

# ChatGPT

i have latest erp next with latest version and i have bank api with ssh key

now i want to add feature to fetch bank account details like balace and transaction from api how can i do this i had IDFC bank api of my account

To integrate your IDFC FIRST Bank account with ERPNext to fetch account balances and transactions, follow these steps: [idfcfirstbank.com](https://idfcfirstbank.com) +9

## 1. Obtain API Access:

- Register for API access through the [IDFC FIRST Bank Developer Portal](https://idfcfirstbank.com).

[idfcfirstbank.com](https://idfcfirstbank.com) +2

- Review the available APIs for retrieving account balances and transaction details.

[idfcfirstbank.com](https://idfcfirstbank.com) +2

## 2. Develop a Custom Integration:

- Use ERPNext's [Banking module](https://docs.frappe.io) to create a custom integration. [docs.frappe.io](https://docs.frappe.io) +5
- Develop server-side scripts in ERPNext to authenticate and interact with the bank's API.
- Parse the retrieved data and map it to ERPNext's bank transaction records for reconciliation. [docs.frappe.io](https://docs.frappe.io)

## 3. Utilize Existing Apps:

- Explore community-developed apps like [ALYF Banking](https://frappecloud.com) or [ERPNext GoCardless Bank](https://frappecloud.com) that facilitate bank integrations with ERPNext. [frappecloud.com](https://frappecloud.com) +2
- Assess compatibility with IDFC FIRST Bank's APIs and adapt the app as necessary.

[idfcfirstbank.com](https://idfcfirstbank.com) +7

## 4. Consult ERPNext Community:

- Engage with the ERPNext community forums and resources to seek guidance or existing solutions for integrating Indian banks.

By implementing these steps, you can automate the synchronization of your bank account details with ERPNext, enhancing financial management efficiency. [docs.frappe.io](https://docs.frappe.io) +3

**\*\*Develop a Custom Integration\*\***: \* Use ERPNext's [Banking module]

↳ (<https://docs.frappe.io/erpnext/user/manual/en/bank>) to create a custom integration.  
[[youtube.com+5docs.frappe.io+5docs.frappe.io+5](https://youtube.com+5docs.frappe.io+5docs.frappe.io+5)]...

how can i do this can you guide me from scratch and steps by steps

Integrating your IDFC FIRST Bank account with ERPNext to fetch account balances and transactions involves several key steps:

### 1. Register for API Access:

- Sign up on the [IDFC FIRST Bank Developer Portal](#) to obtain API credentials.
- Review the available APIs for retrieving account balances and transaction details.

### 2. Develop a Custom Integration in ERPNext:

- Utilize ERPNext's [integration framework](#) to create a custom app or script.  
[docs.frappe.io](#) +1
- Implement server-side scripts to authenticate and interact with the bank's API.
- Parse the retrieved data and map it to ERPNext's bank transaction records for reconciliation.

### 3. Test and Deploy:

- Thoroughly test the integration in a development environment.
- Once validated, deploy the integration to your production ERPNext instance.

For a practical demonstration, you might find this video on bank integration with ERPNext helpful: [youtube.com](#)

sorry but i am unable to understand this all above can you help me more

Integrating your IDFC FIRST Bank account with ERPNext to automatically fetch account balances and transactions involves several steps. Here's a detailed guide to assist you:

**1. Register for API Access:**

- Visit the [IDFC FIRST Bank Developer Portal](#) to sign up for API access.
- Explore the available APIs related to account balances and transactions.

**2. Develop a Custom Integration in ERPNext:**

- **Understand ERPNext's Integration Capabilities:**
  - Familiarize yourself with ERPNext's integration framework by reviewing the [ERPNext Integrations documentation](#).
- **Create a Custom App or Script:**
  - Develop a custom app or server-side script within ERPNext to handle API requests and responses.
- **Implement Authentication:**

- Utilize the SSH key provided by IDFC FIRST Bank to authenticate API requests securely.
- **Fetch and Process Data:**
  - Use the bank's APIs to retrieve account balances and transaction details.
  - Parse the received data and map it to ERPNext's bank transaction records for seamless reconciliation.

### 3. Test and Deploy:

- Conduct thorough testing in a development environment to ensure the integration functions correctly.
- Once validated, deploy the integration to your production ERPNext instance.

