File ▾ Edit ▾ Insert ▾ Generate Server ▾ Generate Client ▾ About ▾ Try our new Editor ↗

Gacha Game Backend API

1.0.0 OAS 3.0

API for managing gacha rolls, user inventories, auctions, transactions, and user authentication.

Servers

http://localhost:8080 - Local development server

Authorize

PLAYER

Player-specific operations and authentication

POST `/users/auth/login` Login a player

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "username": "player123",
  "password": "securepassword"
}
```

Responses

Code	Description	Links
200	Successful login	*No links*

Code	Description	Links
401	Invalid credentials	*No links*
500	Database error	*No links*
501	Unexpected error	*No links*

POST `/users/auth/register` Register a new player

Parameters Try it out

No parameters

Request body required multipart/form-data

username
`string`

password
`string`

email
`string`

image
`object`

Responses

Code	Description	Links
200	Player successfully registered	*No links*
400	Missing required fields	*No links*
409	Username or email already exists	*No links*

Code	Description	Links
500	Unexpected error	*No links*

DELETE `/users/auth/logout` Logout a player

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Successful logout	*No links*
401	Invalid user type	*No links*
500	Unexpected error	*No links*

DELETE `/users/auth/delete` Delete a user account

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Account deleted successfully	*No links*
400	Session token is required	*No links*
408	Session token not found	*No links*
500	Unexpected error	*No links*

PUT `/users/player/update` Update player account details

Parameters Try it out

No parameters

Request body required multipart/form-data

username
`string`

password
`string`

email
`string`

image
`object`

Responses

Code	Description	Links
200	Account successfully updated	*No links*
400	Invalid user type	*No links*
408	User not found	*No links*
500	Unexpected error	*No links*

GET `/users/player/my_gacha_list` Retrieve the list of gacha items for a user

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	List of gacha items retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*
404	No gacha items found for the user Media type **application/json** **Example Value** Schema	*No links*

```
[
  {
    "gacha_id": "1",
    "name": "Auto",
    "rarity": "epic",
    "status": "available",
    "description": "Fast auto",
    "image": "iVBORw0KGgoAAAANSUhEUgAA...",
    "locked": "unlocked",
    "acquired_date": "2023-10-01T12:00:00Z"
  }
]
```

```
{
  "error": "No gacha items found for the user"
}
```

GET `/users/player/gacha/{user_id}/{gacha_id}`

Retrieve information about a specific gacha item for a user

Parameters

Try it out

Name	Description
user_id * required `string` *(path)*	*Example* : 1 1
gacha_id * required `string`	*Example* : 1 1

Name	Description
(path)	

Responses

Code	Description	Links
200	Gacha item information retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "gacha_id": "1", "name": "Auto", "rarity": "epic", "status": "available", "description": "Fast auto", "image": "iVBORw0KGgoAAAANSUhEUgAA..." }`	*No links*
404	User not found or gacha item not found	*No links*

POST `/users/player/real_money_transaction` Handle real money transactions

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "amount": 100
}
```

Responses

Code	Description	Links
200	Account topped up successfully	*No links*
400	Missing amount for transaction or invalid amount	*No links*

GET `/users/player/get_user_balance` Retrieve the balance of a user

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	User balance retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "currency_balance": 1500 }`	*No links*
400	Invalid user type	*No links*
408	User not found	*No links*
500	Unexpected error	*No links*

GET `/users/player/get_user` Get user information

Retrieve information for the authenticated user. Only users with the 'PLAYER' type can access this endpoint.

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	User information retrieved successfully. Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "user_id": 0, "username": "string", "email": "string", "currency_balance": 0, "image": {}, "session_token": "string" }`	*No links*
404	User not found	*No links*
500	Unexpected error	*No links*

GET `/users/player/get_user/{user_id}` Get user information by ID (Admin only)

Retrieve user information by ID. This endpoint is restricted to admin users and is hidden from the API documentation.

