standard chartered

Guide to implementing ISO 20022 and Bank standards for Payment APIs

Table of contents

Background

Due to the ISO 20022 migration for Cross-Border Payments and Reporting Plus (CBPR+) and High Value Payment Systems (HVPS+), we have updated our Payment APIs to ensure our services are secure and fully compliant with regulatory requirements.

What it means for you

Not all the updates may apply to you. We encourage you to start aligning current live/going-to-live payment requests early to the updated specifications to ensure your payment requests can continue smoothly without any interruptions. Any payment requests not aligned with the published specifications will be rejected from November 2026.

Payment APIs in scope

All the Move Money - Payment APIs used to initiate a payment instruction to the Bank are in scope. The list of APIs as per the below. You would need to check which of these APIs you are using and make the required alignments for every API that you identified.

- **Payment Initiation**
  - https://api.standardchartered.com/openapi/payments/v2/initiate
  - https://api.standardchartered.com/openapi/payments/v2/payee/initiate
- **Bulk Transfer**
  - https://api.standardchartered.com/openapi/payments/v2/bulk/initiate
- **Market-specific initiation endpoints**

  If you are currently using the market-specific endpoints below, you are encouraged to switch to the Payment Initiation API (https://api.standardchartered.com/openapi/payments/v2/initiate) which supports a single payment instruction for all available markets and payment types and has the same specifications as the market-specific endpoints.

  - https://api.standardchartered.com/openapi/payments/v2/fps/bank-transfer/initiate
  - https://api.standardchartered.com/openapi/payments/v2/fps/proxy/initiate
  - https://api.standardchartered.com/openapi/payments/v2/fps/proxy/initiate-with-validation
  - https://api.standardchartered.com/openapi/payments/v2/fast/initiate
  - https://api.standardchartered.com/openapi/payments/v2/paynow/initiate
  - https://api.standardchartered.com/openapi/payments/v2/paynow/initiatewithproxyvalidation
  - https://api.standardchartered.com/openapi/payments/v2/imps/initiate
  - https://api.standardchartered.com/openapi/payments/v2/bifast/bank-transfer/initiate
  - https://api.standardchartered.com/openapi/payments/v2/duitnow/initiate
  - https://api.standardchartered.com/openapi/payments/v2/promptpay/bank-transfer/initiate

- **Obsolete initiation endpoints**

  If you are currently using the obsolete endpoints below, you must switch to the Payment Initiation API (https://api.standardchartered.com/openapi/payments/v2/initiate) which supports a single payment instruction for all available markets and payment types and has the same specifications as the obsolete endpoints.

  - https://api.standardchartered.com/openapi/duitnow/payment/initiate
  - https://api.standardchartered.com/openapi/payments/v2/crossborder/initiate

Note: The current supported version for Move Money – Payment APIs is version 2 (v2) since 2021. The next iteration, version 3 (v3), which contains alignments to other ISO 20022 non-address related updates, is scheduled for release in 2027. Details regarding this iteration will be communicated closer to date of release.

Testing and validation

A sandbox environment has been made available on Standard Chartered's Open Banking Marketplace (OBM), where you can input your payment request payloads via the UI, and the response will indicate if there are any residual errors in your payload that need correcting.

Summary of the process

1. Identify all the in-scope Move Money - Payment APIs you have implemented with Standard Chartered.
2. Review the payment instructions and amend them according to the areas listed in this document.
3. Register for an account in OBM. We will review your registration and send you an email to set up your account when it is approved.
4. Once your account is set up, log in to OBM.
5. Test your amended payment instructions using our OBM sandbox. Refer to the details below.
6. Ensure you receive a success 200 response. If not, review, amend and test your payment instructions until a success 200 response is received.
7. Confirm with your Client Service Group that you have completed the necessary changes and go live with them.

Sandbox testing

1. Go to the Open Banking Marketplace sandbox and select "Payment Initiation" endpoint under "Cash management > Move Money".
2. We have prepared two test cases with blank payment instructions for each of the use case below. Determine your use case and select the API and test case accordingly.

