Dragon Vault Bank API Documentation

Base URL: https://dragonVault.site/bank-api

Overview

The Dragon Vault Banking System provides a comprehensive API for both account holders and bank tellers. This document outlines the available endpoints, request parameters, and response formats for all operations.

Account Holder APIs

1. Register / Create Account

Creates a new bank account for a customer.

Endpoint: `/register`
 Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
first_name	String	Yes	First name of the account holder
middle_initial	String	Yes	Middle initial of the account holder (use "N/A" if none)
last_name	String	Yes	Last name of the account holder
password	String	Yes	Strong password for account security
email_address	String	Yes	Email address for notifications and communications
mobile_number	String	Yes	Mobile number for OTP and transaction notifications

Success Response:

Status Code: `200 OK`

```
{
  "account_successfully_created": true,
```

```
  "bearer_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...",
  "account_number": 1234567890
}
```

Error Response:

Status Code: `400 Bad Request`

```
{
  "error_message": "Email address already registered"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/register \
 -H "Content-Type: application/json" \
 -d '{
   "first_name": "John",
   "middle_initial": "D",
   "last_name": "Smith",
   "password": "SecureP@ssw0rd123",
   "email_address": "john.smith@example.com",
   "mobile_number": "09123456789"
 }'
```

2. Authenticate Login

Authenticates an account holder using their account number and password.

Endpoint: `/login`
 Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
account_number	Integer	Yes	The unique account number
password	String	Yes	Account password

Success Response:

Status Code: 200 OK

```
{
  "login_successful": true,
  "bearer_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...",
  "user_id": "abc123"
}
```

Error Response:

Status Code: 401 Unauthorized

```
{
  "error_message": "Invalid account number or password"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/login \
  -H "Content-Type: application/json" \
  -d '{
    "account_number": 1234567890,
    "password": "SecureP@ssw0rd123"
  }'
```

3. Display Account Balance

Retrieves the current balance for an authenticated account holder.

Endpoint: /account/balance
 Method: GET

Request Parameters:

Parameter	Type	Required	Description
authorization	String	Yes	Bearer token in header (JWT provided after login)
account_number	Integer	No	Optional if tied to authenticated user

Success Response:

Status Code: `200 OK`

```
{
  "account_number": 1234567890,
  "balance": 5000.00,
  "currency": "PHP",
  "as_of": "2025-05-02T08:30:45Z"
}
```

Error Response:

Status Code: `401 Unauthorized`

```
{
  "error_message": "Unauthorized access, please log in"
}
```

Example Request:

```
curl -X GET https://dragonVault.site/bank-api/account/balance \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ..."
```

4. Transfer Funds Internally

Transfers funds between two accounts within Dragon Vault.

Endpoint: `/fund-transfer-internal`
 Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
authorization	String	Yes	Bearer token in header (JWT provided after login)
transaction_amount	Number	Yes	Amount to be transferred
source_account_no	Integer	Yes	The account number of the sender

recipient_account_no	Integer	Yes	The account number of the recipient (within Dragon Vault)
redirect_url	String	Yes	URL to redirect after completion

Success Response:

Status Code: 200 OK

```
{
  "fund_transfer_success": true,
  "transaction_id": 2343241,
  "otp_required": true,
  "otp_session_id": "abc123xyz"
}
```

The system will then redirect to the OTP verification endpoint.

Error Response:

Status Code: 400 Bad Request

```
{
  "error_message": "Insufficient balance for transfer"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/fund-transfer-internal \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ..." \
  -H "Content-Type: application/json" \
  -d '{
    "transaction_amount": 1000.00,
    "source_account_no": 1234567890,
    "recipient_account_no": 9876543210,
    "redirect_url": "https://dragonVault.site/transfer-result"
  }'
```

5. Transfer Funds Externally

Transfers funds from a Dragon Vault account to an account at another bank.

Endpoint: `/fund-transfer-external`
Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
authorization	String	Yes	Bearer token in header (JWT provided after login)
transaction_amount	Number	Yes	Amount to be transferred
source_account_no	Integer	Yes	The account number of the sender
recipient_account_no	Integer	Yes	The account number at the external bank
recipient_bank_code	String	Yes	Bank code of the recipient's bank
redirect_url	String	Yes	URL to redirect after completion

Success Response:

Status Code: `200 OK`

```
{
  "fund_transfer_success": true,
  "transaction_id": 75464342,
  "otp_required": true,
  "otp_session_id": "ext123xyz"
}
```

The system will then redirect to the OTP verification endpoint.

Error Response:

Status Code: `400 Bad Request`

```
{
  "error_message": "Insufficient balance for external transfer"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/fund-transfer-external \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ..." \
  -H "Content-Type: application/json" \
  -d '{
    "transaction_amount": 2000.00,
    "source_account_no": 1234567890,
    "recipient_account_no": 9876543210,
    "recipient_bank_code": "OTHRBNK001",
    "redirect_url": "https://dragonVault.site/transfer-result"
  }'
```

6. OTP Validation

Validates the One-Time Password sent to the user's mobile number for transaction authorization.

Endpoint: `/validate-otp`
Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
otp_code	String	Yes	6-digit OTP code received via SMS
otp_session_id	String	Yes	Session ID provided during transfer initiation
transaction_id	Integer	Yes	ID of the transaction to be validated

Success Response:

Status Code: `200 OK`

```
{
  "validation_successful": true,
  "transaction_id": 2343241,
  "redirect_url":
"https://dragonVault.site/transfer-result?fund_transfer_success=true&transaction_id=2343241"
}
```

Error Response:

Status Code: `400 Bad Request`

```
{
  "error_message": "Invalid OTP code",
  "attempts_remaining": 2
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/validate-otp \
  -H "Content-Type: application/json" \
  -d '{
    "otp_code": "123456",
    "otp_session_id": "abc123xyz",
    "transaction_id": 2343241
  }'
```

7. Receive Funds API

Handles incoming funds from external banks.

