

GOG API Documentation — GOG-API 0.1

documentation

Welcome to the unofficial documentation of the APIs used by the GOG website and Galaxy client. It's a very young project, so don't be surprised if something is missing. But now get ready for a wild ride into a world where GET and POST don't mean anything and consistency is a lucky mistake.

Contents

- [Authentication](#)
 - [Introduction](#)
 - [Auth-Flow](#)
 - [Authorizing a Request](#)
 - [Methods](#)
- [Account Management](#)
 - [User](#)
 - [Games & Movies](#)
 - [Wishlist](#)
 - [Tags](#)
 - [Settings](#)
 - [Friends](#)
- [Listing](#)
- [Store](#)
 - [Cart](#)
 - [Order](#)
 - [History](#)
 - [Wallet](#)
- [Reviews](#)
- [GOG Connect](#)
- [Galaxy APIs](#)
 - [api.gog.com](#)
 - [auth.gog.com](#)
 - [cfg.gog.com](#)
 - [chat.gog.com](#)
 - [content-system.gog.com](#)
 - [cdn.gog.com](#)

- gameplay.gog.com
- presence.gog.com
- users.gog.com
- [Game ID List](#)

Links

- [Source on Github](#)
- [My Python implementation](#)
- [GOG Downloader Documentation by Sude](#)

Contributors


- Yepoleb
- Barefoot_Monkey

Account Management — GOG-API 0.1 documentation

Account Management

Methods used to manage the user' s account.

User

GET /userData.json 

Information about the logged in user.

GET /userData.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "country": "AT", "currencies": [ { "code": "EUR", "symbol": "€" }, { "code": "USD", "symbol": "$" } ],
  "selectedCurrency": { "code": "EUR", "symbol": "€" }, "preferredLanguage": { "code": "en", "name": "English" },
  "ratingBrand": "PEGI", "isLoggedIn": true, "checksum": { "cart": null, "games":
  "c1fc44f3808bd755560e1b00d34451a1", "wishlist": "fcfd279ac1042f8baf8e659729ab1b89", "reviews_votes":
  null, "games_rating": null }, "updates": { "messages": 0, "pendingFriendRequests": 0, "unreadChatMessages":
  0, "products": 0, "forum": 0, "total": 0 }, "userId": "48628349971017", "username": "Yepoleb", "email":
  "mail@example.com", "personalizedProductPrices": [], "personalizedSeriesPrices": [] }
```

GET /user/set-redirect-url 

Sets URL to redirect to after login. You shouldn't need this with a native client which is always logged in.

| Query Parameters: |

- **url** – Redirect URL, the only accepted value seems to be `checkout`

|

Example request:

GET /user/set-redirect-url?url=checkout HTTP/1.1 Host: embed.gog.com

Example response:

No content

GET /user/reviewTipsStatus.json [1](#)

Checks if the user has read the tips that pop up before you can write your first review.

Example request:

GET /user/reviewTipsStatus.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "userId": "48628349971017", "readTips": false }
```

GET /users/info/ (int: user_id) [1](#)

Returns the public information about a user.

| Query Parameters: |

- **expand** (*str*) – Additional sections to request. Possible values: `friendStatus`, `wishlistStatus`, `blockedStatus`

|| Response JSON Object: |

- **friendStatus.status** (*int*) –
 - ANONYMOUS_USER = 0: No special relationship with this user.
 - INVITED_USER = 1: You have sent this user a friend request.
 - INVITED_BY_USER = 2: You have received a friend request from this user.
 - FRIEND = 3: You are friends with this user.
- **friendStatus.dateCreated** (*int*) – Timestamp of when a friend request was sent or `null`.
- **friendStatus.dateAccepted** (*int*) – Timestamp of when a friend request was accepted or `null`.
- **wishlistStatus.sharing** (*int*) –
 - WISHLIST_PRIVATE = 0
 - WISHLIST_PUBLIC = 1
 - WISHLIST_FOR_FRIENDS = 2

|

Example request:

