

openapi: 3.0.0

info:

title: User Service API

version: 1.0.0

description: >

This API manages user authentication, player profile and game interactions,
as well as administrative tasks.

It includes microservices:

- AUTH: Handles user registration, login, password reset, and account management.
- USER: Manages player-specific operations, such as inventory, bids, currency, and profile information.
- ADMIN: Provides administrative functionality for managing users, updating their inventory, and modifying currency balances.
- TRANSACTION: Handles in-game currency transactions, including top-ups, payments, and transaction history
- AUCTIONS: Manages the creation, management, and bidding on auctions.
- GACHA: Handles gacha items, inventory management, and gacha rolls.

servers:

- url: http://localhost:5000

description: Local development server

tags:

- name: AUTH

description: Authentication and account management

- name: USER

description: Player-specific operations

- name: ADMIN

description: Administrative functions for managing users and system resources

- name: TRANSACTION

description: Handles in-game currency transactions for players and administrators

- name: AUCTIONS

description: Handles the creation and management of auctions and bids.

- name: GACHA

description: Handles gacha items and inventory management.

paths:

/auth/register:

post:

tags:

- AUTH

summary: Register a new user

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: new_user123

password:

type: string

example: securepassword

email:

type: string

example: new_user@example.com

responses:

'201':

description: User successfully registered

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: User registered successfully

'400':

description: Invalid request (e.g., missing fields or malformed data)

/auth/login:

post:

tags:

- AUTH

summary: Login a user and retrieve a JWT token

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: user123

password:

type: string

example: securepassword

responses:

'200':

description: Successful login and returns a JWT token

content:

application/json:

schema:

type: object

properties:

token:

type: string

example: jwt_token

'401':

description: Invalid credentials

/auth/logout:

post:

tags:

- AUTH

summary: Logout the user

responses:

'200':

description: Successful logout

'401':

description: Unauthorized access (invalid or missing token)

/auth/reset_pwd:

post:

tags:

- AUTH

summary: Request a password reset for the user

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

email:

type: string

example: user123@example.com

responses:

'200':

description: Password reset link sent to the user's email

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Password reset email sent

'404':

description: User with the specified email not found

/auth/delete_account:

delete:

tags:

- AUTH

summary: Delete the user's account permanently

responses:

'200':

description: User account deleted successfully

'401':

description: Unauthorized access (invalid or missing token)

'404':

description: User not found or account already deleted

/user/player_inventory:

get:

tags:

- USER

summary: Retrieve the player's current inventory of gacha items

responses:

'200':

description: Returns the list of gacha items the player owns

content:

application/json:

schema:

type: object

properties:

inventory:

type: array

items:

type: object

properties:

item_id:

type: string

example: gacha_001

name:

type: string

example: Dragon Sword

rarity:

type: string

example: Epic

example:

- item_id: gacha_001

name: Dragon Sword

rarity: Epic

- item_id: gacha_002

name: Mystic Shield

rarity: Rare

'401':

description: Unauthorized access (invalid or missing token)

/user/player_bids:

get:

tags:

- USER

summary: Get the list of active bids the player has placed on auctions

responses:

'200':

description: Returns a list of player's active bids

content:

application/json:

schema:

type: object

properties:

bids:

type: array

items:

type: object

properties:

auction_id:

type: string

example: auction_123

bid_amount:

type: integer

example: 500

bid_time:

type: string

format: date-time

example: '2023-12-01T15:45:00Z'

example:

- auction_id: auction_123

bid_amount: 500

bid_time: '2023-12-01T15:45:00Z'

'401':

description: Unauthorized access (invalid or missing token)

/user/player_currency:

get:

tags:

- USER

summary: Retrieve the player's current in-game currency balance

responses:

'200':

description: Returns the player's currency balance

content:

application/json:

schema:

type: object

properties:

currency_balance:

type: integer

example: 2500

'401':

description: Unauthorized access (invalid or missing token)

/user/player_profile:

get:

tags:

- USER

summary: Retrieve the player's profile information

responses:

'200':

description: Returns the player's profile information

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: user123

email:

type: string

example: user123@example.com

currency_balance:

type: integer

example: 150

gacha_collection:

type: array

items:

type: string

example:

- gacha_001

- gacha_002

'401':

description: Unauthorized access (invalid or missing token)

/admin/update_user/{username}:

put:

tags:

- ADMIN

summary: Update user profile information (Admin only)

parameters:

- name: username

in: path

required: true

schema:

type: string

description: Username of the user to update

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

email:

type: string

example: updated_user@example.com

password:

type: string

example: newpassword

responses:

'200':

description: User profile updated successfully

'401':

description: Unauthorized access (invalid or missing token)

'403':

description: Access denied (requires admin role)

