Peter Gustafson writing sample
Epic Games
July 19, 2021

Lobby Interface

Lobbies provide a connection between users for the purpose of sharing game and user state with real-time updates.

Users can create or join lobbies to form teams, select pre-game options, and wait for additional players to join in before playing together. Using the Lobby Interface, your users can create, join, leave, and manage lobbies.

Lifecycle
The lifecycle of a lobby generally follows this sequence:

 1. A user creates the lobby. This user automatically joins the lobby and becomes both the first member and the owner.

 2. The owner sets up the lobby's initial state, and may invite users to join.

 3. Other users join and leave the lobby. While connected, members update state information about themselves and can invite other users.

 4. The lobby owner can update data specific to the state of the game.

 5. The owner can remove members from the lobby or transfer ownership status to another member.

 6. Users can play multiple rounds of the game without leaving or destroying the lobby. Your game's logic determines the lifetime of a lobby.

 7. Lastly, the owner destroys the lobby.

Access
All Lobby Interface functions require that you use a handle as the first parameter. You must ensure that the `EOS_HPlatform` handle is ticking for callbacks to trigger when operations complete.

To access the Lobby Interface, get a `EOS_HLobby` handle through the Platform Interface function `EOS_Platform_GetLobbyInterface`.

Get Handle
To see lobby information an authenticated `EOS_HLobbyDetails` handle is required. Below are three functions to obtain `EOS_HLobbyDetails` based on the local user's connection to the lobby:

Function	Description
`EOS_Lobby_CopyLobbyDetailsHandle`	Available to users currently connected to the lobby.
`EOS_Lobby_CopyLobbyDetailsHandleByInviteId`	Requires an invitation to the lobby.
`EOS_LobbySearch_CopySearchResultByIndex`	Works with lobbies that a local user found in a search.

After successfully using these operations, you receive the `EOS_HLobbyDetails` handle, which provides access to the following information:

Get Lobby Information
Once your handle is set-up, the following lobby information is accessible:

Lobby Info	Description
`LobbyId`	The unique identifier for the lobby.
Lobby Owner	The current owner of the lobby.
Permission Level	Restrictions on who can find or join the lobby.
Space Available	The number of open slots currently available in the lobby.
Max Members	The maximum number of users allowed in the lobby.
Lobby Attributes	Key-value pairs associated with the lobby.
Lobby Member Attributes	Key-value pairs associated with each individual user.

Lobby Properties
All lobby members retain application-specific, key-value attributes supported by the `EOS_Lobby_Attribute` data type. These attributes are represented as numerical, string, or boolean data. Below are lobby properties:

Field	Description
`Key`	Identifying string for the attribute. The system will compare against this string when you search for an attribute.
`Value`	The numerical, string, or boolean data associated with the Key.
`ValueType`	The type of data in the Value.
`Visibility`	The attribute advertised or stored locally with the session `EOS_ELobbyAttributeVisibility`.

- A single user within a lobby, or the lobby itself, can have at most `EOS_LOBBYMODIFICATION_MAX_ATTRIBUTES` attributes.

- Attribute names in the Key field must not exceed

`EOS_LOBBYMODIFICATION_MAX_ATTRIBUTE_LENGTH` characters.

Notifications
Users receive invitations in real time as long as they have registered a callback through `EOS_Lobby_AddNotifyLobbyInviteReceived`.

- The callback data includes all of the relevant information about the invitation. Use`EOS_Lobby_CopyLobbyDetailsHandleByInviteId` to retrieve the invitation in the form of an `EOS_HLobbyDetails`, which the user must have to join the lobby.

- When shutting down the application or going offline, use `EOS_Lobby_RemoveNotifyLobbyInviteReceived` to remove your callback from the notification list.

Create Lobby
To create a lobby make a call to `EOS_Lobby_CreateLobby` the following parameters:
Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_CREATELOBBY_API_LATEST`.
`LocalUserId`	The user creating the lobby.
`MaxLobbyMembers`	The maximum number of users who can be in the lobby at any time.
`Visibility`	Controls if the lobby is visible to the public or is private.
`Options`	Pointer to `EOS_Lobby_CreateLobbyOptions`.

