Table of Contents

SUGAR-Unity Client

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

Online Documentation

You can fine 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.

This Section includes

- **Creating an Account**: How to set up your first account to get started with SUGAR
- **Quick Start**: Step by step guide to get your game setup using SUGAR

Once you have followed the steps in the pages above, see SUGARManager for details on how to use SUGAR in your unity game. The SUGAR Manager handles all of the logic and interactions between game and server.

Creating a new SUGAR account

The admin panel is built for admins to manage their games, so does not allow for accounts to be created unless you are logged in already. Therefore, to get started, you must create a new account by either:

- Logging in to the admin panel as an admin and creating new users at: Users > Create New User.
- Creating a new account using the C# API.

After creating a new SUGAR account, you will be able to sign in to both the admin panel and your game and see groups and games that are publicly available.

Quick Start

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

See creating accounts for more details.

Import the SUGAR Unity Client

Downlaod the SUGAR Unity package from the Unity Asset Store here.
OR
If you have downloaded the source code, follow the build instructions.

Add SUGAR to your game

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.

The SUGAR Unity Manager script component on the SUGAR prefab holds universal information required by other SUGAR components. Configuring this component is necessary to connect to SUGAR.

- Base Address - web address of SUGAR server (e.g. `http://localhost:59400/` or `http://www.mysugarserver.com`). This is overwritten by the value set inside the config.json file (found in "Assets/StreamingAssets").
- Game Token - name of the <xref:game> used for database lookup.
- Game Id - database row Id of the <xref:game>, returned from token lookup.
- Source - Account Source to log in to. Use "SUGAR" as default to get started quickly or if you don't need your own account source.

Creating your Game in SUGAR

New games can be created through the admin panel or through the C# API, but the simplest is to use a game seed through Unity. The seed file allows you to define the name, achievements, leaderboards and skills at the same time.

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.

Using SUGAR in your game

Once you have set up your game in SUGAR, you are now ready to log in and start using SUGAR features in your game. SUGAR functionality is controlled through the SUGARManager.

Before using SUGAR functionality during your game, a user should be logged in.

Logging in to SUGAR

By default when the SUGAR gameobject is present in the scene and no user is logged in, the login panel will be shown to the user. In the example scene this will happen as soon as the project is run. Alternatively you can set auto login values (for testing purposes).

Setting 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 + J	Evaluation.DisplaySkillList
T + A	Evaluation.ForceNotification
T + S	Unity.StartSpinner
T + H	Unity.StopSpinner
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*.

Development

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

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

Saving Data

SUGAR Unity includes a class which handles saving and retrieving of data, similar to the way in which Player Prefs are set, retrieved and deleted. The class uses the following prefix: **SUGAR_PREFS_** followed by the key provided.

Current Usage

Currently the saving and retrieving of data is used for storing player login tokens, allowing for players login details to be saved so they can log in without entering details every time

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 availabe 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

Use this to Sign In, Register, Logout and manage other account functionality

ActorResponseRelationshipStatus<T>

ActorResponse with additional information on the relationship between the current user and the actor.

ActorUnityClient

Unity Client for getting and updating groups and users and for creating and deleting groups.

BaseAccountInterface

Base abstract class for controlling the interface related to Account functionality.

BaseEvaluationListInterface

Base abstract class for controlling the interface related to displaying evaluation progress.

BaseEvaluationPopupInterface

Base abstract class for controlling the interface related to display evaluation notifications when an evalaution is completed.

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 interface related to displaying the latest standings for a leaderboard.

BaseLeaderboardListInterface

Base abstract class for controlling the interface related to displaying a list of leaderboards.

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

Use this for gathering evaluation progress and notifications when an evaluation is completed.

GameDataUnityClient

Use this to GET and POST data related to the game.

GroupMemberUnityClient

Use this for actions related to group member lists.

GroupResponseRelationshipStatus

ActorResponse with additional information on the relationship between the current user and the actor.

LeaderboardListUnityClient

Use this to get a list of leaderboards for this game

LeaderboardUnityClient

Use this to get the current standings for a leaderboard

ResourceUnityClient

Use this to get current resources, add resources and send resources to other users

ResponseHandler

SavedPrefsHandler

SUGARManager

Access point for SUGAR related classes.

SUGARUnityManager

Class for managing Unity elements of the asset

UserFriendUnityClient

Use this to get current user's list of friends and send and handle friend requests and other friend related actions

UserGroupUnityClient

Use this to get current user's list of groups and send and handle group requests

UserResponseRelationshipStatus

ActorResponse with additional information on the relationship between the current user and the actor.

Enums

RelationshipStatus

The different relationship states two actors can be in related to each other

Class AccountUnityClient

Use this to Sign In, Register, Logout and manage other account functionality

Inheritance

System.Object

AccountUnityClient

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class AccountUnityClient : MonoBehaviour
```

Properties

HasInterface

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Has an interface been provided for this Unity Client in the current orientation

HasSavedLogin

Declaration

```
public bool HasSavedLogin { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	Whether there are login details that were saved by a previously using "remember me".

IsActive

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Is there an interface and if so is it currently active

Methods

DisplayLogInPanel(Action<Boolean>)

Displays interface if provided and allowAutoLogin is false. Attempts automatic sign in using provided details if allowAutoLogin is

true.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Whether the user successfully signed in.

