

Coventry Building Society

Sandbox API Specification V2.1

Contents

Coventry Building Society.....	0
Sandbox API Specification V2.1	0
Version control	5
Release Note	5
Overview	5
Document Overview	5
Design Principles	6
RESTful APIs.....	6
Standards	6
Scope	6
Basics	7
Overview	7
TPP Information Steps	7
Sequence Diagram.....	9
Actors	10
Character Encoding	12
Date Formats.....	12
Resource URI Path Structure.....	12
Headers.....	13
Request Headers.....	13
Response Headers	14
Return & Error Codes.....	15
400 (Bad Request) v/s 404 (Not Found)	16
403 (Forbidden)	16
429 (Too Many Requests).....	17
Pre-Conditions.....	17
Pre-conditions for TPPs.....	17
Filtering.....	17
Pagination	19
All available Sandbox Endpoints.....	19
POST /token.....	20
POST open-banking-sandbox/v2.0/account-requests	21
Account Request Status	21
Mocked Responses	21

Data Dictionary - Response.....	22
GET open-banking-sandbox/v2.0/account-requests/{AccountRequestId}	23
Account Request Status	23
Mocked Responses	23
Data Dictionary – Response.....	24
DELETE open-banking-sandbox/v2.0/account-requests/{AccountRequestId}.....	25
Mocked Responses	25
GET open-banking-sandbox/v2.0/accounts	26
Data Dictionary - Response.....	26
GET open-banking-sandbox/v2.0/accounts/{AccountId}	27
Mocked Responses	28
Data Dictionary - Response.....	28
GET open-banking-sandbox/v2.0/accounts/{AccountId}/balances.....	29
Mocked Responses	29
GET open-banking-sandbox/v2.0/accounts/{AccountId}/beneficiaries	30
Mocked Responses	30
GET open-banking-sandbox/v2.0/accounts/{AccountId}/direct-debits	32
Mocked Responses	32
GET open-banking-sandbox/v2.0/accounts/{AccountId}/standing-orders	33
Mocked Responses	34
GET open-banking-sandbox/v2.0/accounts/{AccountId}/transactions.....	37
Mocked Responses	38
GET open-banking-sandbox/v2.0/accounts/{AccountId}/product	40
Mocked Responses	40
GET open-banking-sandbox/v2.0/accounts/{AccountId}/scheduled-payments	41
Mocked Responses	42
Data Dictionary- Response.....	42
Mandate Payment Status	44
Mocked Responses	44
Data Dictionary - Response.....	48
Mandate PaymentId Status	52
Mocked Responses	53
Payment Status	58
Mocked Responses	58
Data Dictionary - Response.....	62

PaymentId Status	66
Mocked Responses	66
Mandate Payment Submission Status	71
Mandate Payment Status	72
Mocked Responses	72
PaymentSubmissionId Status	77
Mocked Responses	77
Payment Status	79
Mocked Responses	79
PaymentSubmissionId Status	84
Mocked Responses	84
Funds Confirmation Consent Status	87
Funds Confirmation Consent Status	87
Mocked Responses	87
Funds Confirmation ConsentId Status	89
Data Payload – Enumerations	94
TransactionEntryCode	94
CreditDebitCode	94
BalanceTypeCode	95
ExternalAccountIdentification2Code	95
ExternalDirectDebitStatus1Code	95
ExternalPermissions1Code	96
ExternalProductType1Code	96
ExternalRequestStatus1Code	96
TransactionEntryCode	96
Usage Examples	99
All Permissions Granted	99
Setup Account Request	99
Status - AwaitingAuthorisation	100
Status - Authorised	101
Accounts - <i>Bulk</i>	104
Accounts - Specific Account	106
Balances - Specific Account	107
Beneficiaries - Specific Account	108
Direct Debits - Specific Account	109

Product - Specific Account	110
Standing Orders - Specific Account	111
Transactions - Specific Account	112
Scheduled Payment - Specific Account	113
Delete Account Request.....	113
Pagination	114
Payments Examples	116
Setup Payment Request.....	116
Setup Payment Submission Request for Single or Immediate Payments	124
Mandates Examples.....	129
Setup Mandate Payment Request for Future or Regular Dated Payment.....	129
Setup Mandate Submission Request for Regular or Future Dated Payments.....	136
Confirmation of Funds Examples.....	139
Setup Funds Confirmation Request	139
Status - AwaitingAuthorisation	140
Status - Authorised	141
Alternate and Error Flows	146
Missing or Expired Access Token	146
Incomplete or Malformed Request Payload	147
Missing or Invalid Access Token Scope.....	148
Failed Authorisation Consent	148

Version control

Version	Date	Updated by	Changes made
2.0	28 Jan 2019	Coventry Building Society	Baseline version
2.1	12 Feb 2020	Coventry Building Society	Add reference to eIDAS QWAC and OB WAC certificates

Release Note

This release note explains what's new in The Sandbox API Specifications v2.0 between versions.

V2.1 Baseline version – minor wording changes to include reference to eIDAS QWAC and OB WAC certificates.

Overview

This specification describes the Sandbox API flows and payloads. It applies to AISP, CBPIL and PISP third parties.

The API endpoints described here allow an AISP or PISP to use the Sandbox API's to test connectivity and functionality.

Document Overview

This document consists of the following parts:

Overview: Provides an overview of the scope of the API and the key decisions and principles that contributed to the specification.

Basics: The section identifies the resources, operations that are permitted on those resources, and various special cases.

Security & Access Control: Specifies the means for TPPs and PSUs to authenticate themselves and provide consent.

Swagger Specifications: Provides links to the swagger specifications for the APIs.

Data Model: Describes the data model for the API payloads.

Usage Examples: Examples for normal flows, and alternate flows.

Design Principles

RESTful APIs

The API adheres to RESTful API concepts where possible and sensible to do so.

However, the priority is to have an API that is simple to understand and easy to use. In instances where following RESTful principles would be convoluted and complex, the principles have not been followed.

References:

- The highest level Data Description Language used is the JSON Schema : <http://json-schema.org/>
- Best Practice has also been taken from the Data Description Language for APIs; JSON API : <http://jsonapi.org/>
- The Interface Description Language used is the Swagger Specification version 2.0 (also known as Open API) : <http://swagger.io/> and <https://github.com/OAI/OpenAPI-Specification>

Standards

The CBS principles for developing the new API standards:

- CBS will adopt existing standards where relevant/appropriate.
- CBS has adopted Open Banking Standards in the development and use of the APIs <https://www.openbanking.org.uk/standards/>

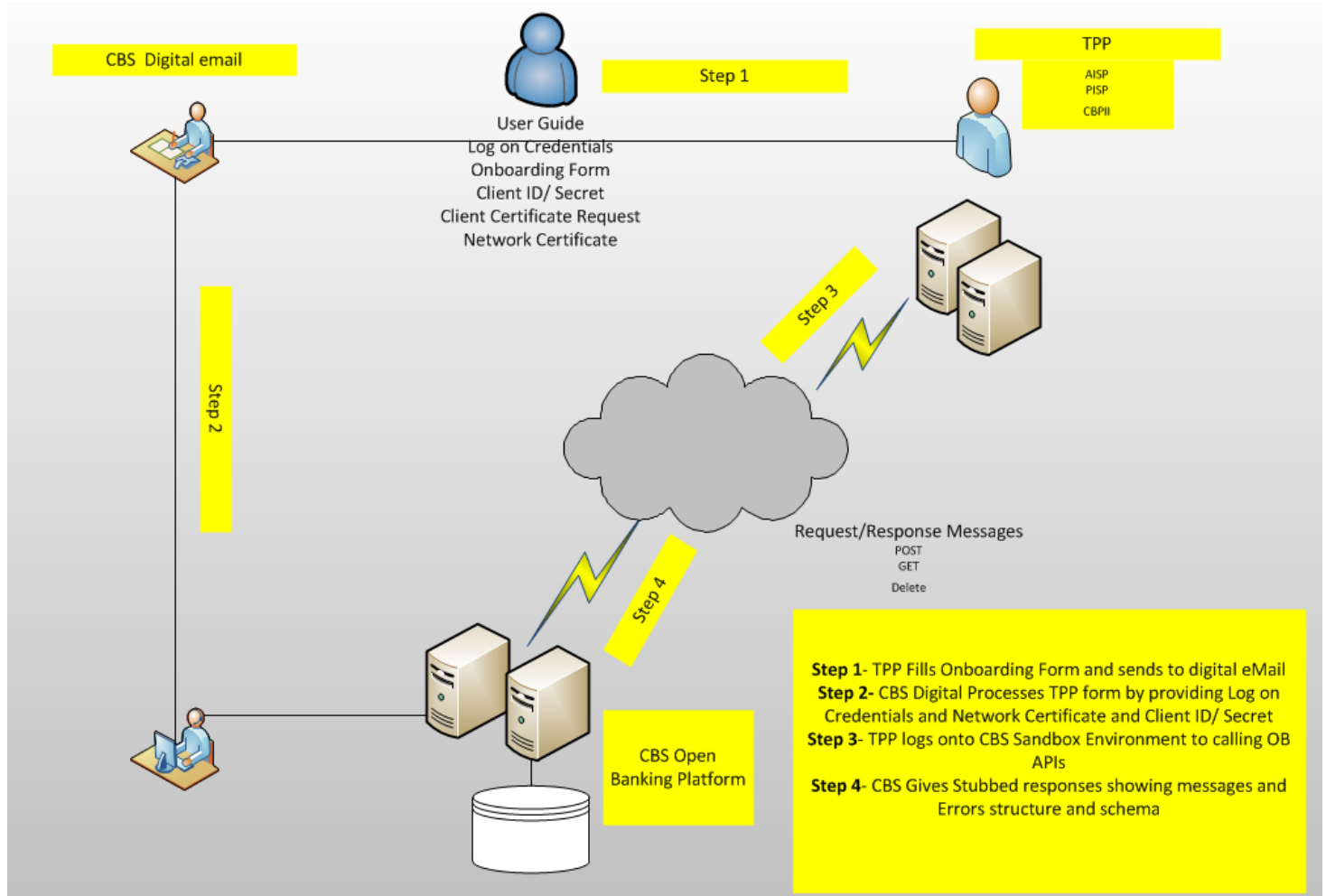
Scope

The APIs specified in this document provide the ability for AISPs, CBPIIs or PISPs to access Sandbox versions of the production APIs. All Sandbox APIs return mocked data; they are all accessible using a valid Access Token which was retrieved using the token endpoint via client credentials.

Basics

Overview

The figure below provides a general outline of a TPP information request and flow using the Account Info APIs as an example.



TPP Information Steps

Step 1: Request Access Token

- This flow begins with a TPP requesting an Access Token from our token endpoint:

<https://resourcema.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>

Request must include:

```
grant_type=client_credentials
scope=openid accounts
```


client_id={clientId provided by CBS when TPP on-boarded}
client_secret={client secret provided by CBS when TPP on-boarded}

Step 2: Setup Account Request

- The TPP connects to CBS and creates a **request** for resource. CBS responds with an identifier for the resource (e.g. AccountRequestId - which is the intent identifier). For the sandbox the AccountRequestId is always **6495080e-cc91-4171-a560-4ec9522740ee**
- This step is carried out by making a **POST** request to <https://resource.ma.coventrybuildingsociety.co.uk/pd/open-banking/open-banking-sandbox/v2.0/account-requests> endpoint
- The account-requests setup payload must include these fields:
 - Permissions - a list of data clusters that have been consented for access
 - Expiration Date - an optional expiration for when the AISP will no longer have access to the PSU's data
 - Transaction Validity Period - the From/To date range which specifies a transaction history period which can be accessed by the AISP

