<EvaluationDataResponse>>	success	Callback which will return the list of gathered results.
System.String[]	keys	Optional parameter. If provided, only GameData with a matching key will be returned.

GetCount(String, EvaluationDataType, Action<EvaluationDataResponse>)

Get the count of recorded values for the currently signed in user for the key and dataType provided.

Declaration

```
public void GetCount(string key, EvaluationDataType dataType, Action<EvaluationDataResponse> success)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	success	Callback which will contain the gathered result.

GetCumulative(String, EvaluationDataType, Action<EvaluationDataResponse>)

Get the cumulative value for the currently signed in user for the key and dataType provided.

## Declaration

```
public void GetCumulative(string key, EvaluationDataType dataType, Action<EvaluationDataResponse> success)
```

## Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	success	Callback which will contain the gathered result.

GetEarliest(String, EvaluationDataType, Action<EvaluationDataResponse>)

Get the earliest recorded data for the currently signed in user for the key and dataType provided.

## Declaration

```
public void GetEarliest(string key, EvaluationDataType dataType, Action<EvaluationDataResponse> success)
```

## Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	success	Callback which will contain the gathered result.

GetHighest(String, EvaluationDataType, Action<EvaluationDataResponse>)

Get the data related to the highest value recorded for the currently signed in user for the key and dataType provided.

## Declaration

```
public void GetHighest(string key, EvaluationDataType dataType, Action<EvaluationDataResponse> success)
```

## Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.

TYPE	NAME	DESCRIPTION
Action<EvaluationDataResponse>	success	Callback which will contain the gathered result.

**GetLatest(String, EvaluationDataType, Action<EvaluationDataResponse>)**

Get the latest recorded data for the currently signed in user for the key and dataType provided.

Declaration

```
public void GetLatest(string key, EvaluationDataType dataType, Action<EvaluationDataResponse> success)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	success	Callback which will contain the gathered result.

**GetLowest(String, EvaluationDataType, Action<EvaluationDataResponse>)**

Get the data related to the lowest value recorded for the currently signed in user for the key and dataType provided.

Declaration

```
public void GetLowest(string key, EvaluationDataType dataType, Action<EvaluationDataResponse> success)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	success	Callback which will contain the gathered result.

**Send(String, Boolean)**

Record GameData with EvaluationDataType Bool with the key and value provided.

Declaration

```
public void Send(string key, bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.Boolean	value	The Bool value that'll be recorded.

Send(String, Int64)

Record GameData with EvaluationDataType Long with the key and value provided.

Declaration

```
public void Send(string key, long value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.Int64	value	The Long value that'll be recorded.

Send(String, Single)

Record GameData with EvaluationDataType Float with the key and value provided.

Declaration

```
public void Send(string key, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.Single	value	The Float value that'll be recorded.

Send(String, String)

Record GameData with EvaluationDataType String with the key and value provided.

Declaration

```
public void Send(string key, string value)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.String	value	The String value that'll be recorded.



# Class GroupMemberUnityClient

Unity client for calls related to group member lists.

Inheritance

System.Object  
BaseUnityClient<BaseGroupMemberInterface>  
GroupMemberUnityClient

Inherited Members

BaseUnityClient<BaseGroupMemberInterface>.\_landscapeInterface  
BaseUnityClient<BaseGroupMemberInterface>.\_portraitInterface  
BaseUnityClient<BaseGroupMemberInterface>.\_interface  
BaseUnityClient<BaseGroupMemberInterface>.HasInterface  
BaseUnityClient<BaseGroupMemberInterface>.IsActive  
BaseUnityClient<BaseGroupMemberInterface>.Update()  
BaseUnityClient<BaseGroupMemberInterface>.Hide()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class GroupMemberUnityClient : BaseUnityClient<BaseGroupMemberInterface>
```

## Properties

### CurrentGroup

Currently selected/displayed group.

Declaration

```
public ActorResponse CurrentGroup { get; }
```

Property Value

TYPE	DESCRIPTION
ActorResponse	

## Members

Member list for the current group.