Remarks

- allowAutoLogin is set to false after automatic sign in is first attempted.
- If there is no interface provided callback will return false

Hide()

Hide the AccountPanel game object

Declaration

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

Logout(Action<Boolean>)

Sign out the currently signed in user.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Whether the currently signed in user successfully signed out.

Remarks

- If no user is currently signed in, callback returns false

Class ActorResponseRelationshipStatus<T>

ActorResponse with additional information on the relationship between the current user and the actor.

Inheritance

System.Object

ActorResponseRelationshipStatus<T>

GroupResponseRelationshipStatus

UserResponseRelationshipStatus

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public abstract class ActorResponseRelationshipStatus<T> : object where T : ActorResponse
```

Type Parameters

NAME	DESCRIPTION
T	

Constructors

ActorResponseRelationshipStatus(T, RelationshipStatus)

Constructor

Declaration

```
public ActorResponseRelationshipStatus(T actor, RelationshipStatus status)
```

Parameters

TYPE	NAME	DESCRIPTION
T	actor	
RelationshipStatus	status	

Properties

Actor

Declaration

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

Property Value

TYPE	DESCRIPTION
T	ActorResponse contains the actor ID and Name.

RelationshipStatus

Declaration

```
public RelationshipStatus RelationshipStatus { get; set; }
```

Property Value

TYPE	DESCRIPTION
RelationshipStatus	Current status of the relationship between this actor and the current user

Class ActorUnityClient

Unity Client for getting and updating groups and users and for creating and deleting groups.

Inheritance

System.Object
ActorUnityClient

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class ActorUnityClient : object
```

Methods

CreateGroup(String, Action<GroupResponse>)

Create a group with the provided name.

Declaration

```
public void CreateGroup(string name, Action<GroupResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	Name the newly created group should have.
Action<GroupResponse>	onComplete	Callback with a GroupResponse result for the newly created group.

CreateGroup(String, String, Action<GroupResponse>)

Create a group with the provided name and description.

Declaration

```
public void CreateGroup(string name, string description, Action<GroupResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	Name the newly created group should have.
System.String	description	Description the newly created group should have.
Action<GroupResponse>	onComplete	Callback with a GroupResponse result for the newly created group.

DeleteGroup(Int32, Action<Boolean>)

Delete the group with the provided Id.

Declaration

```
public void DeleteGroup(int id, Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the group to delete.
Action<System.Boolean>	onComplete	Callback with a bool for whether the deletion was successful.

GetAllGroups(Action<IEnumerable<GroupResponse>>)

Get a list of all groups that have been created.

Declaration

```
public void GetAllGroups(Action<IEnumerable<GroupResponse>> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<IEnumerable<GroupResponse>>	onComplete	Callback with a list of gathered GroupResponse results.

GetGroupById(Int32, Action<GroupResponse>)

Get the group whose id matches the id provided.

Declaration

```
public void GetGroupById(int id, Action<GroupResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	Id used to get the group.
Action<GroupResponse>	onComplete	Callback with a GroupResponse result.

GetGroupsControlledByCurrentUser(Action<IEnumerable<GroupResponse>>)

Get a list of all groups that the current user has permissions over.

Declaration

```
public void GetGroupsControlledByCurrentUser(Action<IEnumerable<GroupResponse>> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<IEnumerable<GroupResponse>>	onComplete	Callback with a list of gathered GroupResponse results.

GetUserById(Int32, Action<UserResponse>)

Get the user whose id matches the id provided.

Declaration

```
public void GetUserById(int id, Action<UserResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	Id used to get the user.
Action<UserResponse>	onComplete	Callback with a UserResponse result.

SearchGroupsByName(String, Action<IEnumerable<GroupResponse>>)

Get a list of all groups whose name contains the string provided.

Declaration

```
public void SearchGroupsByName(string name, Action<IEnumerable<GroupResponse>> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	String to search by.
Action<IEnumerable<GroupResponse>>	onComplete	Callback with a list of gathered GroupResponse results.

SearchUsersByName(String, Action<IEnumerable<UserResponse>>)

Get a list of all users whose name contains the string provided.

Declaration

```
public void SearchUsersByName(string name, Action<IEnumerable<UserResponse>> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	String to search by.

TYPE	NAME	DESCRIPTION
Action<IEnumerable<UserResponse>>	onComplete	Callback with a list of gathered UserResponse results.

UpdateGroup(Int32, String, String, Action<GroupResponse>)

Update the group with the provided id to now have the provided name and description.

Declaration

```
public void UpdateGroup(int id, string name, string description, Action<GroupResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the group to update.
System.String	name	Name the group should now have.
System.String	description	Description the group should now have.
Action<GroupResponse>	onComplete	Callback with a GroupResponse result for the newly updated group.

UpdateGroupDescription(Int32, String, Action<GroupResponse>)

Update the group with the provided id to now have the provided description.

Declaration

```
public void UpdateGroupDescription(int id, string description, Action<GroupResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the group to update.
System.String	description	Description the group should now have.
Action<GroupResponse>	onComplete	Callback with a GroupResponse result for the newly updated group.

UpdateGroupName(Int32, String, Action<GroupResponse>)

Update the group with the provided id to now have the provided name.

Declaration

```
public void UpdateGroupName(int id, string name, Action<GroupResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the group to update.
System.String	name	Name the group should now have.
Action<GroupResponse>	onComplete	Callback with a GroupResponse result for the newly updated group.

