Lobbies Interface

Create and manage online lobbies.

Learn to use this **Beta** feature, but use caution when shipping with it.

The **Online Services Lobbies Interface** provides an API for creating and managing lobbies. A **lobby** is a group of players within the context of a client application where all of the players have some shared state. This could mean that they are all playing in the same online match together or all waiting for the same match to begin. Both the lobby and its members have attributes used to share this state. Changes to a lobby's attributes or the attributes of any of its members are seen by all joined lobby members in real-time.

The lobby leader may change a lobby's attributes, and each member (including the leader) may change their own attributes at any time. A **lobby schema** defines these attributes, along with their types and value constraints, that are then used to create a lobby. This lobby schema is defined in the application config files and is validated when the application starts. You are not required to define all attributes to create a lobby, but if you define an attribute that fails schema validation, lobby creation will fail.

A common use of Lobbies is to group players together before proceeding into a match. Several game matches may be played within the lifetime of a single lobby. Below is an example lifecycle of a lobby:

- A player creates a new lobby with the desired privacy settings and attributes. This player is designated the lobby leader.
- Additional players locate and join the lobby through public search, invitation, or social presence.
- Players share attributes within the lobby through attribute updates.
- Once the lobby chooses to play a match, the lobby leader records the match session ID as a lobby attribute.
- Other lobby members see this session ID and join the same game session.
- Players join or leave the lobby throughout the lifetime of multiple game matches.

API Overview

Functions

The following table provides a high-level overview of the functions provided by the Lobbies Interface:

Function	Description
Operations	
`CreateLobby`	Create and join a new lobby.
`FindLobbies`	Search for lobbies that fit the provided parameters.
`RestoreLobbies`	Try to rejoin previously joined lobbies.
`JoinLobby`	Join a lobby with the provided lobby ID.
`LeaveLobby`	Leave a joined lobby.

Function	Description
`InviteLobbyMember`	Invite a player to join a lobby.
`DeclineLobbyInvitation`	Decline an invitation to join a lobby.
`KickLobbyMember`	Kick a member from the target lobby.
`PromoteLobbyMember`	Promote another lobby member to leader. The local player calling `PromoteLobbyMember` must be the current lobby leader to promote another member.
Mutations	
`ModifyLobbySchema`	Change the schema applied to the lobby and member attributes.Only the lobby leader may change the schema. Existing attributes not present in the new schema will be cleared.
`ModifyLobbyJoinPolicy`	Change the join policy applied to the lobby. Only the lobby leader may change the join policy.
`ModifyLobbyAttributes`	Change the attributes applied to the lobby. Only the lobby leader may change the lobby attributes. Attributes are validated against the lobby schema before an update can succeed.
`ModifyLobbyMemberAttributes`	Change the attributes applied to a lobby member. Lobby members may only change their own attributes. Attributes are validated against the lobby schema before an update can succeed.
Accessors	
`GetJoinedLobbies`	Retrieve the list of joined lobbies for the target local player.

Function	Description
`GetReceivedInvitations`	Retrieve the list of received invitations for the target local player.
Event Listening	
`OnLobbyJoined`	Event triggered when a player joins a lobby.
`OnLobbyLeft`	Event triggered when all local members leave a lobby.
`OnLobbyMemberJoined`	Event triggered when a lobby member joins, either as a result of a local player creating or joining a lobby, or when a remote player joins.
`OnLobbyMemberLeft`	Event triggered when a lobby member leaves a joined lobby.
`OnLobbyLeaderChanged`	Event triggered when the lobby's leadership changes.
`OnLobbySchemaChanged`	Event triggered when the lobby's attribute schema changes.
`OnLobbyAttributesChanged`	Event triggered when a lobby's attributes change.
`OnLobbyMemberAttributesChanged`	Event triggered when a lobby member's attributes change.
`OnLobbyInvitationAdded`	Event triggered when a lobby member receives an invitation.
`OnLobbyInvitationRemoved`	Event triggered when a lobby member addresses an invitation or the invitation expires.
`OnUILobbyJoinRequested`	Event triggered when a player requests to join the lobby through an external mechanism.

Enumeration Classes

The Lobbies Interface provides enumeration classes to reflect Lobby Join Policy and Lobby Member Leave Reasons through two enumeration classes: `ELobbyJoinPolicy` and `ELobbyMemberLeaveReason`.

ELobbyJoinPolicy

Enumerator	Description
`PublicAdvertised`	Users can find the lobby through searches based on attribute matching, lobby ID, or invitation.
`PublicNotAdvertised`	Users can join the lobby by lobby ID or invitation.
`InvitationOnly`	Users can only join the lobby by invitation.

