

## Documentação da API - Carteira Financeira

### Base URL

http://localhost:8000/api

---

### Autenticação

A API utiliza **token-based authentication com Laravel Sanctum**. Após o login, você deve incluir o token em todas as requisições protegidas.

#### Headers obrigatórios nas rotas autenticadas:

Authorization: Bearer {SEU\_TOKEN}

Accept: application/json

---

## ROTAS

### 1. Login

#### Endpoint:

POST /api/login

#### Body (JSON):

```
{  
  "email": "usuario@email.com",  
  "password": "sua_senha"  
}
```

#### Resposta (200):

```
{  
  "token": "1|a1b2c3d4..."  
}
```

#### Erros (401):

```
{  
  "message": "Credenciais inválidas"  
}
```

---

## ♦ 2. Depósito em Carteira

### Endpoint:

POST /api/wallet/deposit

### Headers:

Authorization: Bearer {token}

Accept: application/json

### Body (JSON):

```
{  
  "amount": 100.00  
}
```

### Regras:

- Apenas o dono da carteira pode depositar (policy de acesso).
- Não é permitido depositar em carteira com saldo negativo.

### Resposta (201):

```
{  
  "message": "Depósito realizado com sucesso.",  
  "transaction": {  
    "id": 1,  
    "type": "deposit",  
    "amount": 100.00,  
    ...  
  }  
}
```

### Erros (422):

```
{  
  "errors": {  
    "balance": ["Não se pode depositar em carteira com saldo negativo."]  
  }  
}
```

```
}
```

```
}
```

#### **Erros (403 - acesso negado):**

```
{
```

```
  "message": "Esta ação não é autorizada."
```

```
}
```

---

### **♦ 3. Transferência entre carteiras**

#### **Endpoint:**

POST /api/wallet/transfer

#### **Body (JSON):**

```
{
```

```
  "receiver_id": 2,
```

```
  "amount": 50.00
```

```
}
```

#### **Regras:**

- Apenas o dono da carteira pode transferir.
- O destinatário deve existir.
- A carteira deve possuir saldo suficiente.

#### **Resposta (201):**

```
{
```

```
  "message": "Transferência feita com sucesso.",
```

```
  "transaction": {
```

```
    "id": 5,
```

```
    "type": "transfer",
```

```
    "amount": 50.00,
```

```
    ...
```

```
}
```

```
}
```

**Erros (422):**

```
{
```

```
  "errors": {
```

```
    "balance": ["Saldo insuficiente para a transferência."]
```

```
  }
```

```
}
```

**Erros (403):**

```
{
```

```
  "message": "Esta ação não é autorizada."
```

```
}
```

---

**◆ 4. Reversão de Transação****Endpoint:**

POST /api/wallet/reverse

**Body (JSON):**

```
{
```

```
  "transaction_id": 10
```

```
}
```

**Regras:**

- Somente o dono da carteira pode reverter.
- Só é possível reverter uma vez.
- Só podem ser revertidas transações do próprio usuário.

**Resposta (200):**

```
{
```

```
  "message": "Transação revertida com sucesso.",
```

```
  "transaction": {
```

```
    "id": 10,
```

```
"type": "reversal",  
"amount": 50.00,  
...  
}
```

```
}
```

#### **Erros (422):**

```
{  
  "errors": {  
    "transaction": ["Transação já foi revertida ou inválida."]  
  }  
}
```

#### **Erros (403):**

```
{  
  "message": "Esta ação não é autorizada."  
}
```

---

### **Autorização (Policies)**

- **Deposit:** apenas o dono da carteira pode depositar.
- **Transfer:** apenas o dono da carteira pode transferir.
- **Reverse:** apenas o dono da transação original pode reverter.

---

### **Segurança**

- auth:sanctum protege todas as rotas principais.
- throttle:60,1 limita a 60 requisições por minuto por usuário.
- Policies verificam o proprietário da carteira.
- Validações são feitas com Form Requests.
- Respostas padronizadas para erros e exceções.
- Logging implementado em todos os services (transfer, reverse, erros, etc)