'404':

description: User not found

/admin/update_inventory/{username}:

put:

tags:

- ADMIN

summary: Update a user's gacha inventory (Admin only)

parameters:

- name: username

in: path

required: true

schema:

type: string

description: Username of the user to update inventory for

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

inventory:

type: array

items:

type: string

example:

- gacha_001

- gacha_002

responses:

'200':

description: User inventory updated successfully

'401':

description: Unauthorized access (invalid or missing token)

'403':

description: Access denied (requires admin role)

'404':

description: User not found

/admin/update_currency/{username}:

put:

tags:

- ADMIN

summary: Update a user's currency balance (Admin only)

parameters:

- name: username

in: path

required: true

schema:

type: string

description: Username of the user whose currency balance will be updated

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

currency_balance:

type: integer

example: 5000

responses:

'200':

description: User currency balance updated successfully

'401':

description: Unauthorized access (invalid or missing token)

'403':

description: Access denied (requires admin role)

'404':

description: User not found

/transaction/top_up:

post:

tags:

- TRANSACTION

summary: Top up the user's in-game currency balance

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: user123

amount:

type: integer

example: 1000

responses:

'200':

description: Successful top-up

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Account successfully topped up

new_balance:

type: integer

example: 2500

'400':

description: Invalid request (e.g., negative amount)

'401':

description: Unauthorized (token missing or invalid)

/transaction/user/{username}:

get:

tags:

- TRANSACTION

summary: >-

Retrieve the transaction history for a user (players can only view their own history, admins can view all)

parameters:

- name: username

in: path

required: true

schema:

type: string

description: Username of the user whose transactions you want to retrieve

responses:

'200':

description: Returns the transaction history for the specified user

content:

application/json:

schema:

type: object

properties:

transactions:

type: array

items:

type: object

properties:

transaction_id:

type: string

example: trans_001

type:

type: string

example: roll_purchase

amount:

type: integer

example: 200

date:

type: string

format: date-time

example: '2023-10-05T14:48:00Z'

example:

- transaction_id: trans_001

type: roll_purchase

amount: 200

date: '2023-10-05T14:48:00Z'

- transaction_id: trans_002

type: auction_bid

amount: 500

date: '2023-10-05T15:12:00Z'

'401':

description: Unauthorized (token missing or invalid)

'403':

description: Access forbidden (players can only view their own transactions)

'404':

description: User not found

/transaction/make_payment:

post:

tags:

- TRANSACTION

summary: Make a payment for a gacha roll purchase or an auction bid

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: user123

transaction_type:

type: string

enum:

- roll_purchase

- auction_bid

example: roll_purchase

amount:

type: integer

example: 300

responses:

'200':

description: Payment successful, balance updated

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Payment successful

new_balance:

type: integer

example: 1200

'400':

description: Invalid request (e.g., insufficient funds)

'401':

description: Unauthorized (token missing or invalid)

'403':

description: Forbidden (user does not have enough funds)

/transaction/auction_lock:

post:

tags:

- TRANSACTION

summary: >-

Locked service to ensure a user can't spend more than their balance
during auctions

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: user123

bid_amount:

type: integer

example: 500

responses:

'200':

description: Bid accepted, user has enough funds

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Bid accepted

'401':

description: Unauthorized (token missing or invalid)

'403':

description: Insufficient funds for the bid

/transaction/auction_lock/{username}/{auction_id}:

delete:

tags:

- TRANSACTION

summary: Remove a lock from a user currency

parameters:

- name: username

in: path

required: true

schema:

type: string

description: Username of the user to unlock

- name: auction_id

in: path

required: true

schema:

type: string

description: Id of the auction

responses:

'200':

description: Gacha successfully deleted from the user's inventory

'401':

description: Unauthorized (token missing or invalid)

'403':

description: Access forbidden (admin-only endpoint)

'404':

description: User or gacha not found

/auctions/all:

get:

tags:

- AUCTIONS

summary: Retrieve all active or expired auctions

parameters:

- name: status

in: query

required: false

schema:

type: string

enum:

- active

- expired

description: Filter auctions based on status (active or expired)

responses:

'200':

description: Returns a list of all auctions

content:

application/json:

schema:

type: object

properties:

auctions:

type: array

items:

type: object

properties:

auction_id:

type: string

example: auction_001

gacha_id:

type: string

example: gacha_001

base_price:

type: integer

example: 200

highest_bid:

type: integer

example: 500

status:

type: string

example: active

end_time:

type: string

format: date-time

example: '2023-10-05T14:48:00Z'

example:

- auction_id: auction_001

gacha_id: gacha_001

base_price: 200

highest_bid: 500

status: active

end_time: '2023-10-05T14:48:00Z'