Endpoint: `/receive-external-transfer`
 Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
transaction_amount	Number	Yes	Amount being transferred
source_account_no	Integer	Yes	Sender's account number at external bank
source_bank_code	String	Yes	Bank code of the sender's bank
recipient_account_no	Integer	Yes	Recipient's account number at Dragon Vault

Success Response:

Status Code: `200 OK`

```
{
  "fund_transfer_success": true,
  "transaction_id": 29485420343
}
```

Error Response:

Status Code: `404 Not Found`

```
{
  "error_message": "Recipient account not found"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/receive-external-transfer \
  -H "Content-Type: application/json" \
  -d '{
    "transaction_amount": 3000.00,
    "source_account_no": 9876543210,
    "source_bank_code": "EXTBNK001",
    "recipient_account_no": 1234567890
  }'
```

Bank Teller APIs

1. Teller Authentication

Authenticates a bank teller to access the system.

Endpoint: `/teller/login`
 Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
teller_id	String	Yes	Unique identifier for the teller
password	String	Yes	Teller's password

Success Response:

Status Code: `200 OK`

```
{
  "login_successful": true,
  "bearer_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...",
  "teller_name": "Jane Doe",
  "branch_code": "DV001"
}
```

Error Response:

Status Code: 401 Unauthorized

```
{
  "error_message": "Invalid teller ID or password"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/teller/login \
  -H "Content-Type: application/json" \
  -d '{
    "teller_id": "T12345",
    "password": "SecureTeller@123"
  }'
```

2. Process Deposit

Allows a teller to deposit funds into a customer's account.

Endpoint: /teller/transaction
 Method: POST

Request Parameters:

Parameter	Type	Required	Description
authorization	String	Yes	Bearer token in header (JWT provided after login)
teller_id	String	Yes	ID of the teller performing the transaction
account_no	Integer	Yes	Customer's account number
transaction_type	String	Yes	Must be "deposit"

transaction_amount	Number	Yes	Amount to deposit

Success Response:

Status Code: 200 OK

```
{
  "transaction_success": true,
  "transaction_id": 998877,
  "account_no": 1234567890,
  "previous_balance": 5000.00,
  "new_balance": 6000.00,
  "transaction_timestamp": "2025-05-02T10:15:30Z"
}
```

Error Response:

Status Code: 404 Not Found

```
{
  "error_message": "Account not found or invalid teller credentials"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/teller/transaction \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ..." \
  -H "Content-Type: application/json" \
  -d '{
    "teller_id": "T12345",
    "account_no": 1234567890,
    "transaction_type": "deposit",
    "transaction_amount": 1000.00
  }'
```

3. Process Withdrawal

Allows a teller to withdraw funds from a customer's account.

Endpoint: /teller/transaction
Method: POST

Request Parameters:

Parameter	Type	Required	Description
authorization	String	Yes	Bearer token in header (JWT provided after login)
teller_id	String	Yes	ID of the teller performing the transaction
account_no	Integer	Yes	Customer's account number
transaction_type	String	Yes	Must be "withdrawal"
transaction_amount	Number	Yes	Amount to withdraw

Success Response:

Status Code: 200 OK

```
{
  "transaction_success": true,
  "transaction_id": 998878,
  "account_no": 1234567890,
  "previous_balance": 6000.00,
  "new_balance": 5000.00,
  "transaction_timestamp": "2025-05-02T11:45:20Z"
}
```

Error Response:

Status Code: 400 Bad Request

```
{
  "error_message": "Insufficient balance for withdrawal"
}
```

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/teller/transaction \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ..." \
  -H "Content-Type: application/json" \
```

```
-d '{
  "teller_id": "T12345",
  "account_no": 1234567890,
  "transaction_type": "withdrawal",
  "transaction_amount": 1000.00
}'
```

4. Teller OTP Validation

For high-value transactions, tellers may need to validate using OTP.

Endpoint: `/teller/validate-otp`
 Method: `POST`

Request Parameters:

Parameter	Type	Required	Description
authorization	String	Yes	Bearer token in header (JWT provided after login)
teller_id	String	Yes	ID of the teller performing the validation
otp_code	String	Yes	6-digit OTP code received via SMS
transaction_id	Integer	Yes	ID of the transaction to be validated

Success Response:

Status Code: `200 OK`

```
{
  "validation_successful": true,
  "transaction_id": 998878,
  "transaction_status": "completed"
}
```

Error Response:

Status Code: `400 Bad Request`

```
{
  "error_message": "Invalid OTP code",
  "attempts_remaining": 2
```

}

Example Request:

```
curl -X POST https://dragonVault.site/bank-api/teller/validate-otp \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ..." \
  -H "Content-Type: application/json" \
  -d '{
    "teller_id": "T12345",
    "otp_code": "123456",
    "transaction_id": 998878
  }'
```

Error Codes

Code	Description
400	Bad Request - Invalid parameters or request format
401	Unauthorized - Authentication required or failed
403	Forbidden - User does not have permission for the requested action
404	Not Found - Requested resource does not exist
409	Conflict - Request conflicts with current state
422	Unprocessable Entity - Request validation failed
429	Too Many Requests - Rate limit exceeded
500	Internal Server Error - Something went wrong on the server