S/N	Use case	API	Test case name
1.	Single payment	Cash management > Move Money > Payment Initiation > Payment Initiation	[ISO 20022 validation] Initiate a payment
2.	Bulk payments	Cash management > Move Money > Payment Initiation > Bulk Transfer Initiation (Deprecated)	[ISO 20022 validation] Initiate a bulk payment

3. Follow the test case instructions.
4. Copy and paste your amended payment instructions under **Request parameters** > **Body**.
5. Ensure that you have:

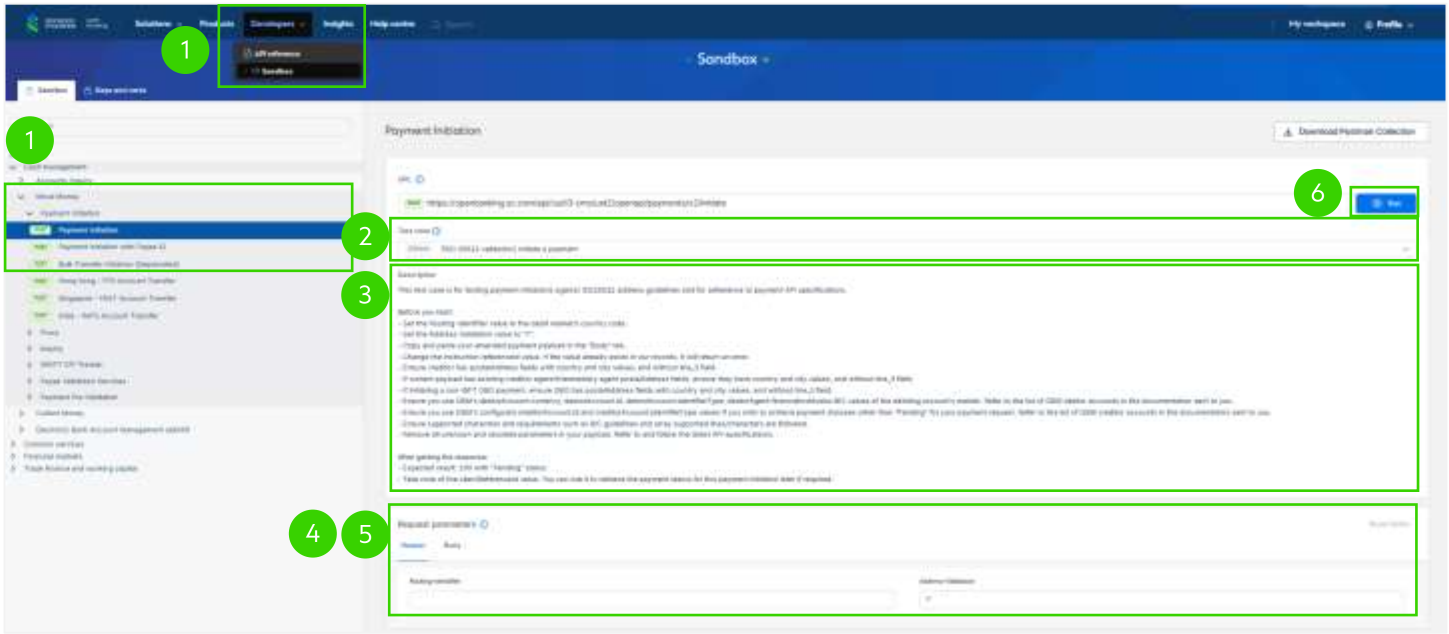
   a. Set the `Routing-Identifier` (HTTP header) value to the debit market's country code.

   b. Set the `Address-Validation` (HTTP header) value to **Y**.

   c. Changed the `debtorAccount`, `debtorAccount.currency` and `debtorAccount.identifierType` values to OBM's debtor account and supported currency as per the debit market. Refer to the list of **OBM debtor accounts** in the **Appendix**.

6. Click **Run**.

Note: A successful 200 response will always have a **Pending** status. If you wish to have other statuses for your payment instruction when retrieving the status via the Payment Status Inquiry API, refer to **OBM creditor accounts** in the **Appendix**.



Changes required

1. ISO 20022 address guidelines adoption

The following changes for ISO 20022 address standards are mandatory for all payment types as the financial industry moves to ISO 20022 standards with more enriched and structured information, enabling greater transaction data quality and transparency, achieving increased efficiency, interoperability and strengthening compliance controls.

The Bank has introduced a mandatory field in the header to help clients who are ready with the change to self-validate with ease.

HTTP header

S/N	Field	Remarks
1.	`Address-Validation`	A new HTTP header field to indicate the payment instruction is ready for address format validation. This will stimulate an additional validation on whether address elements (`country` and `city`) are provided for the applicable parties and agents in payment instruction as per the conditions described in the **Request body** table. If the conditions are not met, an error 400 response will be returned. Another validation will also be stimulated for address line 3 as it should not be used. If provided, it will return an error 400 response. Use this value: Y

According to the ISO 20022 address guidelines, town name and country are required fields. In our Payment API specifications, town name is denoted by the `city` field. Therefore, you need to include both `city` and `country` fields in your payment instructions. Refer to the table below for the specific fields affected.