UpdateUser(Int32, String, String, Action<UserResponse>)

Update the user with the provided id to now have the provided name and description.

Declaration

```
public void UpdateUser(int id, string name, string description, Action<UserResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the user to update.
System.String	name	Name the user should now have.
System.String	description	Description the user should now have.
Action<UserResponse>	onComplete	Callback with a UserResponse result for the newly updated user.

UpdateUserDescription(Int32, String, Action<UserResponse>)

Update the user with the provided id to now have the provided description.

Declaration

```
public void UpdateUserDescription(int id, string description, Action<UserResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the user to update.

TYPE	NAME	DESCRIPTION
System.String	description	Description the user should now have.
Action<UserResponse>	onComplete	Callback with a UserResponse result for the newly updated user.

UpdateUserName(Int32, String, Action<UserResponse>)

Update the user with the provided id to now have the provided name.

Declaration

```
public void UpdateUserName(int id, string name, Action<UserResponse> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the user to update.
System.String	name	Name the user should now have.
Action<UserResponse>	onComplete	Callback with a UserResponse result for the newly updated user.

Class BaseAccountInterface

Base abstract class for controlling the interface related to Account functionality.

Inheritance

System.Object

BaseAccountInterface

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

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

Fields

_closeButton

Declaration

```
protected Button _closeButton
```

Field Value

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

_errorText

Declaration

```
protected Text _errorText
```

Field Value

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

_loginButton

Declaration

```
protected Button _loginButton
```

Field Value

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

_name

Declaration

```
protected InputField _name
```

Field Value

TYPE	DESCRIPTION
InputField	Input field used for providing usernames. Required.

_password

Declaration

```
protected InputField _password
```

Field Value

TYPE	DESCRIPTION
InputField	Input field used for providing passwords. Required.

_registerButton

Declaration

```
protected Button _registerButton
```

Field Value

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

_rememberMeToggle

Declaration

```
protected Toggle _rememberMeToggle
```

Field Value

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

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 displaying evaluation progress.

Inheritance

System.Object
BaseInterface
BaseEvaluationListInterface

Inherited Members

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

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	Was the data successfully loaded?

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 display evaluation notifications when an evalaution is completed.

Inheritance

System.Object

BaseEvaluationPopupInterface

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

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

Fields

_evaluationQueue

Declaration

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

Field Value

TYPE	DESCRIPTION
List<EvaluationNotification>	Queue of notifications to be displayed.

_image

Declaration

```
protected Image _image
```

Field Value

TYPE	DESCRIPTION
Image	Image displayed alongside notification. Can be left null.

_name

Declaration

```
protected Text _name
```

Field Value

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

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	Notification which will be displayed.

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()
BaseInterface.Reload()

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

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

Fields

_groupName

Declaration

```
protected Text _groupName
```

Field Value

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

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
BaseUserFriendInterface
BaseUserGroupInterface

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

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

Fields

_closeButton

Declaration

```
protected Button _closeButton
```

Field Value

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

_errorText

Declaration

```
protected Text _errorText
```

Field Value

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

_signinButton

Declaration

```
protected Button _signinButton
```

Field Value

TYPE	DESCRIPTION

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

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	Was the data successfully loaded?

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 interface.

Declaration

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

PreDisplay()

Functionality triggered before displaying the interface.

Declaration

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

Reload()

Logic to perform when the UI is reloaded.

Declaration

```
protected abstract void Reload()
```

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	Was the data successfully loaded?

Class BaseLeaderboardInterface

Base abstract class for controlling the interface related to displaying the latest standings for a leaderboard.

Inheritance

System.Object
BaseInterface
BaseLeaderboardInterface

Inherited Members

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

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

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

Fields

_alliancesButton

Declaration

```
protected Button _alliancesButton
```

Field Value

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

_friendsButton

Declaration

```
protected Button _friendsButton
```

Field Value

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

_leaderboardName

Declaration

```
protected Text _leaderboardName
```

Field Value

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

_leaderboardType

Declaration

```
protected Text _leaderboardType
```

Field Value

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

_membersButton

Declaration

```
protected Button _membersButton
```

Field Value

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

_nearButton

Declaration

```
protected Button _nearButton
```

Field Value

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

_topButton

Declaration

```
protected Button _topButton
```

Field Value

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

Methods

Awake()

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	Was the data successfully loaded?

Overrides

BaseInterface.ErrorDraw(Boolean)

HideInterfaces()

Hides Account, Evaluation, UserFriend, GroupMember and UserGroup UI objects. Makes filter buttons interactable. Set leaderboard related text.

Declaration

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

Overrides

BaseInterface.HideInterfaces()

IsValid(ActorResponse, ActorType)

If the response actor isn't null, does the leaerboard ActorType match the actorType provided or the Combined ActorType

Declaration

```
protected bool IsValid(ActorResponse response, ActorType actorType)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	response	Actor that would be used for the filter
ActorType	actorType	Valid actorType for the leaderboard if response is used as the basis

