Split Payment App API Documentation

Backend Development Team

January 17, 2026

Contents

1 Introduction

This document outlines the REST API endpoints for the Split Payment application. The API handles User Authentication, Group Management, Expense Tracking, and Settlement Logic.

Base URL

`http://127.0.0.1:8000/api/`

Authentication

All endpoints (except Login/Register) require Token Authentication.

- **Header Key:** `Authorization`
- **Header Value:** `Token <your_token_here>`

2 User Authentication

2.1 Register User

Creates a new user account and returns an authentication token.

- **URL:** `/users/register/`
- **Method:** `POST`
- **Auth Required:** No

Request Body:

```
{
    "username": "rudra",
    "email": "rudra@example.com",
    "password": "securepassword123",
    "name": "Rudra Pratap"
}
```

Response (201 Created):

```
{
    "token": "9944b09199c62bcf9418ad846dd0e4bbdfc6ee4b",
    "user_id": 1,
    "username": "rudra",
    "email": "rudra@example.com"
}
```

2.2 Login

Authenticates existing users.

- **URL:** `/users/login/`
- **Method:** `POST`
- **Auth Required:** No

Request Body:

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

Response (200 OK):

```
{
    "token": "9944b09199c62bcf9418ad846dd0e4bbdfc6ee4b",
    "user_id": 1,
    "name": "Rudra Pratap"
}
```

3 Group Management

3.1 Create Group

Creates a new group. The creator is automatically added as a member.

- **URL:** /expenses/groups/
- **Method:** POST

Request Body:

```
{
    "name": "Goa Trip 2025"
}
```

Response (201 Created):

```
{
    "id": 1,
    "name": "Goa Trip 2025",
    "created_by": 1,
    "created_at": "2025-01-16T10:00:00Z"
}
```

3.2 Join Group

Adds the authenticated user to an existing group.

- **URL:** /expenses/groups/<group_id>/join/
- **Method:** POST

Response (200 OK):

```
{
    "message": "Joined group successfully"
}
```

3.3 Get Group Members

Fetches all members of a specific group (useful for populating ”Split With” dropdowns).

- **URL:** /expenses/groups/<group_id>/members/
- **Method:** GET

Response (200 OK):

```json
[
    {
        "id": 1,
        "username": "rudra",
        "name": "Rudra Pratap",
        "email": "rudra@example.com"
    },
    {
        "id": 2,
        "username": "alice",
        "name": "Alice Wonderland",
        "email": "alice@example.com"
    }
]
```

4 Expenses

4.1 Add Expense

Creates a new transaction. **Note:** The sum of ‘shares‘ must exactly match the ‘amount‘.

- **URL:** `/expenses/groups/<group_id>/expenses/`
- **Method:** `POST`

Request Body:

```
{
    "description": "Lunch at Cafe",
    "amount": 100.00,
    "shares": [
        {
            "user_id": 1,
            "amount": 50.00
        },
        {
            "user_id": 2,
            "amount": 50.00
        }
    ]
}
```

Response (201 Created):

```
{
    "id": 101,
    "description": "Lunch at Cafe",
    "amount": "100.00",
    "date": "2025-01-16T12:30:00Z",
    "category": "EXPENSE",
    "paid_by": 1,
    "paid_by_name": "rudra",
    "splits": [
        { "user_id": 1, "user_name": "rudra", "amount": "50.00" },
        { "user_id": 2, "user_name": "alice", "amount": "50.00" }
    ]
}
```

4.2 List Expenses (History)

Fetches transactions for a group. Results are **paginated** and ordered by date (newest first).

- **URL:** `/expenses/groups/<group_id>/expenses/?page=1`
- **Method:** `GET`

Response (200 OK):

```
{
    "count": 52,
    "next": "http://.../expenses/?page=2",
    "previous": null,
    "results": [
        {
            "id": 101,
            "description": "Lunch at Cafe",
            "amount": "100.00",
            "paid_by_name": "rudra",
            "splits": [...]
        },
        ...
    ]
}
```

5 Settlement & Balances

5.1 Get Balances (Detail View)

Calculates how much the current user owes (or is owed) by every other member in the group.

- **URL:** `/expenses/groups/<group_id>/balances/`
- **Method:** `GET`

Response (200 OK):

```
[
    {
        "user": "alice",
        "user_id": 2,
        "amount": -50.00,
        "status": "You owe"
    },
    {
        "user": "bob",
        "user_id": 3,
        "amount": 25.00,
        "status": "You are owed"
    }
]
```

5.2 Settle Up (Pay User)

Records a payment from the current user to another user to settle debts.

- **URL:** `/expenses/groups/<group_id>/settle/`
- **Method:** `POST`

Request Body:

```
{
    "to_user_id": 2,
    "amount": 50.00
}
```

Response (201 Created):

```
{
    "message": "Paid 50.00 to alice"
}
```