Request body

S/N	Party/Agent	Field	Mandatory/Optional	Specifications
1.	Creditor	`instruction.creditor.postalAddress.country`	Mandatory	2-char ISO country code
2.		`instruction.creditor.postalAddress.city`	Mandatory	Max length: 35
3.	On-Behalf-Of (OBO)	`instruction.onBehalfOfInfo.postalAddress.country` (existing mandatory field)	Mandatory if: Initiating OBO payments	2-char ISO country code
4.		`instruction.onBehalfOfInfo.postalAddress.city`		Max length: 35

5.	Creditor agent	`instruction.creditorAgent.financialInstitution.postalAddress.country`	Mandatory if: Current payment instruction has creditor agent with `postalAddress` fields	2-char ISO country code
6.		`instruction.creditorAgent.financialInstitution.postalAddress.city`		Max length: 35
7.	Intermediary agent	`instruction.intermediaryAgent.financialInstitution.postalAddress.country`	Mandatory if: Current payment instruction has intermediary agent with `postalAddress` fields	2-char ISO country code
8.		`instruction.intermediaryAgent.financialInstitution.postalAddress.city`		Max length: 35

`creditor.postalAddress` sample

```
"creditor": {
  "postalAddress": {
    "country": "SG",
    "city": "Singapore"
  }
}
```

Notes:

a. Ensure both `country` and `city` (town name) fields are always provided for the applicable parties and agents in the payment instruction.

b. If you currently do not use `creditorAgent` (e.g. proxy payment), `intermediaryAgent`, or `onBehalfOfInfo` (e.g. not transacting an OBO payment) objects in your payment instruction, it is not required to add the `country` and `city` fields for these parties/agents.

c. If you currently do not provide any `postalAddress` fields in `creditorAgent` and/or `intermediaryAgent` object in your payment instruction, it is not required to add the `country` and `city` fields for these agents.

d. For China domestic payments, use only the `country` and `city` fields for `creditor` `postalAddress`, else the payment instruction will be rejected to meet local data regulations.

e. There is no change to all other existing `postalAddress` specifications (e.g. `line_1`, `line_2`, `state`, `postcode`).

2. Adherence to Payment APIs schema

Our Payment APIs have been updated as part of our commitment to maintaining compliance with payment processing standards. The Bank has identified several areas that requires your attention and amendment to comply with these standards. Please review the following in detail to ensure that your payment instruction requests will not be rejected, and your applications that utilises our APIs can continue to run smoothly after November 2026.

2.1. New HTTP header mandatory fields

Note that these two fields are HTTP header fields and not a field in the API request.

New HTTP header field	Specification
`Routing-Identifier`	2-char ISO country code indicating booking location For example: - Use **IN** for payment requests to be processed in India as the booking location - Use **HK** for payment requests to be processed in Hong Kong as the booking location
`Address-Validation`	Refer to **ISO 20022 address guidelines adoption** section above.

2.2. Payment type

It is now mandatory to indicate the payment type in your payment instruction request via `instruction.paymentType`.

Field	Specification
`instruction.paymentType`	The transaction payment type such as ACH, RTGS, BT, TT, XACH, XBT, IBFT, PAY, etc.

If you are also using `instruction.paymentTypePreference`, ensure the value provided in `instruction.paymentType` follows the logic below.

Field	Specification
`instruction.paymentTypePreference`	• For **Fastest**, use only **IBFT** in `instruction.paymentType` • For **Cheapest**, use only **ACH** in `instruction.paymentType` • For **Explicit** or when not provided, use any payment type in `instruction.paymentType`

IBFT payment sample

```
"paymentType": "IBFT",
"paymentTypePreference": "Fastest"
```

ACH payment sample

```
"paymentType": "ACH",
"paymentTypePreference": "Cheapest"
```

Non-IBFT/non-ACH payment sample

```
"paymentType": "RTGS",
"paymentTypePreference": "Explicit"
```

When paymentTypePreference is not provided

```
"paymentType": "TT",
"paymentTypePreference": ""
```

2.3. Array data type fields

Do not exceed the maximum supported lines and characters for the following array fields.

Unstructured remittance information

Field	Note to clients
`instruction.remittanceInfo.multiUnstructured[]`	• Max 2 lines, each line max 70 characters • Do not use `instruction.remittanceInfo.unstructured` as it is no longer supported. Refer to the **Obsolete fields** section for more details.

Local language supported fields

Field	Note to clients
`instruction.creditorExtension.nameLocalLanguage[]`	Max 2 lines, each line max 80 UTF-8 characters
`instruction.creditorExtension.addressLinesLocalLanguage[]`	Max 4 lines, each line max 80 UTF-8 characters
`instruction.remittanceInfoExtension.detailsLineLocalLanguage[]`	Max 2 lines, each line max 70 characters

2.4. Supported characters

Use only supported characters for these fields.