Returns

TYPE	DESCRIPTION
System.Boolean	

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()

SetInteractable(Button, Boolean)

Set the button's interactable value

Declaration

```
protected void SetInteractable(Button button, bool interactable = true)
```

Parameters

TYPE	NAME	DESCRIPTION
Button	button	Button to enable/disable the use of
System.Boolean	interactable	Value to change the button's interactable boolean to

UpdateFilter(LeaderboardFilterType)

Change the leaderboard filter currently being used

Change the leaderboard filter currently being used

Declaration

```
protected void UpdateFilter(LeaderboardFilterType filter)
```

Parameters

TYPE	NAME	DESCRIPTION
LeaderboardFilterType	filter	The filter to use for display leaderboard standings

UpdateMultiplePerActor(Boolean)

Change the multiple per actor setting currently being used

Declaration

```
protected void UpdateMultiplePerActor(bool multiplePerActor)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	multiplePerActor	Setting that determines if actors can appear on a leaderboard multiple times

Class BaseLeaderboardListInterface

Base abstract class for controlling the interface related to displaying a list of leaderboards.

Inheritance

System.Object
BaseInterface
BaseLeaderboardListInterface

Inherited Members

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

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

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

Fields

_combinedButton

Declaration

```
protected Button _combinedButton
```

Field Value

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

_groupButton

Declaration

```
protected Button _groupButton
```

Field Value

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

_leaderboardType

Declaration

```
protected Text _leaderboardType
```

Field Value

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

_userButton

Declaration

```
protected Button _userButton
```

Field Value

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

Methods

Awake()

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. Set leaderboard type related text.

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()

UpdateFilter(ActorType)

Chnage the filter currently being used to get leaderboard for a particular type of actor

Declaration

```
protected void UpdateFilter(ActorType filter)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorType	filter	The filter to use for display leaderboard standings

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

Declaration

```
protected T _landscapeInterface
```

Field Value

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

_portraitInterface

Declaration

```
protected T _portraitInterface
```

Field Value

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

Properties

_interface

Declaration

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

Property Value

TYPE	DESCRIPTION
T	The interface that is used for the current aspect ratio.

HasInterface

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Has an interface been provided for this Unity Client?

IsActive

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Is there an interface and if so is it currently active?

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()
BaseInterface.Reload()

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()
```

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()
BaseInterface.Reload()

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()
```

GetPendingReceived()

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

Declaration

```
protected void GetPendingReceived()
```

GetPendingSent()

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

Declaration

```
protected void GetPendingSent()
```

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
CommandLineOptions	

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

Use this for gathering evaluation progress and notifications when an evaluation is completed.

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

Declaration

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

Property Value

TYPE	DESCRIPTION
List<EvaluationProgressResponse>	Current completion status for evaluations in this application for the currently signed in user.

Methods

DisplayAchievementList()

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

Declaration

```
public void DisplayAchievementList()
```

DisplayEvaluationList()

Gathers current user achievement and skill completion status and displays the interface if it is provided.

Declaration

```
public void DisplayEvaluationList()
```

DisplayGroupAchievementList()

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

Declaration

```
public void DisplayGroupAchievementList()
```

DisplayGroupEvaluationList()

Gathers current group achievement and skill completion status and displays the interface if it is provided.

Declaration

```
public void DisplayGroupEvaluationList()
```

DisplayGroupSkillList()

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

Declaration

```
public void DisplayGroupSkillList()
```

DisplaySkillList()

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

Declaration

```
public void DisplaySkillList()
```

ForceNotification(String)

Force a notification to be displayed with the provided notification text.

Declaration

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

Parameters

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

Remarks

- This uses the EvaluationPopupInterface to display the text in the application

GetAchievementProgress(ActorResponse, Action<List<EvaluationProgressResponse>>)

Gathers achievement completion status for the actor provided.

Declaration

```
public void GetAchievementProgress(ActorResponse actor, Action<List<EvaluationProgressResponse>> progress)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	actor	The user or group the achievement progress will be gathered for.

TYPE	NAME	DESCRIPTION
Action<List<EvaluationProgressResponse>>	progress	Callback which will return the achievement progress for the actor provided.

GetAchievementProgress(Int32, Action<List<EvaluationProgressResponse>>)

Gathers achievement completion status for the actor id provided.

Declaration

```
public void GetAchievementProgress(int id, Action<List<EvaluationProgressResponse>> progress)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The id of the user or group the achievement progress will be gathered for.
Action<List<EvaluationProgressResponse>>	progress	Callback which will return the achievement progress for the actor provided.

GetEvaluationProgress(ActorResponse, Action<List<EvaluationProgressResponse>>)

Gathers achievement and skill completion status for the actor provided.

Declaration

```
public void GetEvaluationProgress(ActorResponse actor, Action<List<EvaluationProgressResponse>> progress)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	actor	The user or group the achievement and skill progress will be gathered for.
Action<List<EvaluationProgressResponse>>	progress	Callback which will return the achievement and skill progress for the actor provided.

GetEvaluationProgress(Int32, Action<List<EvaluationProgressResponse>>)

Gathers achievement and skill completion status for the actor id provided.

Declaration

```
public void GetEvaluationProgress(int id, Action<List<EvaluationProgressResponse>> progress)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The id of the user or group the achievement and skill progress will be gathered for.