GET /users/info/48628349971017?expand=friendStatus,wishlistStatus,blockedStatus HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "id": "48628349971017", "username": "Yepoleb", "userSince": 1449237763, "avatars": { "small": "https://images.gog.com/3f9e109ac09308f7d52c607c8571e63d5fb482acca499a83e767dfff7f00d57d_avs.jpg", "small2x": "https://images.gog.com/3f9e109ac09308f7d52c607c8571e63d5fb482acca499a83e767dfff7f00d57d_avs2.jpg", "medium": "https://images.gog.com/3f9e109ac09308f7d52c607c8571e63d5fb482acca499a83e767dfff7f00d57d_avm.jpg", "medium2x": "https://images.gog.com/3f9e109ac09308f7d52c607c8571e63d5fb482acca499a83e767dfff7f00d57d_avm2.jpg", "large": "https://images.gog.com/3f9e109ac09308f7d52c607c8571e63d5fb482acca499a83e767dfff7f00d57d_avl.jpg", "large2x": "https://images.gog.com/3f9e109ac09308f7d52c607c8571e63d5fb482acca499a83e767dfff7f00d57d_avl2.jpg" }, "friendStatus": { "id": "48628349971017", "status": 0, "dateCreated": null, "dateAccepted": null }, "wishlistStatus": { "sharing": 2, "url": "https://embed.gog.com/u/Yepoleb/wishlist" }, "blockedStatus": { "blocked": false }, "chatStatus": { "url": "https://embed.gog.com/u/Yepoleb/chat", "isChatRestricted": false } }
```

Games & Movies

GET /user/data/games 


List of games and movies the account owns. Use [GET /account/getFilteredProducts](#) for more than just the IDs.

Example request:

GET /user/data/games HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "owned": [ 1207658691, 1207658713, 1207658805, 1207658924, 1207658930, 1207658945, 1207658957, 1929434313, 1949616134, 1432207890, 1444035366, 1444036272, 1443696086 ] }
```

GET /account/gameDetails/(int: game_id) .json 

Returns detailed information about a game. Seems to work with movies as well, but they have their own method.

Example request:

GET /account/gameDetails/1207658691.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "title": "Unreal Tournament 2004 Editor's Choice Edition", "backgroundImage": "//images-4.gog.com/ebed1d5546a4fa382d7d36db8aee7f298eac7db3a8dc2f4389120b5b7b3155a9", "cdKey": "", "textInformation": "", "downloads": [ [ "English", { "windows": [ { "manualUrl": "/downlink/unreal_tournament_2004_ece/en1installer3", "downloaderUrl":
```

```
"gogdownloader://unreal_tournament_2004_ece/installer_win_en", "name": "Unreal Tournament 2004 Editor\'s Choice Edition (Part 1 of 3)", "version": null, "date": "", "size": "1 MB" }, { "manualUrl": "/downlink/unreal_tournament_2004_ece/en1installer4", "downloaderUrl": "gogdownloader://unreal_tournament_2004_ece/installer_win_en", "name": "Unreal Tournament 2004 Editor\'s Choice Edition (Part 2 of 3)", "version": null, "date": "", "size": "1.5 GB" }, { "manualUrl": "/downlink/unreal_tournament_2004_ece/en1installer5", "downloaderUrl": "gogdownloader://unreal_tournament_2004_ece/installer_win_en", "name": "Unreal Tournament 2004 Editor\'s Choice Edition (Part 3 of 3)", "version": null, "date": "", "size": "507 MB" } ] ] ], "extras": [ { "manualUrl": "/downlink/file/unreal_tournament_2004_ece/6093", "downloaderUrl": "gogdownloader://unreal_tournament_2004_ece/6093", "name": "manual (33 pages)", "type": "manuals", "info": 1, "size": "2 MB" }, { "manualUrl": "/downlink/file/unreal_tournament_2004_ece/6073", "downloaderUrl": "gogdownloader://unreal_tournament_2004_ece/6073", "name": "HD wallpapers", "type": "wallpapers", "info": 12, "size": "115 MB" }, { "manualUrl": "/downlink/file/unreal_tournament_2004_ece/6083", "downloaderUrl": "gogdownloader://unreal_tournament_2004_ece/6083", "name": "avatars", "type": "avatars", "info": 8, "size": "1 MB" } ], "dlcs": [], "tags": [], "isPreOrder": false, "releaseTimestamp": 1227585600, "messages": [], "changelog": null, "forumLink": "https://embed.gog.com/forum/unreal_series", "isBaseProductMissing": false, "missingBaseProduct": null }
```

GET /account/movieDetails/ (int: movie_id) .json [🔗](#)

Returns detailed information about a movie.

Example request:

GET /account/movieDetails/1207665463.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "title": "Art of Playing, The", "backgroundImage": "//images-1.gog.com/a3e48e4e370e2e7e9cdf648b712ff0506b994b404b64993e3fab4934294a4890", "cdKey": "", "textInformation": "", "downloads": [ { "manualUrl": "/downlink/the_art_of_playing/en1video1", "downloaderUrl": "gogdownloader://the_art_of_playing/video_en1video1", "playerUrl": "http://www.gog.com/video/the_art_of_playing/en1video1", "name": "The Art of Playing (1080p)", "size": "1.1 GB" }, { "manualUrl": "/downlink/the_art_of_playing/en1video2", "downloaderUrl": "gogdownloader://the_art_of_playing/video_en1video2", "playerUrl": "http://www.gog.com/video/the_art_of_playing/en1video2", "name": "The Art of Playing (720p)", "size": "382 MB" }, { "manualUrl": "/downlink/the_art_of_playing/en1video3", "downloaderUrl": "gogdownloader://the_art_of_playing/video_en1video3", "playerUrl": "http://www.gog.com/video/the_art_of_playing/en1video3", "name": "The Art of Playing (480p)", "size": "189 MB" } ], "extras": [ { "manualUrl": "/downlink/file/the_art_of_playing/6103", "downloaderUrl": "gogdownloader://the_art_of_playing/6103", "name": "manual (1 page)", "type": "manuals", "info": 1, "size": "1 MB" } ], "dlcs": [], "tags": [], "isPreOrder": false, "releaseTimestamp": 1432207890, "messages": [], "changelog": null, "forumLink": "https://embed.gog.com/forum/general", "isBaseProductMissing": false, "missingBaseProduct": null }
```