Field	Note to clients
`instruction.remittanceInfo.multiUnstructured[]`	a-z A-Z 0-9 /?().,'+:-_
`instruction.onBehalfOfInfo.name`	English characters only

2.5. BIC and country code specification guidelines

Field	Specification
`instruction.debtorAgent.financialInstitution.BIC`	Always specify Standard Chartered Bank's 11-characters BIC based on where your account is opened. Refer to the list in **Appendix** > **OBM debtor accounts**.

instruction.debtorAgent.financialInstitution.postalAddress.country	Use the country code of the market where your account is opened with Standard Chartered. Refer to the list in **Appendix** > **OBM debtor accounts**.
instruction.intermediaryAgent.financialInstitution.BIC	Always provide the intermediary agent's BIC if you are using the intermediaryAgent object

Note: Always provide the BIC value in upper case.

Correct BIC value in upper case

```
"BIC": " SCBLINBBXXX"
```

Incorrect BIC value in lower case

```
"BIC": "scblinbbxxx"
```

2.6. Unknown fields

Remove any fields in your payment requests that are not explicitly defined in the specifications.

Below are some examples of unknown fields that are not in our specifications and the common issues that give rise to the discrepancies. Note that this is only a small sample of unknown fields; the key requirement is to follow our specifications precisely.

You may also use the sandbox validation feature detailed in the **Testing and validation** section to check if your payment instructions have unknown fields. Payment instructions that contain unknown fields may get an error 400 response with error messages such as 'Invalid header.messageID detected'.

Note: Payment instructions containing unknown fields will be rejected in the sandbox now but only rejected in the production environment after November 2026 to enable sufficient time for testing and development.

S/N	Issue	Unknown field	Correct field
1.	Case-sensitive	header.messageID	header.messageId
2.	Wrong spelling	instruction.chargeBearer	instruction.chargerBearer
3.	Missing underscore	instruction.creditor.postalAddress.line1	instruction.creditor.postalAddress.line_1
4.	Unknown field	instruction.creditorAddress	Use the fields inside instruction.creditor.postalAddress object such as: instruction.creditor.postalAddress.country

2.7. Obsolete fields

We have removed fields that are no longer supported under the updated specifications. It is crucial to identify and remove these fields from your payment instruction requests to ensure the continued functionality and data quality of your transactions. Your payment instruction requests will be rejected if these fields are not removed. Adhering to these updates is essential to avoid any disruptions in service.

You may also use the sandbox validation feature detailed in the **Testing and validation** section to check if your payment instructions have these obsolete fields. Payment instructions that contain obsolete fields may get an error

400 response with error messages such as ‘Invalid instruction.discountType detected’.

Note: Payment instructions containing obsolete fields will be rejected in the sandbox now but only rejected in the production environment after November 2026 to enable sufficient time for testing and development.

```
1.  header.messageSender
2.  instruction.debtorAddress.addressType
3.  instruction.debtorAgent.branchCode
4.  instruction.debtorAgent.financialInstitution.branchCode
5.  instruction.creditorAgent.financialInstitution.branchCode
6.  instruction.relatedRemittanceInfo.multiUnstructured[]
7.  instruction.relatedRemittanceInfo.unstructured
8.  instruction.serviceType
9.  instruction.subPaymentType
10. instruction.debtor.contact.empty
11. instruction.debtorAccount.city
12. instruction.debtorAgent.clearingSystemCode
13. instruction.debtorAgent.financialInstitution.type
14. instruction.creditor.contact.empty
15. instruction.creditor.jointBeneficiaryIdentity
16. instruction.creditorAccount.city
17. instruction.creditorAgent.clearingSystemCode
18. instruction.creditorAgent.subBranchCode
19. instruction.creditorExtension.emailLinesLocalLanguage[]
20. instruction.discountType
21. instruction.documents.count
22. instruction.documents.name
23. instruction.documents.referenceId
24. instruction.documents.sequence
25. instruction.fxConversion.fxRateIndicator
26. instruction.fxConversion.quoteRefFXRate
27. instruction.fxConversion.quoteRefId
28. instruction.intermediaryAgent.financialInstitution.branchCode
29. instruction.onBehalfOfInfo.oboAddressCodes.code
30. instruction.onBehalfOfInfo.oboAddressCodes.description
31. instruction.onBehalfOfInfo.oboAddressLineIssuer
32. instruction.onBehalfOfInfo.oboAddressLineIssuerLL
33. instruction.onBehalfOfInfo.partyIdentifierCode
34. instruction.onBehalfOfInfo.partyIdentifierCountry
35. instruction.onBehalfOfInfo.partyIdentifierIssuer
36. instruction.onBehalfOfInfo.partyIdentifierIssuingAuthority
37. instruction.onBehalfOfInfo.partyIdentifierRegisteringAuthority
38. instruction.onBehalfOfInfo.partyIdentifierValue
39. instruction.relatedRemittanceInfo.structured.creditorReferenceDetails
40. instruction.relatedRemittanceInfo.structured.creditorReferenceId
41. instruction.relatedRemittanceInfo.structured.creditorReferenceType
42. instruction.relatedRemittanceInfoExtension.multiUnstructured[]
43. instruction.relatedRemittanceInfoExtension.structured.creditorReferenceDetails
44. instruction.relatedRemittanceInfoExtension.structured.creditorReferenceId
45. instruction.relatedRemittanceInfoExtension.structured.creditorReferenceType
46. instruction.relatedRemittanceInfoExtension.unstructured
47. instruction.remittanceInfoExtension.structured.creditorReferenceDetails
48. instruction.remittanceInfoExtension.structured.creditorReferenceId
49. instruction.remittanceInfoExtension.structured.creditorReferenceType
50. instruction.remittanceInfoExtension.unstructured
```