TYPE	NAME	DESCRIPTION
Action<List<EvaluationProgressResponse>>	progress	Callback which will return the achievement and skill progress for the actor provided.

GetSkillProgress(ActorResponse, Action<List<EvaluationProgressResponse>>)

Gathers skill completion status for the actor provided.

Declaration

```
public void GetSkillProgress(ActorResponse actor, Action<List<EvaluationProgressResponse>> progress)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	actor	The user or group the skill progress will be gathered for.
Action<List<EvaluationProgressResponse>>	progress	Callback which will return the skill progress for the actor provided.

GetSkillProgress(Int32, Action<List<EvaluationProgressResponse>>)

Gathers skill completion status for the actor id provided.

Declaration

```
public void GetSkillProgress(int id, Action<List<EvaluationProgressResponse>> progress)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The id of the user or group the skill progress will be gathered for.
Action<List<EvaluationProgressResponse>>	progress	Callback which will return the skill progress for the actor provided.

Update()

Update the interface to be used when the aspect ration changes

Declaration

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

Overrides

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

Class GameDataUnityClient

Use this to GET and POST data related to the game.

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>> onComplete, string[] keys = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<IEnumerable<EvaluationDataResponse>>	onComplete	Callback with a list of gathered EvaluationDataResponse results.
System.String[]	keys	**Optional** Keys to search and return values for. (default: null)

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> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	onComplete	Callback which contains 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> onComplete)
```

Parameters

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

Remarks

- EvaluationDataType should be a type that can be added together, eg. Long

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> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	onComplete	Callback which contains 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> onComplete)
```

Parameters

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

TYPE	NAME	DESCRIPTION
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	onComplete	Callback which contains 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> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
EvaluationDataType	dataType	EvaluationDataType of the GameData.
Action<EvaluationDataResponse>	onComplete	Callback which contains 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> onComplete)
```

Parameters

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

Send(String, Boolean, Action<Boolean>)

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

Declaration

```
public void Send(string key, bool value, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.Boolean	value	The Bool value that'll be recorded.
Action<System.Boolean>	onComplete	**Optional** Callback returns whther the data was sent successfully (default: null)

Send(String, Int64, Action<Boolean>)

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

Declaration

```
public void Send(string key, long value, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.Int64	value	The Long value that'll be recorded.
Action<System.Boolean>	onComplete	**Optional** Callback returns whther the data was sent successfully (default: null)

Send(String, Single, Action<Boolean>)

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

Declaration

```
public void Send(string key, float value, Action<bool> onComplete = null)
```

Parameters

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

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	**Optional** Callback returns whther the data was sent successfully (default: null)

Send(String, String, Action<Boolean>)

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

Declaration

```
public void Send(string key, string value, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the GameData key.
System.String	value	The String value that'll be recorded.
Action<System.Boolean>	onComplete	**Optional** Callback returns whther the data was sent successfully (default: null)

Class GroupMemberUnityClient

Use this for actions 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

Declaration

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

Property Value

TYPE	DESCRIPTION
ActorResponse	Currently selected/displayed group.

Members

Declaration

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

Property Value

TYPE	DESCRIPTION
List<UserResponseRelationshipStatus>	Members for the current group.

Methods

Display(ActorResponse)

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

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	group	The group which should be set to CurrentGroup

GetGroupMembers(ActorResponse, Action<List<UserResponseRelationshipStatus>>)

Gather member list for the provided group.

Declaration

```
public void GetGroupMembers(ActorResponse group, Action<List<UserResponseRelationshipStatus>> members)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	group	The group for which members will be gathered
Action<List<UserResponseRelationshipStatus>>	members	Callback which will return the list of group members and their current relationship with the signed in user.

GetGroupMembers(Int32, Action<List<UserResponseRelationshipStatus>>)

Gather member list for the group with the provided id.

Declaration

```
public void GetGroupMembers(int groupId, Action<List<UserResponseRelationshipStatus>> members)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	groupId	The id of the group for which members will be gathered
Action<List<UserResponseRelationshipStatus>>	members	Callback which will return the list of group members and their current relationship with the signed in user.

Class GroupResponseRelationshipStatus

ActorResponse with additional information on the relationship between the current user and the actor.

Inheritance

System.Object
ActorResponseRelationshipStatus<ActorResponse>
GroupResponseRelationshipStatus

Inherited Members

ActorResponseRelationshipStatus<ActorResponse>.Actor
ActorResponseRelationshipStatus<ActorResponse>.RelationshipStatus

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class GroupResponseRelationshipStatus : ActorResponseRelationshipStatus<ActorResponse>
```

Constructors

GroupResponseRelationshipStatus(ActorResponse, RelationshipStatus)

Constructor

Declaration

```
public GroupResponseRelationshipStatus(ActorResponse actor, RelationshipStatus status)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	actor	
RelationshipStatus	status	

Methods

Add(Action<Boolean>, Boolean)

Send a relationship request to this Group.

Declaration

```
public void Add(Action<bool> onComplete, bool autoAccept = true)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the request was successfully created
System.Boolean	autoAccept	**Optional** Should the request be automatically accepted

CancelSentRequest(Action<Boolean>)

Cancel a relationship request to this Group.

Declaration

```
public void CancelSentRequest(Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the relationship request was successfully cancelled

Remove(Action<Boolean>)

Cancel the relationship with this Group.

