



OiJodHRwczovL3dmZXIwaGlhcXdwbmtrcWd5dmNpLnN1cGFiYXNlMmNvL2F1dGgydjEiLCJzdWliOil0MzgzMGQxMC0xYzcxLTQ2ZjctYWVhNy01NzNhZmMwYjhmYTUilCJhdWQiOiJhdXRoaW50aWNhdGVkIiwiaXhwLjoxNzQ2OTU5Mzk4L0JpYXQiOjE3NDY5NTU3OTgslmVtYWsljoidGVzdCt1c2VyMUBnbWFpbC5jb20iLCJwaG9uZSI6IiIsImFwcF9tZXRhZGF0YSI6eyJwcm92aWRlcil6ImVtYWslIiwicHJvdmlkZXJzljpbImVtYWslIi19LCJ1c2VyX21ldGFkYXRhIjp7ImVtYWsljoidGVzdCt1c2VyMUBnbWFpbC5jb20iLCJlbWFpbF92ZXJpZmlZCI6dHJ1ZSwicGhvbmVfdmVyaWZpZWQiOmZhbHNILCJzdWliOil0MzgzMGQxMC0xYzcxLTQ2ZjctYWVhNy01NzNhZmMwYjhmYTUifSwicm9sZSI6ImF1dGhIbnRpY2F0ZWQiLCJhYWwiOiJhYWwxiIiwiaW1yIjpbeyJtZXRob2QiOiJwYXNzd29yZCIslInRpbWVzdGFtcC16MTc0Njk1NTc5OH1dLCJzZXNzaW9uX2IkljoiMDIyMjNmNGUtYjYyY00NWU5LWJhNWUtMWMMyNWZkZDM3NzJiliwiaXNfYW5vbmltb3VzljpmYWxzZX0.nvLlbS8xAvkgD\_VbQx86FChPwf8Suib\_xN4zhCBnjzc",

```
"token_type": "bearer",
"expires_in": 3600,
"expires_at": 1746959398,
"refresh_token": "jNz4FQVNkq2g9MvhFfo5rg",
"user": {
  "id": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",
  "aud": "authenticated",
  "role": "authenticated",
  "email": "test+user1@gmail.com",
  "email_confirmed_at": "2025-05-11T09:29:58.085353677Z",
  "phone": "",
  "last_sign_in_at": "2025-05-11T09:29:58.09131949Z",
  "app_metadata": {
    "provider": "email",
    "providers": [
      "email"
    ]
  },
  "user_metadata": {
    "email": "test+user1@gmail.com",
    "email_verified": true,
    "phone_verified": false,
    "sub": "43830d10-1c91-46f7-aaa7-573afc0b8fa5"
  },
  "identities": [
    {
      "identity_id": "c58c2cb7-11c1-4fa2-8b68-25ed529f0c02",
      "id": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",
      "user_id": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",
      "identity_data": {
        "email": "test+user1@gmail.com",
        "email_verified": true,
        "phone_verified": false,
        "sub": "43830d10-1c91-46f7-aaa7-573afc0b8fa5"
      },
      "provider": "email",
      "last_sign_in_at": "2025-05-11T09:29:58.072334449Z",
```



```
"expires_at": 1746959727,
"refresh_token": "yapXFumPpan5-oKIAry3cQ",
"user": {
  "id": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",
  "aud": "authenticated",
  "role": "authenticated",
  "email": "test+user1@gmail.com",
  "email_confirmed_at": "2025-05-11T09:29:58.085353Z",
  "phone": "",
  "confirmed_at": "2025-05-11T09:29:58.085353Z",
  "last_sign_in_at": "2025-05-11T09:35:27.882939733Z",
  "app_metadata": {
    "provider": "email",
    "providers": [
      "email"
    ]
  },
  "user_metadata": {
    "email": "test+user1@gmail.com",
    "email_verified": true,
    "phone_verified": false,
    "sub": "43830d10-1c91-46f7-aaa7-573afc0b8fa5"
  },
  "identities": [
    {
      "identity_id": "c58c2cb7-11c1-4fa2-8b68-25ed529f0c02",
      "id": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",
      "user_id": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",
      "identity_data": {
        "email": "test+user1@gmail.com",
        "email_verified": false,
        "phone_verified": false,
        "sub": "43830d10-1c91-46f7-aaa7-573afc0b8fa5"
      },
      "provider": "email",
      "last_sign_in_at": "2025-05-11T09:29:58.072334Z",
      "created_at": "2025-05-11T09:29:58.072406Z",
      "updated_at": "2025-05-11T09:29:58.072406Z",
      "email": "test+user1@gmail.com"
    }
  ],
  "created_at": "2025-05-11T09:29:58.04943Z",
  "updated_at": "2025-05-11T09:35:27.891735Z",
  "is_anonymous": false
}
```