Appendix

OBM debtor accounts

Use these debtor accounts for your testing. If your debtor market's account's currency is not listed here (for example a TH account but in USD), change USD to THB.

Note: If you are using `instruction.debtorAccount.currency` in your payment instruction, change the currency value here. Otherwise, change the currency value at `instruction.amount.currencyCode`. If both exist in your payment instruction, ensure both values are the same for the testing.

Currency and debtor account sample

```
"amount": {
  "amount": 1,
  "currencyCode": "THB"
},
"debtorAccount": {
  "currency": "THB",
  "id": "00100123456",
  "identifierType": "BBAN"
}
```

S/N	Market	debtorAccount.currency	debtorAccount.id	debtorAccount.identifierType	11-characters BIC
1.	AE	AED	0002201234567	BBAN	SCBLAEADXXX
2.		USD	0102201234678	BBAN	
3.	BH	BHD	0001641234567	BBAN	SCBLBHBMXXX
4.	CN	CNY	501512345678	BBAN	SCBLCNSXSHA
5.	GB	GBP	0001270012345	BBAN	SCBLGB2LXXX
6.	GH	GHS	0105012345678	BBAN	SCBLGHACXXX
7.	HK	HKD	44111234567	BBAN	SCBLHKHHXXX
8.		CNY	44111234568	BBAN	
9.	ID	IDR	30681123456	BBAN	SCBLIDJXXXX
10.	IN	INR	45611001234	BBAN	SCBLINBBXXX
11.	KE	KES	2345678904	BBAN	SCBLKENXXXX
12.		USD	8709875431121	BBAN	

13.	LK	LKR	0018500123456	BBAN	SCBLLKLXXXX
14.	MY	MYR	312123456789	BBAN	SCBLMYKXXXX
15.	NP	NPR	0001101234567	BBAN	SCBLNPKAXXX
16.	PH	PHP	0141123456789	BBAN	SCBLPHMMXXX
17.	PK	PKR	0000001100012345	BBAN	SCBLPKKXXXX
18.	SG	SGD	0106123456	BBAN	SCBLSG22XXX
19.	TH	THB	00100123456	BBAN	SCBLTHBXXXX
20.	UG	UGX	0066512345600	BBAN	SCBLUGKAXXX
21.	US	USD	3582021234001	BBAN	SCBLUS33XXX
22.	VN	VND	VND90212345678	BBAN	SCBLVNVXXXX
23.	ZA	ZAR	00181234567	BBAN	SCBLZAJJXXX

OBM creditor accounts

These creditor accounts are configured for our sandbox to stimulate the various payment status returned when retrieving the payment status via the Payment Status Inquiry API. If you do not use these accounts during testing, the payment status returned via the Payment Status Inquiry API will always be pending.

creditorAccount sample

- IBFT payments
- 69 - Completed status

```
"creditorAccount": {
  "id": "20011101015011",
  "identifierType": "BBAN"
}
```

creditorAccount sample

- Non-IBFT payments
- 64 - Processed status

```
"creditorAccount": {
  "id": "91701015012",
  "identifierType": "Other"
}
```

S/N	creditorAccount.id	creditorAccount.identifierType	Payment status via Payment Status Inquiry API
IBFT payments			
1.	20011101015011	BBAN	69 - Completed
2.	20011101015022	BBAN	44 - Rejected
3.	20011101015033	BBAN	48 - Returned

4.	20011101015044	BBAN	72 - Rejected
Non-IBFT payments			
1.	91701015012	Other	64 - Processed
2.	91701015013	Other	44 - Rejected
3.	91701015014	Other	44 - Rejected
4.	91701015018	Other	72 - Rejected

