

Sage API Wrapper Documentation

Overview

The Sage API Wrapper is a RESTful API service that provides integration with Sage accounting software. It allows external systems to retrieve payment information and post cashbook transactions to Sage.

Base URLs

- **Development:** `http://localhost:8080`
- **Production:** `https://production-domain.com`

Authentication

The API uses API key authentication. To access the protected endpoints, you must include an API key in the request header.

API Key Header

- **Header Name:** `X-API-KEY`
- **Value:** Your API key

Example:

```
X-API-KEY: SampleAPIKey
```

Error Responses

- **401 Unauthorized:** Missing API key
- **403 Forbidden:** Invalid API key

API Endpoints

Payments API

Get Posted Payments by Date Range

Retrieves payments that were posted within a specified date range.

Endpoint: POST /payments/by_date

Request Body:

```
{
  "startDate": 20250101,
  "endDate": 20250731
}
```

Request Parameters:

- **startDate** (integer, required): The start date in YYYYMMDD format
- **endDate** (integer, required): The end date in YYYYMMDD format

Response:

```
{
  "responseCode": "200",
  "responseMessage": "success",
  "timeStamp": "2025-07-30T08:21:00Z",
  "data": [
    {
      "accountNumber": "1234567890",
      "amount": 1000,
      "currency": "USD",
      "remarks": "Payment for invoice #12345",
      "vendorId": "VEND001",
      "accountName": "Vendor Company Ltd",
      "branchCode": "001",
      "sortCode": "123456",
      "currencyCode": "USD",
      "transactionDate": "2025-07-15",
      "transactionType": "CASH",
      "transactionReference": "Payment for invoice #12345",
      "bankName": "Example Bank"
    }
  ]
}
```

Response Fields:

- `responseCode` (string): HTTP status code as a string
- `responseMessage` (string): Description of the response status
- `timeStamp` (string): ISO 8601 timestamp of when the response was generated
- `data` (array): List of payment objects with the following properties:
- `accountNumber` (string): Bank account number
- `amount` (integer): Payment amount
- `currency` (string): Currency of the payment
- `remarks` (string): Additional notes about the payment
- `vendorId` (string): ID of the vendor/payee
- `accountName` (string): Name on the bank account
- `branchCode` (string): Bank branch code
- `sortCode` (string): Bank sort code
- `currencyCode` (string): Currency code
- `transactionDate` (string): Date of the transaction in ISO 8601 format
- `transactionType` (string): Type of transaction
- `transactionReference` (string): Reference for the transaction
- `bankName` (string): Name of the bank

Status Codes:

- `200 OK`: Request successful
- `400 Bad Request`: Invalid date format

Cashbook API

Post Cashbook Transaction

Posts a new transaction to the Sage cashbook.

Endpoint: `POST /api/postCBTxn`

Request Body:

```
{
  "transactionId": "TXN12345",
  "bankCode": "BANK001",
  "Description": "Cash receipt from customer",
  "noEntries": 1,
  "creditAmount": 1000.00,
  "debitAmount": 1000.00,
  "entries": [
```

```

{
  "entryNo": 1,
  "referenceNo": "REF001",
  "customerNo": "CUST001",
  "noDetails": 2,
  "amount": 1000,
  "currency": "USD",
  "details": [
    {
      "entryDescription": "Sales revenue",
      "accountId": "4000",
      "amount": 800,
      "DrCr": "Cr",
      "detailNo": 1
    },
    {
      "entryDescription": "Sales tax",
      "accountId": "2200",
      "amount": 200,
      "DrCr": "Cr",
      "detailNo": 2
    }
  ]
}

```

Request Parameters:

- `transactionId` (string, required): Unique identifier for the transaction
- `bankCode` (string, required): Code identifying the bank
- `Description` (string, required): Description of the transaction
- `noEntries` (integer, required): Number of entries in the transaction
- `creditAmount` (number, required): Total credit amount
- `debitAmount` (number, required): Total debit amount
- `entries` (array, required): List of entry objects with the following properties:
- `entryNo` (integer, required): Entry number
- `referenceNo` (string, required): Reference number for the entry
- `customerNo` (string, required): Customer number
- `noDetails` (integer, required): Number of details in the entry
- `amount` (integer, required): Amount of the entry
- `currency` (string, required): Currency of the entry
- `details` (array, required): List of entry detail objects with the following properties:
- `entryDescription` (string, required): Description of the entry detail
- `accountId` (string, required): Account identifier