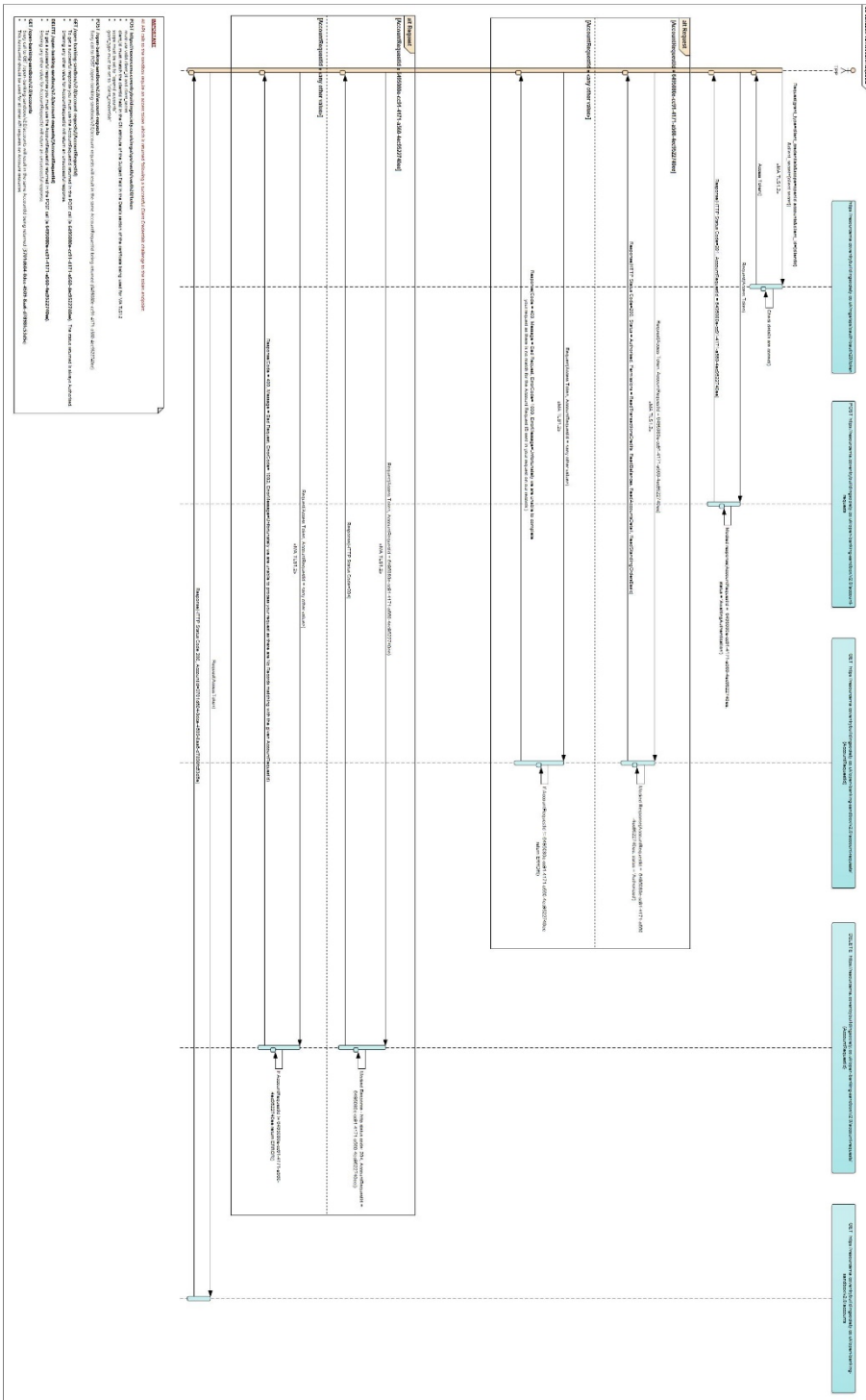
Step 3: Authorise Consent

- We do not provide sandbox endpoints to represent the authorisation of consents by PSU's. Once a valid Access Token has been received using client credentials, this Access Token will give access to all resource APIs.

Step 4: Request Data

- This is carried out by making a **GET** request to the relevant **resource**.
- The unique AccountId(s) that are valid for the account-request will be returned with a call to GET /accounts. **This will always be the first call once an AISP has a valid access token. For the sandbox GET /accounts API the AccountId returned will always be 3701d604-0dce-4509-8aa6-d7898fc53d5e.**

Sequence Diagram



Actors

Actor	Abbreviation	Type	Specializes	Description
Payment Service User	PSU	Person	N/A	A Payment Services User is a natural or legal person making use of a payment service as a payee, payer or both.
Payment Service Provider	PSP	Legal Entity	N/A	A Payment Services Provider is an entity which carries out regulated payment services, including AISPs, PISPs, CBPIIs and ASPSPs.
Account Servicing Payment Service Provider	ASPSP	Legal Entity	PSP	Account Servicing Payment Service Providers provide and maintain a payment account for a payer as defined by the PSRs and are entities that publish Read/Write APIs to permit, with customer consent, payments initiated by third party providers and/or make their customers' account transaction data available to third party providers via their API end points. CBS is an ASPSP.
Third Party Providers / Trusted Third Parties	TPP	Legal Entity	PSP	Third Party Providers are organisations or natural persons that use APIs developed to Standards to access customer's accounts, in order to provide account information services and/or to initiate payments. Third Party Providers are Payment Initiation Service Providers (PISPs) and/or Account Information Service Providers (AISPs) and/or Card-based Payment Instrument Issuers (CBPII)
Payment Initiation Service Provider	PISP	Legal Entity	TPP	A Payment Initiation Services Provider provides an online service to initiate a payment order at the request of the payment service user with respect to a payment account held at another payment service provider.

<p>Account Information Service Provider</p>	<p>AISP</p>	<p>Legal Entity</p>	<p>TPP</p>	<p>An Account Information Service provides account information services as an online service to provide consolidated information on one or more payment accounts held by a payment service user with one or more payment service provider(s).</p>
<p>Card-based Payment Instrument Issuers</p>	<p>CBPII</p>	<p>Legal Entity</p>	<p>TPP</p>	<p>A Card Based Payment Instrument Issuer is a payment services provider that issues card-based payment instruments that can be used to initiate a payment transaction from a payment account held with another payment service provider.</p>

Character Encoding

The API requests and responses **must** use a UTF-8 character encoding. This is the default character encoding for JSON (RFC 7158 - [Section 8.1](#)).

Date Formats

All dates in the JSON payloads are represented in ISO 8601 date-time format. All date-time fields in responses **must** include the timezone. An example is below:

2017-04-05T10:43:07+00:00

All dates in the HTTP headers are represented as [RFC 7231](#) Full Dates. An example is below:

Sun, 10 Sep 2017 19:43:31 UTC

JWT claims are expressed as a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

Resource URI Path Structure

The resources defined by these APIs can be addressed through a path structure consisting of the following parts:

- The version of the APIs expressed as /v[major-version].[minor-version]/
- The resource name

Examples:

- <https://resourcema.coventrybuildingsociety.co.uk/pd/open-banking/open-banking-sandbox/v2.0/account-requests>
- <https://resourcema.coventrybuildingsociety.co.uk/pd/open-banking/open-banking-sandbox/v2.0/accounts>
- <https://resourcema.coventrybuildingsociety.co.uk/pd/open-banking/open-banking-sandbox/v2.0/accounts/{AccountId}/balances>

Headers

Request Headers

The following headers SHOULD be inserted by the TPP in each API call:

Header Value	Notes	POST	GET	DELETE
x-fapi-financial-id	Should be set to "CBSOpenBanking"	Mandatory	Mandatory	Mandatory
x-fapi-customer-last-logged-time	The time when the PSU last logged in with the TPP.	Optional	Optional	Optional
x-fapi-customer-ip-address	The PSU's IP address if the PSU is currently logged in with the TPP.	Optional	Optional	Optional
x-fapi-interaction-id	An RFC4122 UID used as a correlation id. If provided, CBS will "play back" this value in the x-fapi-interaction-id response header.	Optional	Optional	Optional
Authorization	Standard HTTP Header; Allows Credentials to be provided to the Authorisation / Resource Server depending on the type of resource being requested. For OAuth 2.0 / OIDC, this comprises of either the Basic / Bearer Authentication Schemes.	Mandatory	Mandatory	Mandatory
Content-Type	Standard HTTP Header; Represents the format of the payload being provided in the request. This must be set to application/json.	Mandatory	Do not use	Do not use
Accept	Standard HTTP Header; Determine the Content-Type that is required from the Server. If set, it must have the value application/json. If set to any other value, CBS will respond with a 406 Not Acceptable.	Optional	Optional	Optional

(Reference: Section 6.3 - Financial API — Part 1: Read Only API Security Profile (Implementer's Draft).)

Whether the PSU is present or not-present is identified via the x-fapi-customer-ip-address header. If the PSU IP address is supplied, it is inferred that the PSU is present during the interaction.

Response Headers

Header Value	Notes	Mandatory ?
Content-Type	<p>Standard HTTP Header; Represents the format of the payload returned in the response.</p> <p>CBS will return Content-type: application/json as a content header in response to requests that return a HTTP body (all post and get requests)</p>	Conditionally Mandatory
x-fapi-interaction-id	<p>An RFC4122 UID used as a correlation id.</p> <p>This must be the same value provided in the x-fapi-interaction-id request header.</p> <p>Mandatory if provided in the request.</p>	Conditionally Mandatory
Retry-After	<p>Header indicating the time (in seconds) that the TPP should wait before retrying an operation.</p> <p>CBS will include this header along with responses with the HTTP status code of 429 (Too many requests).</p>	Optional

Return & Error Codes

Our Sandbox APIs use mocked data, to ensure that TPPs can test both happy and unhappy paths we will provide specific AccountRequestIds or PaymentIds which will result in a pre-determined response. Each individual Sandbox API will provide the details later in this document.

The following are the HTTP response codes for the different HTTP methods - across all Account Info and Payment Initiation API endpoints.

Situation	HTTP Status	Notes	Returned by POST	Returned by GET	Returned by DELETE
Query completed successfully	200 OK		No	Yes	No
Normal execution. The request has succeeded.	201 Created	The operation results in the creation of a new resource.	Yes	No	No
Delete operation completed successfully	204 No Content		No	No	Yes
Account Request has malformed, missing or non-compliant JSON body or URL parameters	400 Bad Request	The requested operation will not be carried out.	Yes	No	No
Authorization header missing or invalid token	401 Unauthorized	The operation was refused access. Re-authenticating the PSU may result in an appropriate token that can be used.	Yes	Yes	Yes
Token has incorrect scope or a security policy was violated.	403 Forbidden	The operation was refused access. Re-authenticating the PSU is unlikely to remediate the situation.	Yes	Yes	Yes
The TPP tried to access the resource with a method that is not supported.	405 Method Not Allowed		Yes	Yes	Yes
The operation was refused as too many requests have been made within a certain timeframe.	429 Too Many Requests	Throttling is a NFR. CBS will include a Retry-After header in the response indicating how long the TPP must wait before retrying the operation.	Yes	Yes	Yes
Something went wrong on the API gateway or micro-service	500 Internal Server Error	The operation failed.	Yes	Yes	Yes

400 (Bad Request) v/s 404 (Not Found)

When a TPP tries to request a resource URL with a resource Id that does not exist, CBS **will** respond with a 400 (Bad Request) rather than a 404 (Not Found).

E.g., if a TPP tries to GET /accounts/22289 where 22289 is not a valid AccountId, CBS **will** respond with a 400.

When a TPP tries to request a resource URL that results in no business data being returned (e.g. a request to retrieve standing order on an account that does not have standing orders) CBS **will** respond with a 200 (OK) and set the array to be empty.

If the TPP tries to access a URL for a resource that is not defined by these specifications (e.g. GET /card-accounts), CBS **will** respond with a 404 (Not Found).

The table below illustrates some examples of expected behaviour:

Situation	Request	Response
TPP attempts to retrieve an account with an AccountId that does not exist	GET /accounts/1001	400 (Bad Request)
TPP attempts to retrieve a resource that is not defined	GET /credit-cards	404 (Not Found)
TPP attempts to retrieve standing orders for an AccountId that does not exist	GET /accounts/1001/standing-orders	400 (Bad Request)
TPP attempts to retrieve standing orders for an AccountId that exists, but does not have any standing orders	GET /accounts/1000/standing-orders	200 OK { "Data": { "StandingOrder": [] }, "Links": { "Self": "/open-banking/v2.0/accounts/1000/standing-orders/" }, "Meta": {} }

403 (Forbidden)

When a TPP tries to access a resource that it does not have permission to access, CBS **will** return a 403 (Forbidden).

The situation could arise when:

- The TPP uses an access token that does not have the appropriate scope to access the requested resource.
- The TPP does not have a consent authorisation for the AccountId
E.g., an attempt to access GET /accounts/2001 or /accounts/2001/transactions when the PSU has not selected AccountId 2001 for authorisation.
- The TPP does not have a consent authorisation with the right permissions to access the requested resource.
E.g., an attempt to access GET /standing-orders when the ReadStandingOrdersBasic permission was not included in the consent authorisation.

- The TPP attempted to access a resource with an Id that it does not have access to.
E.g., an attempt to access GET /account-requests/1001 where an account-request resource with Id 1001 belongs to another TPP.