Declaration

```
public void Remove(Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the relationship request was successfully cancelled

UpdateRequest(Boolean, Action<Boolean>)

Accept or decline a received relationship request from this Group.

Declaration

```
public void UpdateRequest(bool accept, Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	accept	Accept or decline the request
Action<System.Boolean>	onComplete	Callback for if the relationship was successfully updated

Class LeaderboardListUnityClient

Use this to get a list of leaderboards for this game

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

Declaration

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

Property Value

TYPE	DESCRIPTION
ActorType	Currently used ActorType filter.

Leaderboards

Declaration

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

Property Value

TYPE	DESCRIPTION
Dictionary<ActorType, List<LeaderboardResponse>>	Each ActorType and list of leaderboard responses for this application.

Methods

DisplayGameList(ActorType)

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

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
ActorType	filter	**Optional** The filter type to use (default: ActorType.User)

DisplayGlobalList(ActorType)

Gathers all leaderboards not attached to a game and displays list for current ActorType if interface if provided.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
ActorType	filter	**Optional** The filter type to use (default: ActorType.User)

Class LeaderboardUnityClient

Use this to get the current standings for a leaderboard

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

Declaration

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

Property Value

TYPE	DESCRIPTION
LeaderboardFilterType	Current filter to use for gathering leaderboard standings.

CurrentLeaderboard

Declaration

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

Property Value

TYPE	DESCRIPTION
LeaderboardResponse	Current leaderboard to use for gathering leaderboard standings from.

CurrentStandings

Declaration

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

Property Value

TYPE	DESCRIPTION
List<LeaderboardStandingsResponse>	Last set of standings gathered.

MultiplePerActor

Declaration

```
public bool MultiplePerActor { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	Current setting for whether actors can appear on leaderboards multiple times.

PositionCount

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	Number of results that should be gathered per call.

Methods

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

Gathers information for leaderboard and displays the interface if it has been provided.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.String	token	The unique identifier for the Leaderboard
LeaderboardFilterType	filter	The Filter type to order standings by
System.Boolean	multiplePerActor	If the leaderboard allows for actors to appeard multiple times
System.Int32	pageNumber	**Optional** The page number to start from (default: 0)

TYPE	NAME	DESCRIPTION
System.Boolean	globalLeaderboard	**Optional** Whether the leaderboard is global or in game scope. (default: false)

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

Get standings for the current leaderboard.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	pageNumber	The page number to retrieve
Action<System.Boolean>	onComplete	Whether the standings were retrieved successfully
Action<List<LeaderboardStandingsResponse>>	result	**Optional** the results for the leaderboard standings, null value will save results to CurrentStandings (default: null)

SetPositionCount(Int32)

Set the maximum number of results to get per call.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	count	The Maximum number of results

Enum RelationshipStatus

The different relationship states two actors can be in related to each other

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public enum RelationshipStatus : int
```

Fields

NAME	DESCRIPTION
ExistingRelationship	
NoRelationship	
PendingReceivedRequest	
PendingSentRequest	

Class ResourceUnityClient

Use this to get current resources, add resources and send resources to other users

Inheritance

System.Object
ResourceUnityClient

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class ResourceUnityClient : MonoBehaviour
```

Methods

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

Add the resource with the key provided to 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, long> onComplete, bool globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	key	Name of the resource being transferred
System.Int64	amount	The amount being transferred
Action<System.Boolean, System.Int64>	onComplete	Callback which returns whether the transfer was a success and the current value of the resource that was transferred
System.Boolean	globalResource	**Optional** Setting for if the resource is global rather than for this game. (default: false)

Remarks

If globalResource is true, resource transferred will be global rather than for the game.

GetFromCache(String, Boolean)

Get the current resource amount for the current user from the local cache. Cache is updated at the rate set in the Inspector.

Declaration

```
public long GetFromCache(string key, bool globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	key	Resource key value is being gathered for
System.Boolean	globalResource	**Optional** Get value for a global resource rather than one for this game. (default: false)

Returns

TYPE	DESCRIPTION
System.Int64	

Remarks

- If globalResource is true, resource will be global rather than for the game.

GetFromServer(Action<Boolean, Dictionary<String, Int64>>, String[], Boolean)

Get the resources with the keys provided for the current user directly from the server.

Declaration

```
public void GetFromServer(Action<bool, Dictionary<string, long>> result, string[] keys = null, bool
globalResource = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean, Dictionary<System.String, System.Int64>>	result	Callback which will return whether the call to the server was successful and a dictionary of all the keys and their current values
System.String[]	keys	Resource keys values are being gathered for
System.Boolean	globalResource	**Optional** Get resource values for global resources rather than one for this game. (default: false)

Remarks

- If globalResource is true, resource will be global rather than for the game.

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

Transfer the resource with the key provided from the currently signed in user

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Int32	recipientId	Id of the actor who will receive the resource
System.String	key	Name of the resource being transferred
System.Int64	amount	The amount being transferred
Action<System.Boolean, System.Int64>	onComplete	Callback which returns whether the transfer was a success and the current value of the resource that was transferred
System.Boolean	globalResource	**Optional** Setting for if the resource is global rather than for this game. (default: false)

Remarks

If globalResource is true, resource transferred will be global rather than for the game.

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

Transfer the resource with the key provided to the currently signed in user

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	senderId	Id of the actor who will send the resource
System.String	key	Name of the resource being transferred
System.Int64	amount	The amount being transferred
Action<System.Boolean, System.Int64>	onComplete	Callback which returns whether the transfer was a success and the current value of the resource that was transferred
System.Boolean	globalResource	**Optional** Setting for if the resource is global rather than for this game. (default: false)

Remarks

If globalResource is true, resource transferred will be global rather than for the game.

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
AccountUnityClient	

Actor

Declaration