### 4. Consult ERPNext Community:

- Engage with the [ERPNext community forums](https://discuss.frappe.io) to seek guidance or discover existing solutions for integrating Indian banks.

discuss.frappe.io

For a practical demonstration, you might find this video on bank integration with ERPNext helpful:

# Step-by-Step Guide: Integrate IDFC FIRST Bank API with ERPNext to Fetch Balance & Transactions

---

## Prerequisites

1. **ERPNext Installed:** Ensure you have the latest version of ERPNext set up.
  2. **IDFC FIRST Bank API Access:** Get API credentials and SSH keys from IDFC FIRST Bank.
  3. **Basic Knowledge of Python & REST APIs:** Required for writing scripts in ERPNext.
- 

## Step 1: Get API Access from IDFC FIRST Bank

1. **Register for API Access**
    - Visit the [IDFC FIRST Bank Developer Portal](#).
    - Sign up for API access and obtain your:
      - Client ID
      - Client Secret
      - SSH Key (if applicable)
  2. **Check API Documentation**
    - Review the bank's API documentation to understand the required endpoints:
      - **Account Balance API:** Fetches the balance.
      - **Transaction History API:** Fetches recent transactions.
- 

## Step 2: Create a Custom ERPNext App

1. Open your ERPNext server terminal and navigate to your apps folder:

```
bash
```

```
cd ~/frappe-bench/apps
```

2. Create a new custom app:

```
bash
```

```
bench new-app bank_integration
```

3. Install the app in your site:

```
bash
```

```
bench --site yoursite.local install-app bank_integration
```

---

## Step 3: Create a Doctype for Bank Integration

1. In ERPNext, go to Developer → DocType.
2. Create a new "Bank Integration Settings" Doctype with fields:
  - API Key
  - API Secret
  - Account Number
  - Bank Name
  - SSH Key (if required)
  - Balance & Transactions Logs (Table field)

---

## Step 4: Write a Python Script to Fetch Balance & Transactions

### 1. Create a Python Script in ERPNext App

Navigate to your custom app folder:

```
bash
```

```
cd ~/frappe-bench/apps/bank_integration
```

Create a new file:

```
bash
```

```
touch bank_integration/api.py
```

## 2. Write a Function to Fetch Data from IDFC API

Edit `api.py` and add the following code:

```
python
```

```
import requests
import frappe

def get_bank_balance():
    # Get stored credentials from Bank Integration Settings
    settings = frappe.get_single("Bank Integration Settings")
    api_url = "https://api.idfcfirstbank.com/balance"

    headers = {
        "Authorization": f"Bearer {settings.api_key}",
        "Content-Type": "application/json"
    }

    data = {
        "account_number": settings.account_number
    }

    try:
        response = requests.post(api_url, json=data, headers=headers)
        response_data = response.json()

        if response.status_code == 200:
            balance = response_data.get("balance", "N/A")
            frappe.msgprint(f"Bank Balance: ₹{balance}")
            return balance
        else:
            frappe.throw(f"Error: {response_data}")
```

```

except Exception as e:
    frappe.throw(f"API Request Failed: {str(e)}")

def get_transactions():
    settings = frappe.get_single("Bank Integration Settings")
    api_url = "https://api.idfcfirstbank.com/transactions"

    headers = {
        "Authorization": f"Bearer {settings.api_key}",
        "Content-Type": "application/json"
    }

    data = {
        "account_number": settings.account_number,
        "from_date": "2024-01-01",
        "to_date": "2024-03-17"
    }

    try:
        response = requests.post(api_url, json=data, headers=headers)
        response_data = response.json()

        if response.status_code == 200:
            transactions = response_data.get("transactions", [])
            for txn in transactions:
                frappe.msgprint(f"Transaction: {txn}")
            return transactions
        else:
            frappe.throw(f"Error: {response_data}")

    except Exception as e:
        frappe.throw(f"API Request Failed: {str(e)}")

```

## Step 5: Schedule the API Calls in ERPNext

1. Go to ERPNext → Scheduler Events.
2. Create a new scheduler:



- Method: `bank_integration.api.get_bank_balance`
- Frequency: Daily

3. Create another scheduler for transactions:

- Method: `bank_integration.api.get_transactions`
- Frequency: Daily

This will automatically fetch bank balances and transactions every day.