'400':

description: Invalid request

/auctions/add:

post:

tags:

- AUCTIONS

summary: Add a new auction

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

gacha_id:

type: string

example: gacha_001

base_price:

type: integer

example: 200

end_time:

type: string

format: date-time

example: '2023-12-01T15:45:00Z'

responses:

'201':

description: Auction successfully created

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Auction created successfully

auction_id:

type: string

example: auction_001

'400':

description: Invalid request

'401':

description: Unauthorized (token missing or invalid)

/auctions/end:

post:

tags:

- AUCTIONS

summary: End an auction by auction ID

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

auction_id:

type: string

example: auction_001

responses:

'200':

description: Auction successfully ended

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Auction ended successfully

'401':

description: Unauthorized (token missing or invalid)

'404':

description: Auction not found

/auctions/user/{username}:

get:

tags:

- AUCTIONS

summary: Retrieve all auctions created by a given user

parameters:

- name: username

in: path

required: true

schema:

type: string

description: Username of the user whose auctions you want to retrieve

responses:

'200':

description: Returns a list of all auctions created by the specified user

content:

application/json:

schema:

type: object

properties:

auctions:

type: array

items:

type: object

properties:

auction_id:

type: string

example: auction_001

gacha_id:

type: string

example: gacha_001

base_price:

type: integer

example: 200

status:

type: string

example: active

end_time:

type: string

format: date-time

example: '2023-10-05T14:48:00Z'

example:

- auction_id: auction_001

gacha_id: gacha_001

base_price: 200

status: active

end_time: '2023-10-05T14:48:00Z'

'404':

description: User not found

/auctions/{auction_id}:

get:

tags:

- AUCTIONS

summary: Retrieve an auction by auction ID

parameters:

- name: auction_id

in: path

required: true

schema:

type: string

description: ID of the auction to retrieve

responses:

'200':

description: Returns details of the specified auction

content:

application/json:

schema:

type: object

properties:

auction_id:

type: string

example: auction_001

gacha_id:

type: string

example: gacha_001

base_price:

type: integer

example: 200

highest_bid:

type: integer

example: 500

status:

type: string

example: active

end_time:

type: string

format: date-time

example: '2023-10-05T14:48:00Z'

'404':

description: Auction not found

/auctions/bid:

post:

tags:

- AUCTIONS

summary: Make a bid on an auction by auction ID

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

bid_amount:

type: integer

example: 500

username:

type: string

example: user123

auction_id:

type: string

example: auction_001

responses:

'200':

description: Bid successfully placed

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Bid successfully placed

new_highest_bid:

type: integer

example: 500

'400':

description: Invalid bid amount (e.g., lower than current highest bid)

'401':

description: Unauthorized (token missing or invalid)

'404':

description: Auction not found

put:

tags:

- AUCTIONS

summary: Update an auction by auction ID

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

base_price:

type: integer

example: 250

end_time:

type: string

format: date-time

example: '2023-12-01T18:00:00Z'

responses:

'200':

description: Auction successfully updated

content:

application/json:

schema:

type: object

properties:

message:

type: string

example: Auction updated successfully

'401':

description: Unauthorized (token missing or invalid)

'404':

description: Auction not found

/auctions/gacha/{gacha_id}:

get:

tags:

- AUCTIONS

summary: Retrieve all auctions for a given gacha ID

parameters:

- name: gacha_id

in: path

required: true

schema:

type: string

description: ID of the gacha whose auctions you want to retrieve

responses:

'200':

description: Returns a list of all auctions for the specified gacha

content:

application/json:

schema:

type: object

properties:

auctions:

type: array

items:

type: object

properties:

auction_id:

type: string

example: auction_001

base_price:

type: integer

example: 200

status:

type: string

example: active

end_time:

type: string

format: date-time

example: '2023-10-05T14:48:00Z'

example:

- auction_id: auction_001

base_price: 200

status: active

end_time: '2023-10-05T14:48:00Z'

'404':

description: Gacha not found or no auctions available for the given gacha

/auctions/bids/{auction_id}:

get:

tags:

- AUCTIONS

summary: Retrieve all bids for a given auction by auction ID

parameters:

- name: auction_id

in: path

required: true

schema:

type: string

description: ID of the auction whose bids you want to retrieve

responses:

'200':

description: Returns a list of all bids for the specified auction

content:

application/json:

schema:

type: object

properties:

bids:

type: array

items:

type: object

properties:

bid_id:

type: string

example: bid_001

username:

type: string

example: user123

bid_amount:

type: integer

example: 500

bid_time:

type: string

format: date-time

example: '2023-10-05T14:50:00Z'

example:

- bid_id: bid_001

username: user123

bid_amount: 500

bid_time: '2023-10-05T14:50:00Z'

'404':

description: Auction or bids not found

/gacha/{gacha_id}:

get:

tags:

- GACHA

summary: Retrieve gacha details by gacha ID

parameters:

- name: gacha_id

in: path

required: true

schema:

type: string

description: The ID of the gacha to retrieve

responses:

'200':

description: Returns details of the specified gacha

content:

application/json:

schema:

type: object

properties:

gacha_id:

type: string

example: gacha_001

name:

type: string

example: Jeep '98

rarity:

type: string

example: Epic

description:

type: string

example: A rare jeep with old style model.