When the TPP uses an access token that is no longer valid, the situation could potentially be remedied by asking the PSU to re-authenticate. This should be indicated by a 401 (Unauthorized) status code.

429 (Too Many Requests)

When a TPP tries to access a resource too frequently CBS may return a 429 (Too Many Requests).

This situation could arise when:

- The TPP has not implemented caching, it requests transactions for a PSU account, and constantly re-requests the same transactions
- Similarly, for any of the PSU information endpoints

Pre-Conditions

The following pre-conditions must be satisfied in order to use these APIs:

Pre-conditions for TPPs

1. The TPP must have completed on-boarding with CBS and have been issued with a ClientId and Secret.
2. The TPP must have valid eIDAS QWAC or OB WAC network certificate.

Filtering

Limited support for filtering is provided on the **transactions** resource.

Transactions can be filtered based on their Booking Date using the **fromBookingDateTime** and **toBookingDateTime** parameters

The dates **MUST** be specified in ISO8601 format. The date **MUST NOT** include a timezone.

The filter values will be assumed to refer to the same timezone as the timezone in which the booking date for the account is maintained.

The following are treated as valid input:

- non-working days (e.g. a sunday or a bank holiday) or any other days on which no transactions are recorded
- dates that fall outside the range for which transaction information is provided through APIs
- dates that fall outside the range for which a consent authorisation is available.

In the above situations, CBS will return data for the remaining valid period specified by the filter.

Examples for filtering transactions

// All transactions from 1st Jan, 2015

```
GET /open-banking-sandbox/v2.0/accounts/{AccountId}/transactions?fromBookingDateTime=2015-01-01T00:00:00
```

// All transactions in 2016

```
GET /open-banking-sandbox/v2.0/accounts/{AccountId}/transactions?fromBookingDateTime=2016-01-01T00:00:00&toBookingDateTime=2016-12-31T23:59:59
```

// All transactions in a specific account upto 31-Mar-2017

```
GET /open-banking-sandbox/v2.0/accounts/{AccountId}/transactions?toBookingDateTime=2017-03-31T23:59:59
```

Error Response Reasons and Handling

In Compliance to the Regulatory Technical Standards, all error scenarios caused by an exception or unexpected event during the communication sessions between the TPP and the CBS Open Banking Platform, will respond with a Http 400 Error Response reasons and a description of the reason for the failure,

See below for Example:

Response

Delete Account Requests Response

```
HTTP/1.1 400 Bad Request
x-fapi-financial-id: CBSOpenBanking
Content-Type: application/json

{
  "Code": "400",
  "Message": "Bad Request",
  "Errors": [
    {
      "ErrorCode": "1005",
      "Message": "Unfortunately we've been unable to complete your request as the
PaymentId has expired."
    }
  ]
}
```

This error response schema is applicable to All the APIs on the Open Banking Platform.

Pagination

For paginated responses CBS will return 50 records per page.

- If a subsequent page of resource records exists, CBS will provide a link to the next page of resources in the **Links.Next** field of the response. The absence of a next link would indicate that the current page is the last page of results.
- If a previous page of resource records exists, CBS will provide a link to the previous page of resources in the **Links.Prev** field of the response. The absence of a prev link would indicate that the current page is the first page of results.
- The total number of pages will be populated in the **Meta.TotalPages** field
- CBS will provide a link to the first page of results in the **Links.First** field
- CBS will provide a link to the last page of results in the **Links.Last** field
- CBS will provide a self link to the current page of results in the **Links.Self** field

Note: The underlying data-set may change between two subsequent requests. This may result in situations where the same transaction is returned on more than one page.

All available Sandbox Endpoints

Host: <https://resourcema.coventrybuildingsociety.co.uk/>

Scope	Resource	HTTP Operation	Endpoint
openid accounts OR openid payments	Token	POST	POST mga/sps/oauth/oauth20/token

Host: <https://resourcema.coventrybuildingsociety.co.uk/pd/open-banking>

Scope	Resource	HTTP Operation	Endpoint
Accounts	account-requests	POST	POST open-banking-sandbox/v2.0/account-requests
Accounts	account-requests	DELETE	DELETE open-banking-sandbox/v2.0/account-requests/{AccountRequestId}
Accounts	accounts	GET	GET open-banking-sandbox/ v2.0/accounts
Accounts	accounts	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}
Accounts	balances	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/balances
Accounts	beneficiaries	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/beneficiaries
Accounts	direct-debits	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/direct-debits
Accounts	products	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/product
Accounts	standing-orders	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/standing-orders
Accounts	transactions	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/transactions

Accounts	scheduled-payments	GET	GET open-banking-sandbox/v2.0/accounts/{AccountId}/scheduled-payments
Payments	payments	POST	POST open-banking-sandbox/v2.0/payments
Payments	payments	GET	GET open-banking-sandbox/v2.0/payments/{paymentId}
Payments	payment-submissions	POST	POST open-banking-sandbox/v2.0/payment-submissions
Payments	payment-submissions	GET	GET open-banking-sandbox/v2.0/payment-submissions/{PaymentSubmissionId}
Payments	mandates	POST	POST open-banking-sandbox/v2.0/mandates
Payments	mandates	GET	GET open-banking-sandbox/v2.0/mandates/{paymentId}
Payments	mandate-submissions	POST	POST open-banking-sandbox/v2.0/mandate-submissions
Payments	mandate-submissions	GET	GET open-banking-sandbox/v2.0/mandate-submissions/{PaymentSubmissionId}
Payments	funds-confirmation	GET	GET open-banking-sandbox/v2.0/payments/{PaymentId}/funds-confirmation
fundsconfirmations	funds-confirmation-consent	POST	POST /open-banking-sandbox/v2.0/funds-confirmation-consent
fundsconfirmations	funds-confirmation-consent	GET	GET /open-banking-sandbox/v2.0/funds-confirmation-consents/{ConsentId}
fundsconfirmations	funds-confirmation-consent	DELETE	DELETE /open-banking-sandbox/v2.0/funds-confirmation-consents/{ConsentId}
fundsconfirmations	funds-confirmation	POST	POST /open-banking-sandbox/v2.0/funds-confirmation

POST /token

The API allows the TPP using client credentials grant type to request CBS to generate an access token using its client id and client secret. The access token is required for all subsequent Sandbox API calls.

Note:

- The scope (accounts and/or payments and/or fundsconfirmations) must be valid for the TPP
- The clientId used must match the clientId provided to the TPP when they on-boarded with CBS (note in the case of TPPs who are using OB WAC certificates, this will be the same as the Software Statement ClientId).

Example call:

Request Type	Request Example
Accounts Sandbox APIs	grant_type=client_credentials&scope=openid%20accounts&client_id={clientId}&client_secret={client secret}
Payments Sandbox APIs	grant_type=client_credentials&scope=openid%20payments&client_id={clientId}&client_secret={client secret}

Fundsconfirmation Sandbox APIS	grant_type=client_credentials&scope=openid%20fundsconfirmations&client_id={clientId}&client_secret={client secret}
--------------------------------	--

POST open-banking-sandbox/v2.0/account-requests

The API allows the AISP to ask CBS to create a new **account-request** resource.

- This API effectively allows the AISP to send a copy of the consent to CBS to authorise access to account and transaction information.
- CBS creates the account-request resource and responds with a unique AccountRequestId to refer to the resource.
- Prior to calling the API, the AISP must have an access token issued by CBS using a client credentials grant. (POST <https://resourcema.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

Account Request Status

The account-request resource that is created successfully must have one of the following Status code-list enumerations:

Status	Status Description
1 Rejected	The account request has been rejected.
2 AwaitingAuthorisation	The account request is awaiting authorisation.

Mocked Responses

Given	When	Then
An AISP makes a request to retrieve the status of an Account Data Access request	The request is valid containing <ul style="list-style-type: none"> • Valid access token • A customer is yet to Authorize 	HTTP Status Code: 200 Status: AwaitingAuthorisation
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Header	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request - Invalid token	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary - Request

Name	Card-inality	Data Type	Notes
Request			
Request/Data	1..1		
Request/Data/Permissions	1..n	ExternalPermissions1Code Enumeration (see enumeration section below for details)	This is a list of the data clusters being consented by the PSU, and requested for authorisation with CBS.

Request /Data/ExpirationDateTime	0..1	ISODateTime	Specified date and time the permissions will expire. If this is not populated, the permissions will be open ended.
Request/Data/TransactionFromDateTime	0..1	ISODateTime	Specified start date and time for the transaction query period. If this is not populated, the start date will be open ended, and data will be returned from the earliest available transaction.
Request/Data/TransactionToDateTime	0..1	ISODateTime	Specified end date and time for the transaction query period. If this is not populated, the end date will be open ended, and data will be returned to the latest available transaction.
Request/Risk	1..1	OBRisk2	The Risk section is sent by the initiating party to CBS. It is used to specify additional details for risk scoring for Account Info.

Data Dictionary - Response

Name	Card-inality	Data Type	Notes
Response			
Response/Data	1..1		
Response/Data/AccountRequestId	1..1	Max128Text	Unique identification as assigned to identify the account request resource.
Response/Data/Status	0..1	ExternalRequestStatusCode Enumeration (see enumeration section below for details)	Specifies the status of the account request resource. Enumeration <ul style="list-style-type: none"> • Authorised • AwaitingAuthorisation • Rejected • Revoked
Response/Data/CreationDateTime	1..1	ISODateTime	Date and time at which the resource was created.
Response/Data/Permissions	1..n	ExternalPermissions1Code Enumeration (see enumeration section below for details)	This is a list of the data clusters being consented by the PSU, and requested for authorisation with CBS.
Response/Data/ExpirationDateTime	0..1	ISODateTime	Specified date and time the permissions will expire. If this is not populated, the permissions will be open ended.
Response/Data/TransactionFromDateTime	0..1	ISODateTime	Specified start date and time for the transaction query period.

			If this is not populated, the start date will be open ended, and data will be returned from the earliest available transaction.
Response/Data/TransactionToDateTime	0..1	ISODateTime	Specified end date and time for the transaction query period. If this is not populated, the end date will be open ended, and data will be returned to the latest available transaction.
Response/Risk	1..1	OBRisk2	The Risk section is sent by the initiating party to CBS. It is used to specify additional details for risk scoring for Account Info.

GET open-banking-sandbox/v2.0/account-requests/{AccountRequestId}

An AISP can optionally retrieve an **account-request** resource that they have created to check its status.

Prior to calling the API, the AISP must have an access token issued by CBS using a client credentials grant. ([POST https://resource.ma.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token](https://resource.ma.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token))

Account Request Status

The Sandbox APIs do not include PSU authorisation, to assist TPPs with testing these flows specific AccountRequestIds can be used as detailed in the Mocked Responses section.

The available Status code-list enumerations for the account-request resource are:

Status	Status Description
1 Rejected	The account request has been rejected.
2 AwaitingAuthorisation	The account request is awaiting authorisation.
3 Authorised	The account request has been successfully authorised.
4 Revoked	The account request has been revoked.

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token Valid Request - The CBS customer has Authorized the Request 	HTTP Status Code: 200 Status: Authorised

	<ul style="list-style-type: none"> AccountRequestId = 6495080e-cc91-4171-a560-4ec9522740ee 	
A request	<p>The request is valid containing</p> <ul style="list-style-type: none"> Valid access token Valid Request - The CBS customer is yet to Authorize the Request Account Request ID = d02d3f54-234a-45a9-a097-1ddb1a8d828e 	HTTP Status Code: 200 Status: Pending
A request	<p>The request is valid containing</p> <ul style="list-style-type: none"> Valid access token Valid Request - The CBS Customer Rejects the Request during the Authentication process Account Request ID = dd5447f7-1898-4855-86a6-fd50d37eb42f 	HTTP Status Code: 200 Status: Rejected
A request	<p>The request is valid containing</p> <ul style="list-style-type: none"> Valid access token Valid Request - The account request has been revoked via the ASPSP interface Account Request ID = 397a699a-0c6a-486c-b11f-7249cebfe344 	HTTP Status Code: 200 Status: Revoked
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary – Response

Data Item	Data Type	Occurence	Comments
Data	object	1	
AccountRequestId	string	1	Unique identification as assigned to identify the account request resource.
Status	enumeration	1	Specifies the status of the account request resource.

