

# Handover Document: Wallet Recharge Process

---

This document outlines the step-by-step process for recharging a merchant's wallet in the absence of portal access. The wallet recharge process requires server access and involves fetching the wallet details via API and updating the wallet directly in the database.

---

## Step 1: Retrieve Wallet Information

To retrieve the wallet information of a merchant, you'll need to access the wallet's passbook using an API call.

API Endpoint:

```
https://api.lightspeedpay.in/api/v1/wallet/get-passbook/66323c14eeb28a6027a0a421?limit=100&page=1&status=
```

Parameters:

- **limit**: Can be updated to increase or decrease the number of records returned.
- **page**: Specifies the page number of results.
- **status**: Filters the wallets by status (e.g., active).

Authorization:

The request requires a header with the **x-authorization** token, which can be retrieved by:

1. Logging into the portal.
2. Using the browser's Developer Tools (Inspect Element) to view network requests.
3. Finding the **x-authorization** token in the request headers after refreshing the page.

API Request (Using Postman):

1. **Method**: GET
2. **Endpoint**: Paste the above URL into Postman.
3. **Headers**: Add **x-authorization** token in the headers section.

Sample Response:

```
{
  "status": "success",
  "result": [
    {
      "_id": "66697ac70f3f56edd7d2ae15",
      "merchantId": {
        "_id": "66697ac70f3f56edd7d2ae13",
```

```

        "name": "Vikas",
        "mobile": 9729753351,
        "email": "solutionsothala@gmail.com",
        "company": "OTHALA SOLUTIONS PRIVATE LIMITED"
    },
    "amount": 44420.88850000232,
    "status": "active",
    "alertAt": 0,
    "__v": 0
},
{
    "_id": "66465af54653c90fde1d6da1",
    "merchantId": {
        "_id": "66465af54653c90fde1d6d9f",
        "name": "Vikas",
        "mobile": 1234567890,
        "email": "cust171@lightspeedpay.in",
        "company": "GERTH TECHNOLOGY PRIVATE LIMITED"
    },
    "amount": 29472.60980000292,
    "status": "active",
    "alertAt": 0,
    "__v": 0
},
{
    "_id": "66487c9a39c355989e9d9bf5",
    "merchantId": {
        "_id": "6638736e2d7c247138dec391",
        "name": "Mitra Fintech",
        "mobile": 7667489732,
        "email": "lightspeedpay@mitrafintech.com",
        "company": "LightSpeedPay"
    },
    "amount": 1016.8999999999999,
    "status": "active",
    "alertAt": 0,
    "__v": 0
}
]
}

```

### Important Fields:

- **\_id**: This is the wallet ID you will use to update the wallet balance.
- **merchantId**: Contains merchant information such as name, email, company, etc.
- **amount**: The current wallet balance.

## Step 2: Access the Server

To update the wallet balance, you will need to log in to the server.

1. **Open Terminal.**
2. **Log in via SSH:**

```
ssh -o StrictHostKeyChecking=no root@46.28.44.62
```

---

## Step 3: Open MongoDB Shell

Once logged into the server, open the MongoDB shell to update the wallet amount:

```
mongosh
```

---

## Step 4: Update the Wallet Balance

Use the `updateMany` command to adjust the wallet balance for the specific merchant. You will need to pass the wallet's `_id` and the amount to update.

Command Format:

```
db.wallets.updateMany(  
  { _id: { $in: [ObjectId('WALLET_ID')] } },  
  { $inc: { amount: AMOUNT } }  
)
```

Example:

To update the wallet for the merchant with wallet `_id 66465af54653c90fde1d6da1` and add an amount of ₹29,500:

```
db.wallets.updateMany(  
  { _id: { $in: [ObjectId('66465af54653c90fde1d6da1')] } },  
  { $inc: { amount: 29500 } }  
)
```

- **Note:** Replace the wallet ID and amount accordingly.

To Deduct Amount:

If you need to deduct the wallet balance, use a negative amount:

```
db.wallets.updateMany(  
  { _id: { $in: [ObjectId('WALLET_ID')] } },
```

```
{ $inc: { amount: -AMOUNT } }  
)
```

### Example Deduction:

```
db.wallets.updateMany(  
  { _id: { $in: [ObjectId('66465af54653c90fde1d6da1')] } },  
  { $inc: { amount: -20000 } }  
)
```

---

## Summary of Steps

1. **Retrieve wallet details via API** using the wallet ID.
2. **Log in to the server** using SSH.
3. **Access MongoDB** using `mongosh`.
4. **Update wallet amount** by using the appropriate `updateMany` command.

Be sure to use the correct wallet `_id` and adjust the amount (positive for recharge, negative for deduction).