Declaration

```
public List<ActorResponseAllowableActions> Members { get; }
```

Property Value

TYPE	DESCRIPTION
List<ActorResponseAllowableActions>	

## Methods

### AddFriend(Int32, Boolean)

Send friend request to user with id provided. If reload is true, UI is also redrawn.

#### Declaration

```
public void AddFriend(int id, bool reload = true)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	reload	

#### Display(ActorResponse)

Sets current group and gathers member list for that group. Displays UI object if provided.

#### Declaration

```
public void Display(ActorResponse group)
```

#### Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	group	

# Class LeaderboardListUnityClient

Unity client for calls related to leaderboards for an application.

Inheritance

System.Object  
BaseUnityClient<BaseLeaderboardListInterface>  
LeaderboardListUnityClient

Inherited Members

BaseUnityClient<BaseLeaderboardListInterface>.\_landscapeInterface  
BaseUnityClient<BaseLeaderboardListInterface>.\_portraitInterface  
BaseUnityClient<BaseLeaderboardListInterface>.\_interface  
BaseUnityClient<BaseLeaderboardListInterface>.HasInterface  
BaseUnityClient<BaseLeaderboardListInterface>.IsActive  
BaseUnityClient<BaseLeaderboardListInterface>.Update()  
BaseUnityClient<BaseLeaderboardListInterface>.Hide()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class LeaderboardListUnityClient : BaseUnityClient<BaseLeaderboardListInterface>
```

## Properties

### CurrentActorType

Currently used ActorType filter.

Declaration

```
public ActorType CurrentActorType { get; }
```

Property Value

TYPE	DESCRIPTION
ActorType	

## Leaderboards

List of leaderboards for this application for each ActorType filter.

Declaration

```
public Dictionary<ActorType, List<LeaderboardResponse>> Leaderboards { get; }
```

Property Value

TYPE	DESCRIPTION
Dictionary<ActorType, List<LeaderboardResponse>>	

## Methods

### DisplayGameList(ActorType)

Gathers leaderboards for this application and displays list for current ActorType if UI object if provided.

#### Declaration

```
public void DisplayGameList(ActorType filter = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION
ActorType	filter	

### DisplayGlobalList(ActorType)

Gathers leaderboards for this application and displays list for current ActorType if UI object if provided.

#### Declaration

```
public void DisplayGlobalList(ActorType filter = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION
ActorType	filter	

### SetFilter(ActorType)

Set the ActorType filter to use.

#### Declaration

```
public void SetFilter(ActorType filter)
```

#### Parameters

TYPE	NAME	DESCRIPTION
ActorType	filter	

# Class LeaderboardUnityClient

Unity client for calls related to leaderboard standings.

Inheritance

System.Object  
BaseUnityClient<BaseLeaderboardInterface>  
LeaderboardUnityClient

Inherited Members

BaseUnityClient<BaseLeaderboardInterface>.\_landscapeInterface  
BaseUnityClient<BaseLeaderboardInterface>.\_portraitInterface  
BaseUnityClient<BaseLeaderboardInterface>.\_interface  
BaseUnityClient<BaseLeaderboardInterface>.HasInterface  
BaseUnityClient<BaseLeaderboardInterface>.IsActive  
BaseUnityClient<BaseLeaderboardInterface>.Update()  
BaseUnityClient<BaseLeaderboardInterface>.Hide()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class LeaderboardUnityClient : BaseUnityClient<BaseLeaderboardInterface>
```

## Properties

### CurrentFilter

Current filter to use for gathering leaderboard standings.

Declaration

```
public LeaderboardFilterType CurrentFilter { get; }
```

Property Value

TYPE	DESCRIPTION
LeaderboardFilterType	

### CurrentLeaderboard

Current leaderboard to use for gathering leaderboard standings from.

Declaration

```
public LeaderboardResponse CurrentLeaderboard { get; }
```

Property Value

TYPE	DESCRIPTION
LeaderboardResponse	

### CurrentStandings

Last set of standings gathered.

Declaration