CreationDateTime	dateTime	1	Date and time at which the resource was created.
Permissions	enumeration	1..n	Specifies the Open Banking account request types. This is a list of the data clusters being consented by the PSU, and requested for authorisation with the ASPSP.
ExpirationDateTime	dateTime	1	Specified date and time the permissions will expire.
TransactionFromDate Time	dateTime	0..1	Specified start date and time for the transaction query period. If this is not populated, the start date will be open ended, and data will be returned from the earliest available transaction.
TransactionToDateTime	dateTime	0..1	Specified end date and time for the transaction query period. If this is not populated, the end date will be open ended, and data will be returned to the latest available transaction.
Risk	object	1	The Risk section is sent by the initiating party to CBS. It is used to specify additional details for risk scoring for Account Info. For Account Information APIs no Risk data has currently been identified/specified.
Links	object	1	The Links section is mandatory and will always contain URIs to related resources.
Self	string	1	Link to the current page
Meta	object	1	The Meta section is mandatory, but can be empty
TotalPages	integer	0..1	The total number of pages

DELETE open-banking-sandbox/v2.0/account-requests/{AccountRequestId}

If the PSU revokes consent to data access with the AISP - the AISP should delete the account-request resource.

- This is done by making a call to DELETE the **account-request** resource.
- Prior to calling the API, the AISP must have an access token issued by CBS using a client credentials grant.

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> • Valid access token • AccountRequestId = 6495080e-cc91-4171-a560-4ec9522740ee 	HTTP Status Code: 204
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400

An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500

GET open-banking-sandbox/v2.0/accounts

The first step for an AISP after an account-request is authorised - is to call the GET /accounts endpoint.

This will give the full list of accounts (the AccountId(s)) that the PSU has authorised the AISP to access. The AccountId(s) returned can then be used to retrieve other resources for an account.

Note: CBS current implementation is that only one account is associated to a AccountRequestId, this means that for a PSU to grant TPPs access to multiple accounts they would need to generate a new AccountRequestId for each account.

Prior to calling the API, the AISP must have an access token issued by CBS using a client credentials grant

Mocked Response

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token 	HTTP Status Code: 200
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary - Response

Name	Cardinality	Data Type	Permission Code Restrictions	Notes
Response			ReadAccountsBasic ReadAccountsDetail	
Response/Data	1..1			
Response/Data/Account	0..n		ReadAccountsBasic ReadAccountsDetail	
Response/Data/Account/AccountId	1..1	Max40Text	ReadAccountsBasic ReadAccountsDetail	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/Account/Currency	1..1	^[A-Z]{3,3}\$	ReadAccountsBasic ReadAccountsDetail	Identification of the currency in which the account is held. Usage: Currency should only be used in case one and the same account number covers several currencies and the initiating party needs to identify which currency needs to be used for settlement on the account.
Response/Data/Account/Account	0..1		ReadAccountsDetail	Provides the details to identify an account.
Response/Data/Account/Account/SchemeName	1..1	ExternalAccountIdentification2Code Enumeration (see enumeration section below for details)	ReadAccountsDetail	Name of the identification scheme, in a coded form as published in an external list.
Response/Data/Account/Account/Identification	1..1	Max34Text	ReadAccountsDetail	Identification assigned by an institution to identify an account. This identification is known by the account owner.

GET open-banking-sandbox/v2.0/accounts/{AccountId}

The resource that represents the account to which credit and debit entries are made.

Each account resource will have a unique and immutable AccountId. For the Sandbox environment this will always return a mocked AccountId which is provided in the information below.

An AISP can retrieve the account information resources for the AccountId (which is retrieved in the call to GET /accounts).

The AISP will use an access token associated with the PSU issued through an authorization code grant.

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary - Response

Name	Card-inality	Data Type	Permission Code Restrictions	Notes
Response			ReadAccountsBasic ReadAccountsDetail	
Response/Data	1..1			
Response/Data/Account	0..n		ReadAccountsBasic ReadAccountsDetail	
Response/Data/Account/AccountId	1..1	Max40Text	ReadAccountsBasic ReadAccountsDetail	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/Account/Currency	1..1	^[A-Z]{3,3}\$	ReadAccountsBasic ReadAccountsDetail	Identification of the currency in which the account is held. Usage: Currency should only be used in case one and the same account number covers several currencies and the initiating party needs to identify which currency needs to be used for settlement on the account.
Response/Data/Account/Account	0..1		ReadAccountsDetail	Provides the details to identify an account.
Response/Data/Account/Account/SchemeName	1..1	ExternalAccountIdentification2Code	ReadAccountsDetail	Name of the identification scheme, in a coded form as published in an external list.

			Enumeration (see enumeration section below for details)	
Response/Data/Account/Account/Identification	1..1	Max34Text	ReadAccountsDetail	Identification assigned by an institution to identify an account. This identification is known by the account owner.

GET open-banking-sandbox/v2.0/accounts/{AccountId}/balances

A resource that contains balance information for an account.

- A DateTime element has been used instead of a complex choice element of Date and DateTime. Where time elements do not exist - the time portion of the DateTime element will be defaulted to 00:00:00+00:00

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary - Response

Name	Card-inality	Data Type	Permission Code Required	Notes
Response			ReadBalances	
Response/Data	1..1		ReadBalances	

Response/Data/Balance	1..n		ReadBalances	Set of elements used to define the balance details.
Response/Data/Balance/AccountId	1..1	Max40Text	ReadBalances	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/Balance/Amount	1..1	^d{1,13}\.d{1,5}\$	ReadBalances	Amount of money of the cash balance.
Response/Data/Balance/Amount/Currency	1..1	^[A-Z]{3,3}\$	ReadBalances	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
Response/Data/Balance/CreditDebitIndicator	1..1	CreditDebitCode Enumeration (see enumeration section below for details)	ReadBalances	Indicates whether the balance is a credit or a debit balance. Usage: A zero balance is considered to be a credit balance.
Response/Data/Balance/Type	1..1	BalanceTypeCode Enumeration (see enumeration section below for details)	ReadBalances	Balance type, in a coded form.
Response/Data/Balance/DateTime	1..1	ISODateTime	ReadBalances	Indicates the date (and time) of the balance.

Refer to data payload section for [Balances](#) request and response details.

GET open-banking-sandbox/v2.0/accounts/{AccountId}/beneficiaries

A resource that contains a set of elements that describe the list of trusted beneficiaries linked to a specific account (AccountId). An account can have no trusted beneficiaries set up, or may have multiple beneficiaries set up.

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401

An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary – Response

Name	Cardinality	Data Type	Permission Code Required	Notes
Response			ReadBeneficiariesBasic ReadBeneficiariesDetail	
Response/Data	1..1		ReadBeneficiariesBasic ReadBeneficiariesDetail	
Response/Data/Beneficiary	0..n		ReadBeneficiariesBasic ReadBeneficiariesDetail	
Response/Data/Beneficiary/AccountId	0..1	Max40Text	ReadBeneficiariesBasic ReadBeneficiariesDetail	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/Beneficiary/Reference	0..1	Max35Text	ReadBeneficiariesBasic ReadBeneficiariesDetail	<p>Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction.</p> <p>Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.</p> <p>If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification.</p>
Response/Data/Beneficiary/CreditorAccount	0..1		ReadBeneficiariesDetail	Provides the details to identify the beneficiary account.
Response/Data/Beneficiary/CreditorAccount/Scheme Name	1..1	ExternalAccountIdentification2Code Enumeration (see enumeration section below for details)	ReadBeneficiariesDetail	Name of the identification scheme, in a coded form as published in an external list.
	1..1	Max35Text	ReadBeneficiariesDetail	Identification assigned by an institution to identify an account. This

Response/Data/Beneficiary/CreditorAccount/Identification				identification is known by the account owner.
Response/Data/Beneficiary/CreditorAccount/Name	0..1	Max70Text	ReadBeneficiariesDetail	<p>Name of the account, as assigned by the account servicing institution, in agreement with the account owner in order to provide an additional means of identification of the account.</p> <p>Usage: The account name is different from the account owner name. The account name is used in certain user communities to provide a means of identifying the account, in addition to the account owner's identity and the account number.</p>

GET open-banking-sandbox/v2.0/accounts/{AccountId}/direct-debits

A resource that contains a set of elements that describes the list of direct-debits that have been set up on a specific account (AccountId).

An account (AccountId) can have no direct debits set up, or may have multiple direct debits se

- A DateTime element has been used - so that there is consistency across all API endpoints using dates. Where time elements do not exist - the time portion of the DateTime element will be defaulted to 00:00:00+00:00

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> • Valid access token • Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary - Response

Name	Cardinality	Data Type	Permission Code Required	Notes
Response			ReadDirectDebits	
Response/Data	1..1		ReadDirectDebits	
Response/Data/DirectDebit	0..n		ReadDirectDebits	Account to or from which a cash entry is made.
Response/Data/DirectDebit/AccountId	1..1	Max40Text	ReadDirectDebits	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/DirectDebit/MandateIdentification	1..1	Max35Text	ReadDirectDebits	Direct Debit reference.
Response/Data/DirectDebit/DirectDebitStatusCode	0..1	ExternalDirectDebitStatus1Code Enumeration (see enumeration section below for details)	ReadDirectDebits	Specifies the status of the direct debit in code form.
Response/Data/DirectDebit/Name	1..1	Max70Text	ReadDirectDebits	Name of Service User.
Response/Data/DirectDebit/PreviousPaymentDateTime	0..1	ISODateTime	ReadDirectDebits	Date of most recent direct debit collection.
Response/Data/DirectDebit/PreviousPaymentAmount	0..1	^\d{1,13}\.\d{1,5}\$	ReadDirectDebits	The amount of the most recent direct debit collection.
Response/Data/DirectDebit/PreviousPaymentAmount/Currency	1..1	^[A-Z]{3,3}\$	ReadDirectDebits	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".

GET open-banking-sandbox/v2.0/accounts/{AccountId}/standing-orders

A resource that contains a set of elements that describe the list of standing-orders that have been set up on a specific account (AccountId).

An account (AccountId) can have no standing orders set up, or may have multiple standing orders set up.

- A DateTime element has been used - so that there is consistency across all API endpoints using dates. Where time elements do not exist - the time portion of the DateTime element will be defaulted to 00:00:00+00:00

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary- Response

Name	Cardinality	Data Type	Permission Code Required	Notes
Response			ReadStandingOrdersBasic ReadStandingOrdersDetail	
Response/Data	1..1		ReadStandingOrdersBasic ReadStandingOrdersDetail	
Response/Data/Standing Order	0..n		ReadStandingOrdersBasic ReadStandingOrdersDetail	Account to or from which a cash entry is made.
Response/Data/Standing Order/AccountId	1..1	Max40Text	ReadStandingOrdersBasic ReadStandingOrdersDetail	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/Standing Order/Frequency	1..1	Max35Text ^(EvryDay)\$ ^(Evry WorkgDay)\$ ^(Intrvl WkDay:0[1-9]:0[1-7])\$ ^(WkInMnthDay:0[1-5]:0[1-7])\$ ^(IntrvlMnthDay:(0[1-6] 12 24):(-0[1-5] 0[1-9] 12][0-9] 3[01]))\$ ^(QtrDay:(ENGLISH SCOTTISH RECEIVED))\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	Individual Definitions: EvryDay - Every day EvryWorkgDay - Every working day IntrvlWkDay - An interval specified in weeks (01 to 09), and the day within the week (01 to 07)

WkInMnthDay - A monthly interval, specifying the week of the month (01 to 05) and day within the week (01 to 07)

IntrvlMnthDay - An interval specified in months (between 01 to 06, 12, 24), specifying the day within the month (-5 to -1, 1 to 31)

QtrDay - Quarterly (either ENGLISH, SCOTTISH, or RECEIVED)

Individual Patterns:

EvryDay (ScheduleCode)

EvryWorkgDay (ScheduleCode)

IntrvlWkDay:IntervalInWeeks:DayInWeek (ScheduleCode + IntervalInWeeks + DayInWeek)

WkInMnthDay:WeekInMonth:DayInWeek (ScheduleCode + WeekInMonth + DayInWeek)

IntrvlMnthDay:IntervalInMonths:DayInMonth (ScheduleCode + IntervalInMonths + DayInMonth)

QtrDay: + either (ENGLISH, SCOTTISH or RECEIVED) ScheduleCode + QuarterDay

The regular expression for this element combines five smaller versions for each permitted pattern. To aid legibility - the components are presented individually here:

EvryDay

EvryWorkgDay

IntrvlWkDay:0[1-9]:0[1-7]