 - Upon success your `EOS_Lobby_OnCreateLobbyCallback` run with the`EOS_Lobby_CreateLobbyCallbackInfo` data structure.

 - The`ResultCode` indicates success. The `LobbyId` contains the new lobby's ID value. You need this ID value to interact with the lobby further.

Find Lobby Owner ID
To find the ID of a lobby's owner make a call to `EOS_LobbyDetails_GetLobbyOwner`.

- `EOS_LobbyDetails_Info` contains information about the lobby and is available through the `EOS_LobbyDetails_CopyInfo` function.

- The `EOS_LobbyDetails_Info` that you receive will be a copy of the EOS SDK's cached information. You own this data and must release it with `EOS_LobbyDetails_Info_Release` function to avoid leaking memory.

- The data is a snapshot and will not update if any information about the lobby changes after you receive it.

Invite User to Lobby

Users who are in a lobby can invite others to join them with the `EOS_Lobby_SendInvite` function.

- The `EOS_Lobby_OnSendInviteCallback` triggers locally once the invitation is successfully sent.

- The target user receives a notification when the invitation arrives.

- The game uses `EOS_Lobby_AddNotifyLobbyInviteAccepted` to monitor when the player clicks Accept in the overlay.

Reject Invitation
To reject an invitation make a call to `EOS_Lobby_RejectInvite` with these parameters:

Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_REJECTINVITE_API_LATEST`.
`LobbyId`	The ID of the lobby associated with the invitation.
`LocalUserId`	The ID of the user who is rejecting the invitation.
`Options`	Pointer to `EOS_Lobby_RejectInviteOptions`.

Once complete, your `EOS_Lobby_OnRejectInviteCallback` function triggers locally with an `EOS_Lobby_RejectInviteCallbackInfo` data structure. The invitation is then permanently deleted from the system.

Update All Invitations
To ensure that your local cache contains all pending invitations make a call to `EOS_Lobby_QueryInvites` and use these parameters:

Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_QUERYINVITES_API_LATEST`.
`LocalUserId`	The ID of the user whose invitations you want to retrieve.
`Options`	Pointer to `EOS_Lobby_QueryInvitesOptions`

- When the operation finishes, your `EOS_Lobby_OnQueryInvitesCallback` runs with an `EOS_Lobby_QueryInvitesCallbackInfo` data structure.

- On success, the local cache contains all pending invitations for the specified user that you can search. This is useful at startup to discover invitations that were sent while the user was offline.

- During play, users with a registered notification should rarely need to call this function.

Retrieve Invitation via Cache

To see a cached invitation make a call to `EOS_Lobby_GetInviteCount` with the parameters below:

Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_GETINVITECOUNT_API_LATEST`.
`LobbyId`	The ID of the lobby associated with the invitation.
`LocalUserId`	The local user whose invitations you want to count.
`Options`	Pointer to `EOS_Lobby_GetInviteCountOptions`.

`EOS_Lobby_GetInviteCount` runs against your local cache and returns `uint32_t` indicating the number of pending invitations in the cache.

Retrieve Invitation via Index
Make a call to `EOS_Lobby_GetInviteIdByIndex` with the parameters in the table:

Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_GETINVITEIDBYINDEX_API_LATEST`.
`LocalUserId`	The local user who owns the invitation.
`Index`	The index of the invitation you want to retrieve.
`Options`	Pointer to `EOS_Lobby_GetInviteIdByIndexOptions`.

On success, the output parameters contain a copy of the invitation data and its size. You are responsible for freeing the buffer when you no longer need it.

Remove Lobby Member
A lobby owner may remove a lobby member at any time. To remove a lobby member make a call to `EOS_Lobby_KickMember` with the following parameters:

Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_KICKMEMBER_API_LATEST`.
`LobbyId`	The ID of the lobby from which the member should be removed.
`LocalUserId`	The ID of the lobby member requesting the removal. This must be the lobby owner.
`Options`	Pointer to `EOS_Lobby_KickMemberOptions`.
`EOS_ProductUserId`	The `TargetUserId`.