```
public List<LeaderboardStandingsResponse> CurrentStandings { get; }
```

#### Property Value

TYPE	DESCRIPTION
List<LeaderboardStandingsResponse>	

#### PositionCount

Number of results that should be gathered per call.

#### Declaration

```
public int PositionCount { get; }
```

#### Property Value

TYPE	DESCRIPTION
System.Int32	

#### Methods

##### Display(String, LeaderboardFilterType, Int32, Boolean)

Gathers information on the leaderboard with the token provided and gets current standings based on the filter and page number provided, with the UI object displayed if provided.

#### Declaration

```
public void Display(string token, LeaderboardFilterType filter, int pageNumber = 0, bool globalLeaderboard = false)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	token	
LeaderboardFilterType	filter	
System.Int32	pageNumber	
System.Boolean	globalLeaderboard	

##### GetLeaderboardStandings(Int32, Action<Boolean>, Action<List<LeaderboardStandingsResponse>>)

Get standings for the current leaderboard. A request for results to be returned means that the standings gotten will not be stored. Otherwise, they will be saved into CurrentStandings.

#### Declaration

```
public void GetLeaderboardStandings(int pageNumber, Action<bool> success, Action<List<LeaderboardStandingsResponse>> result = null)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Int32	pageNumber	
Action<System.Boolean>	success	
Action<List<LeaderboardStandingsResponse>>	result	

SetPositionCount(Int32)

Set the number of results to get at most per call.

Declaration

```
public void SetPositionCount(int count)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	count	

# Class ResourceUnityClient

Unity client for calls related to resources.

Inheritance

System.Object

ResourceUnityClient

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class ResourceUnityClient : MonoBehaviour
```

## Properties

### GlobalUserResources

Resources for the user not tied to any game.

Declaration

```
public Dictionary<string, long> GlobalUserResources { get; }
```

Property Value

TYPE	DESCRIPTION
Dictionary<System.String, System.Int64>	

### UserGameResources

Resources for the currently signed in user for this game.

Declaration

```
public Dictionary<string, long> UserGameResources { get; }
```

Property Value

TYPE	DESCRIPTION
Dictionary<System.String, System.Int64>	

## Methods

### Add(String, Int64, Action<Boolean>, Boolean)

Add the resource with the key provided from the currently signed in user. If globalResource is true, resource transferred will be global rather than for the game.

Declaration

```
public void Add(string key, long amount, Action<bool> success, bool globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	



TYPE	NAME	DESCRIPTION
System.Int64	amount	
Action<System.Boolean>	success	
System.Boolean	globalResource	

**Get(Action<List<ResourceResponse>>, String[], Boolean)**

Get the resources with the keys provided for the signed in user. If globalResource is true, resources will be global rather than for the game.

Declaration

```
public void Get(Action<List<ResourceResponse>> result, string[] keys = null, bool globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<List<ResourceResponse>>	result	
System.String[]	keys	
System.Boolean	globalResource	

**Transfer(Int32, String, Int64, Action<Boolean>, Boolean)**

Transfer the resource with the key provided from the currently signed in user to the user with the provided id. If globalResource is true, resource transferred will be global rather than for the game.

Declaration

```
public void Transfer(int recipientId, string key, long amount, Action<bool> success, bool globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	recipientId	
System.String	key	
System.Int64	amount	
Action<System.Boolean>	success	
System.Boolean	globalResource	

**TryTake(Int32, String, Int64, Action<Boolean>, Boolean)**

Attempt to Take a resource from an actor. If globalResource is true, resource transferred will be global rather than for the game.