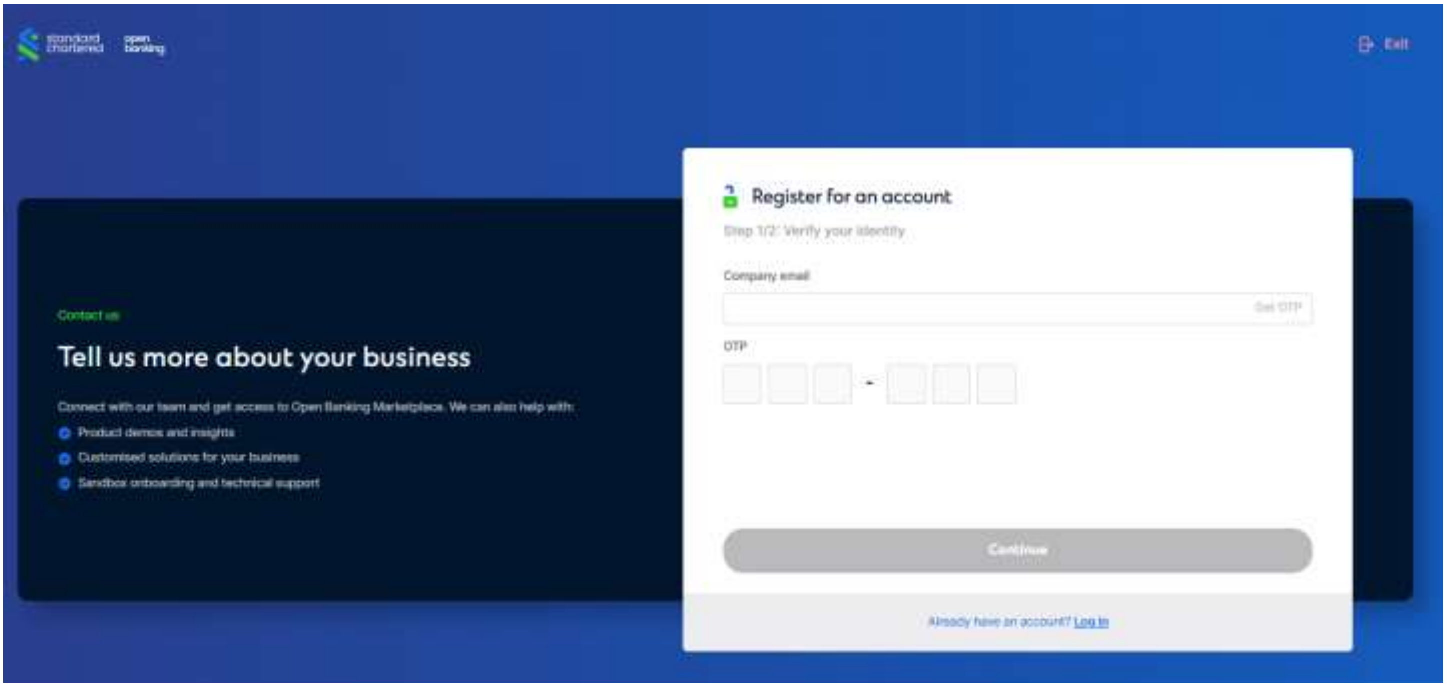
FAQs

What is the Open Banking Marketplace (OBM)?

OBM is a platform for you to discover, test, and implement Standard Chartered's open banking APIs. You can find the latest API specifications here as well as test the APIs in the sandbox environment.

How can I access OBM?

Your corporate email has been pre-approved. You can click here to register for an account via OBM.



Tell us more about your business

Connect with our team and get access to Open Banking Marketplace. We can also help with:

- Product demos and insights
- Customised solutions for your business
- Sandbox onboarding and technical support

Register for an account

Step 1/2: Verify your identity

Already have an account? Log in

Where can I find the latest specifications?

You can find the latest specifications of the Move Money – Payment APIs on Open Banking Marketplace.

Note: If the API you are currently using is not listed there, please refer to the Payment Initiation API for the latest specifications and conduct your testing using this API.