Parameters

Try it out

Name	Description
user_id * required `integer` *(path)*	The ID of the user to retrieve. user_id

Responses

Code	Description	Links
200	User information retrieved successfully. Media type application/json Controls `Accept` header. **Example Value** Schema	*No links*
404	User not found	*No links*
500	Unexpected error	*No links*

```
{
  "user_id": 0,
  "username": "string",
  "email": "string",
  "currency_balance": 0,
  "image": {},
  "session_token": "string"
}
```

PUT `/users/player/update_balance/{user_type}` Update the balance of a user

Parameters Try it out

Name	Description
user_type * required `string` *(path)*	*Available values* : PLAYER, ADMIN *Example* : PLAYER PLAYER

Request body required application/json

Example Value Schema

```
{
  "user_id": "1",
  "amount": "100",
  "type": "credit"
}
```

Responses

Code	Description	Links
200	Balance updated successfully	*No links*
400	Missing data to update balance	*No links*
404	User not found	*No links*
500	Unexpected error	*No links*

ADMIN Administrative functions and authentication

POST `/users/admin_auth/register` Register a new admin

Parameters Try it out

No parameters

Request body required multipart/form-data

username
`string`

password
`string`

email
`string`

Responses

Code	Description	Links
201	Admin successfully registered	*No links*
400	Missing required fields	*No links*

Code	Description	Links
409	Username or email already exists	*No links*
500	Unexpected error	*No links*

POST `/users/admin_auth/login` Login an admin and retrieve a JWT token

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "username": "admin123",
  "password": "securepassword"
}
```

Responses

Code	Description	Links
200	Successful login and returns a JWT token Media type application/json Controls `Accept` header. **Example Value** Schema `{ "token": "jwt_token", "message": "Login successful", "user_id": "1" }`	*No links*
400	Invalid user type	*No links*
401	Invalid credentials	*No links*
500	Database error	*No links*

DELETE `/users/admin_auth/logout` Logout the admin

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Successful logout	*No links*
401	Invalid user type	*No links*
500	Unexpected error	*No links*

TRANSACTION

Handles in-game currency transactions for players and administrators

POST `/transaction/add_transaction` Add a new transaction

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "user_id": "1",
  "amount": 100,
  "type": "credit"
}
```

Responses

Code	Description	Links
200	Transaction added successfully	*No links*
500	Failed to add transaction	*No links*

GET `/transaction/get_user_transactions/{user_id}` Retrieve all transactions for a user

Parameters Try it out

Name	Description
user_id * required `string` *(path)*	*Example* : 1 1

Responses

Code	Description	Links
200	Transactions retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*
400	User_id not found	*No links*
404	No transactions found	*No links*

```
[
  {
    "transaction_id": "txn_001",
    "user_id": "1",
    "amount": 100,
    "type": "credit",
    "timestamp": "2023-10-01T12:00:00Z"
  }
]
```

GET `/transaction/get_user_transactions` Retrieve all transactions for current user

Parameters

Try it out

No parameters

Responses

Code	Description	Links

200	Transactions retrieved successfully
	Media type
	application/json
	Controls Accept header.
	Example Value Schema
	{
	"transaction_id": "txn_001",
	"user_id": "1",
	"amount": 100,
	"type": "credit",
	"timestamp": "2023-10-01T12:00:00Z"
	}
		No links

400	User_id not found	*No links*
403	Admin can not get their own transactions	*No links*
404	No transactions found	*No links*

GET `/transaction/get_transaction` Retrieve a specific transaction

Parameters

Try it out

Name	Description
transaction_id * required `string` *(query)*	*Example* : txn_001 txn_001

Responses

Code	Description	Links
200	Transaction retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "transaction_id": "txn_001", "user_id": "1", "amount": 100, "type": "credit", "timestamp": "2023-10-01T12:00:00Z" }`	*No links*
404	Transaction not found	*No links*

GET `/transaction/all` Retrieve all transactions

Retrieve all transactions from the database.