- `amount` (integer, required): Amount of the entry detail
- `DrCr` (string, required): Debit or Credit indicator ("Dr" or "Cr")
- `detailNo` (integer, required): Detail number for this entry

Response:

```
{
  "responseCode": 200,
  "responseMessage": "saved successfully"
}
```

Response Fields:

- `responseCode` (integer): HTTP status code
- `responseMessage` (string): Description of the response status

Status Codes:

- `200 OK`: Transaction saved successfully
- `409 Conflict`: Transaction already processed
- `500 Internal Server Error`: Error processing transaction

Error Handling

Common Error Responses

Invalid Date Format

```
{
  "responseCode": "400",
  "responseMessage": "Invalid date format: [error details]",
  "timestamp": "2025-07-30T08:21:00Z"
}
```

Transaction Already Processed

```
{
  "responseCode": 409,
  "responseMessage": "Already Reported"
}
```

```
}
```

Server Error

```
{  
  "responseCode": 500,  
  "responseMessage": "Error processing transaction: [error details]"  
}
```

Data Models

paymentsRequest

```
{  
  "startDate": 20250101,  
  "endDate": 20250731  
}
```

responseMain

```
{  
  "responseCode": "200",  
  "responseMessage": "success",  
  "timeStamp": "2025-07-30T08:21:00Z",  
  "data": []  
}
```

paymentsResponse

```
{  
  "accountNumber": "1234567890",  
  "amount": 1000,  
  "currency": "USD",  
  "remarks": "Payment for invoice #12345",  
  "vendorId": "VEND001",  
  "accountName": "Vendor Company Ltd",  
  "branchCode": "001",  
  "sortCode": "123456",  
  "currencyCode": "USD",  
  "transactionDate": "2025-07-15",  
}
```

```
"transactionType": "CHK",  
"transactionReference": "Payment for invoice #12345",  
"bankName": "Example Bank"  
}
```

receiptRequest

```
{  
  "transactionId": "TXN12345",  
  "bankCode": "BANK001",  
  "Description": "Cash receipt from customer",  
  "noEntries": 1,  
  "creditAmount": 1000.00,  
  "debitAmount": 1000.00,  
  "entries": []  
}
```

entry

```
{  
  "entryNo": 1,  
  "referenceNo": "REF001",  
  "customerNo": "CUST001",  
  "noDetails": 2,  
  "amount": 1000,  
  "currency": "USD",  
  "details": []  
}
```

entryDetails

```
{  
  "entryDescription": "Sales revenue",  
  "accountId": "4000",  
  "amount": 800,  
  "DrCr": "Cr",  
  "detailNo": 1  
}
```

TransactionResponse

```
{
  "responseCode": 200,
  "responseMessage": "saved successfully"
}
```

Rate Limiting

The Sage API Wrapper implements rate limiting to ensure system stability and fair usage. The following limits apply:

- **Requests per minute:** 60 requests per minute per API key
- **Requests per day:** 10,000 requests per day per API key

When you exceed these limits, the API will return a 429 Too Many Requests response with a Retry-After header indicating when you can resume making requests.

Rate Limit Headers

The following headers are included in each response to help you manage your rate limits:

- **X-RateLimit-Limit:** The maximum number of requests you're permitted to make per minute
- **X-RateLimit-Remaining:** The number of requests remaining in the current rate limit window
- **X-RateLimit-Reset:** The time at which the current rate limit window resets in UTC epoch seconds

Example Rate Limit Exceeded Response

```
{
  "responseCode": "429",
  "responseMessage": "Too many requests. Please try again later.",
  "timeStamp": "2025-07-30T08:41:00Z"
}
```

Support

For support with the Sage API Wrapper, please contact:

- **Email:** mupumamgtmgsdev@gmail.com
- **Support Hours:** Monday to Friday, 8:00 AM to 6:00 PM (UTC)
- **Response Time:** Within 24 business hours

Reporting Issues

When reporting issues, please include:

1. Your API key (masked except for the last 4 characters)
2. Request timestamp
3. Request endpoint
4. Request payload (with sensitive data removed)
5. Response received
6. Expected behavior