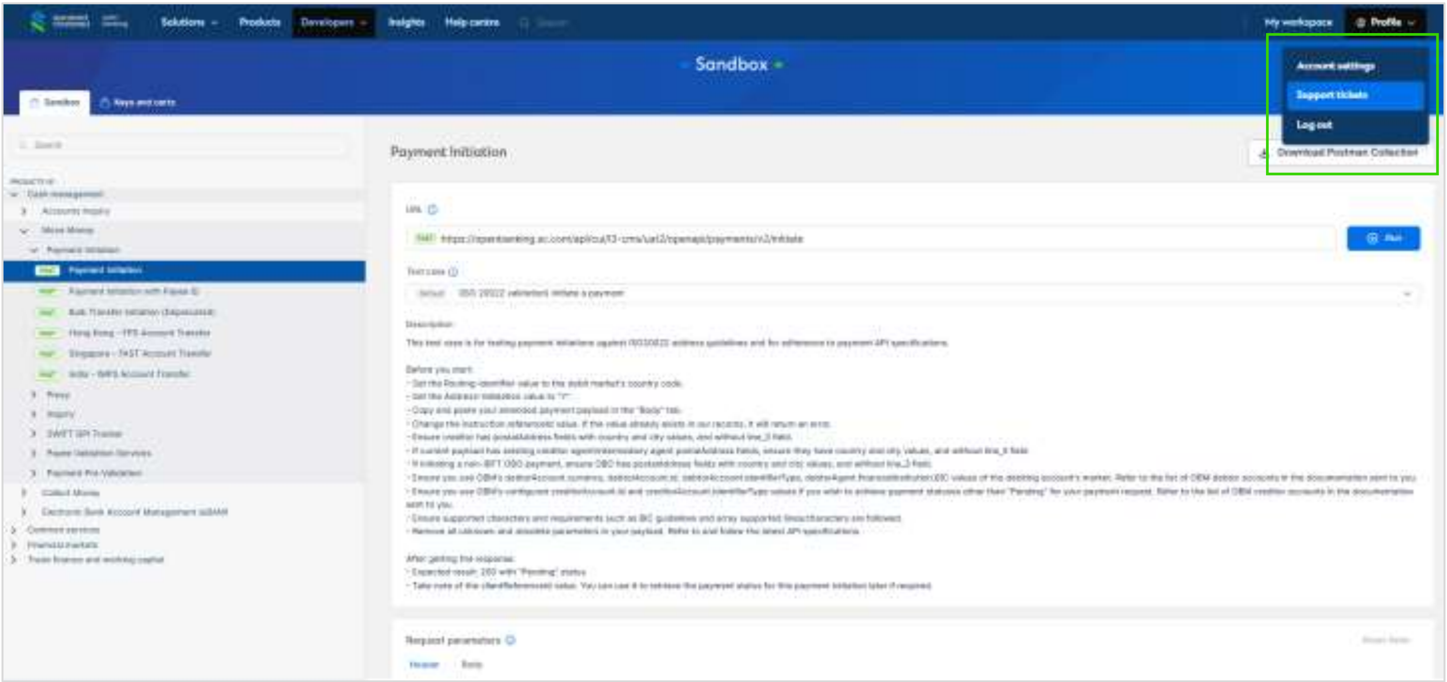
What validations do the OBM sandbox/UAT cover?

The OBM sandbox/UAT provides testing validations for JSON format, generic global API specifications, and the address guidelines specified in this document.

It does not cover functional validations, market/payment type nuances, and other end-to-end system validations.

Who can I contact if I require technical support of OBM sandbox/UAT?

You can raise a support ticket to us in OBM via **Profile** > **Support tickets**.



If my debtor market is not in the OBM debtor account list, what can I do?

You can raise a support ticket to us in OBM via **Profile** > **Support tickets** and let us know the debtor market and corresponding currency required.

If my debtor account has currencies that are not in your list of OBM debtor accounts, what can I do?

If your system currently constructs different set of fields in the payment instructions because of the debtor account currency, you are recommended to test all these different types of payment instructions by simply changing the currency value to match the debtor market's currency value when testing in the sandbox/UAT. For example, if your debtor market is TH with USD currency, change USD to THB, when run the updated API payloads in Sandbox.

What happens if I get an error 400 response?

If an error 400 response is returned, read the error message, amend your payment instruction request accordingly and try again.

What do these error messages mean?

- **Invalid [field name] detected**

  Getting this error message means that your payment instruction contains unknown/obsolete fields that are not in the payment API specifications. Common issues could include spelling mistakes, missing underscores, and case-sensitive field names. Check and follow the payment API specifications precisely by amending or removing these fields and you should be able to resolve this error.

- **[field name] is not supported for the payment type**

Getting this error message means that your payment instruction contains fields that are not supported for the payment type indicated in your payment instruction. Remove these fields and you should be able to resolve this error.

- **Invalid value for debtorAccount.id**

  Getting this error message means that you did not use OBM's debtor account details in your payment instruction for your testing in sandbox/UAT. Change the debtor account details such as the debtor account ID and supported currency as per the debit market in your payment instruction and you should be able to resolve this error. You can refer to the list of OBM debtor accounts in this document at **Testing and validation** > **Sandbox testing** > **OBM debtor accounts**.