Declaration

```
public void TryTake(int senderId, string key, long amount, Action<bool> success, bool globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	senderId	
System.String	key	
System.Int64	amount	
Action<System.Boolean>	success	
System.Boolean	globalResource	

# Class ResponseHandler

Inheritance

System.Object

ResponseHandler

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class ResponseHandler : MonoBehaviour
```

# Class SavedPrefsHandler

Inheritance

System.Object

SavedPrefsHandler

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class SavedPrefsHandler : ISavedPrefsHandler
```

## Properties

### Prefix

Declaration

```
public string Prefix { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

## Methods

### Delete(String)

Declaration

```
public void Delete(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

### Get<T>(String)

Declaration

```
public T Get<T>(string key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	

Returns

TYPE	DESCRIPTION
T	

Type Parameters

NAME	DESCRIPTION
T	

Save<T>(String, T)

Declaration

```
public void Save<T>(string key, T value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	
T	value	

Type Parameters

NAME	DESCRIPTION
T	

# Class SUGARManager

Access point for SUGAR related classes.

Inheritance

System.Object

SUGARManager

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public static class SUGARManager : object
```

## Properties

### Account

Unity client for calls related to accounts

Declaration

```
public static AccountUnityClient Account { get; }
```

Property Value

TYPE	DESCRIPTION
<a href="#">AccountUnityClient</a>	

### ClassId

Group name gathered from auto sign in.

Declaration

```
public static string ClassId { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

### Client

Class for contacting SUGAR client functionality

Declaration

```
public static SUGARClient Client { get; }
```

Property Value

TYPE	DESCRIPTION
SUGARClient	

### CurrentGroup

Currently signed in user's primary group.

## Declaration

```
public static ActorResponse CurrentGroup { get; }
```

## Property Value

TYPE	DESCRIPTION
ActorResponse	

## CurrentUser

Currently signed in user.

## Declaration

```
public static ActorResponse CurrentUser { get; }
```

## Property Value

TYPE	DESCRIPTION
ActorResponse	

## Evaluation

Unity client for calls related to evaluations

## Declaration

```
public static EvaluationUnityClient Evaluation { get; }
```

## Property Value

TYPE	DESCRIPTION
<a href="#">EvaluationUnityClient</a>	

## GameData

Unity client for calls related to gamedata

## Declaration

```
public static GameDataUnityClient GameData { get; }
```

## Property Value

TYPE	DESCRIPTION
<a href="#">GameDataUnityClient</a>	

## GameId

GameId for this application.

## Declaration

```
public static int GameId { get; }
```

## Property Value

TYPE	DESCRIPTION
System.Int32	

GameLeaderboard

Unity client for calls related to leaderboard lists

Declaration

```
public static LeaderboardListUnityClient GameLeaderboard { get; }
```

Property Value

TYPE	DESCRIPTION
LeaderboardListUnityClient	

GroupMember

Unity client for calls related to group members

Declaration

```
public static GroupMemberUnityClient GroupMember { get; }
```

Property Value

TYPE	DESCRIPTION
GroupMemberUnityClient	

Leaderboard

Unity client for calls related to leaderboard standings

Declaration

```
public static LeaderboardUnityClient Leaderboard { get; }
```

Property Value

TYPE	DESCRIPTION
LeaderboardUnityClient	

Resource

Unity client for calls related to resources

Declaration

```
public static ResourceUnityClient Resource { get; }
```

Property Value

TYPE	DESCRIPTION
ResourceUnityClient	



## Unity

Class for managing Unity elements of the asset

Declaration

```
public static SUGARUnityManager Unity { get; }
```

Property Value

TYPE	DESCRIPTION
<a href="#">SUGARUnityManager</a>	

## UserFriend

Unity client for calls related to friend lists

Declaration

```
public static UserFriendUnityClient UserFriend { get; }
```

Property Value

TYPE	DESCRIPTION
<a href="#">UserFriendUnityClient</a>	

## UserGroup

Unity client for calls related to user groups

Declaration

```
public static UserGroupUnityClient UserGroup { get; }
```

Property Value

TYPE	DESCRIPTION
<a href="#">UserGroupUnityClient</a>	

## UserSignedIn

Is there a user currently signed in.

Declaration

```
public static bool UserSignedIn { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