```
public static ActorUnityClient Actor { get; }
```

Property Value

TYPE	DESCRIPTION
ActorUnityClient	Unity client for calls related to groups and users

ClassId

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	Group name gathered from auto sign in.

Client

Declaration

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

Property Value

TYPE	DESCRIPTION
SUGARClient	Class for contacting SUGAR client functionality

CurrentGroup

Declaration

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

Property Value

TYPE	DESCRIPTION
ActorResponse	Currently signed in user's primary group.

CurrentUser

Declaration

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

Property Value

TYPE	DESCRIPTION
ActorResponse	Currently signed in user.

Evaluation

Declaration

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

Property Value

TYPE	DESCRIPTION
EvaluationUnityClient	Unity client for calls related to evaluations

GameData

Declaration

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

Property Value

TYPE	DESCRIPTION
GameDataUnityClient	Unity client for calls related to gamedata

GameId

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	GameId for this application.

GameLeaderboard

Declaration

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

Property Value

TYPE	DESCRIPTION
LeaderboardListUnityClient	Unity client for calls related to leaderboard lists

GroupMember

Declaration

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

Property Value

TYPE	DESCRIPTION
GroupMemberUnityClient	Unity client for calls related to group members

Leaderboard

Declaration

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

Property Value

TYPE	DESCRIPTION
LeaderboardUnityClient	Unity client for calls related to leaderboard standings

Resource

Declaration

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

Property Value

TYPE	DESCRIPTION
ResourceUnityClient	Unity client for calls related to resources

Unity

Declaration

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

Property Value

TYPE	DESCRIPTION
SUGARUnityManager	Class for managing Unity elements of the asset

UserFriend

Declaration

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

Property Value

TYPE	DESCRIPTION
UserFriendUnityClient	Unity client for calls related to friend lists

UserGroup

Declaration

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

Property Value

TYPE	DESCRIPTION
UserGroupUnityClient	Unity client for calls related to user groups

UserSignedIn

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Is there a user currently signed in.

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

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Is any piece of SUGAR UI currently active?

SpinnerActive

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Boolean	Whether the spinner UI is currently active

Methods

CreateSUGARClient(String)

Create a SUGAR Client from a string

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.String	baseAddress	uri to create SUGAR Client from

Returns

TYPE	DESCRIPTION
SUGARClient	new SUGARClient

DisableObject(GameObject)

Disable a piece of SUGAR UI.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
GameObject	activeObject	The object that should be disabled

Remarks

- This should be used instead of SetActive to ensure UI and blocker ordering is correct.

EnableObject(GameObject)

Enable a piece of SUGAR UI.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
GameObject	activeObject	The object that should be enabled

Remarks

- This should be used instead of SetActive to ensure UI and blocker ordering is correct.

GameValidityCheck()

Check if the current game is valid by the current gameToken

Declaration

```
public bool GameValidityCheck()
```

Returns

TYPE	DESCRIPTION
System.Boolean	Whether the _gameToken returns a valid game

SetBlocker(Boolean, Boolean)

Setup for blocker

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	use	Whether the blocker should be used
System.Boolean	block	Whether clicking on the blocker should close the current UI

SetSpinner(Boolean, Int32)

Setup the spinner

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	clockwise	Whether the spinner should rotate clockwise or not
System.Int32	speed	The speed of the rotation

StartSpinner(String)

Start the loading spinner.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	text	**Optional** Text to display with the spinner. (default: "")

Remarks

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

StopSpinner(String, Single)

Stop the loading spinner.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	**Optional** Text to display when the spinner stops. (default: "")
System.Single	stopDelay	**Optional** The time, in seconds, the text should be displayed for before disabling (default: 0)

Remarks

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

Class UserFriendUnityClient

Use this to get current user's list of friends and send and handle friend requests and other friend related actions

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

Relationships

Declaration

```
public List<UserResponseRelationshipStatus> Relationships { get; }
```

Property Value

TYPE	DESCRIPTION
List<UserResponseRelationshipStatus>	Users with some sort of relationship with the currently signed in user.

Methods

AddFriend(Int32, Action<Boolean>)

Send friend request to another user

Declaration

```
public void AddFriend(int id, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The id of the user to add
Action<System.Boolean>	onComplete	**Optional** Callback for if the request was successfully performed

CancelSentFriendRequest(Int32, Action<Boolean>)

Cancel a friend request sent by the current user

Declaration

```
public void CancelSentFriendRequest(int id, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the user who received the request
Action<System.Boolean>	onComplete	**Optional** Callback for if the request was successfully cancelled

Display()

Gathers updated versions of each type of relationship and displays interface UI object if it has been provided.

Declaration