				WkInMnthDay:0[1-5]:0[1-7] IntrvlMnthDay:(0[1-6] 12 24):(-0[1-5] 0[1-9] 12 0-9 3 01) QtrDay:(ENGLISH SCOTTISH RECEIVED)
Response/Data/Standing Order/Reference	0..1	Max35Text	ReadStandingOrdersBasic ReadStandingOrdersDetail	Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money. If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification.
Response/Data/Standing Order/ FirstPaymentDateTime	0..1	ISODateTime	ReadStandingOrdersBasic ReadStandingOrdersDetail	The date on which the first payment for a Standing Order schedule will be made.
Response/Data/Standing Order/FirstPaymentAmount	0..1	^d{1,13}\.d{1,5}\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	The amount of the first Standing Order
Response/Data/Standing Order/FirstPaymentAmount/Currency	1..1	^[A-Z]{3,3}\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
Response/Data/Standing Order/NextPaymentDateTime	1..1	ISODateTime	ReadStandingOrdersBasic ReadStandingOrdersDetail	The date on which the next payment for a Standing Order schedule will be made.
Response/Data/Standing Order/NextPaymentAmount	1..1	^d{1,13}\.d{1,5}\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	The amount of the next Standing Order

Response/Data/Standing Order/NextPaymentAmount/Currency	1..1	^[A-Z]{3,3}\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
Response/Data/Standing Order/FinalPaymentDate Time	0..1	ISODateTime	ReadStandingOrdersBasic ReadStandingOrdersDetail	The date on which the final payment for a Standing Order schedule will be made.
Response/Data/Standing Order/FinalPaymentAmount	0..1	^d{1,13}\.ld{1,5}\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	The amount of the final Standing Order
Response/Data/Standing Order/ FinalPaymentAmount/Currency	1..1	^[A-Z]{3,3}\$	ReadStandingOrdersBasic ReadStandingOrdersDetail	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
Response/Data/Standing Order/CreditorAccount	0..1		ReadStandingOrdersDetail	Provides the details to identify the beneficiary account.
Response/Data/Standing Order/CreditorAccount/SchemeName	1..1	ExternalAccountIdentification2Code Enumeration (see enumeration section for details)	ReadStandingOrdersDetail	Name of the identification scheme, in a coded form as published in an external list.
Response/Data/Standing Order/CreditorAccount/Identification	1..1	Max34Text	ReadStandingOrdersDetail	Beneficiary account identification.
Response/Data/Standing Order/CreditorAccount/Name	0..1	Max70Text	ReadStandingOrdersDetail	Name of the account, as assigned by the account servicing institution, in agreement with the account owner in order to provide an additional means of identification of the account. Usage: The account name is different from the account owner name. The account name is used in certain user communities to provide a means of identifying the account, in addition to the account owner's identity and the account number.

Refer to data payload section for [Standing Orders](#) request and response details.

GET open-banking-sandbox/v2.0/accounts/{AccountId}/transactions

A resource that describes a posting to an account that results in an increase or decrease to a balance.

For a specific date range - an account (AccountId) can have no transactions booked, or can have multiple transactions booked.

- A DateTime element has been used instead of a complex choice element of Date and DateTime. Where time elements do not exist - the time portion of the DateTime element will be defaulted to 00:00:00+00:00
- The BookingDateTime is the date the transaction is booked (or posted) and becomes immutable - which is not the date the transaction took place.

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> • Valid access token • Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary - Response

Name	Card-inality	Data Type	Permission Code Required	Notes
Response			ReadTransactionsBasic ReadTransactionsDetail	
Response/Data	1..1		ReadTransactionsBasic ReadTransactionsDetail	
Response/Data/Transaction	0..n		ReadTransactionsBasic ReadTransactionsDetail	Provides further details on an entry in the report.
Response/Data/Transaction/AccountId	1..1	Max40Text	ReadTransactionsBasic ReadTransactionsDetail	A unique and immutable identifier used to identify the account resource. This identifier has no

				meaning to the account owner.
Response/Data/Transaction/TransactionReference	0..1	Max35Text	ReadTransactionsBasic ReadTransactionsDetail	Unique reference for the transaction. This reference is optionally populated, and may as an example be the FPID in the Faster Payments context.
Response/Data/Transaction/Amount	1..1	^\d{1,13}\.\d{1,5}\$	ReadTransactionsBasic ReadTransactionsDetail	Amount of money in the cash transaction entry.
Response/Data/Transaction/Amount/Currency	1..1	^[A-Z]{3,3}\$	ReadTransactionsBasic ReadTransactionsDetail	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
Response/Data/Transaction/CreditDebitIndicator	1..1	CreditDebitCode Enumeration (see enumeration section for details)	ReadTransactionsBasic ReadTransactionsDetail	Indicates whether the transaction is a credit or a debit entry.
Response/Data/Transaction/Status	1..1	TransactionEntryCode Enumeration (see enumeration section for details)	ReadTransactionsBasic ReadTransactionsDetail	Status of a transaction entry on the books of the account servicer.
Response/Data/Transaction/BookingDateTime	1..1	ISODatetime	ReadTransactionsBasic ReadTransactionsDetail	Date and time when a transaction entry is posted to an account on the account servicer's books. Usage: Booking date is the expected booking date, unless the status is booked, in which case it is the actual booking date.
Response/Data/Transaction/TransactionInformation	0..1	Max500Text	ReadTransactionsDetail	Further details of the transaction. This is the transaction narrative, which is unstructured text.
Response/Data/Transaction/Balance	0..1		ReadTransactionsDetail	Set of elements used to define the balance as a

				numerical representation of the net increases and decreases in an account after a transaction entry is applied to the account.
Response/Data/Transaction/Balance/Amount	1..1	^\d{1,13}\.\d{1,5}\$	ReadTransactionsDetail	Amount of money of the cash balance after a transaction entry is applied to the account..
Response/Data/Transaction/Balance/Amount/Currency	1..1	^[A-Z]{3,3}\$	ReadTransactionsDetail	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
Response/Data/Transaction/Balance/CreditDebitIndicator	1..1	CreditDebit Enumeration (see enumeration section for details)	ReadTransactionsDetail	Indicates whether the balance is a credit or a debit balance. Usage: A zero balance is considered to be a credit balance.
Response/Data/Transaction/Balance/Type	1..1	BalanceTypeCode Enumeration (see enumeration section for details)	ReadTransactionsDetail	Balance type, in a coded form.

GET open-banking-sandbox/v2.0/accounts/{AccountId}/product

A resource that contains a set of elements that describe the product details specific to the account (AccountId).

An account (AccountId) can only have a single product.

- An AccountId will only have **one product** - so the single account endpoint will return only one product (for /accounts/{AccountId}/product)

Mocked Responses

Given	When	Then
A request	The request is valid containing	HTTP Status Code: 200

	<ul style="list-style-type: none"> Valid access token Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary - Response

Name	Cardinality	Data Type	Permission Code Required	Notes
Response			ReadProducts	
Response/Data	1..1		ReadProducts	
Response/Data/Product	0..1		ReadProducts	
Response/Data/Product/AccountId	1..1	Max40Text	ReadProducts	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
Response/Data/Product/ProductIdentifier	1..1	Max40Text	ReadProducts	Identifier within the parent organisation for the product. Must be unique in the organisation.
Response/Data/Product/ProductType	1..1	ExternalProductType1Code Enumeration (see enumeration section below for details)	ReadProducts	Descriptive code for the product category.

GET open-banking-sandbox/v2.0/accounts/{AccountId}/scheduled-payments

A resource that contains a set of elements that describe the scheduled payments specific to the account (AccountId).

An account (AccountId) can numerous scheduled payments

- An AccountId can have **numerous scheduled payments** - so the single account endpoint will return all the scheduled payments (for /accounts/{AccountId}/scheduled-payments)

Mocked Responses

Given	When	Then
A request	The request is valid containing <ul style="list-style-type: none"> Valid access token Valid Request = 3701d604-0dce-4509-8aa6-d7898fc53d5e 	HTTP Status Code: 200
An invalid request	Null Account Request ID	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (alpha / numeric characters)	HTTP Status Code: 400
An invalid request	Invalid Account Request ID (special characters)	HTTP Status Code: 400
An invalid request	Invalid Header within the Request	HTTP Status Code: 400
An invalid request	Null Headers within the Request	HTTP Status Code: 400
An invalid request	Invalid Certificate within the Request	HTTP Status Code: 401
An invalid request	Payment token instead of an Accounts Token within the Request	HTTP Status Code: 403
An invalid request	Invalid Method within the Request	HTTP Status Code: 405
An invalid request	Server is disabled	HTTP Status Code: 500
An invalid request	Rate limit reached (> 500)	HTTP Status Code: 429

Data Dictionary- Request

Not required

Data Dictionary- Response

Data Item	Data Type	Occurrence	Notes
Data	object	1..1	
ScheduledPayment	object	0..n	
AccountId	string	1..1	A unique and immutable identifier used to identify the account resource. This identifier has no meaning to the account owner.
ScheduledPaymentDate	date time	1..1	The date on which the scheduled payment will be made.
ScheduledType	enum	1..1	Specifies the scheduled payment date type requested
Reference	string	0..1	Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to

				enable reconciliation by the creditor upon receipt of the amount of money. If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification.
InstructedAmount	object		1..1	
Amount	string		1..1	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party. Usage: This amount has to be transported unchanged through the transaction chain.
Currency	enum		1..1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
CreditorAccount	object		0..1	Provides the details to identify the beneficiary account.
SchemeName	enum		1..1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string		1..1	Beneficiary account identification.
Name	string		0..1	Name of the account, as assigned by the account servicing institution, in agreement with the account owner in order to provide an additional means of identification of the account. Usage: The account name is different from the account owner name. The account name is used in certain user communities to provide a means of identifying the account, in addition to the account owner's identity and the account number.

POST - /open-banking-extensions-sandbox/v2.0/mandates

The API allows the PISP to ask CBS to create a new **mandate payment** resource.

- This API effectively allows the PISP to send a copy of the consent to CBS to authorise THE SETTING UP OF A Mandate payment

- CBS creates the mandate payment resource and responds with a unique PaymentId to refer to the resource.
- Prior to calling the API, the PISP must have an access token issued by CBS using a client credentials grant. (POST <https://resource.ma.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

Mandate Payment Status

The Mandate payment resource that is created successfully must have one of the following Status code-list enumerations:

Status	Status Description
Pending	Payment initiation or Individual transaction included in the payment initiation is pending. Further checks and status update will be performed
Rejected	Payment initiation or Individual transaction included in the payment initiation has been rejected
AcceptedTechnicalValidation	Authentication and Syntactical and Semantic validation are successful

Mocked Responses

Given	When	Then
An PISP makes a request to create a new Mandate payment resource	The request is valid containing <ul style="list-style-type: none"> • Valid access token 	HTTP Status Code: 200 Status:AcceptedTechnicalValidation
Valid Request	Payment type Future Dated • Valid access token	HTTP Status Code: 201
Valid Request	Payment type Regular • Valid access token	HTTP Status Code: 201
Invalid Request	Invalid Future Payment Date	HTTP Status Code: 400
Invalid Request	Invalid Regular Payment Date	HTTP Status Code: 400
Invalid Request	Invalid Payment Amount = 0.00	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Invalid request schema	HTTP Status Code: 500

Data Dictionary - Request

Name	Data Type	Occurrence	Notes
Data	complex type	1	
PaymentType	enumeration	1	Categorises the different types of payment methods within CBS for processing purposes.
Initiation	complex type	1	
InstructionIdentification	string	1	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction. Usage: the instruction identification is a point to point reference that can be used between the instructing party and

			the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.
EndToEndIdentification	string	0..1	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain. Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.
FuturePaymentDateTime	ISODateTime	1	Date at which the initiating party requests the clearing agent to process the payment. Usage: This is the date on which the debtor's account is to be debited.
ValidFromDateTime	ISODateTime	1	The date on which the first payment for a Standing Order schedule will be made.
ValidToDateTime	ISODateTime	0..1	The date on which the final payment for a Standing Order schedule will be made.
Frequency	enumeration	1	The number of times something happens within a particular period.
InstructedAmount	complex type	1	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.
Usage: This amount has to be transported unchanged through the transaction chain.			
Amount	string	1	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Currency	string	1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".

DebitorAccount	complex type	0..1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account. OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.

Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
RemittanceInformation	complex type	0..1	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts' receivable system.
Reference	string	0..1	Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.
			If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification. OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.
Unstructured	string	0..1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as

			defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
StreetName	string	0..1	Name of a street or thoroughfare
BuildingNumber	string	0..1	Number that identifies the position of a building on a street.
PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

Data Dictionary - Response

Name	Card-inality	Data Type	Notes
Data	complex type	1	
PaymentType	enumeration	1	Categorises the different types of payment methods within CBS for processing purposes.
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
Status	enumeration	1	Determines the status of a payment consent as defined by Open Banking.
CreationDateTime	datetime	1	Date and time at which the resource was created.
Initiation	complex type	1	
InstructionIdentification	string	1	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction. Usage: the instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the

			individual instruction. It can be included in several messages related to the instruction.
EndToEndIdentification	string	0..1	<p>Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.</p> <p>Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.</p>
FuturePaymentDateTime	ISODatetime	0..1	Date at which the initiating party requests the clearing agent to process the payment. Usage: This is the date on which the debtor's account is to be debited.
ValidFromDateTime	ISODatetime	0..1	The date on which the first payment for a Standing Order schedule will be made.
ValidToDateTime	ISODatetime	0..1	The date on which the final payment for a Standing Order schedule will be made.
Frequency	enumeration	0..1	The number of times something happens within a particular period.
InstructedAmount	complex type	1	<p>Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.</p> <p>Usage: This amount has to be transported unchanged through the transaction chain.</p>
Amount	string	1	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Currency	string	1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".

DebitorAccount	complex type	0..1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account. OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing

			institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
RemittanceInformation	complex type	0..1	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts' receivable system.
Reference	string	0..1	Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money. If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification. OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.
Unstructured	string	0..1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.

DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
Street Name	string	0..1	Name of a street or thoroughfare
BuildingNumber	string	0..1	Number that identifies the position of a building on a street.
PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

GET - /open-banking-extensions-sandbox/v2.0/mandates/{PaymentId}

A PISP can optionally retrieve a **payment** resource that they have created to check its status. Prior to calling the API, the PISP must have an access token issued by CBS using a client credentials grant. ([POST https://resource.ma.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token](https://resource.ma.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token))

Mandate PaymentId Status

The Sandbox APIs do not include PSU authorisation, to assist TPPs with testing these flows specific PaymentIDs can be used as detailed in the Mocked Responses section.

The available Status code-list enumerations for the mandates resource are:

Status	Status Description
Pending	Payment initiation or Individual transaction included in the payment initiation is pending. Further checks and status update will be performed
Rejected	Payment initiation or Individual transaction included in the payment initiation has been rejected
AcceptedTechnicalValidation	Authentication and Syntactical and Semantic validation are successful
AcceptedCustomerProfile	Preceding check of technical validation was successful. Customer profile check was also successful.

Mocked Responses

Given	When	Then
Valid Request	Mandates Payment ID for Regular Mandate for AcceptedTechnicalValidation response <ul style="list-style-type: none"> Valid access token PaymentID = 219e279c-a1dd-4f80-9d52-822853dff653 	HTTP Status Code: 200 Status: AcceptedTechnicalValidation
Valid Request	Mandates Payment ID for Regular Mandate for AcceptedCustomerProfile response <ul style="list-style-type: none"> Valid access token PaymentID = 20dc7be4-5c28-48a9-a1eb-670e76e741f5 	HTTP Status Code: 200 Status: AcceptedCustomerProfile
Valid Request	Mandates Payment ID for Regular Mandate for Pending response <ul style="list-style-type: none"> Valid access token PaymentID = 4876eedd-575f-4208-a821-2ebfea73042c 	HTTP Status Code: 200 Status: Pending
Valid Request	Mandates Payment ID for Regular Mandate for Rejected response <ul style="list-style-type: none"> Valid access token PaymentID = 6f02ee22-19cb-4dd3-b900-21639472b97d 	HTTP Status Code: 200 Status: Rejected
Valid Request	Mandates Payment ID for Future Dated Mandate for AcceptedTechnicalValidation response <ul style="list-style-type: none"> Valid access token PaymentID = 172cf16c-5819-40cf-aa3f-fae1b16b8f4 	HTTP Status Code: 200 Status: AcceptedTechnicalValidation
Valid Request	Mandates Payment ID for Future Dated Mandate for AcceptedCustomerProfile response <ul style="list-style-type: none"> Valid access token PaymentID = fec0ef90-2c27-4057-9730-d4121915d943 	HTTP Status Code: 200 Status: AcceptedCustomerProfile
Valid Request	Mandates Payment ID for Future Dated Mandate for Pending response <ul style="list-style-type: none"> Valid access token PaymentID = 75e48600-7236-4e97-95c2-25f24a08f478 	HTTP Status Code: 200 Status: Pending
Valid Request	Mandates Payment ID for Future Dated Mandate for Rejected response <ul style="list-style-type: none"> Valid access token PaymentID = 8408d906-8751-4fd7-9fb2-0b99cb917e1c 	HTTP Status Code: 200 Status: Rejected
Invalid Request	Null Payment ID	HTTP Status Code: 400
Invalid Request	Invalid Payment ID	HTTP Status Code: 400
Invalid Request	Null Header	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403

Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary - Request

Not required

Data Dictionary – Response

Data Item	Data Type	Cardinality	Comments
Data	complex type	1	
PaymentType	enumeration	1	<p>Categorises the different types of payment methods within CBS for processing purposes.</p> <ul style="list-style-type: none"> • Regular • FutureDated
PaymentId	string	1	<p>Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows:</p> <ul style="list-style-type: none"> - Payment - Payment ID - Account - Account Request ID
Status	enumeration	1	<p>Determines the status of a payment consent as defined by Open Banking:</p> <ul style="list-style-type: none"> • AcceptedCustomerProfile • AcceptedTechnicalValidation • Pending • Rejected
CreationDateTime	datetime	1	Date and time at which the resource was created.
Initiation	complex type	1	
InstructionIdentification	string	1	<p>Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction.</p> <p>Usage: the instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.</p>

EndToEndIdentification	string	0..1	<p>Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.</p> <p>Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.</p>
FuturePaymentDateTime	ISODateTime	0..1	<p>Date at which the initiating party requests the clearing agent to process the payment.</p> <p>Usage: This is the date on which the debtor's account is to be debited.</p>
ValidFromDateTime	ISODateTime	0..1	<p>The date on which the first payment for a Standing Order schedule will be made.</p>
ValidToDateTime	ISODateTime	0..1	<p>The date on which the final payment for a Standing Order schedule will be made.</p>
Frequency	enumeration	0..1	<p>The number of times something happens within a particular period.</p>
InstructedAmount	complex type	1	<p>Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.</p> <p>Usage: This amount has to be transported unchanged through the transaction chain.</p>
Amount	string	1	<p>A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.</p>
Currency	string	1	<p>A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".</p>
DebitorAccount	complex type	0..1	<p>Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.</p>
SchemeName	enumeration	1	<p>Name of the identification scheme, in a coded form as published in an external list.</p>

Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account. OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
RemittanceInformation	complex type	0..1	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle,

			such as commercial invoices in an accounts' receivable system.
Reference	string	0..1	<p>Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction.</p> <p>Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.</p> <p>If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification.</p> <p>OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.</p>
Unstructured	string	0..1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
StreetName	string	0..1	Name of a street or thoroughfare
BuildingNumber	string	0..1	Number that identifies the position of a building on a street.

PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

POST - /open-banking-sandbox/v2.0/payments

The API allows the PISP to ask CBS to create a new **single or immediate payment** resource.

- This API effectively allows the PISP to send a copy of the consent to CBS to authorise the setting up of a single or immediate payment
- CBS creates the payment resource and responds with a unique PaymentId to refer to the resource.
- Prior to calling the API, the PISP must have an access token issued by CBS using a client credentials grant.
(POST <https://resourcema.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

Payment Status

The Mandate payment resource that is created successfully must have one of the following Status code-list enumerations:

Status	Status Description
Pending	Payment initiation or Individual transaction included in the payment initiation is pending. Further checks and status update will be performed
Rejected	Payment initiation or Individual transaction included in the payment initiation has been rejected
AcceptedTechnicalValidation	Authentication and Syntactical and Semantic validation are successful

Mocked Responses

Given	When	Then
Valid Request	The request is valid containing · Valid access token	HTTP Status Code: 201
Invalid Request	Invalid Amount	HTTP Status Code: 400
Invalid Request	Null Header	HTTP Status Code: 400
Invalid Request	Invalid Header	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429

Invalid Request	Internal server error	HTTP Status Code: 500
Invalid Request	Invalid request schema	HTTP Status Code: 500

Data Item	Data Type	Cardinality	Comments
Data	Object	1...1	
Initiation	Object	1...1	The Initiation payload is sent by the initiating party to the ASPSP. It is used to request movement of funds from the debtor account to a creditor.
InstructionIdentification	string	1...1	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction.
EndToEndIdentification	String	1...1	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.
InstructedAmount	Object	1...1	
Amount	String	1...1	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.
Currency	String	1...1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
CreditorAccount	Object	1...1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	Enumeration	1...1	Name of the identification scheme, in a coded form as published in an external list.
Identification	String	1...1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	String	1...1	Name of the account, as assigned by the account servicing institution, in agreement with the

			<p>account owner in order to provide an additional means of identification of the account.</p> <p>Usage: The account name is different from the account owner name. The account name is used in certain user communities to provide a means of identifying the account, in addition to the account owner's identity and the account number.</p> <p>ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.</p>
SecondaryIdentification	String	0...1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
DebtorAgent	Object	0...1	Financial institution servicing an account for the debtor.
SchemeName	String	1...1	Name of the identification scheme, in a coded form as published in an external list.
Identification	String	1...1	Unique and unambiguous identification of a person.
DebtorAccount	Object	0...1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	Enumeration	1...1	Name of the identification scheme, in a coded form as published in an external list.
Identification	String	1...1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	String	1...1	Name of the account, as assigned by the account servicing institution, in agreement with the account owner in order to provide an additional means of identification of the account.
SecondaryIdentification	String	0...1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a

			sort code and account number combination).
CreditorAgent	Object	0...1	Financial institution servicing an account for the creditor.
SchemeName	Enumeration	1...1	Name of the identification scheme, in a coded form as published in an external list.
Identification	String	1...1	Unique and unambiguous identification of a person.
RemittanceInformation	Object	0...1	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts' receivable system.
Reference	string	1...1	Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money. If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification. OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.
Unstructured	string	1...1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	Object	1...1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	Enumeration	0...1	Specifies the payment context
MerchantCategoryCode	string	0...1	Category code conform to ISO 18245, related to the type of services or goods the merchant provides for the transaction.

MerchantCustomerIdentification	string	0...1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	Object	0...1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	Array	0...1	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
StreetName	string	0...1	Name of a street or thoroughfare.
BuildingNumber	string	0...1	Number that identifies the position of a building on a street.
PostCode	string	0...1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail.
TownName	string	0...1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	Array	0...1	Identifies a subdivision of a country, for instance state, region, county.
Country	String	0...1	Nation with its own government, occupying a particular territory.

Data Dictionary - Response

Name	Data Type	Occurrence	Notes
Data	complex type	1	
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
Status	enumeration	1	Determines the status of a payment consent as defined by Open Banking.
CreationDateTime	datetime	1	Date and time at which the resource was created.
Initiation	complex type	1	
InstructionIdentification	string	1	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the

			instruction. Usage: the instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.
EndToEndIdentification	string	0..1	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain. Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.
InstructedAmount	complex type	1	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party. Usage: This amount has to be transported unchanged through the transaction chain.
Amount	string	1	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Currency	string	1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
DebitorAccount	complex type	0..1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account.

			This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account. OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
RemittanceInformation	complex type	0..1	Information supplied to enable the matching of an entry with the items that the transfer is intended

			to settle, such as commercial invoices in an accounts' receivable system.
Reference	string	0..1	<p>Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.</p> <p>If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification. OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.</p>
Unstructured	string	0..1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
Street Name	string	0..1	Name of a street or thoroughfare

BuildingNumber	string	0..1	Number that identifies the position of a building on a street.
PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

GET - /open-banking-sandbox/v2.0/payments/{PaymentId}

A PISP can optionally retrieve a **payment** resource that they have created to check its status. Prior to calling the API, the PISP must have an access token issued by CBS using a client credentials grant. (POST <https://resourcemema.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

PaymentId Status

The Sandbox APIs do not include PSU authorisation, to assist TPPs with testing these flows specific PaymentIDs can be used as detailed in the Mocked Responses section.

The available Status code-list enumerations for the account-request resource are:

Status	Status Description
Pending	Payment initiation or Individual transaction included in the payment initiation is pending. Further checks and status update will be performed
Rejected	Payment initiation or Individual transaction included in the payment initiation has been rejected
AcceptedTechnicalValidation	Authentication and Syntactical and Semantic validation are successful
AcceptedCustomerProfile	Preceding check of technical validation was successful. Customer profile check was also successful.