- When the operation finishes, you receive a callback to your `EOS_Lobby_OnKickMemberCallback` function with an `EOS_Lobby_KickMemberCallbackInfo` data structure.

- All remaining lobby members will receive notification of the `EOS_LMSC_KICKED` event.

Search
Searching lobbies using different attributes is easy. Below are your options:

- `EOS_Lobby_CreateLobbySearch` creates a handle `EOS_HLobbySearch` that is used to search for lobbies that are set up to be discoverable .

- Use `EOS_LobbySearch_Find` to execute a search. Multiple search operations can run simultaneously with different handles.

- After a search completes, use `EOS_LobbySearch_GetSearchResultCount` to get the number of results for a given `EOS_HLobbySearch` handle.

- To request copies of individual results use`EOS_LobbySearch_CopySearchResultByIndex`.

- When finished with your copy of the data, dispose of it with `EOS_LobbyDetails_Release` to reclaim the memory it was using.

By Attributes
The most robust way to find lobbies is to search for a desired subset of settings.

 - `EOS_LobbySearch_SetParameter` the user to set up `EOS_Lobby_AttributeData` .

 - `EOS_EComparisonOp` is returned dictating how the attribute is compared against all of the existing lobbies stored with the service.

 - One or more of these calls will set the search parameters with an implicit boolean.

By Lobby ID
Search for a lobby ID by making a call to `EOS_LobbySearch_SetLobbyId`.

By User ID
Search for a User ID by making a call to `EOS_LobbySearch_SetTargetUserId`.

Destroy Lobby
To destroy a lobby make a call to `EOS_Lobby_DestroyLobby` using the parameters below:
Parameter	Description
`ApiVersion`	Set to `EOS_LOBBY_DESTROYLOBBY_API_LATEST`.
`LocalUserId`	The owner of the lobby requesting destruction of it.

|
| `LobbyId` | The ID of the lobby to destroy. |

 - After a successful response the`EOS_Lobby_OnDestroyLobbyCallback` callback runs with the `EOS_Lobby_DestroyLobbyCallbackInfo` data structure indicating the lobby ID is destroyed.

 - When the lobby closes, it automatically removes all remaining members and triggers a member status update with state `EOS_LMS_CLOSED`.

Operators
Below are operators that work on all attribution types:

Operator	Description
`EOS_CO_EQUAL`	Search value is the attribute.
`EOS_CO_NOTEQUAL`	Search value is not equal to the attribute.

Below are operators that work on number types:
Operator	Description
`EOS_CO_GREATERTHAN`	Attribute is greater than the search value.
`EOS_CO_GREATERTHANOREQUAL`	Attribute is greater than or equal to the search value.
`EOS_CO_LESSTHAN`	Attribute is less than the search value.
`EOS_CO_LESSTHANOREQUAL`	Attribute is less than or equal to the search value.
`EOS_CO_DISTANCE`	Attribute is close to the search value, following the formula abs(attribute - searchvalue) = 0.

Below are operators that work on string types:
Operator	Description
`EOS_CO_ANYOF`	The attribute is any of in a list of string values separated by semicolons.
`EOS_CO_NOTANYOF`	The attribute is not any in a list of string values.

Limit Search Results
To limit the maximum number of search results that you retrieve, use `EOS_LobbySearch_SetMaxResults`.

Permission Levels
A lobby's permission level controls the conditions that users can discover and join the lobby. The lobby owner can change these at any time by calling `EOS_LobbyModification_SetPermissionLevel`. Below are details about permissions:
Permission Level (macro)	Lobby Behavior

`EOS_LPL_PUBLICADVERTISED`	A public lobby. Any user can find or use it at any time.
`EOS_LPL_JOINVIAPRESENCE`	A friends-only lobby. The lobby is hidden from the public and only friends of the owner can join the lobby. This lobby is not private and can be found if the lobby ID is known.
`EOS_LPL_INVITEONLY`	A private lobby that is hidden from the public. Users join only by invitation from an active user inside the lobby.