ELobbyMemberLeaveReason

Enumerator	Description
`Left`	Lobby member chose to leave the lobby.
`Kicked`	Lobby member was kicked out of the lobby by the lobby owner.
`Disconnected`	Lobby member left unexpectedly.
`Closed`	Lobby was destroyed by the online services and all members have left.

Primary Structs

The Lobby Interface functionality is primarily communicated through two structs: `FLobbyMember` and `FLobby`, in addition to the function specific structs for passing parameters and return values.

FLobbyMember

Member	Type	Description
`AccountId`	`FAccountId`	Account ID of this lobby member.
`PlatfromAccountId`	`FAccountId`	Platform account ID of this lobby member.
`PlatfromDisplayName`	`FString`	Platform display name of this lobby member.
`Attributes`	`TMap<FSchemaAttributeId, FSchemaVariant>`	Attributes of this lobby member as defined in the configuration.
`bIsLocalMember`	`bool`	Whether this lobby member is a local player on this client. (Default value is `false`.)

FLobby

Member	Type	Description
`LobbyId`	`FLobbyId`	ID of this lobby.
`OwnerAccountId`	`FAccountId`	Account ID of the lobby member that is the current owner of this lobby.
`LocalName`	`FName`	Local name of this lobby.
`SchemaId`	`FSchemaId`	ID of the schema applied to this lobby.

Member	Type	Description
`MaxMembers`	`int32`	Maximum number of members who can be in this lobby at any given time.
`JoinPolicy`	`ELobbyJoinPolicy`	Join policy setting for this lobby.
`Attributes`	`TMap<FSchemaAttributeId, FSchemaVariant>`	Attributes of this lobby as defined in the configuration.
`Members`	`TMap<FAccountId, TSharedRef<const FLobbyMember>>`	Dictionary of lobby members where the keys are account IDs and the values are pointers to their corresponding lobby member struct.

Configuration

Lobbies use a schema system to define the structure and properties of the lobby as well as lobby member attributes. A game may declare multiple individual schema definitions to support many different types of lobbies. All lobby schemas must derive from the `LobbyBase` schema.

The `LobbyBase` schema contains all attributes players can use to search for a lobby. This inheritance structure means that the Lobbies interface knows which game-provided schema to apply to a search result. `SchemaCompatibilityId` is a special attribute within `LobbyBase` that exists to ensure that a schema is compatible between two clients.

A game must declare a definition for each schema attribute present in a schema definition. Schemas are defined in a project's configuration files (`*.ini`) files. These definitions include an attribute's type, maximum size, and visibility, as well as additional behaviors such as whether the attribute is available as a search parameter.

In general, a schema has categories which contain attribute definitions. The Lobbies interface schema has two categories: `Lobby` and `LobbyMember`. These categories contain attribute definitions which apply to the lobby object and lobby member objects, respectively.

Example

Here is an example configuration for the Lobbies interface:

DefaultEngine.ini

```
[OnlineServices.Lobbies]
+SchemaDescriptors=(Id="GameLobby", ParentId="LobbyBase")

!SchemaCategoryAttributeDescriptors=ClearArray
+SchemaCategoryAttributeDescriptors=(SchemaId="LobbyBase", CategoryId="Lobby", Att
+SchemaCategoryAttributeDescriptors=(SchemaId="LobbyBase", CategoryId="LobbyMember

+SchemaCategoryAttributeDescriptors=(SchemaId="GameLobby", CategoryId="Lobby", Att
+SchemaCategoryAttributeDescriptors=(SchemaId="GameLobby", CategoryId="LobbyMember

+SchemaAttributeDescriptors=(Id="ExampleSearchableLobbyAttribute", Type="String", 
+SchemaAttributeDescriptors=(Id="GameMode", Type="String", Flags=("Public"), MaxSi
+SchemaAttributeDescriptors=(Id="GameSessionId", Type="String", Flags=("Private"),
+SchemaAttributeDescriptors=(Id="MapName", Type="String", Flags=("Public"), MaxSiz
+SchemaAttributeDescriptors=(Id="MatchTimeout", Type="Double", Flags=("Public"))
+SchemaAttributeDescriptors=(Id="Appearance", Type="String", Flags=("Public"), Max
```

Copy full snippet

Process Flow

Create

A player initiates lobby creation, and will be designated the lobby leader once the lobby is created. The creating player decides the initial lobby settings. These settings can include:

- Visibility
- Join Policy
- Schema ID
- Lobby Attributes
- Lobby Member Attributes

Other players can see the public attributes when they search for a lobby. The lobby leader has additional privileges as outlined in the Leader Actions section of this page. All lobby members, including the leader, can change their personal attributes or invite players to join the lobby. These actions are summarized in the Member Actions section.