## Methods

### SetClassId(String)

Set the ClassId for the currently signed in user

Declaration

```
public static void SetClassId(string classid)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	classid	

SetCurrentGroup(ActorResponse)

Set the 'primary' group for the currently signed in user

Declaration

```
public static void SetCurrentGroup(ActorResponse group)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	group	

# Class SUGARUnityManager

Class for managing Unity elements of the asset

Inheritance

System.Object

SUGARUnityManager

Namespace: [PlayGen.SUGAR.Unity](#)

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class SUGARUnityManager : MonoBehaviour
```

Fields

CustomInterfaces

Declaration

```
public Dictionary<string, GameObject> CustomInterfaces
```

Field Value

TYPE	DESCRIPTION
Dictionary<System.String, GameObject>	

Properties

AnyActiveUI

Is any piece of SUGAR UI currently active?

Declaration

```
public bool AnyActiveUI { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

SpinnerActive

Declaration

```
public bool SpinnerActive { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

CreateSUGARClient(String)

Declaration

```
protected virtual SUGARClient CreateSUGARClient(string baseAddress)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	baseAddress	

#### Returns

TYPE	DESCRIPTION
SUGARClient	

### DisableObject(GameObject)

Disable a piece of SUGAR UI. This should be used instead of SetActive to ensure UI and blocker ordering is correct.

#### Declaration

```
public void DisableObject(GameObject activeObject)
```

#### Parameters

TYPE	NAME	DESCRIPTION
GameObject	activeObject	

### EnableObject(GameObject)

Enable a piece of SUGAR UI. This should be used instead of SetActive to ensure UI and blocker ordering is correct.

#### Declaration

```
public void EnableObject(GameObject activeObject)
```

#### Parameters

TYPE	NAME	DESCRIPTION
GameObject	activeObject	

### GameValidityCheck()

#### Declaration

```
public bool GameValidityCheck()
```

#### Returns

TYPE	DESCRIPTION
System.Boolean	

### SetBlocker(Boolean, Boolean)

Set if the blocker should be used and if it should close objects when clicked

#### Declaration

```
public void SetBlocker(bool use, bool block)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	use	
System.Boolean	block	

### SetSpinner(Boolean, Int32)

Set the direction and speed of the loading spinner

#### Declaration

```
public void SetSpinner(bool clockwise, int speed)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	clockwise	
System.Int32	speed	

### StartSpinner(String)

Start the loading spinner. This method should be used instead of directly calling Loading.Start to ensure UI and blocker ordering is correct.

#### Declaration

```
public void StartSpinner(string text = "")
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	text	

### StopSpinner(String, Single)

Stop the loading spinner. This method should be used instead of directly calling Loading.Stop to ensure UI and blocker ordering is correct.

#### Declaration

```
public void StopSpinner(string text = "", float stopDelay = 0F)
```

#### Parameters

TYPE	NAME	DESCRIPTION
System.String	text	
System.Single	stopDelay	

# Class UserFriendUnityClient

Unity client for calls related to user friends.

Inheritance

System.Object  
BaseUnityClient<BaseUserFriendInterface>  
UserFriendUnityClient

Inherited Members

BaseUnityClient<BaseUserFriendInterface>.\_landscapeInterface  
BaseUnityClient<BaseUserFriendInterface>.\_portraitInterface  
BaseUnityClient<BaseUserFriendInterface>.\_interface  
BaseUnityClient<BaseUserFriendInterface>.HasInterface  
BaseUnityClient<BaseUserFriendInterface>.IsActive  
BaseUnityClient<BaseUserFriendInterface>.Update()  
BaseUnityClient<BaseUserFriendInterface>.Hide()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class UserFriendUnityClient : BaseUnityClient<BaseUserFriendInterface>
```

## Properties

### Friends

Friends list for currently signed in user.

Declaration

```
public List<ActorResponseAllowableActions> Friends { get; }
```

Property Value

TYPE	DESCRIPTION
List<ActorResponseAllowableActions>	

### PendingReceived

Pending received friend requests for currently signed in user.

