Greenworks API

Greenworks.initAPI()

Returns true if Steam API was successfully initialized or false if not.

You need to be connected and logged into Steam or using the offline mode for cache operations (where available).

Greenworks.getCloudQuotas()

Returns 2 integers (Steam Cloud Quota): nTotalBytes, nAvailableBytes

success function(int nTotalBytes, int nAvailableBytes)
Callback if the method call has been successful

error function()
Callback if the method call encountered an error

Greenworks.saveTextToFile(string fileName, string content, func success, func error)

fileName string
Name of the text file to write to the cloud.

content string
Content to write into the text file.

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

Greenworks.readTextFromFile(string fileName, func success, func error)

fileName string
Name of the text file to read from the cloud.

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

Greenworks.activateAchievement(string achievementId, func success, func error)

achievementId string
Id of the achievement to activate

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

Greenworks.enableCloud()

Enables / Disabled the Cloud feature for the current app.

Greenworks.isCloudEnabled()

Checks if Cloud is Enabled for the current app.

Returns: bool

Greenworks.isCloudEnabledForUser()

Checks if Cloud is Enabled for the current user account.

Returns: bool

Greenworks.getNumberOfPlayers()

Returns the current number of players on Steam.

Returns: int

Greenworks.ugcPublish(string fileName, string title, string description, string imageFile, func successCallback, func errorCallback, func progressCallback)

---

fileName string
Name of the file to publish (preview image)

title string
Title of the workshop item

description string
Description of the workshop item

imageFile string
Name of the image file to publish (preview image)

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})
Callback on a download progress

Greenworks.ugcPublishUpdate(int publishedFileId, string fileName, string title, string description, string imageFile, func successCallback, func errorCallback, func progressCallback)

---

publishedFileId int
Id of the published workshop item you wish to update (as obtained through ugcGetItems or ugcGetUserItems)

fileName string
Name of the file to publish (preview image)

title string
Title of the workshop item

description string
Description of the workshop item

imageFile string
Name of the image file to publish (preview image)

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})
Callback on a download progress

Greenworks.ugcGetItems(int type, int sort, func successCallback, func errorCallback, func progressCallback)

---

type int
Type corresponds to k_EUGCMatchingUGCType
EUGCMatchingUGCType_Items = 0,
EUGCMatchingUGCType_ItemsMtx = 1,
EUGCMatchingUGCType_ItemsReadyToUse = 2,
EUGCMatchingUGCType_Collections = 3,
EUGCMatchingUGCType_Artwork = 4,
EUGCMatchingUGCType_Videos = 5,
EUGCMatchingUGCType_Screenshots = 6,
EUGCMatchingUGCType_AllGuides = 7,
EUGCMatchingUGCType_WebGuides = 8,
EUGCMatchingUGCType_IntegratedGuides = 9,
EUGCMatchingUGCType_UsableInGame = 10,
EUGCMatchingUGCType_ControllerBindings = 11

sort int
Sort corresponds to k_EUGCQuery
EUGCQuery_RankedByVote = 0,
EUGCQuery_RankedByPublicationDate = 1,
EUGCQuery_AcceptedForGameRankedByAcceptanceDate = 2,
EUGCQuery_RankedByTrend = 3,
EUGCQuery_FavoritedByFriendsRankedByPublicationDate = 4,
EUGCQuery_CreatedByFriendsRankedByPublicationDate = 5,
EUGCQuery_RankedByNumTimesReported = 6,
EUGCQuery_CreatedByFollowedUsersRankedByPublicationDate = 7,
EUGCQuery_NotYetRated = 8,
EUGCQuery_RankedByTotalVotesAsc = 9,
EUGCQuery_RankedByVotesUp = 10,
EUGCQuery_RankedByTextSearch = 11

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})
Callback on a download progress

Greenworks.ugcGetUserItems(int type, int sort, int filter, func successCallback, func errorCallback, func progressCallback)

---

type int

Type corresponds to k_EUGCMatchingUGCType

EUGCMatchingUGCType_Items	= 0,
EUGCMatchingUGCType_ItemsMtx	= 1,
EUGCMatchingUGCType_ItemsReadyToUse	= 2,
EUGCMatchingUGCType_Collections	= 3,
EUGCMatchingUGCType_Artwork	= 4,
EUGCMatchingUGCType_Videos	= 5,
EUGCMatchingUGCType_Screenshots	= 6,
EUGCMatchingUGCType_AllGuides	= 7,
EUGCMatchingUGCType_WebGuides	= 8,
EUGCMatchingUGCType_IntegratedGuides	= 9,
EUGCMatchingUGCType_UsableInGame	= 10,
EUGCMatchingUGCType_ControllerBindings	= 11

