BANKING CIRCLE

Integration Guide

ISO 20022 pain.001 and pain.002 via API

Version Control

Version	Date	Summary of changes
1.0	Sept 2022	Document Created
1.1	Oct 2023	Updated with Idempotency key usage

Contents

1. Contents of this guide

This document describes how to send and receive ISO 20022 Payment initiation messages to and from Banking Circle API.

ISO 20022 Messages	Banking Circle API
Customer Credit Transfer Initiation V03 pain.001.001.03 xml file **Customer Payment Status Report V03** pain.002.001.03 xml format	**Request** **Customer Credit Transfer Initiation V03** pain.001.001.03 *POST /api/v1/payments/iso20022/customer-credit-transfer-initiation* xml file **Response** **Customer Payment Status Report V03** pain.002.001.03 xml format

2. A note on authentication and access tokens

Integrating to Banking Circle API: To send and receive data from the Banking Circle API, you must be authenticated and have a valid access token. For more details on how to generate access tokens and authenticate, please see our API Portal: Connecting to the API (https://docs.bankingcircleconnect.com/guides/connecting-to-the-api) and API Reference - Authorization (https://docs.bankingcircleconnect.com/api-reference/m2m-user-token).

2.1 Technical support

For additional technical support not covered in this document, please reach out to our Integration Team by email at integration@bankingcircle.com

3. ISO 20022 pain.001 initiation using API

The Payments initiation messages are set of messages exchanged between a debtor (or buyer) and its bank or between a creditor (or seller) and its bank to initiate, collect, manage, and monitor payments.

We support below Payments initiation messages:

Message	Message Definition
pain.001.001.03 CustomerCreditTransferInitiationV3	The CustomerCreditTransferInitiation message is sent by the initiating party to the forwarding agent or debtor agent. It is used to request movement of funds from the debtor account to a creditor.
pain.002.001.03 CustomerPaymentStatusReportV3	The CustomerPaymentStatusReport message is sent by an instructed agent to the previous party in the payment chain. It is used to inform this party about the positive or negative status of an instruction (either single or file). It is also used to report on a pending instruction.

3.1 API Request for Customer Credit Transfer Initiation pain.001

To initiate a pain.001 payment, a POST request should be sent to:

POST /api/v1/payments/iso20022/customer-credit-transfer-initiation

The Credit Transfer endpoint is designed for processing single and multiple payments in the pain.001 xml format. This endpoint is suited for companies operating with ISO20022 industry changes for singles and large volumes of payments.

<u>Note</u>: For initiating customer credit transfers, you need a pain.001.001.03 xml file. Banking Circle only supports the following version: pain.001.001.03.

Below are the specifications for the parameters in the header and body:
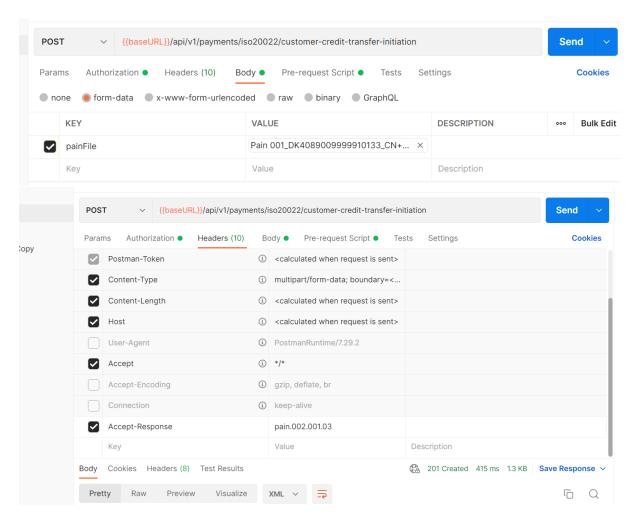
Content Specifications Header

Name	Required	Format	Description	Content
Accept-Response	Mandatory	String	Requested Customer Payment status report pain.002	Value: pain.002.001.03
Authorization	Mandatory	String	Bearer token	Bearer eyJ0eXAiOiJKV1Q iLCJhb…

Host	Mandatory	String	Target host	Test: sandbox.bankingcircleconnect.com Production: bankingcircleconnect.com
Accept	Mandatory	String	The Accept header is used to inform the server by the client that which content type is understandable by the client expressed as MIME-types.	Valid values: */* application/* application/xml
Content-Type	Mandatory	String	Media type of input resource. Boundary is a delimiter of data parts in the input.	Value: Multipart/form-data; Example: boundary=---xyz
X-Content-Type	Optional	String	X-ContentType is not required but may be in the future. Developers are encouraged to set the value to text/pain001.	Value: text/pain001
Content-Length	Mandatory	Number	Indicates the size of the message body in bytes.	Example: 2277
Idempotency-Key	Optional	String	Unique value generated by the user which the API uses to recognize subsequent retries of the same request. This field is limited to a maximum of 100 characters, if not upheld the API will return with an error 400 - Bad Request.	Example: 2b7b05ad-4411-4a19-8c95-2508b17e0fd8

