

GiftList REST API

Complete Endpoint Documentation

A comprehensive guide to the GiftList backend API

Built with PHP + MySQL + JWT Authentication

Overview

- **Base URL:** `http://localhost:8000`
- **Authentication:** Bearer Token (JWT)
- **Content-Type:** `application/json`
- **Total Endpoints:** 16

Endpoint Categories

1. **Health** - GET /health - Server status check
2. **Authentication** - User registration and login
3. **Recipients** - Manage gift recipients
4. **Gifts** - Manage gifts for recipients



Authentication Endpoints

POST /auth/register

Description: Register a new user account

Authentication: ✗ Not Required

Request Body:

```
{  
  "name": "John Doe",  
  "email": "john@example.com",  
  "password": "password123"  
}
```

POST /auth/register (continued)

Response (201 Created):

```
{  
  "success": true,  
  "data": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com"  
  }  
}
```

cURL Example:

```
curl -X POST http://localhost:8000/auth/register \  
  -H "Content-Type: application/json" \  
  -d '{"name": "John Doe", "email": "john@example.com", "password": "password123"}'
```

POST /auth/login

Description: Login and receive JWT token

Authentication: ✗ Not Required

Request Body:

```
{  
  "email": "john@example.com",  
  "password": "password123"  
}
```

POST /auth/login (continued)

Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "token": "eyJ0eXAiOiJKV1QiLCJhbGc...",  
    "user": {  
      "id": 1,  
      "name": "John Doe",  
      "email": "john@example.com"  
    }  
  }  
}
```

POST /auth/login - Example

PowerShell:

```
Invoke-RestMethod -Uri "http://localhost:8000/auth/login" `  
-Method POST -ContentType "application/json" `  
-Body (@{email="john@example.com";password="pass123"}|ConvertTo-Json)
```

GET /auth/me

Description: Get current authenticated user information

Authentication:  Required

Headers:

```
Authorization: Bearer YOUR_JWT_TOKEN
```

Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "id": 1,  
    "name": "John Doe",  
    "email": "john@example.com",  
    "created_at": "2025-11-07 10:30:00"  
  }  
}
```



Recipients Endpoints

GET /recipients

Description: Get all recipients for the authenticated user

Authentication:  Required

Response (200 OK):

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "user_id": 1,  
      "name": "Mom",  
      "relationship": "Mother",  
      "notes": "Birthday in June",  
      "created_at": "2025-11-07 10:35:00"  
    }  
  ]  
}
```

GET /recipients/:id

Description: Get a specific recipient by ID

Authentication:  Required

URL Parameters:

- `id` (integer) - Recipient ID

Example: GET /recipients/1

Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "id": 1,  
    "user_id": 1,  
    "name": "Mom",  
    "relationship": "Mother",  
    "status": "Active"  
  }  
}
```

POST /recipients

Description: Create a new recipient

Authentication:  Required

Request Body:

```
{  
  "name": "Dad",  
  "relationship": "Father",  
  "notes": "Loves technology"  
}
```

Required Fields: name

Optional Fields: relationship , notes

POST /recipients (continued)

Response (201 Created):

```
{  
  "success": true,  
  "data": {  
    "id": 2,  
    "user_id": 1,  
    "name": "Dad",  
    "relationship": "Father",  
    "notes": "Loves technology",  
    "created_at": "2025-11-07 10:40:00"  
  }  
}
```

POST /recipients - Example

PowerShell:

```
Invoke-RestMethod -Uri "http://localhost:8000/recipients" `  
-Method POST -Headers @{Authorization="Bearer $token"} `  
-ContentType "application/json" `  
-Body (@{name="Dad";relationship="Father"}|ConvertTo-Json)
```

PUT /recipients/:id

Description: Update an existing recipient

Authentication:  Required

URL Parameters: `id` (integer)

Request Body (all fields optional):

```
{  
  "name": "Mom",  
  "relationship": "Mother",  
  "notes": "Birthday in June - loves gardening"  
}
```

Response (200 OK): Returns updated recipient object

DELETE /recipients/:id

Description: Delete a recipient and all associated gifts

Authentication:  Required

URL Parameters: `id` (integer)

Example: `DELETE /recipients/1`

Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "message": "Recipient deleted"  
  }  
}
```



Gifts Endpoints

GET /gifts

Description: Get all gifts for the authenticated user

Authentication: Required

Optional Query Parameters:

- recipientId (integer) - Filter gifts by recipient

Examples:

- GET /gifts - All gifts
- GET /gifts?recipientId=1 - Gifts for recipient 1

GET /gifts (continued)

Response (200 OK):

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "recipient_id": 1,  
      "name": "Garden Tools Set",  
      "description": "Premium garden tools",  
      "price": 89.99,  
      "url": "https://example.com/product",  
      "purchased": 0,  
      "created_at": "2025-11-07 10:45:00"  
    }  
  ]  
}
```

GET /gifts/:id

Description: Get a specific gift by ID

Authentication:  Required

URL Parameters: `id` (integer)

Example: `GET /gifts/1`

Response (200 OK): Returns single gift object

GET /recipients/:recipientId/gifts

Description: Get all gifts for a specific recipient

Authentication:  Required

URL Parameters: `recipientId` (integer)

Example: `GET /recipients/1/gifts`

Response (200 OK): Returns array of gift objects

POST /gifts

Description: Create a new gift

Authentication:  Required

Request Body:

```
{  
  "recipient_id": 1,  
  "name": "Wireless Headphones",  
  "description": "Noise cancelling",  
  "price": 199.99,  
  "url": "https://example.com/headphones",  
  "purchased": 0  
}
```

Required Fields: recipient_id , name

Optional Fields: description , price , url , purchased (0 or 1)

POST /gifts (continued)

Response (201 Created):

```
{  
  "success": true,  
  "data": {  
    "id": 2,  
    "recipient_id": 1,  
    "name": "Wireless Headphones",  
    "description": "Noise cancelling",  
    "price": 199.99,  
    "url": "https://example.com/headphones",  
    "purchased": 0,  
    "created_at": "2025-11-07 10:50:00"  
  }  
}
```

PUT /gifts/:id

Description: Update an existing gift

Authentication:  Required

URL Parameters: `id` (integer)

Request Body (all fields optional):

```
{  
  "name": "Wireless Headphones Pro",  
  "description": "Noise cancelling with case",  
  "price": 249.99,  
  "purchased": 1  
}
```

Response (200 OK): Returns updated gift object

DELETE /gifts/:id

Description: Delete a gift

Authentication:  Required

URL Parameters: `id` (integer)

Example: `DELETE /gifts/1`

Response (200 OK):

```
{  
  "success": true,  
  "data": {  
    "message": "Gift deleted"  
  }  
}
```