Parameters Try it out

Name	Description
user_id `string` *(query)*	*Example* : 1 1

Responses

Code	Description	Links
200	A JSON array of all transactions Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*

Code	Description	Links

	[
	  {
	  "transaction_id": 1,
	  "user_id": 123,
	  "amount": 100,
	  "date": "2023-10-01T12:00:00Z"
	  }
	]	

404	No transactions found	*No links*

AUCTIONS Handles the creation and management of auctions and bids

POST `/auctions/add` Add a new auction

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "gacha_id": "1",
  "base_price": 100
}
```

Responses

Code	Description	Links
201	Auction created successfully	*No links*
400	Missing data to create auction	*No links*
403	Unauthorized access	*No links*
404	Gacha is locked or not exist	*No links*

Code	Description	Links
500	Unexpected error	*No links*

GET `/auctions/check_auction_status` Check and update the status of auctions

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Auction status updated successfully Media type application/json Controls `Accept` header. **Example Value** Schema `{ "message": "Auction status updated successfully" }`	*No links*
500	An error occurred while checking auction status Media type application/json **Example Value** Schema `{ "error": "An error occurred while checking auction status" }`	*No links*

GET `/auctions/all` Retrieve all auctions or filter by auction status

This endpoint retrieves all auctions from the database. Optionally, it can filter the auctions based on their status (active or expired) using a query parameter.

Parameters Try it out

Name	Description
status `string` *(query)*	The status of the auctions to filter by. Can be "active" or "expired". *Available values* : active, expired, completed --

Responses

Code	Description	Links
200	A JSON response containing a list of auctions. Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "auctions": [ { "additionalProp1": {} } ] }`	*No links*
500	Unexpected error occurred	*No links*

GET `/auctions/all_active` Retrieve all auctions or filter by auction status

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	A JSON response containing a list of auctions. Media type **application/json**	*No links*

Code	Description	Links
	Controls `Accept` header. **Example Value** Schema `{ "auctions": [ { "additionalProp1": {} } ] }`	
500	Unexpected error occurred	*No links*

GET `/auctions/get_gacha_auctions` Retrieve gacha auctions based on gacha_id

This endpoint handles GET requests to fetch auctions associated with a specific gacha_id.

Parameters Try it out

Name	Description
gacha_id * required `string` *(query)*	The ID of the gacha to retrieve auctions for. gacha_id

Responses

Code	Description	Links
200	A JSON response containing the list of auctions. Media type **application/json** Controls `Accept` header. **Example Value** Schema `[ { "additionalProp1": {} } ]`	*No links*

Code	Description	Links
400	Missing gacha_id parameter	*No links*
404	No auctions found for the gacha	*No links*
500	Database error occurred	*No links*

POST `/auctions/bid` Place a bid on an auction

This endpoint allows a user to place a bid on an auction.

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "auction_id": "string",
  "bid_amount": 0
}
```

Responses

Code	Description	Links
200	Bid placed successfully Media type application/json Controls `Accept` header. **Example Value** Schema `{ "message": "string" }`	*No links*
400	Missing data for bid	*No links*

Code	Description	Links
403	You can't place a bid as an admin	*No links*
408	Auction not found or already ended	*No links*
409	Failed to place bid	*No links*
500	Database error occurred	*No links*

GET `/auctions/bids` Retrieve bids for a specific auction

This endpoint handles GET requests to retrieve bids for a specific auction.

Parameters Try it out

Name	Description
auction_id * required `string` *(query)*	The ID of the auction to retrieve bids for. auction_id

Responses

Code	Description	Links
200	A JSON response containing the list of bids. Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "bids": [ { "additionalProp1": {} } ] }`	*No links*
400	Missing auction_id parameter	*No links*

Code	Description	Links
500	Database error occurred	*No links*

GACHA Handles gacha items and inventory management

POST `/gacha/add` Add a new gacha item

Parameters Try it out

No parameters

Request body required multipart/form-data

name
`string`

rarity
`string`

status
`string`

description
`string`

image
`string($binary)`

Responses

Code	Description	Links
201	Gacha item added successfully Media type application/json	*No links*