Declaration

```
public List<ActorResponseAllowableActions> PendingReceived { get; }
```

Property Value

TYPE	DESCRIPTION
List<ActorResponseAllowableActions>	

### PendingSent

Pending sent friend requests for currently signed in user.

Declaration

```
public List<ActorResponseAllowableActions> PendingSent { get; }
```

Property Value

TYPE	DESCRIPTION
List< <a href="#">ActorResponseAllowableActions</a> >	

## SearchResults

Last set of search results.

Declaration

```
public List<ActorResponseAllowableActions> SearchResults { get; }
```

Property Value

TYPE	DESCRIPTION
List< <a href="#">ActorResponseAllowableActions</a> >	

## Methods

### AddFriend(Int32, Boolean)

Send friend request to user with id provided. If reload is true, UI is also redrawn.

Declaration

```
public void AddFriend(int id, bool reload = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	reload	

### Display()

Updates lists and displays UI object if provided.

Declaration

```
public void Display()
```

### GetFriendsList(Action<Boolean>)

Get friends list for the currently signed in user.

Declaration

```
public void GetFriendsList(Action<bool> success)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	success	

ManageFriendRequest(Int32, Boolean, Boolean, Boolean)

Resolve friend request to user with id provided. If reload is true, UI is also redrawn. Reverse should be true if cancelling sent request. Accept and reverse cannot both be true.

Declaration

```
public void ManageFriendRequest(int id, bool accept, bool reverse = false, bool reload = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	accept	
System.Boolean	reverse	
System.Boolean	reload	

RemoveFriend(Int32, Boolean)

Remove user with id provided from friends list. If reload is true, UI is also redrawn.

Declaration

```
public void RemoveFriend(int id, bool reload = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	reload	



# Class UserGroupUnityClient

Unity client for calls related to group lists.

## Inheritance

System.Object  
BaseUnityClient<BaseUserGroupInterface>  
UserGroupUnityClient

## Inherited Members

BaseUnityClient<BaseUserGroupInterface>.\_landscapeInterface  
BaseUnityClient<BaseUserGroupInterface>.\_portraitInterface  
BaseUnityClient<BaseUserGroupInterface>.\_interface  
BaseUnityClient<BaseUserGroupInterface>.HasInterface  
BaseUnityClient<BaseUserGroupInterface>.IsActive  
BaseUnityClient<BaseUserGroupInterface>.Update()  
BaseUnityClient<BaseUserGroupInterface>.Hide()

Namespace: PlayGen.SUGAR.Unity  
Assembly: PlayGen.SUGAR.Unity.dll

## Syntax

```
public class UserGroupUnityClient : BaseUnityClient<BaseUserGroupInterface>
```

## Properties

### Groups

List of groups that the currently signed in user is a member of.

#### Declaration

```
public List<ActorResponseAllowableActions> Groups { get; }
```

#### Property Value

TYPE	DESCRIPTION
List<ActorResponseAllowableActions>	

### PendingSent

List of groups that the currently signed in user has applied to join.

#### Declaration

```
public List<ActorResponseAllowableActions> PendingSent { get; }
```

#### Property Value

TYPE	DESCRIPTION
List<ActorResponseAllowableActions>	

### SearchResults

List of groups that matched the last search string.

#### Declaration

```
public List<ActorResponseAllowableActions> SearchResults { get; }
```

Property Value

TYPE	DESCRIPTION
List< <a href="#">ActorResponseAllowableActions</a> >	

## Methods

### AddGroup(Int32, Boolean)

Send group membership request to group with id provided. If reload is true, UI is also redrawn.

Declaration

```
public void AddGroup(int id, bool reload = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	reload	

### Display()

Gathers updated versions of each list and displays UI object if provided.

Declaration

```
public void Display()
```

### GetGroupsList(Action<Boolean>)

Get group list for the currently signed in user.

Declaration

```
public void GetGroupsList(Action<bool> success)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	success	

### ManageGroupRequest(Int32, Boolean)

Cancel sent membership request to group with id provided. If reload is true, UI is also redrawn.