Content Specifications Body

Parameter	Required	Allowed Input	description	Value
painFile	Mandatory	UTF8 file	File containing pain.001.001.03 xml request.	Your pain.001 file

3.1.1 Examples

For details on how to build your pain.001 xml message, read more the Banking Circle - ISO 20022 Implementation Guide, sections 2.1.2. Message definition, 2.1.3. Message sample.

For details on the payment solutions, you can test and how to formulate your pain.001 xml message, read Banking Circle - ISO 20022 Implementation Guide, Chapter 3 Payment Solutions.

3.2 API Response to Customer Credit Transfer Initiation pain.001

For every initiation pain.001 xml message file, we respond with a pain.002.001.03 xml format or an error response. The ISO 20022 Customer Payment Status Report pain.002.001.03 message is an output status message for the initiated pain.001.001.03 and gives information on credit transfers.

<u>Note</u>: Banking Circle only supports the following version: pain.002.001.03.

Banking Circle API performs API, then XSD, and upfront validations, and returns one of the following responses:

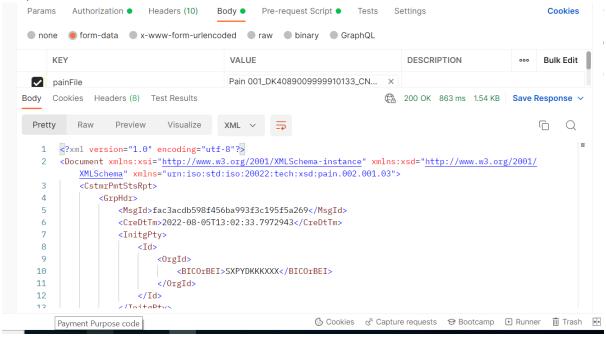| Response Code	Header	Description
200	Content-Type: application/xml	OK Request is not sent for processing; all payments are rejected. pain.002 gets generated (Refer <u>Banking Circle - ISO 20022 Implementation Guide</u> section 2.2.4. Message sample)
	X-Correlation-ID	GUID Correlates HTTP requests between a client and server.
	Request-Context	Used for cross-component correlation when 2 of your applications use different instrumentation keys
201	Content-Type: application/xml	Success: Created Request is sent for processing pain.002 gets generated (Refer <u>Banking Circle - ISO 20022 Implementation Guide</u> section 2.2.4. Message sample)
	X-Correlation-ID	GUID Correlates HTTP requests between a client and server.
	Request-Context	Used for cross-component correlation when 2 of your applications use different instrumentation keys
403	Content-Type: application/json	Forbidden. The server understood the request but is refusing it or the access is not allowed, i.e., if resource exists but calling user does not have rights to access it. No pain.002 generation.

400	Content-Type: application/json	Bad request. Request is not correctly formulated.
		No pain.002 generation.
		
		In case the Header, Query, or Body are not correctly formulated, the response is:
		
		{
		  "Id": "f7004420-0780-4ed3-ba25-56ed8920a921",
		  "Type": "Pain001",
		  "Errors": [
		    {

		``
		      "ErrorCode": "InvalidFile",
		    }
		  ]
		}
		``
		
		In case XSD validations fail, there will not be a pain.002 generated. Instead, the response is:
		
		``
		{
		  "Id": "f7004420-0780-4ed3-ba25-56ed8920a921",
		  "Type": "Pain001",
		  "Errors": [
		    {
		      "ErrorCode": "InvalidFile",
		      "ErrorDescription": "The element 'PmtInf' in namespace 'urn:iso:std:iso:20022:tech:xsd:pain.001.001.03' has invalid child element 'PmtInfIdx' in namespace 'urn:iso:std:iso:20022:tech:xsd:pain.001.001.03'. List of possible elements expected: 'PmtInfId' in namespace 'urn:iso:std:iso:20022:tech:xsd:pain.001.001.03'"
		    }
		  ]
		}
		``
		
		Refer Banking Circle - ISO 20022 Implementation Guide section 2.2.4. Message sample: Message does not pass XSD validation.

404	Content-Type: application/json	Not found. There is no resource behind the URL, i.e., requesting data on an account that does not exist or don't have permissions to.
		