Code	Description	Links
	Controls `Accept` header. **Example Value** Schema `{ "message": "string" }`	
400	Missing data to add gacha item	*No links*
500	Database error	*No links*

POST `/gacha/roll` Perform a gacha roll for a random item

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Gacha roll successful Media type **application/json** Controls `Accept` header. **Example Value** Schema `{ "message": "string", "gacha_id": "string", "name": "string", "rarity": "string" }`	*No links*
400	Missing data for gacha roll	*No links*
403	Insufficient funds for gacha roll	*No links*
500	Failed to add gacha item to user inventory	*No links*

GET `/gacha/inventory/{user_id}` Retrieve a user's gacha inventory

Parameters

Try it out

Name	Description
user_id * required `string` *(path)*	user_id

Responses

Code	Description	Links
200	User inventory retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*
404	No gacha items found for user	*No links*

```
{
  "inventory": [
    {
      "gacha_id": "string",
      "name": "string",
      "rarity": "string",
      "status": "string",
      "description": "string",
      "acquired_date": "string",
      "locked": true,
      "image": "string"
    }
  ]
}
```

POST `/gacha/inventory/add` Add a gacha item to a user's inventory

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "gacha_id": "string"
}
```

Responses

Code	Description	Links
201	Gacha item successfully added to user's inventory Media type application/json Controls `Accept` header. **Example Value** Schema `{ "message": "string" }`	*No links*
400	Missing data to add gacha to inventory	*No links*
404	User or gacha item not found	*No links*

GET `/gacha/all` Retrieve all gacha items

Parameters

Try it out

Name	Description
offset `integer` *(query)*	The number of items to skip offset

Responses

Code	Description	Links
200	Gacha items retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*
404	No gacha items found	*No links*

```
{
  "message": [
    {
      "gacha_id": "string",
      "name": "string",
      "rarity": "string",
      "status": "string",
      "description": "string",
      "image": "string"
    }
  ]
}
```

PUT `/gacha/update` Update a gacha item

Parameters Try it out

No parameters

Request body required multipart/form-data

gacha_id
`string`

name
`string`

rarity
`string`

status
`string`

description

string

image

string($binary)

Responses

Code	Description	Links

200	Gacha item updated successfully
	Media type
	application/json
	Controls Accept header.
	Example Value Schema
	{
	"message": "string"
	}
		No links

400	Missing data to update gacha item	*No links*
404	Gacha item not found	*No links*
500	Failed to update gacha item	*No links*

GET /gacha/get/{gacha_id} Retrieve information about a specific gacha item

Parameters

Try it out

Name	Description
gacha_id * required string *(path)*	gacha_id

Responses

Code	Description	Links
200	Gacha item retrieved successfully Media type **application/json** Controls `Accept` header. **Example Value** Schema	*No links*
404	Gacha item not found	*No links*

```
{
  "gacha_id": "string",
  "name": "string",
  "rarity": "string",
  "status": "string",
  "description": "string",
  "image": "string"
}
```

GET `/gacha/get/{user_id}/{gacha_id}`

Retrieve details of a specific gacha item for a specific user

Parameters Try it out

Name	Description
user_id * required `string` *(path)*	user_id
gacha_id * required `string` *(path)*	gacha_id

Responses

Code	Description	Links
200	Gacha item retrieved successfully Media type **application/json** Controls `Accept` header.	*No links*

Code	Description	Links

Example Value Schema

```
{
  "gacha_id": "string",
  "name": "string",
  "rarity": "string",
  "status": "string",
  "description": "string",
  "acquired_date": "string",
  "locked": true,
  "image": "string"
}
```

404	User not found or gacha item not found in user inventory	*No links*

GET `/gacha/is_gacha_unlocked/{user_id}/{gacha_id}`

Check if a gacha item is unlocked for a user

Parameters

Try it out

Name	Description
user_id * required `string` *(path)*	user_id
gacha_id * required `string` *(path)*	gacha_id

Responses

Code	Description	Links
200	Gacha item is unlocked Media type **application/json** Controls `Accept` header.	*No links*

Example Value Schema