Declaration

```
public void ManageGroupRequest(int id, bool reload = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	reload	

RemoveGroup(Int32, Boolean)

Leave group with id provided. If reload is true, UI is also redrawn.

Declaration

```
public void RemoveGroup(int id, bool reload = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	
System.Boolean	reload	

# Namespace PlayGen.SUGAR.Unity.Editor

## Classes

[AutoLogin](#)

[EditGameSeed](#)

[EditGameSeedWindow](#)

[SeedGame](#)

[SeedGameWindow](#)

[SetEditorAutoLogin](#)

[SetEditorAutoLogin.AutoLoginOption](#)

[SetEditorAutoLogin.BoolValue](#)

[SetEditorAutoLogin.StringValue](#)

# Class AutoLogIn

Inheritance

System.Object

AutoLogIn

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public class AutoLogIn : EditorWindow
```

# Class EditGameSeed

Inheritance

System.Object

EditGameSeed

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public static class EditGameSeed : object
```

## Methods

ShowEditGameSeed()

Declaration

```
public static void ShowEditGameSeed()
```

# Class EditGameSeedWindow

Inheritance

System.Object

EditGameSeedWindow

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public class EditGameSeedWindow : EditorWindow
```

## Methods

SetGameSeed(TextAsset)

Declaration

```
public void SetGameSeed(TextAsset gameSeedText)
```

Parameters

TYPE	NAME	DESCRIPTION
TextAsset	gameSeedText	

# Class SeedGame

Inheritance

System.Object

SeedGame

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public static class SeedGame : object
```

## Properties

### DefaultGameSeed

Declaration

```
public static TextAsset DefaultGameSeed { get; }
```

Property Value

TYPE	DESCRIPTION
TextAsset	

## Methods

### ShowSeedGameWindow()

Declaration

```
public static void ShowSeedGameWindow()
```

### TryApplySeed(String, String, TextAsset, out List<String>)

Declaration

```
public static bool TryApplySeed(string username, string password, TextAsset gameSeedText, out List<string> errors)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	username	
System.String	password	
TextAsset	gameSeedText	
List<System.String>	errors	

Returns

TYPE	DESCRIPTION
System.Boolean	



# Class SeedGameWindow

Inheritance

System.Object

SeedGameWindow

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public class SeedGameWindow : EditorWindow
```

## Methods

SetGameSeed(TextAsset)

Declaration

```
public void SetGameSeed(TextAsset gameSeed)
```

Parameters

TYPE	NAME	DESCRIPTION
TextAsset	gameSeed	

# Class SetEditorAutoLogin

Inheritance

System.Object

SetEditorAutoLogin

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public static class SetEditorAutoLogin : object
```

## Fields

### AutoLoginOptions

Declaration

```
public static List<SetEditorAutoLogin.AutoLoginOption> AutoLoginOptions
```

Field Value

TYPE	DESCRIPTION
List< <a href="#">SetEditorAutoLogin.AutoLoginOption</a> >	

## Methods

### DependentValue(String)

Declaration

```
public static bool DependentValue(string dependingValueKey)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	dependingValueKey	

Returns

TYPE	DESCRIPTION
System.Boolean	

### SetAutoLogIn()

Declaration

```
public static void SetAutoLogIn()
```

# Class SetEditorAutoLogin.AutoLoginOption

Inheritance

System.Object  
SetEditorAutoLogin.AutoLoginOption  
[SetEditorAutoLogin.BoolValue](#)  
[SetEditorAutoLogin.StringValue](#)

Namespace: [PlayGen.SUGAR.Unity.Editor](#)  
Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public class AutoLoginOption : object
```

## Fields

### AutoLoginPrefix

Declaration

```
public string AutoLoginPrefix
```

Field Value

TYPE	DESCRIPTION
System.String	

### DependsOnValue

depends on value name must be the name of a boolean

Declaration

```
public string DependsOnValue
```

Field Value

TYPE	DESCRIPTION
System.String	

## Key

Declaration