		No pain.002 generation.
		
		``
		{
		  "errors": [
		    {
		      "httpStatus": 404,
		      "errorCode": null,
		      "keyOrMessage": "Not Found",
		      "interpolations": [],
		      "details": null,
		      "exception": null
		    }
		  ],
		  "account": null,
		  "_links": null
		}
		``

3.2.1 Examples

Response 201

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
        XMLSchema" xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03">
3       <CstmrPmtStsRpt>
4           <GrpHdr>
5               <MsgId>a7e30f7d3706497191537638e07967e6</MsgId>
6               <CreDtTm>2022-08-05T12:41:07.887261</CreDtTm>
7               <InitgPty>
8                   <Id>
9                       <OrgId>
10                          <BICOrBEI>SXPYDKKKXXX</BICOrBEI>
11                      </OrgId>
12                  </Id>
13              </InitgPty>
14          </GrpHdr>
```
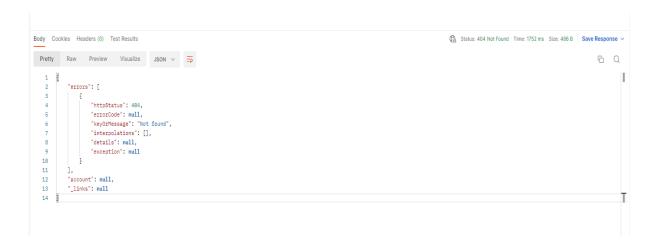
201 Created   415 ms   1.3 KB

Body | Cookies | Headers (8) | Test Results

Pretty | Raw | Preview | Visualize | XML

Save Response

Response 200

Params | Authorization ● | Headers (10) | Body ● | Pre-request Script ● | Tests | Settings | Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION	ooo	Bulk Edit
☑	painFile	Pain 001_DK4089009999910133_CN... ✕			

Body | Cookies | Headers (8) | Test Results

200 OK   863 ms   1.54 KB

Pretty | Raw | Preview | Visualize | XML

Save Response

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
        XMLSchema" xmlns="urn:iso:std:iso:20022:tech:xsd:pain.002.001.03">
3       <CstmrPmtStsRpt>
4           <GrpHdr>
5               <MsgId>fac3acdb598f456ba993f3c195f5a269</MsgId>
6               <CreDtTm>2022-08-05T13:02:33.7972943</CreDtTm>
7               <InitgPty>
8                   <Id>
9                       <OrgId>
10                          <BICOrBEI>SXPYDKKKXXX</BICOrBEI>
11                      </OrgId>
12                  </Id>
13                  </InitgPty>
```

Payment Purpose code

Cookies | Capture requests | Bootcamp | Runner | Trash

Response 400


```json
{
    "Id": "de630ffe-bf06-4515-b329-a51230f6ec41",
    "Type": "Pain001",
    "Errors": [
        {
            "ErrorCode": "InvalidFile",
            "ErrorDescription": "The 'FinInstnId' start tag on line 50 position 7 does not match the end
                tag of 'FinInstnId--'. Line 53, position 8."
        }
    ]
}
```

Response 404


```json
{
    "errors": [
        {
            "httpStatus": 404,
            "errorCode": null,
            "keyOrMessage": "Not found",
            "interpolations": [],
            "details": null,
            "exception": null
        }
    ],
    "account": null,
    "_links": null
}
```

For details on the content of the Customer Payment Status Report pain.002, refer to the Banking Circle - ISO 20022 Implementation Guide, sections 2.2.2. Message definition, 2.2.4. Message sample.

For details on the validation types, refer to Banking Circle - ISO 20022 Implementation Guide, section 2.2.3. Validations.

For details on the how to check the final status, read 'Checking status after core validations' in the Banking Circle ISO 20022 Implementation Guide, section 2.2.3. Validations.

4. ISO 20022 Payment Initiation on the Sandbox API

For building your API requests follow the guidelines as in the API Request Customer Credit Transfer Initiation pain.001 section of this document. You can expect the same type of responses as mentioned API Response to Customer Credit Transfer Initiation pain.001 .

For details on how to build your pain.001 xml message, read more the Banking Circle - ISO 20022 Implementation Guide, sections 2.1.2. Message definition, 2.1.3. Message sample.

For details on solutions, you can test and how to formulate your pain.001 xml message, read Banking Circle - ISO 20022 Implementation Guide, chapter 3 Payment Solutions.

Note that the Sandbox environment, the final status of ISO 20022 payments will follow the payment simulator which relies on currency crosses or creditor accounts as described in the Payments section (https://docs.bankingcircleconnect.com/guides/api-sandbox) of the API Sandbox.