'404':

description: Gacha not found

/gacha/roll:

post:

tags:

- GACHA

summary: Roll a gacha for a random item

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

username:

type: string

example: user123

roll_cost:

type: integer

example: 100

responses:

'200':

description: Successfully rolled a gacha item

content:

application/json:

schema:

type: object

properties:

gacha_id:

type: string

example: gacha_001

name:

type: string

example: Dragon Sword

rarity:

type: string

example: Epic

message:

type: string

example: Gacha roll successful

'400':

description: Insufficient funds for the roll or other request error

'401':

description: Unauthorized (token missing or invalid)

/gacha/all:

get:

tags:

- GACHA

summary: Retrieve all gacha items with optional filters

parameters:

- name: rarity

in: query

required: false

schema:

type: string

description: Filter gacha by rarity (e.g., common, rare, epic)

- name: status

in: query

required: false

schema:

type: string

description: Filter gacha by status (e.g., available, out of stock)

responses:

'200':

description: Returns a list of all gacha items matching the filters

content:

application/json:

schema:

type: object

properties:

gachas:

type: array

items:

type: object

properties:

gacha_id:

type: string

example: gacha_001

name:

type: string

example: Dragon Sword

rarity:

type: string

example: Epic

status:

type: string

example: available

example:

- gacha_id: gacha_001

name: Dragon Sword

rarity: Epic

status: available

- gacha_id: gacha_002

name: Mystic Shield

rarity: Rare

status: out of stock

/gacha/inventory/{user_id}:

get:

tags:

- GACHA

summary: Retrieve the gacha inventory of a given user

parameters:

- name: user_id

in: path

required: true

schema:

type: string

description: The ID of the user whose gacha inventory is being retrieved

responses:

'200':

description: Returns the gacha inventory for the specified user

content:

application/json:

schema:

type: object

properties:

inventory:

type: array

items:

type: object

properties:

gacha_id:

type: string

example: gacha_001

name:

type: string

example: Dragon Sword

rarity:

type: string

example: Epic

example:

- gacha_id: gacha_001

name: Dragon Sword

rarity: Epic

- gacha_id: gacha_002

name: Mystic Shield

rarity: Rare

'404':

description: User or inventory not found

/gacha/inventory:

post:

tags:

- GACHA

summary: Add a gacha to a user's inventory

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

gacha_id:

type: string

example: gacha_001

username:

type: string

example: user123

responses:

'201':

description: Gacha successfully added to user's inventory

'401':

description: Unauthorized (token missing or invalid)

'404':

description: User or gacha not found

/gacha/inventory/{username}/{gacha_id}:

delete:

tags:

- GACHA

summary: Delete a gacha from a user's inventory (admin only)

parameters:

- name: username

in: path

required: true

schema:

type: string

description: The username of the user whose gacha has to be removed

- name: gacha_id

in: path

required: true

schema:

type: string

description: The gacha_id that has to be removed

responses:

'200':

description: Gacha successfully deleted from the user's inventory

'401':

description: Unauthorized (token missing or invalid)

'403':

description: Access forbidden (admin-only endpoint)

'404':

description: User or gacha not found

/gacha/admin/add:

post:

tags:

- GACHA

summary: Add a new gacha item to the system (admin only)

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

gacha_id:

type: string

example: gacha_001

name:

type: string

example: Jeep '98

rarity:

type: string

example: Epic

description:

type: string

example: A rare jeep with old style model.

responses:

'201':

description: New gacha successfully added

'400':

description: Invalid input or duplicate gacha ID

'401':

description: Unauthorized (token missing or invalid)

/gacha/admin/update/{gacha_id}:

put:

tags:

- GACHA

summary: Update an existing gacha item (admin only)

parameters:

- name: gacha_id

in: path

required: true

schema:

type: string

description: The ID of the gacha to update

requestBody:

required: true

content:

application/json:

schema:

type: object

properties:

name:

type: string

example: Jeep '98

rarity:

type: string

example: Epic

description:

type: string

example: A rare jeep with old style model.

status:

type: string

example: available

responses:

'200':

description: Gacha item successfully updated

'401':

description: Unauthorized (token missing or invalid)

'404':

description: Gacha not found