Change Ownership
The owner of a lobby may promote another member of the lobby to owner status by making a call to `EOS_Lobby_PromoteMember`. All members will receive notification of this event.

Modify Lobby
A lobby owner can modify a lobby and a user's information by making a call to `EOS_Lobby_UpdateLobbyModification` to get a lobby modification handle `EOS_HLobbyModification`.

A lobby owner can make changes to the lobby using the following functions:
Function	Description
`EOS_LobbyModification_SetPermissionLevel`	Used to set the lobby's Permission Level.
`EOS_LobbyModification_SetMaxMembers`	Changes the maximum number of users allowed in a lobby at the same time. The acceptable range for this value is between the number of users currently in the lobby and `EOS_LOBBY_MAX_LOBBY_MEMBERS`.
`EOS_LobbyModification_AddAttribute`	Adds a new key-value pair to the lobby data.
`EOS_LobbyModification_RemoveAttribute`	Removes a key-value pair from the lobby data.

Modify User Data
Any connected user can modify their own data with the following functions:
Function	Description
`EOS_LobbyModification_AddMemberAttribute`	Adds a new key-value pair to a specific member of the lobby.
`EOS_LobbyModification_RemoveMemberAttribute`	Removes a key-value pair to a specific member of the lobby.

- After making all desired attribute modifications, you must commit the changes to the lobby for them to take effect. You make the commit by making a call to `EOS_Lobby_UpdateLobby`.

- When the operation finishes, the modifications will have been made

and other lobby members will receive notification. You also receive a call to your callback function.

Event Notifications
Users have a persistent connection to the lobby and are notified of events that happen during a lobby's lifetime. Registering for a notification can be done at startup since enough information is provided with each callback to fully describe the lobby affected.

Lobby Data Update
Any time the lobby owner makes a change to the properties of the lobby, all lobby members receive a notification.

- The notification simply states that something has changed.

- The game should evaluate all the available data in the lobby and reapply it to the game.

- Any time a lobby member makes a change to the properties of their own data, all lobby members receive a notification. The notification states that something has changed.

- The game must evaluate all the available data for the lobby member and reapply it to the game.

The following activities of users in the lobby trigger notifications to go out to all other users in the lobby as they occur:

Event Type	Description
`EOS_LMS_JOINED`	A new user has joined the lobby.
`EOS_LMS_LEFT`	A user has left the lobby and their data has been removed.
`EOS_LMS_DISCONNECTED`	A user disconnected.
`EOS_LMS_KICKED`	The lobby owner kicked out a specific user.
`EOS_LSM_PROMOTED`	A user was promoted by the previous owner since the lobby owner left.
`EOS_LMS_CLOSED`	The lobby is closed.

Lobby Usage Limitations
The following are limitations for lobbies and players:
Feature	Service Limit
String attribute length	1000 characters
Connect	30 requests per minute
Find lobby	30 requests per minute
Get lobby ID	100 requests per minute
Find lobbies by user	30 requests per minute
Find lobby by invitation	30 requests per minute
Find invitations by user	30 requests per minute

Join lobby	30 requests per minute
Invite player	30 requests per minute
Create lobby	30 requests per minute
Delete lobby	30 requests per minute
Delete invitation	30 requests per minute
Read lobby data	100 requests per minute
Update lobby attributes	100 requests per minute
Update member attributes	100 requests per minute
Change lobby settings	30 requests per minute
Kick out player	30 requests per minute
Promote player to lobby owner	30 requests per minute
Max players in a lobby	64
Max session attributes	100
Max member attributes	100
Max lobbies at one time	16 per user
String attribute length	1000 characters
Max number of search results for a query	256

Learn more about usage quotas and best practices by reading the [Epic Games Service Usage Limitations](https://dev.epicgames.com/docs/services/en-US/Overview/index.html#serviceusagelimitations) page.