Upon successful lobby creation, the creating player receives an `OnLobbyJoined` event followed by an `OnLobbyMemberJoined` for themselves, since they are now the designated lobby leader. As long as this player is designated as the lobby leader, they will receive additional `OnLobbyMemberJoined` and `OnLobbyMemberLeft` events as players join and leave the lobby.

Find

To join a lobby, a player first needs to know the lobby's ID. Players can find the ID by:

- Search
- Invite

Both search and invitation provide a snapshot of the lobby data to the client application. Attributes within the snapshot are not updated until the player has joined the lobby.

Search

Lobby search allows a player to specify attributes to find lobbies which match their desired configuration. Players can search for lobbies by:

- Attribute filters
- Target player
- Specific lobby ID

Invite

Depending on a lobby's privacy setting, lobby members may invite additional players to join the lobby. When an invitation is sent, the target player receives an `OnLobbyInvitationAdded` event to notify them of the waiting invitation. The `OnLobbyInvitationRemoved` event triggers when the invitation is addressed or expires.

Social Presence

Some online service implementations allow players to join a friend's lobby through a social user interface, such as the Online Services Presence Interface. When a player chooses to join in this way, an `OnUILobbyJoinRequested` event triggers the application to indicate the player's desire to join the lobby.

Join

Once a player knows the ID of the lobby they wish to join, calling `JoinLobby` starts the process of adding the local player. The local player must provide their initial `LobbyMember` attributes, which are shared with the other existing lobby members upon joining the lobby.

After successfully joining a lobby, the player receives an `OnLobbyJoined` event. Every existing member of the lobby receives an `OnLobbyMemberJoined` event notifying them a new player has joined. While a player remains in this lobby, they receive additional `OnLobbyMemberJoined` and `OnLobbyMemberLeft` events as other players join and leave the lobby.

Leave

When a player no longer wishes to remain in a lobby, `LeaveLobby` removes the player from the lobby with the supplied lobby ID and stops notifications. Once a player leaves a lobby, all other players in the lobby receive an `OnLobbyMemberLeft` notification. This notifies them a player has left the lobby. These notifications are followed by an `OnLobbyLeft` event sent to the player who has left the lobby..

Restore

`RestoreLobbies` restores all lobbies previously joined by the calling player. This is typically done by a game at application startup to rejoin the local player to lobbies they were in when the application last exited.

Leader Actions

The lobby leader has additional, exclusive privileges they can use to help maintain the lobby.

Promote Member

The lobby leader can promote another lobby member to become the leader with a call to `PromoteLobbyMember`. This action triggers an `OnLobbyLeaderChanged` event for all lobby members. The player who was previously the leader becomes a normal lobby member.

Kick Member

The lobby leader can remove a target member from the lobby with a call to `KickLobbyMember`. This action triggers an `OnLobbyLeft` event for the kicked lobby member and an `OnLobbyMemberLeft` event for all other lobby members.

Update Lobby Attributes

The lobby leader can change the attributes of the lobby object. This is handled with a call to `ModifyLobbyAttributes`. When the lobby attributes change, all lobby members receive an `OnLobbyAttributesChanged` event to notify them of the changed attributes.

Change Lobby Join Policy

The lobby's join policy affects whether the lobby appears in search results, is invitation only, or can be joined by social presence. The lobby leader can change this setting with a call to `ModifyLobbyJoinPolicy`.

Member Actions

Lobby members are limited to two actions. Members can update their own attributes and invite other members to join the lobby.

Change Member Attributes

Lobby members can change their own attributes with a call to `ModifyLobbyMemberAttributes`. An `OnLobbyMemberAttributesChanged` event notifies other lobby members of this change.

Invite Players to Join the Lobby

If the lobby join policy currently allows invitations, lobby members can invite additional players with a call to `InviteLobbyMember`. The target of the invitation receives an `OnLobbyInvitationAdded` event that notifies them of the pending invitation. The invited player chooses to join the lobby with a call to `JoinLobby` or rejects the invitation with `DeclineLobbyInvitation`.

Converting Code from Online Subsystem

Lobbies is a new interface for the Online Services and does not have a direct counterpart from the **Online Subsystem**.

More Information

Header File

Consult the `Lobbies.h` header file directly for more information as needed. The Lobbies Interface header file `Lobbies.h` is located in the directory:

```
Engine\Plugins\Online\OnlineServices\Source\OnlineServicesInterface\Public\Online
```

Copy full snippet

For instructions on how to obtain the UE source code, see our documentation on Downloading Unreal Engine Source Code.

Function Parameters and Return Types

See the Functions section of the Online Services Overview page for an explanation of function parameters and return types, including how to pass parameters and processing the results when functions return.