```
{
  "message": "string"
}
```

Code	Description	Links
400	Missing data to check gacha item	*No links*
403	Gacha item is locked	*No links*

PUT `/gacha/update_gacha_status` Update the status of a gacha item for a user

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "user_id": "string",
  "gacha_id": "string",
  "status": "string"
}
```

Responses

Code	Description	Links
200	Gacha status updated successfully Media type application/json Controls `Accept` header. **Example Value** Schema `{ "message": "string" }`	*No links*
400	Missing data to update gacha status	*No links*
408	Gacha item not found	*No links*
409	User not found	*No links*

Code	Description	Links
500	Failed to update gacha status	*No links*

PUT `/gacha/update_gacha_owner` Update the owner of a gacha item

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "buyer_id": "string",
  "seller_id": "string",
  "gacha_id": "string",
  "status": "string"
}
```

Responses

Code	Description	Links
200	Gacha owner updated successfully Media type application/json Controls `Accept` header. **Example Value** Schema `{ "message": "string" }`	*No links*
400	Missing data to update gacha owner	*No links*
404	Buyer not found or gacha item not found	*No links*
500	Failed to update gacha owner	*No links*

DELETE `/gacha/delete/{gacha_id}` Delete a gacha item from the system (admin only)

Parameters Try it out

Name	Description
gacha_id * required `string` *(path)*	gacha_id

Responses

Code	Description	Links
200	Gacha item successfully deleted	*No links*
404	Gacha not found	*No links*
500	Failed to delete gacha item	*No links*

Auctions

GET `/auctions/my` Retrieve all auctions for the authenticated user

Retrieves auctions associated with the authenticated user. Only non-admin users can access this endpoint.

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Successfully retrieved user's auctions.	*No links*
403	Admins cannot access this endpoint.	*No links*
500	Internal server error.	*No links*

GET `/auctions/get_auction` Retrieve auction details

Retrieve details for a specific auction using either auction_id or user_id.

Parameters

Try it out

Name	Description
auction_id `string` *`(query)`*	The ID of the auction. auction_id
user_id `string` *`(query)`*	The ID of the user. user_id

Responses

Code	Description	Links
200	Successfully retrieved auction details.	*No links*
400	Missing or conflicting parameters.	*No links*
404	Auction not found.	*No links*
500	Database or internal error.	*No links*

GET `/auctions/highest_bid` Retrieve the highest bid for a specific auction

Retrieve the highest bid amount for a specific auction using the gacha_id parameter.

Parameters Try it out

Name	Description
gacha_id * required `string` *(query)*	The Gacha ID of the auction. gacha_id

Responses

Code	Description	Links
200	Successfully retrieved the highest bid.	*No links*
400	Missing gacha_id parameter.	*No links*
404	Auction not found.	*No links*
500	Database or internal error.	*No links*

PUT `/auctions/update` Update an auction

Update an existing auction using the provided data. The auction must belong to the user.

Parameters Try it out

No parameters

Request body required application/json

Example Value Schema

```
{
  "auction_id": "string",
  "base_price": 0,
```

```
  "end_time": "2024-11-22T17:18:42.724Z"
}
```

Responses

Code	Description	Links
200	Successfully updated the auction.	*No links*
400	Missing or invalid parameters.	*No links*
403	Unauthorized access to the auction.	*No links*
404	Auction not found.	*No links*
500	Database or internal error.	*No links*

DELETE `/auctions/delete` Delete an auction

Delete an existing auction using the gacha_id parameter.

Parameters

Try it out

Name	Description
gacha_id * required `string` *(query)*	The Gacha ID of the auction to delete. gacha_id

Responses

Code	Description	Links
200	Successfully deleted the auction.	*No links*
400	Missing or invalid gacha_id parameter.	*No links*

Code	Description	Links
404	Auction not found.	*No links*
500	Database or internal error.	*No links*