Mocked Responses

Given	When	Then
Valid Request	PaymentId of SingleImmediate for AcceptedTechnicalValidation response · Valid access token PaymentId = b58fe243-4189-45bf-ac69-e071c78bf8ae	HTTP Status Code: 200 Status: AcceptedTechnicalValidation
Valid Request	PaymentId of SingleImmediate for AcceptedCustomerProfile response · Valid access token PaymentId = 4fcbf613-a220-4c1a-ad55-02a12cb02734	HTTP Status Code: 200 Status: AcceptedCustomerProfile

Valid Request	PaymentId of SingleImmediate for AcceptedTechnicalValidation response · Valid access token PaymentId = e738bd3b-985b-4c84-9e99-b06a8e8a0c2f	HTTP Status Code: 200 Status: Pending
Valid Request	PaymentId of SingleImmediate for AcceptedTechnicalValidation response · Valid access token PaymentId = ea5a0af4-f058-40c3-be52-f93391571222	HTTP Status Code: 200 Status: Rejected
Invalid Request	Null Payment ID	HTTP Status Code: 400
Invalid Request	Invalid Payment ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary - Request

Not required

Data Dictionary – Response

Data Item	Data Type	Cardinality	Comments
Data	complex type	1	
PaymentType	enumeration	1	Categorises the different types of payment methods within CBS for processing purposes.
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
Status	enumeration	1	Determines the status of a payment consent as defined by Open Banking: AcceptedCustomerProfile AcceptedTechnicalValidation

				Pending
				Rejected
CreationDateTime	datetime	1		Date and time at which the resource was created.
Initiation	complex type	1		
InstructionIdentification	string	1		<p>Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction.</p> <p>Usage: the instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.</p>
EndToEndIdentification	string	0..1		<p>Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain.</p> <p>Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.</p>
Frequency	enumeration	0..1		The number of times something happens within a particular period.
InstructedAmount	complex type	1		<p>Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party.</p> <p>Usage: This amount has to be transported unchanged through the transaction chain.</p>
Amount	string	1		A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.

Currency	string	1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
DebitorAccount	complex type	0..1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an

			<p>account level. The account name is not the product name or the nickname of the account.</p> <p>OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.</p>
Secondaryidentification	string	0..1	<p>This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).</p>
RemittanceInformation	complex type	0..1	<p>Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts' receivable system.</p>
Reference	string	0..1	<p>Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction.</p> <p>Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money.</p> <p>If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification.</p> <p>OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.</p>
Unstructured	string	0..1	<p>Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.</p>

Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
StreetName	string	0..1	Name of a street or thoroughfare
BuildingNumber	string	0..1	Number that identifies the position of a building on a street.
PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

POST - /open-banking-extensions-sandbox/v2.0/mandate-submissions

Once the mandate payment has been authorised by the PSU for future or regular payments, the PISP can proceed to submitting the payment for processing:

- This is done by making a POST request to the **mandate-submissions** resource.
- This request is an instruction to the ASPSP to begin the single immediate payment journey. The payment will be submitted immediately, however, there are some scenarios where the payment may not happen immediately (e.g. busy periods at the ASPSP).
- The PISP **must** ensure that the Initiation and Risk sections of the payment submission match the corresponding Initiation and Risk sections of the original payment resource. If the two do not match, the ASPSP **must not** process the request and **must** respond with a 400 (Bad Request).
- Any operations on the payment-submission resource will not result in a Status change for the payment resource.
- CBS creates the payment resource and responds with a unique **PaymentSubmissionId** to refer to the resource.

Mandate Payment Submission Status

A payment-submission can only be created if its corresponding payment resource has the status of 'AcceptedCustomerProfile'.

The payment-submission resource that is created successfully must have one of the following PaymentStatusCode code-set enumerations:

Mandate Payment Status

The Mandate payment resource that is created successfully must have one of the following Status code-list enumerations:

Status	Status Description
Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
AcceptedSettlementInProgress	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.

Mocked Responses

Given	When	Then
Valid Request	Payment type Future Dated · Valid access token PaymentId = 172cf16c-5819-40cf-aa3f-faee1b16b8f4	HTTP Status Code: 201 Status: AcceptedSettlementInProgress
Valid Request	Payment type Regular · Valid access token PaymentId = 219e279c-a1dd-4f80-9d52-822853dff653	HTTP Status Code: 201 Status: AcceptedSettlementInProgress
Invalid Request	Null Payment ID	HTTP Status Code: 400
Invalid Request	Invalid Payment ID	HTTP Status Code: 400
Invalid Request	Null Header	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Data Item	Data Type	Occurrence	Comments
Data	complex type	1	
PaymentType	enumeration	1	Categorises the different types of payment methods within CBS for processing purposes.
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
Initiation	complex type	1	
InstructionIdentification	string	1	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction. Usage: the instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.
EndToEndIdentification	string	0..1	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain. Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.
FuturePaymentDateTime	ISODatetime	0..1	Date at which the initiating party requests the clearing agent to process the payment. Usage: This is the date on which the debtor's account is to be debited.
ValidFromDateTime	ISODatetime	0..1	The date on which the first payment for a Standing Order schedule will be made.

ValidToDateTime	ISODatetime	0..1	The date on which the final payment for a Standing Order schedule will be made.
Frequency	enumeration	0..1	The number of times something happens within a particular period.
InstructedAmount	complex type	1	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party. Usage: This amount has to be transported unchanged through the transaction chain.
Amount	string	1	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Currency	string	1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
DebitorAccount	complex type	0..1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).

CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account. OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
RemittanceInformation	complex type	0..1	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts' receivable system.
Reference	string	0..1	Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money. If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment remittance identification should be quoted in the end-to-end transaction identification. OB: The

			Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.
Unstructured	string	0..1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
StreetName	string	0..1	Name of a street or thoroughfare
BuildingNumber	string	0..1	Number that identifies the position of a building on a street.
PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

GET - /open-banking-extensions-sandbox/v2.0/mandate-submissions/{PaymentSubmissionId}

A PISP can optionally retrieve a **mandate payment submission** resource that they have created to check its status. Prior to calling the API, the PISP must have an access token issued by CBS using a client credentials grant. (POST <https://resourcema.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

PaymentSubmissionId Status

The Sandbox APIs do not include PSU authorisation, to assist TPPs with testing these flows specific PaymentSubmissionIDs can be used as detailed in the Mocked Responses section.

The available Status code-list enumerations for the account-request resource are:

Status	Status Description
Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
AcceptedSettlementInProgress	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
AcceptedSettlementCompleted	Settlement on the debtor's account has been completed.

Mocked Responses

Given	When	Then
Valid Request	PaymentSubmissionId for AcceptedSettlementInProgress response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = 13b27b4a-78b1-487c-9c08-522167d2eba7 	HTTP Status Code: 200 Status: AcceptedSettlementInProgress
Valid Request	PaymentSubmissionId for AcceptedSettlementInProgress response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = 20a95b03-8bd7-489d-ae13-68b3676553ce 	HTTP Status Code: 200 Status: Pending
Valid Request	PaymentSubmissionId for Rejected response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = 5d4778b7-0b8e-4bc4-8b32-a268257fed1f 	HTTP Status Code: 200 Status: Rejected
Valid Request	PaymentSubmissionId for AcceptedSettlementCompleted response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = 0c866382-945b-4371-b9bb-d84acf98ecc7 	HTTP Status Code: 200 Status: AcceptedSettlementCompleted
Invalid Request	Null Payment ID	HTTP Status Code: 400
Invalid Request	Invalid Payment ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401

Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Not Required

Data Dictionary- Response

Data Item	Data Type	Cardinality	Comments
Data	complex type	1	
CreationDateTime	datetime	1	Date and time at which the submission resource was created
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
PaymentSubmissionId	string	1	Unique identification as assigned by the ASPSP, for example CBS, to uniquely identify the payment instruction resource.
Status	enumeration	1	Specifies the status of the payment submission resource, as per OB standard codes

POST - /open-banking-sandbox/v2.0/payment-submissions

Once the payment has been authorised by the PSU for future or regular payments, the PISP can proceed to submitting the payment for processing:

- This is done by making a POST request to the **payment-submissions** resource.
- This request is an instruction to the ASPSP to begin the single immediate payment journey. The payment will be submitted immediately, however, there are some scenarios where the payment may not happen immediately (e.g. busy periods at the ASPSP).
- The PISP **must** ensure that the Initiation and Risk sections of the payment submission match the corresponding Initiation and Risk sections of the original payment resource. If the two do not match, the ASPSP **must not** process the request and **must** respond with a 400 (Bad Request).
- Any operations on the payment-submission resource will not result in a Status change for the payment resource.
- CBS creates the payment resource and responds with a unique **PaymentSubmissionId** to refer to the resource.

Payment Submission Status

A payment-submission can only be created if its corresponding payment resource has the status of 'AcceptedCustomerProfile'.

The payment-submission resource that is created successfully must have one of the following PaymentStatusCode code-set enumerations:

Payment Status

The Mandate payment resource that is created successfully must have one of the following Status code-list enumerations:

Status	Status Description
Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
AcceptedSettlementInProgress	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.

Mocked Responses

Given	When	Then
Valid Request	Valid access token PaymentId = b58fe243-4189-45bf-ac69-e071c78bf8ae	HTTP Status Code: 201 Status: AcceptedSettlementInProgress
Invalid Request	Null Payment ID	HTTP Status Code: 400
Invalid Request	Invalid Payment ID	HTTP Status Code: 400
Invalid Request	Null Header	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Data Item	Data Type	Occurrence	Comments
Data	complex type	1	
PaymentType	enumeration	1	Categorises the different types of payment methods within CBS for processing purposes.
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
Initiation	complex type	1	
InstructionIdentification	string	1	Unique identification as assigned by an instructing party for an instructed party to unambiguously identify the instruction. Usage: the instruction identification is a point to point reference that can be used between the instructing party and the instructed party to refer to the individual instruction. It can be included in several messages related to the instruction.
EndToEndIdentification	string	0..1	Unique identification assigned by the initiating party to unambiguously identify the transaction. This identification is passed on, unchanged, throughout the entire end-to-end chain. Usage: The end-to-end identification can be used for reconciliation or to link tasks relating to the transaction. It can be included in several messages related to the transaction. OB: The Faster Payments Scheme can only access 31 characters for the EndToEndIdentification field.
FuturePaymentDateTime	ISODateTime	0..1	Date at which the initiating party requests the clearing agent to process the payment. Usage: This is the date on which the debtor's account is to be debited.

ValidFromDateTime	ISODatetime	0..1	The date on which the first payment for a Standing Order schedule will be made.
ValidToDateTime	ISODatetime	0..1	The date on which the final payment for a Standing Order schedule will be made.
Frequency	enumeration	0..1	The number of times something happens within a particular period.
InstructedAmount	complex type	1	Amount of money to be moved between the debtor and creditor, before deduction of charges, expressed in the currency as ordered by the initiating party. Usage: This amount has to be transported unchanged through the transaction chain.
Amount	string	1	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Currency	string	1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds".
DebitorAccount	complex type	0..1	Unambiguous identification of the account of the debtor to which a debit entry will be made as a result of the transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts

			with a roll number (in addition to a sort code and account number combination).
CreditorAccount	complex type	1	Unambiguous identification of the account of the creditor to which a credit entry will be posted as a result of the payment transaction.
SchemeName	enumeration	1	Name of the identification scheme, in a coded form as published in an external list.
Identification	string	1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	string	1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account. OB: ASPSPs may carry out name validation for Confirmation of Payee, but it is not mandatory.
Secondaryidentification	string	0..1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).
RemittanceInformation	complex type	0..1	Information supplied to enable the matching of an entry with the items that the transfer is intended to settle, such as commercial invoices in an accounts' receivable system.
Reference	string	0..1	Unique reference, as assigned by the creditor, to unambiguously refer to the payment transaction. Usage: If available, the initiating party should provide this reference in the structured remittance information, to enable reconciliation by the creditor upon receipt of the amount of money. If the business context requires the use of a creditor reference or a payment remit identification, and only one identifier can be passed through the end-to-end chain, the creditor's reference or payment

			remittance identification should be quoted in the end-to-end transaction identification. OB: The Faster Payments Scheme can only accept 18 characters for the ReferenceInformation field - which is where this ISO field will be mapped.
Unstructured	string	0..1	Information supplied to enable the matching/reconciliation of an entry with the items that the payment is intended to settle, such as commercial invoices in an accounts' receivable system, in an unstructured form.
Risk	complex type	0..1	The Risk section is sent by the initiating party to the ASPSP. It is used to specify additional details for risk scoring for Payments.
PaymentContextCode	enumeration	0..1	Specifies the payment context
MerchantCategoryCode	string	0..1	Category code conforms to ISO 18245, related to the type of services or goods the merchant provides for the transaction
MerchantCustomerIdentification	string	0..1	The unique customer identifier of the PSU with the merchant.
DeliveryAddress	complex type	0..1	Information that locates and identifies a specific address, as defined by postal services or in free format text.
AddressLine	string	0..2	Information that locates and identifies a specific address, as defined by postal services, that is presented in free format text.
StreetName	string	0..1	Name of a street or thoroughfare
BuildingNumber	string	0..1	Number that identifies the position of a building on a street.
PostCode	string	0..1	Identifier consisting of a group of letters and/or numbers that is added to a postal address to assist the sorting of mail
TownName	string	0..1	Name of a built-up area, with defined boundaries, and a local government.
CountrySubDivision	string	0..2	Identifies a subdivision of a country, for instance state, region, county
Country	string	0..1	Nation with its own government, occupying a particular territory

GET - /open-banking-extensions-sandbox/v2.0/payment-submissions/{PaymentSubmissionId}

A PISP can optionally retrieve a **payment submission** resource that they have created to check its status. Prior to calling the API, the PISP must have an access token issued by CBS using a client credentials grant. (POST <https://resource.ma.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

PaymentSubmissionId Status

The Sandbox APIs do not include PSU authorisation, to assist TPPs with testing these flows specific PaymentSubmissionIDs can be used as detailed in the Mocked Responses section.