GET /user/games_rating.json [🔗](#)

Returns the products the account has rated. Rating numbers are stars * 10

Example request:

GET /user/games_rating.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "games_rating": { "1207658957": 40, "1207659032": 50 }, "checksum": "175d07086bff9322646f1dad2749483e" }
```

GET /user/review_votes.json [🔗](#)

Returns review IDs the user has voted on.

Example request:

GET /user/review_votes.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "reviews": [ 123456, 1112223 ], "checksum": "76c03aa67251e46db3271adf4641b815" }
```

POST /account/hideProduct/ (*int: product_id*) [🔗](#)

Hides a product from your library.

Example request:

POST /user/hideProduct/1430740458 HTTP/1.1 Host: embed.gog.com

Example response:

```
{ }
```

POST /account/unhideProduct/ (*int: product_id*) [🔗](#)

Unhides a product from your library.

Example request:

POST /user/unhideProduct/1430740458 HTTP/1.1 Host: embed.gog.com

Example response:

```
{ }
```

GET /account/getFilteredProducts [🔗](#)

Returns a list of products with filters.

| Query Parameters: |

- **hidden** (*bool*) – Include hidden products
- **mediaType** (*str*) – Filter by media type. Possible values: game, movie
- **page** (*int*) – Page number
- **sortBy** (*str*) – Sort by. Possible values: date_purchased, title, release_date
- **sortOrder** (*str*) – Sort order. Possible values: asc, desc

|

Example request:

GET /account/getFilteredProducts?hidden=false&mediaType=game&page=1&sortBy=title&sortOrder=asc
HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "page": 1, "totalResults": 1, "products": [ { "id": 1207658691, "title": "Unreal Tournament 2004 Editor\'s Choice Edition", "mediaType": "game", "releaseDate": 1227585600, "purchaseDate": 1430740458, "isHidden": false, "rating": { "score": 40, "votes": 100 }, "image": "//images-4.gog.com/ebed1d5546a4fa382d7d36db8aee7f298eac7db3a8dc2f4389120b5b7b3155a9", "url": "https://embed.gog.com/game/unreal_tournament_2004_editors_choice_edition" } ] }
```

Wishlist [🔗](#)

GET /user/wishlist.json [🔗](#)

Returns the user's wishlist.

Example request:

GET /user/wishlist.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "products": [ { "id": 1207658691, "title": "Unreal Tournament 2004 Editor\'s Choice Edition", "image": "//images-4.gog.com/ebed1d5546a4fa382d7d36db8aee7f298eac7db3a8dc2f4389120b5b7b3155a9", "url": "https://embed.gog.com/game/unreal_tournament_2004_editors_choice_edition", "releaseDate": 1227585600, "isPreOrder": false, "isComingSoon": false, "price": { "amount": "9.99", "currency": "EUR", "symbol": "€", "isFree": false }, "isDiscounted": false, "discountPercentage": 0, "discountAmount": "0.00", "basePrice": { "amount": "9.99", "currency": "EUR", "symbol": "€", "isFree": false }, "isWishlisted": true } ], "checksum": "fcfd279ac1042f8baf8e659729ab1b89" }
```

POST /user/wishlist/add/ (*int: product_id*) [🔗](#)

Adds a product to the user's wishlist.