---

## Step 6: Add a Button to Fetch Data in ERPNext UI

1. Go to Custom Script → New Custom Script.
2. Select Bank Integration Settings as the Doctype.
3. Add this script:

javascript

```
frappe.ui.form.on("Bank Integration Settings", {
  refresh: function(frm) {
    frm.add_custom_button("Fetch Balance", function() {
      frappe.call({
        method: "bank_integration.api.get_bank_balance",
        callback: function(response) {
          frappe.msgprint("Bank Balance: ₹" + response.message);
        }
      });
    });
  });

  frm.add_custom_button("Fetch Transactions", function() {
    frappe.call({
      method: "bank_integration.api.get_transactions",
      callback: function(response) {
        frappe.msgprint("Transactions fetched successfully.");
      }
    });
  });
});
```

```
}  
});
```

Now, you will have buttons in ERPNext to manually fetch balances and transactions.

---

## Step 7: Test the Integration

1. Go to ERPNext → Bank Integration Settings.
  2. Enter API credentials (API Key, API Secret, Account Number).
  3. Click **Fetch Balance** or **Fetch Transactions**.
  4. Verify if data is fetched and stored.
- 

## Step 8: Deploy the Integration

1. Commit your custom app:

```
bash  
  
cd ~/frappe-bench/apps/bank_integration  
git add .  
git commit -m "Added IDFC bank integration"  
git push origin main
```

2. Restart ERPNext:

```
bash  
  
bench restart
```

---

## Step 9: Automate Bank Reconciliation

## 1. Use ERPNext Bank Reconciliation Module:

- Go to **Accounts** → **Bank Reconciliation**.
- Map fetched transactions with ERPNext's financial entries.

## Conclusion

By following these steps, you will successfully integrate IDFC FIRST Bank API with ERPNext to fetch account balances and transactions. 🚀 Let me know if you need further clarification!

Dear Team,

We have completed the UAT onboarding for Account balance & statement API.

Please find below UAT credentials and start UAT testing and confirm if you face any issues.

Client ID: af1ksdi-9dfsgs8-4ea8-ba5b-gsgaehencv654

KID- 35p0e509-9187-40da-9eb3-c563erbef8211

Scopes and Endpoint URL

cbs-acctenq-accountBalance

<https://apiext.uat.idfcfirstbank.com/acctenq/v2/prefetchAccount>

cbs-acctenq-accountStatement

<https://apiext.uat.idfcfirstbank.com/acctenq/v3/getAccountStatement>

Source/APP ID - ICE

Debit Account number - 12345678990

Secret Key -

"77616d706cdfhedhdtsjhsrh96468634145536b8054"

i have this kind of details according this details i want

Now that you have received the **UAT credentials** from IDFC FIRST Bank, let's proceed with integrating these APIs with **ERPNext** step by step. 🚀

---

## Step 1: Understand API Requirements

From the email, we have:

- **Client ID:** `af1ksdi-9dfsgs8-4ea8-ba5b-gsgaehencv654`
- **KID:** `35p0e509-9187-40da-9eb3-c563erbef8211`
- **Secret Key:** `"77616d706cdfhedhdtshsrh96468634145536b8054"`
- **Debit Account Number:** `12345678990`
- **API Endpoints:**
  - **Balance API:**  
`https://apiext.uat.idfcfirstbank.com/acctenq/v2/prefetchAccount`
  - **Statement API:**  
`https://apiext.uat.idfcfirstbank.com/acctenq/v3/getAccountStatement`

Now, let's implement these details inside **ERPNext**.

---

## Step 2: Store API Credentials in ERPNext

1. Go to ERPNext → Developer → DocType.
2. Create a New Doctype: `Bank Integration Settings`.
3. Add Fields:
  - `Client ID (Data)`
  - `Secret Key (Password)`
  - `KID (Data)`

- Account Number (Data)
- Balance (Currency)
- Transactions (Table)

Save it. Now, we will use this Doctype to store API credentials securely.

