

Economix API Specification

Authentication Endpoints

POST /api/register

Register a new user

Request Body:

```
{
  "username": "new_user",
  "password": "securepassword123"
}
```

Success Response (201):

```
{
  "success": true
}
```

Error Responses:

409 Conflict - Username exists

400 Bad Request - Invalid username format

429 Too Many Requests - Account creation spam

POST /api/login

Authenticate user

Request Body:

```
{
  "username": "existing_user",
  "password": "securepassword123",
  "code": "2fa-backup-code" // Optional
}
```

Success Response:

```
{
  "success": true,
  "token": "bearer-token"
}
```

Error Responses:

401 Unauthorized - Invalid credentials

429 Too Many Requests - Failed login attempts

2FA Endpoints

POST /api/setup_2fa

Enable 2FA (Requires authentication)

Success Response:

```
{
  "success": true,
  "provisioning_uri": "otpauth://...",
  "backup_code": "ABCD-1234"
}
```

GET /api/2fa_qrcode

Get 2FA QR code (Requires authentication)

Returns: PNG image

POST /api/verify_2fa

Verify 2FA setup (Requires authentication)

Request Body:

```
{
  "code": "123456"
}
```

Success Response:

```
{
  "success": true
}
```

POST /api/disable_2fa

Disable 2FA (Requires authentication)

Success Response:

```
{
  "success": true
}
```

User Management

GET /api/account

Get current user profile (Requires authentication)

Success Response:

```
{
  "username": "testuser",
  "tokens": 150.5,
  "items": [...],
  "pets": [...],
  "level": 5,
  "exp": 1200,
  "2fa_enabled": false,
  "banned_until": null,
  "muted": false
}
```

POST /api/delete_account

Delete user account (Requires authentication)

Success Response:

```
{
  "success": true
}
```

Items System

POST /api/create_item

Create new item (Requires authentication, 1m cooldown)

Success Response:

```
{
  "id": "item-uuid",
  "name": {
    "adjective": "Shiny",
    "material": "Gold",
    "noun": "Sword",
    "suffix": "of Destruction",
    "number": "1234",
    "icon": "✕"
  },
  "rarity": 0.5,
  "level": "Godlike"
}
```

POST /api/recycle_item

Recycle item for tokens

Request Body:

```
{
  "item_id": "target-item-uuid"
}
```

Success Response:

```
{
  "success": true
}
```

Pets System

POST /api/buy_pet

Purchase a pet (Requires authentication)

Success Response:

```
{
  "id": "pet-uuid",
  "name": "Fluffy",
  "level": 1,
  "health": "healthy"
}
```

POST /api/feed_pet

Feed a pet (Costs 10 tokens)

Request Body:

```
{
  "pet_id": "target-pet-uuid"
}
```

Success Response:

```
{
  "success": true
}
```

Market System

GET /api/market

List items for sale

```
Success Response:
[
  {
    "id": "item-uuid",
    "price": 499.99,
    "owner": "seller123",
    "name": {...}
  }
]
```

POST /api/sell_item

List/Unlist item on market

```
Request Body:
{
  "item_id": "target-item-uuid",
  "price": 100.0
}
```

```
Success Response:
{
  "success": true
}
```

POST /api/buy_item

Purchase item from market

```
Request Body:
{
  "item_id": "target-item-uuid"
}
```

```
Success Response:
{
  "success": true
}
```

POST /api/take_item

Claim item using secret code

```
Request Body:
{
  "item_secret": "item-secret-code"
}
```

```
Success Response:
{
  "success": true
}
```

Messaging System

POST /api/send_message

Send chat message

Request Body:

```
{
  "room": "global",
  "message": "Hello world!"
}
```

Success Response:

```
{
  "success": true
}
```

GET /api/get_messages

Retrieve chat history

Query Parameters:

?room=global

Success Response:

```
{
  "messages": [
    {
      "username": "testuser",
      "message": "Hello world!",
      "timestamp": 1689012345,
      "type": "user"
    }
  ]
}
```

POST /api/send_dm

Send direct message

Request Body:

```
{
  "recipient": "target_user",
  "message": "Private message"
}
```

Success Response:

```
{
  "success": true
}
```

GET /api/get_dms

Get DM conversation

Query Parameters:

?recipient=target_user

Success Response:

```
{
  "messages": [
    {
      "sender": "testuser",
      "message": "Hi there!",
      "timestamp": 1689012345
    }
  ]
}
```

Leaderboard & Stats

GET /api/leaderboard

Get top 10 players

```
Success Response:
{
  "leaderboard": [
    {
      "username": "top_player",
      "tokens": 15000,
      "place": "1st"
    }
  ]
}
```

GET /api/stats

Get system statistics

```
Success Response:
{
  "stats": {
    "total_tokens": 150000,
    "total_accounts": 1000,
    "total_items": 50000
  }
}
```

Admin/Moderation Endpoints

POST /api/reset_cooldowns

Reset user cooldowns (Admin)

```
Request Body:
{
  "username": "target_user"
}
```

```
Success Response:
{
  "success": true
}
```

POST /api/edit_tokens

Modify user tokens (Admin)

```
Request Body:
{
  "username": "target_user",
  "tokens": 1000
}
```

```
Success Response:
{
  "success": true
}
```

POST /api/ban_user

Ban user (Admin)

```
Request Body:
{
  "username": "rulebreaker",
  "length": "7d",
  "reason": "Cheating"
}

Success Response:
{
  "success": true
}
```

POST /api/delete_message

Delete message (Mod+)

```
Request Body:
{
  "message_id": "message-uuid"
}

Success Response:
{
  "success": true
}
```

Creator Codes

POST /api/redeem_creator_code

Redeem creator code

```
Request Body:
{
  "code": "CREATOR123"
}

Success Response:
{
  "success": true,
  "tokens": 500,
  "pets": 1
}
```

Miscellaneous

GET /api/get_banner

Get current banner

```
Success Response:
{
  "banner": "https://example.com/banner.png"
}
```

GET /

Root endpoint

```
Redirects to official website
```

Error Responses

Common error structure:

```
{
  "error": "Descriptive message",
  "code": "error-code",
  "remaining": 123  // Optional cooldown timer
}
```

Common Error Codes:

- `invalid-credentials` : Invalid username/password
- `cooldown-active` : Action on cooldown
- `not-enough-tokens` : Insufficient balance
- `user-muted` : Account muted
- `content-spam` : Blocked by automod
- `2fa-required` : 2FA authentication needed
- `ip-blocked` : IP address temporarily blocked