Example request:

POST /user/wishlist/add/1207658691 HTTP/1.1 Host: embed.gog.com

Example response:

```
{ }
```

POST /user/wishlist/remove/ (*int: product_id*) [🔗](#)

Removes a product from the user's wishlist.

Example request:

POST /user/wishlist/remove/1207658691 HTTP/1.1 Host: embed.gog.com

Example response:

```
{ }
```

Tags [🔗](#)

GET /user/tags.json [🔗](#)

Returns the user's tags.

Example request:

GET /user/tags.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "tags": [ { "id": 1, "name": "Action", "productCount": 10 }, { "id": 2, "name": "Adventure", "productCount": 5 } ],  
  "checksum": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6" }
```

POST /user/tags/add/ (*int: product_id*)/` (*int: tag_id*) [🔗](#)

Adds a tag to a product.

Example request:

POST /user/tags/add/1207658691/1 HTTP/1.1 Host: embed.gog.com

Example response:

```
{ }
```

POST /user/tags/remove/ (*int: product_id*)/` (*int: tag_id*) [🔗](#)

Removes a tag from a product.

Example request:

POST /user/tags/remove/1207658691/1 HTTP/1.1 Host: embed.gog.com

Example response:

```
{ }
```

Settings [🔗](#)

GET /user/settings.json [🔗](#)

Returns the user's settings.

Example request:

GET /user/settings.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "settings": { "language": "en", "currency": "EUR", "newsletter": true, "emailNotifications": true }, "checksum":  
  "q1w2e3r4t5y6u7i8o9p0a1s2d3f4g5h6" }
```

POST /user/settings/update [🔗](#)

Updates the user's settings.

| Query Parameters: |

- **language** (*str*) – Language code (e.g., en, de, fr)
- **currency** (*str*) – Currency code (e.g., EUR, USD)
- **newsletter** (*bool*) – Newsletter subscription status
- **emailNotifications** (*bool*) – Email notifications status

|

Example request:

POST /user/settings/update?language=en¤cy=EUR&newsletter=true&emailNotifications=true HTTP/1.1
Host: embed.gog.com

Example response:

```
{}
```

Friends [🔗](#)

GET /user/friends.json [🔗](#)

Returns the user's friends.

Example request:

GET /user/friends.json HTTP/1.1 Host: embed.gog.com

Example response:

```
{ "friends": [ { "id": 12345, "username": "Friend1", "avatar": "https://images.gog.com/avatar1.jpg" }, { "id": 67890, "username": "Friend2", "avatar": "https://images.gog.com/avatar2.jpg" } ], "checksum": "z1x2c3v4b5n6m7q8w9e0r1t2y3u4i5o6" }
```

POST /user/friends/add/ (*int: user_id*) [🔗](#)

Adds a user as a friend.

Example request:

POST /user/friends/add/12345 HTTP/1.1 Host: embed.gog.com

Example response:

```
{}
```

POST /user/friends/remove/ (*int: user_id*) [🔗](#)

Removes a user from friends.

Example request:

POST /user/friends/remove/12345 HTTP/1.1 Host: embed.gog.com

Example response:

```
{}
```

```
POST /user/friends/accept/ (int: user_id) 🔗
```

Accepts a friend request.

Example request:

```
POST /user/friends/accept/12345 HTTP/1.1 Host: embed.gog.com
```

Example response:

```
{}
```

```
POST /user/friends/decline/ (int: user_id) 🔗
```

Declines a friend request.

Example request:

```
POST /user/friends/decline/12345 HTTP/1.1 Host: embed.gog.com
```

Example response:

```
{}
```

```
POST /user/friends/block/ (int: user_id) 🔗
```

Blocks a user.

Example request:

```
POST /user/friends/block/12345 HTTP/1.1 Host: embed.gog.com
```

Example response:

```
{}
```

```
POST /user/friends/unblock/ (int: user_id) 🔗
```

Unblocks a user.

Example request:

```
POST /user/friends/unblock/12345 HTTP/1.1 Host: embed.gog.com
```

Example response:

```
{}
```