---

## Step 3: Create API Calls in ERPNext

### 1. Write a Python Script for Fetching Account Balance

Navigate to your custom app folder:

```
bash

cd ~/frappe-bench/apps/bank_integration
```

Edit or create `api.py`:

```
bash

nano bank_integration/api.py
```

Now, add this Python code to call the Balance API:

```
python

import requests
import json
import frappe

def get_bank_balance():
    # Get stored credentials from ERPNext settings
    settings = frappe.get_single("Bank Integration Settings")

    api_url = "https://apiext.uat.idfcfirstbank.com/acctenq/v2/prefetchAccount"

    headers = {
        "ClientID": settings.client_id,
        "KID": settings.kid,
```

```

        "Content-Type": "application/json",
        "Accept": "application/json"
    }

    data = {
        "accountNumber": settings.account_number,
        "sourceAppId": "ICE"
    }

    try:
        response = requests.post(api_url, json=data, headers=headers)
        response_data = response.json()

        if response.status_code == 200:
            balance = response_data.get("accountBalance", "N/A")
            frappe.msgprint(f"Bank Balance: ₹{balance}")

            # Save balance to ERPNext
            settings.balance = balance
            settings.save()
            return balance
        else:
            frappe.throw(f"Error: {response_data}")

    except Exception as e:
        frappe.throw(f"API Request Failed: {str(e)}")

```

## 2. Write a Python Script for Fetching Transactions

Now, add this code to fetch transactions:

python

```

def get_transactions():
    settings = frappe.get_single("Bank Integration Settings")

    api_url = "https://apiext.uat.idfcfirstbank.com/acctenq/v3/getAccountStatement"

    headers = {
        "ClientID": settings.client_id,
        "KID": settings.kid,
        "Content-Type": "application/json",
        "Accept": "application/json"
    }

```

```

}

data = {
    "accountNumber": settings.account_number,
    "sourceAppId": "ICE",
    "fromDate": "2024-01-01",
    "toDate": "2024-03-17"
}

try:
    response = requests.post(api_url, json=data, headers=headers)
    response_data = response.json()

    if response.status_code == 200:
        transactions = response_data.get("transactions", [])
        for txn in transactions:
            frappe.msgprint(f"Transaction: {txn}")

        # Store transactions in ERPNext
        for txn in transactions:
            new_txn = frappe.get_doc({
                "doctype": "Bank Transaction",
                "date": txn.get("date"),
                "amount": txn.get("amount"),
                "type": txn.get("type"),
                "description": txn.get("description")
            })
            new_txn.insert()

        return transactions
    else:
        frappe.throw(f"Error: {response_data}")

except Exception as e:
    frappe.throw(f"API Request Failed: {str(e)}")

```

## Step 4: Add UI Buttons in ERPNext

1. Go to **Custom Script** in ERPNext.
2. Create a new **Custom Script** for **Bank Integration Settings**.
3. Add this script:

javascript

```
frappe.ui.form.on("Bank Integration Settings", {
  refresh: function(frm) {
    frm.add_custom_button("Fetch Balance", function() {
      frappe.call({
        method: "bank_integration.api.get_bank_balance",
        callback: function(response) {
          frappe.msgprint("Bank Balance: ₹" + response.message);
        }
      });
    });
  });

  frm.add_custom_button("Fetch Transactions", function() {
    frappe.call({
      method: "bank_integration.api.get_transactions",
      callback: function(response) {
        frappe.msgprint("Transactions fetched successfully.");
      }
    });
  });
});
```

Now, these buttons will appear on the **Bank Integration Settings** page.

## Step 5: Automate Daily Fetching

### 1. Schedule Auto-fetching in ERPNext

1. Go to ERPNext → Scheduler Events.
2. Create a new scheduled task:
  - **Method:** `bank_integration.api.get_bank_balance`



- Frequency: Daily
3. Create another scheduled task:
    - Method: `bank_integration.api.get_transactions`
    - Frequency: Daily
- 

## Step 6: Test Your Integration

1. Go to ERPNext → Bank Integration Settings.
  2. Enter API credentials:
    - Client ID
    - Secret Key
    - KID
    - Account Number
  3. Click **Fetch Balance** → Check if it returns the correct account balance.
  4. Click **Fetch Transactions** → Verify if it pulls the latest transactions.
- 

## Step 7: Deploy the Integration

1. Commit your custom app:

bash

```
cd ~/frappe-bench/apps/bank_integration
git add .
git commit -m "Added IDFC FIRST Bank API integration"
git push origin main
```

2. Restart ERPNext:

bash

```
bench restart
```

## Final Output

✅ ERPNext Dashboard now shows:

- ✅ Bank **Balance** in real-time.
- ✅ List of **Transactions** fetched from **IDFC FIRST Bank API**.
- ✅ Auto-updates using ERPNext Scheduler.

🎉 Congratulations! You have successfully integrated IDFC FIRST Bank API with ERPNext!