sort int

Sort corresponds to EUserUGCListSortOrder

EUserUGCListSortOrder::CreationOrderDesc	= 0,
EUserUGCListSortOrder::CreationOrderAsc	= 1,
EUserUGCListSortOrder::TitleAsc	= 2,
EUserUGCListSortOrder::LastUpdatedDesc	= 3,
EUserUGCListSortOrder::SubscriptionDateDesc	= 4,
EUserUGCListSortOrder::VoteScoreDesc	= 5,
EUserUGCListSortOrder::ForModeration	= 6,

filter int

Filter corresponds to EUserUGCList

EUserUGCList::Published	= 0,
EUserUGCList::VotedOn	= 1,
EUserUGCList::VotedUp	= 2,
EUserUGCList::VotedDown	= 3,
EUserUGCList::WillVoteLater	= 4,
EUserUGCList::Favorited	= 5,
EUserUGCList::Subscribed	= 6,
EUserUGCList::UsedOrPlayed	= 7,
EUserUGCList::Followed	= 8,

success function()

Callback if the method call has been successful

error function(msg)

Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})

Callback on a download progress

Greenworks.ugcDownloadItem(string fileName, int hFile, string targetFolder, func successCallback, func errorCallback, func progressCallback)

---

fileName string
Name of the file to download (as obtained through ugcGetItems or ugcGetUserItems)

hFile int
Handle to the file you wish to download (as obtained through ugcGetItems or ugcGetUserItems)

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})
Callback on a download progress

Greenworks.ugcSynchronizeItems(string targetFolder, func success, func error, func progress)

---

targetFolder string
Destination folder to download and sync WS items (usually a custom cache for better control)

success function(obj:{items: array_UGC_DETAILS, count: array length})
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})
Callback on a download progress

Greenworks.getCurrentGameLanguage()

Returns the current language from Steam specifically set for the Game

Returns: string

Greenworks.getCurrentUILanguage()

Returns the current language from Steam set in the UI

Returns: string

Greenworks.getSteamId()

Returns extensive information from the Steam ID object of the current user.

Returns: object

flags	(object) Boolean flags describing type of user information
anonymous	*Is this an anonymous account?*
anonymousGameServer	*Is this an anonymous game server account?*
anonymousGameServerLogin	*Is this an anonymous game server account login request?*
anonymousUser	*Is this an anonymouse user account?*
chat	*Is this a chat account?*
clan	*Is this a clan account?*
consoleUser	*Is this a console user (PSN) account?*
contentServer	*Is this a content server account?*
gameServer	*Is this a game server account?*
individual	*Is this an individual account?*
gameServerPersistent	*Is this a persistent game server account?*
lobby	*Is this a lobby (chat) account?*

type	(object) Object describing type of user account
name	*Name of the resulting enum value (i.e. k_EAccountTypeClan)*
value	*Value of the resulting enum value (i.e. 0)*

accountId	(int)	Account ID (Steam ID)
staticAccountId	(int)	Static int64 representation of a Steam ID
screenName	(string)	Steam Screen Name
level	(int)	Steam Level
isValid	(boolean)	Is it is a valid account

Greenworks.ugcShowOverlay(optional workshopItemId)

Shows the Steam overlay pointed to the app's workshop page or to the optionally specified workshop item.

Greenworks.ugcUnsubscribe(int publishedFileId, func success, func error, func progress)

Shows the Steam overlay pointed to the app's workshop page or to the optionally specified workshop item.

publishedFileId int
Id of the published workshop item that should be unsubscribed

success function()
Callback if the method call has been successful

error function(msg)
Callback if the method call encountered an error

progress function(obj{status: string, reason: string, value: int})
Callback on a unsubscribe progress

Greenworks.getCurrentGameInstallDir()

Not implemented.

Greenworks.runCallbacks()

Internal. Calls underlying SteamAPI_RunCallbacks().

Greenworks.Utils

Accesses the GreenUtils class that provides a set of useful utilities.

Greenworks.Utils.createArchive(string zipFile, string sourceDir, string password, int compressionLevel, func success, func error)

Creates a zip archive.

Greenworks.Utils.extractArchive(string zipFile, string targetDir, string password, func success, func error)

Extracts a zip archive.

Greenworks.Utils.sleep(int ms)

Process Sleep for specified milliseconds.

Greenworks.Utils.getOS(int ms)

Returns the current operating system (linux, win, apple).

Returns: string

Greenworks.Utils.move(string sourceFolder, string targetFolder)

Moves the specified source folder to a target folder (within the same device!).
Use this method to assist you in moving workshop contents around.

Returns: string

Greenworks.Utils.enableConsole()

Enables output to JS Console

Greenworks.Utils.disableConsole()

Disables output to JS Console

Greenworks.Utils.enableWriteToLog(string targetFile)

Enables output to the specified log file (make sure you have proper write permissions).

Greenworks.Utils.disableWriteToLog()

Disables previous enabled output to a log file.