```
public void Display()
```

GetFriendsList(Action<List<ActorResponse>>)

Get friends list for the currently signed in user.

Declaration

```
public void GetFriendsList(Action<List<ActorResponse>> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<List<ActorResponse>>	onComplete	Callback which contains the list of friends for the current user

ManageFriendRequest(Int32, Boolean, Action<Boolean>)

Resolve a friend request sent to the current user

Declaration

```
public void ManageFriendRequest(int id, bool accept, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the user who sent the request
System.Boolean	accept	Whether the request has been accepted

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	**Optional** Callback for if the request was successfully resolved

RefreshRelationships(Action<Boolean>)

Refresh the Relationship list with up to date information

Declaration

```
public void RefreshRelationships(Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the update was successfully performed

RemoveFriend(Int32, Action<Boolean>)

Remove a relationship between the currently signed in user and another user.

Declaration

```
public void RemoveFriend(int id, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id for the user which the current signed in user wishes to remove
Action<System.Boolean>	onComplete	**Optional** Callback for if the friendship was successfully cancelled

Class UserGroupUnityClient

Use this to get current user's list of groups and send and handle group requests

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

Relationships

Declaration

```
public List<GroupResponseRelationshipStatus> Relationships { get; }
```

Property Value

TYPE	DESCRIPTION
List<GroupResponseRelationshipStatus>	Groups with some sort of relationship with the currently signed in user.

Methods

AddGroup(Int32, Action<Boolean>)

Send group membership request to a group

Declaration

```
public void AddGroup(int id, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The id of the group to send the request to
Action<System.Boolean>	onComplete	**Optional** Callback for if the request was successfully performed

CancelSentGroupRequest(Int32, Action<Boolean>)

Cancel a group membership request sent by the current user

Declaration

```
public void CancelSentGroupRequest(int id, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the group that received the request
Action<System.Boolean>	onComplete	**Optional** Callback for if the request was successfully cancelled

Display()

Gathers updated versions of each type of relationship and displays interface UI object if it has been provided.

Declaration

```
public void Display()
```

GetGroupsList(Action<List<ActorResponse>>)

Get list of groups the currently signed in user is a member of.

Declaration

```
public void GetGroupsList(Action<List<ActorResponse>> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<List<ActorResponse>>	onComplete	Callback which contains the list of groups for the current user

LeaveGroup(Int32, Action<Boolean>)

Leave a group the current user is a member of

Declaration

```
public void LeaveGroup(int id, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id for the group which the current signed in user wishes to leave
Action<System.Boolean>	onComplete	**Optional** Callback for if the group membership was successfully cancelled

ManageGroupRequest(Int32, Boolean, Action<Boolean>)

Resolve a group membership request sent to the current user

Declaration

```
public void ManageGroupRequest(int id, bool accept, Action<bool> onComplete = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	id	The Id of the group that sent the request
System.Boolean	accept	Whether the request has been accepted
Action<System.Boolean>	onComplete	**Optional** Callback for if the request was successfully resolved

RefreshRelationships(Action<Boolean>)

Refresh the Relationship list with up to date information

Declaration

```
public void RefreshRelationships(Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the update was successfully performed

Class UserResponseRelationshipStatus

ActorResponse with additional information on the relationship between the current user and the actor.

Inheritance

System.Object
ActorResponseRelationshipStatus<ActorResponse>
UserResponseRelationshipStatus

Inherited Members

ActorResponseRelationshipStatus<ActorResponse>.Actor
ActorResponseRelationshipStatus<ActorResponse>.RelationshipStatus

Namespace: PlayGen.SUGAR.Unity

Assembly: PlayGen.SUGAR.Unity.dll

Syntax

```
public class UserResponseRelationshipStatus : ActorResponseRelationshipStatus<ActorResponse>
```

Constructors

UserResponseRelationshipStatus(ActorResponse, RelationshipStatus)

Constructor

Declaration

```
public UserResponseRelationshipStatus(ActorResponse actor, RelationshipStatus status)
```

Parameters

TYPE	NAME	DESCRIPTION
ActorResponse	actor	
RelationshipStatus	status	

Methods

Add(Action<Boolean>, Boolean)

Send a relationship request to this User.

Declaration

```
public void Add(Action<bool> onComplete, bool autoAccept = true)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the request was successfully created
System.Boolean	autoAccept	**Optional** Should the request be automatically accepted

CancelSentRequest(Action<Boolean>)

Cancel a relationship request to this User.

Declaration

```
public void CancelSentRequest(Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the relationship request was successfully cancelled

Remove(Action<Boolean>)

Cancel the relationship with this User.

Declaration

```
public void Remove(Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
Action<System.Boolean>	onComplete	Callback for if the relationship request was successfully cancelled

UpdateRequest(Boolean, Action<Boolean>)

Accept or decline a received relationship request from this User.

Declaration

```
public void UpdateRequest(bool accept, Action<bool> onComplete)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	accept	Accept or decline the request
Action<System.Boolean>	onComplete	Callback for if the relationship was successfully updated

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 void TryApplySeed(string username, string password, TextAsset gameSeedText, out List<string>
messages)
```

Parameters

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

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<SetEditorAutoLogin.AutoLoginOption>	

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	