  If you have changed the debtor account details and this error persists, you can also verify whether the `instruction.debtorAgent.financialInstitution.BIC` uses Standard Chartered Bank's 11-characters BIC based on the 'Market' of the OBM debtor account used in your payment instruction.

I am getting 5XX error response codes. What should I do?

5XX error response codes indicate that the server encountered an unexpected condition that prevented it from fulfilling the request. Sometimes, the issue is temporary. You can wait a while and try making the request again. If the issue persists, you can raise a support ticket to us in OBM via **Profile** > **Support tickets**.

Why is my payment status always pending?

The payment status for the initial payment instruction will always be pending. If you wish to test out other payment statuses such as processed, rejected, and so on, you will need to replace your creditor account values to the ones we have configured for our sandbox/UAT environments in your payment instructions. For more information, refer to **Appendix** > **OBM creditor accounts** in this document.

Why is sandbox/UAT different from production?

These latest API validation rules will be deployed to the sandbox/UAT environment before production to facilitate more efficient testing for every client.

The Bank's sandbox/UAT environment has more restricted validations to accept your API request. If your payload works in sandbox/UAT, you may also expect a 200 response for that API request when you go live in production and verify the subsequent payment processing which are subject to screening, monitoring etc.

If you are currently under the implementation phase running continuous or regression testing in the Bank's sandbox/UAT environment, please expect that the sandbox/UAT environment is stricter than the production environment and work with your client integration partner for the specific changes to smoothen your transition.

What if I am unable to complete the amendments and testing and go live by November 2026?

Any payment requests not aligned with the published specifications will be rejected from November 2026. Therefore, we encourage you to start aligning your payment requests early to the updated specifications to ensure your payment requests can continue smoothly without any interruptions.

Why are these changes necessary?

All Payment Initiation API payloads are being aligned to comply with ISO20022 standards for Cross-Border Payments and Reporting Plus (CBPR+) as well as High Value Payment Systems (HVPS+).

As part of this exercise, non-essential data such as obsolete and unsupported fields should also be removed from your payload. This will help to improve data quality, processing efficiency and regulatory compliance.

After November 2026, API requests not meeting specifications will be rejected — which may potentially lead to delays. We encourage you to start this transition early to ensure your payments can be processed without interruptions. Please note that this is part of the existing process.

Thank you for helping us maintain these standards to stay aligned with regulatory expectations.

Disclaimer

This material has been prepared by one or more members of SC Group, where "SC Group" refers to Standard Chartered Bank and each of its holding companies, subsidiaries, related corporations, affiliates, representative and branch offices in any jurisdiction, and their respective directors, officers, employees and/or any persons connected with them. Standard Chartered Bank is authorised by the United Kingdom's Prudential Regulation Authority and regulated by the United Kingdom's Financial Conduct Authority and Prudential Regulation Authority.

This material has been produced for reference and information purposes only, is not independent research material, and does not constitute an invitation, recommendation or offer to subscribe for or purchase any of the products or services mentioned or to enter into any transaction.

Some of the information herein may have been obtained from public sources and while SC Group believes such information to be reliable, SC Group has not independently verified the information. Information contained herein is subject to change at any time without notice. Any opinions or views of third parties expressed in this material are those of the third parties identified, and not of SC Group. While all reasonable care has been taken in preparing this material, SC Group makes no representation or warranty as to its accuracy or completeness, and no responsibility or liability is accepted for any errors of fact, omission or for any opinion expressed herein. The members of SC Group may not have the necessary licenses to provide services or offer products in all countries, and/or such provision of services or offer of products may be subject to the regulatory requirements of each jurisdiction. This material has not been prepared with regard to the specific objectives, situation or particular needs of any particular person, and is not meant to be comprehensive. SC Group does not provide any information technology, accounting, legal, regulatory, tax or investment advice and this material should not be relied on as such. Any comments on information technology, accounting, legal, regulatory, tax or investment matters contained in this material should not be relied on or used as a basis to ascertain the various results or implications arising from the matters contained herein, and you need to exercise your own independent judgment (with the advice of your information technology, accounting, legal, regulatory, tax, investment and other professional advisers as necessary) with respect to the risks and consequences of any matter contained herein. SC Group expressly disclaims any liability and responsibility whether arising in tort or contract or otherwise for any damage or losses you may suffer from your use of or reliance of the information contained herein.

You may wish to refer to the incorporation details of Standard Chartered PLC, Standard Chartered Bank and their subsidiaries by visiting http://www.sc.com/en/incorporation-details.html.

This material is not for distribution to any person to which, or any jurisdiction in which, its distribution would be prohibited.