## 1. Create a Transaction (POST)

Creates a new transaction in the database.

- **Method:** POST
- **URL:** /rest/v1/transactions

**HEADERS:** Add +

**Authorization:** Bearer {{access\_token}} \*

**Prefer:** return=representation

**Body:**

```
{  
  "from_user": "{{user_id}}",  
  "to_user": "recipient@test.com",  
  "amount": 10.5,  
  "status": "failed"  
}
```

**Response:**

- **Status Code:** 201 (Created)

**Body:**

```
[  
  {  
    "id": "be565b15-8973-4738-b9d7-21e3b6028605",  
    "from_user": "43830d10-1c91-46f7-aaa7-573afc0b8fa5",  
    "to_user": "recipient@test.com",  
    "amount": 10.5,  
    "status": "failed",  
    "created_at": "2025-05-11T09:48:10.992661+00:00"  
  }  
]
```

## 2. Get a Transaction by ID (GET)

Retrieves a specific transaction by its ID.

- **Method:** GET
- **URL:** /rest/v1/transactions?id=eq.a9bc2a22-d590-400a-8876-c5ec5a105322

**HEADERS:** Add +

**Authorization:** Bearer {{access\_token}} \*

**Prefer:** return=representation

- **Response:**
  - **Status Code:** 200 (OK)

**Body:**

```
[
  {
    "id": "a9bc2a22-d590-400a-8876-c5ec5a105322",
    "from_user": "b4512a52-382d-446f-a6ed-103de32b0b2b",
    "to_user": "recipient@example.com",
    "amount": 100.5,
    "status": "pending",
    "created_at": "2025-04-28T13:07:01.449576+00:00"
  }
]
```

### 3. Update a Transaction (PATCH)

Updates the status of a specific transaction by its ID.

- **Method:** PATCH
- **URL:** /rest/v1/transactions?id=eq.{{transaction\_id}}

**HEADERS:** Add +

**Authorization:** Bearer {{access\_token}} \*

**Prefer:** return=representation

**Body:**

```
{
  "status": "completed"
}
```

- **Response:**
  - **Status Code:** 200 (OK)

**Body:**

```
[
  {
    "id": "a9bc2a22-d590-400a-8876-c5ec5a105322",
    "from_user": "b4512a52-382d-446f-a6ed-103de32b0b2b",
    "to_user": "recipient@example.com",
    "amount": 100.5,
    "status": "completed",
  }
]
```

```
"created_at": "2025-04-28T13:07:01.449576+00:00"
}
]
```

#### 4. Delete a Transaction (DELETE)

Deletes a specific transaction by its ID.

- **Method:** DELETE
- **URL:** /rest/v1/transactions?id=eq.{{transaction\_id}}
- **Response:**
  - **Status Code:** 200 (OK)

**HEADERS:** Add +

**Authorization:** Bearer {{access\_token}} \*

**Body:**

```
[
  {
    "id": "a9bc2a22-d590-400a-8876-c5ec5a105322",
    "from_user": "b4512a52-382d-446f-a6ed-103de32b0b2b",
    "to_user": "recipient@example.com",
    "amount": 100.5,
    "status": "pending",
    "created_at": "2025-04-28T13:07:01.449576+00:00"
  }
]
```

#### Notes

- The id field in the URL query parameter (id=eq.<transaction\_id>) is used to filter transactions for GET, PATCH, and DELETE requests.
- The from\_user field in the response is automatically populated by the backend (e.g., with the authenticated user's ID).
- The created\_at field is automatically generated by the server upon transaction creation.
- Ensure proper authentication is set up, as the API requires a logged-in user.