```
public string Key
```

Field Value

TYPE	DESCRIPTION
System.String	

## Label

Declaration

```
public string Label
```

Field Value

TYPE	DESCRIPTION
System.String	

Required

Declaration

```
public bool Required
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

SugarRefName

Declaration

```
public string SugarRefName
```

Field Value

TYPE	DESCRIPTION
System.String	

# Class SetEditorAutoLogin.BoolValue

Inheritance

System.Object

[SetEditorAutoLogin.AutoLoginOption](#)

SetEditorAutoLogin.BoolValue

Inherited Members

[SetEditorAutoLogin.AutoLoginOption.Label](#)

[SetEditorAutoLogin.AutoLoginOption.Key](#)

[SetEditorAutoLogin.AutoLoginOption.SugarRefName](#)

[SetEditorAutoLogin.AutoLoginOption.Required](#)

[SetEditorAutoLogin.AutoLoginOption.AutoLoginPrefix](#)

[SetEditorAutoLogin.AutoLoginOption.DependsOnValue](#)

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public class BoolValue : SetEditorAutoLogin.AutoLoginOption
```

## Constructors

BoolValue(Boolean)

Declaration

```
public BoolValue(bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	value	

BoolValue(String, String, String, String, Boolean, Boolean)

Declaration

```
public BoolValue(string label, string key, string sugarRefName, string autoLoginPrefix, bool required = false, bool value = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	
System.String	key	
System.String	sugarRefName	
System.String	autoLoginPrefix	
System.Boolean	required	
System.Boolean	value	

Fields

Value

Declaration

```
public bool Value
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

# Class SetEditorAutoLogin.StringValue

Inheritance

System.Object

[SetEditorAutoLogin.AutoLoginOption](#)

SetEditorAutoLogin.StringValue

Inherited Members

[SetEditorAutoLogin.AutoLoginOption.Label](#)

[SetEditorAutoLogin.AutoLoginOption.Key](#)

[SetEditorAutoLogin.AutoLoginOption.SugarRefName](#)

[SetEditorAutoLogin.AutoLoginOption.Required](#)

[SetEditorAutoLogin.AutoLoginOption.AutoLoginPrefix](#)

[SetEditorAutoLogin.AutoLoginOption.DependsOnValue](#)

Namespace: [PlayGen.SUGAR.Unity.Editor](#)

Assembly: PlayGen.SUGAR.Unity.Editor.dll

Syntax

```
public class StringValue : SetEditorAutoLogin.AutoLoginOption
```

## Constructors

### StringValue(String)

Declaration

```
public StringValue(string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	

### StringValue(String, String, String, String, String, Boolean, Boolean, String)

Declaration

```
public StringValue(string label, string key, string sugarRefName, string autoLoginPrefix, string dependsOnValue = "", bool required = false, bool hidden = false, string value = "")
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	
System.String	key	
System.String	sugarRefName	
System.String	autoLoginPrefix	
System.String	dependsOnValue	
System.Boolean	required	

TYPE	NAME	DESCRIPTION
System.Boolean	hidden	
System.String	value	

Fields

Hidden

Declaration

```
public bool Hidden
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Value

Declaration

```
public string Value
```

Field Value

TYPE	DESCRIPTION
System.String	



# Namespace PlayGen.SUGAR.Unity.WebGL

## Classes

[UnityWebGLHttpHandler](#)

# Class UnityWebGLHttpHandler

Inheritance

System.Object

UnityWebGLHttpHandler

Namespace: [PlayGen.SUGAR.Unity.WebGL](#)

Assembly: PlayGen.SUGAR.Unity.WebGL.dll

Syntax

```
public class UnityWebGLHttpHandler : IHttpHandler
```

## Methods

HandleRequest(HttpRequest)

Declaration

```
public HttpResponse HandleRequest(HttpRequest request)
```

Parameters

TYPE	NAME	DESCRIPTION
HttpRequest	request	

Returns

TYPE	DESCRIPTION
HttpResponse	