The available Status code-list enumerations for the account-request resource are:

Status	Status Description
Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
AcceptedSettlementInProgress	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
AcceptedSettlementCompleted	Settlement on the debtor's account has been completed.

Mocked Responses

Given	When	Then
Valid Request	PaymentSubmissionId for AcceptedSettlementInProgress response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = 2a087a2d-47cd-4054-90b2-2f320cc654f2 	HTTP Status Code: 200 Status: AcceptedSettlementInProgress
Valid Request	PaymentSubmissionId for AcceptedSettlementInProgress response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = e0c20b12-3ff6-4aad-90dd-30a70fe22ac5 	HTTP Status Code: 200 Status: Pending
Valid Request	PaymentSubmissionId for Rejected response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = cf66b703-05d9-40fc-9b19-67460b1f1166 	HTTP Status Code: 200 Status: Rejected
Valid Request	PaymentSubmissionId for AcceptedSettlementCompleted response <ul style="list-style-type: none"> · Valid access token · PaymentSubmissionID = 6c20f296-77a8-4bbc-98da-b8c130c421a7 	HTTP Status Code: 200 Status: AcceptedSettlementCompleted
Invalid Request	Null Payment ID	HTTP Status Code: 400
Invalid Request	Invalid Payment ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401

Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Not Required

Data Dictionary- Response

Data Item	Data Type	Occurrence	Notes
CreationDateTime	datetime	1	Date and time at which the submission resource was created
PaymentId	string	1	Unique reference to identify consent throughout the authorisation process. This identifier reflects the PSD2 ids as follows: - Payment - Payment ID - Account - Account Request ID
PaymentSubmissionId	string	1	Unique identification as assigned by the ASPSP, for example CBS, to uniquely identify the payment instruction resource.
Status	enumeration	1	Specifies the status of the payment submission resource, as per OB standard codes

GET /open-banking-sandbox/v2.0/payments/{ConsentId}/funds-confirmation

If the PISP would like to confirm funds with the ASPSP, it should create a new **funds-confirmation** resource, and check the funds available flag in the response.

- The ASPSP creates the **funds-confirmation** resource and responds with a unique FundsConfirmationId to refer to the resource, and a flag confirming if funds are available.
- The PISP **must** use a token issued via authorization code grant and specify the ConsentId in the request payload.
- ThisPISP **must** use a currency of the account.

Mocked Responses

Given	When	Then
Valid Request	The request is valid containing · Valid access token	HTTP Status Code: 200 Status: Yes

	· ConsentId = c1e09335-ad8b-440f-8557-c38163984cca	
Valid Request	The request is valid containing · Valid access token · ConsentId = 50a23b84-9675-458d-b327-5e05fba1bda7	HTTP Status Code: 200 Status: No
Invalid Request	Null Consent ID	HTTP Status Code: 400
Invalid Request	Invalid Consent ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Not required

Data Dictionary- Response

Data Item	Data Type	Occurrence	Notes
Data Item	Data Type	Cardinality	Comments
Data	Object	1..1	
FundsAvailableResult	Object	0..1	Result of a funds availability check.
FundsAvailableDateTime	DateTime	1..1	Date and time at which the funds availability check was generated.
FundsAvailable	Enumeration	1..1	Flag to indicate the availability of funds given the Amount in the consent request. Yes or No
SupplementaryData	Object	0..1	Additional information that can not be captured in the structured fields and/or any other specific block.
Links	Object	1..1	

POST - /open-banking-sandbox/v2.0/funds-confirmation-consent

The API allows the CBPII to ask an ASPSP to create a new **funds-confirmation-consent** resource.

- This endpoint allows the CBPII to propose a consent to be agreed between the ASPSP and PSU, to authorise the CBPII access to confirm funds are available.
- The ASPSP creates the **funds-confirmation-consent** resource and responds with a unique Funds Confirmation ConsentId to refer to the resource.

- Prior to calling the operation, the CBPII must have an access token issued by the ASPSP using a client credentials grant.

Funds Confirmation Consent Status

The PSU **must** authenticate with the ASPSP and agree the **funds-confirmation-consent** with the ASPSP, for the **funds-confirmation-consent** to be successfully setup.

Funds Confirmation Consent Status

The **funds-confirmation-consent** resource that is created successfully must have one of the following Status code-list enumerations:

Status	Status Description
AwaitingAuthorisation	The Funds Confirmation Consent is awaiting agreement.

Mocked Responses

Given	When	Then
Valid Request	The request is valid containing Valid access token	HTTP Status Code: 201 Status: AwaitingAuthorisation
Invalid Request	Null Consent ID	HTTP Status Code: 400
Invalid Request	Invalid Consent ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Data Item	Data Type	Occurence	Notes
Data	Object	1.1	
ExpirationDateTime	DateTime	0.1	Specified date and time the funds confirmation authorisation will expire. If this is not populated, the authorisation will be open ended.
DebtorAccount	Object	1.1	Unambiguous identification of the account of the debtor to which a confirmation of funds consent will be applied.
SchemeName	String	1.1	Name of the identification scheme, in a coded form as published in an external list.

Identification	String	1.1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	String	0.1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account
SecondaryIdentification	String	0.1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).

Data Dictionary- Response

Data Item	Data Type	Cardinality	Comments
Data	Object	1.1	
ConsentId	String	1.1	Unique identification as assigned to identify the funds confirmation consent resource.
CreationDateTime	DateTime	1.1	Date and time at which the resource was created.
Status	String	1.1	Specifies the status of consent resource in code form.
StatusUpdateDateTime	DateTime	1.1	Date and time at which the resource status was updated.
ExpirationDateTime	DateTime	0.1	Specified date and time the funds confirmation authorisation will expire. If this is not populated, the authorisation will be open ended.
DebtorAccount	Object	1.1	Unambiguous identification of the account of the debtor to which a confirmation of funds consent will be applied.
SchemeName	String	1.1	Name of the identification scheme, in a coded form as published in an external list.
Identification	String	1.1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	String	0.1	Name of the account, as assigned by the account servicing

			institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account
SecondaryIdentification	String	0.1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).

GET - /open-banking-sandbox/v2.0/funds-confirmation-consents/{ConsentId}

A CBPII may optionally retrieve an **funds-confirmation-consent** resource that they have created to check its status. Prior to calling the operation, the CBPII must have an access token issued by the ASPSP using a client credentials grant.

Prior to calling the API, the CBPII must have an access token issued by CBS using a client credentials grant. (POST <https://resourcema.coventrybuildingsociety.co.uk/mga/sps/oauth/oauth20/token>)

Funds Confirmation ConsentId Status

The Sandbox APIs do not include PSU authorisation, to assist TPPs with testing these flows specific FundsConfirmationConsentIDs can be used as detailed in the Mocked Responses section.

The available Status code-list enumerations for the Funds Confirmation consent resource are:

Status	Status Description
Rejected	The Funds Confirmation Consent has been rejected.
AwaitingAuthorisation	The Funds Confirmation Consent is awaiting agreement.
Authorised	The Funds Confirmation Consent has been successfully agreed.
Revoked	The Funds Confirmation Consent has been revoked via the ASPSP interface.

Mocked Responses

Given	When	Then
Valid Request	The request is valid containing <ul style="list-style-type: none"> Valid access token ConsentID = 514c144a-3288-4d04-b467-61772f80db27 	HTTP Status Code: 200 Status: AwaitingAuthorisation

Valid Request	The request is valid containing · Valid access token · ConsentID = 625d255b-3288-4d04-b467-61772f80db27	HTTP Status Code: 200 Status: Authorised
Valid Request	The request is valid containing · Valid access token · ConsentID = 736e366c-3288-4d04-b467-61772f80db27	HTTP Status Code: 200 Status: Rejected
Valid Request	The request is valid containing · Valid access token · ConsentID = 847f477d-3288-4d04-b467-61772f80db27	HTTP Status Code: 200 Status: Revoked
Invalid Request	Null Consent ID	HTTP Status Code: 400
Invalid Request	Invalid Consent ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Not required

Data Dictionary- Response

Data Item	Data Type	Occurence	Notes
Data	Object	1.1	
ConsentId	String	1.1	Unique identification as assigned to identify the funds confirmation consent resource.
CreationDateTime	DateTime	1.1	Date and time at which the resource was created.
Status	String	1.1	Specifies the status of consent resource in code form.
StatusUpdateDateTime	DateTime	1.1	Date and time at which the resource status was updated.
ExpirationDateTime	DateTime	0.1	Specified date and time the funds confirmation authorisation will expire.

			If this is not populated, the authorisation will be open ended.
DebtorAccount	Object	1.1	Unambiguous identification of the account of the debtor to which a confirmation of funds consent will be applied.
SchemeName	String	1.1	Name of the identification scheme, in a coded form as published in an external list.
Identification	String	1.1	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Name	String	0.1	Name of the account, as assigned by the account servicing institution. Usage: The account name is the name or names of the account owner(s) represented at an account level. The account name is not the product name or the nickname of the account
SecondaryIdentification	String	0.1	This is secondary identification of the account, as assigned by the account servicing institution. This can be used by building societies to additionally identify accounts with a roll number (in addition to a sort code and account number combination).

DELETE - /open-banking-sandbox/v2.0/funds-confirmation-consents/{ConsentId}

If the PSU revokes consent to confirm funds with the CBPII - the CBPII **must** delete the **funds-confirmation-consent** resource.

- This is done by making a call to DELETE the **funds-confirmation-consent** resource.
- Prior to calling the operation, the CBPII must have an access token issued by the ASPSP using a client credentials grant

Mocked Response

Given	When	Then
Valid Request	The request is valid containing <ul style="list-style-type: none"> · Valid access token · ConsentID = 514c144a-3288-4d04-b467-61772f80db27 	HTTP Status Code: 204
Invalid Request	Null Consent ID	HTTP Status Code: 400
Invalid Request	Invalid Consent ID	HTTP Status Code: 400

Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

POST - /open-banking-sandbox/v2.0/funds-confirmations

If the CBPII would like to confirm funds with the ASPSP, it should create a new **funds-confirmation** resource, and check the funds available flag in the response.

- The ASPSP creates the **funds-confirmation** resource and responds with a unique FundsConfirmationId to refer to the resource, and a flag confirming if funds are available.
- The CBPII **must** use a token issued via authorization code grant and specify the ConsentId in the request payload.
- This CBPII **must** use a currency of the account.

Mocked Responses

Given	When	Then
Valid Request	The request is valid containing <ul style="list-style-type: none"> · Valid access token · Amount = 200.00 	HTTP Status Code: 201 Status: Yes
Valid Request	The request is valid containing <ul style="list-style-type: none"> · Valid access token · Amount < 200.00 	HTTP Status Code: 201 Status: Yes
Valid Request	The request is valid containing <ul style="list-style-type: none"> · Valid access token · Amount > 200.00 	HTTP Status Code: 201 Status: No
Invalid Request	Null Consent ID	HTTP Status Code: 400
Invalid Request	Invalid Consent ID	HTTP Status Code: 400
Invalid Request	Null Headers	HTTP Status Code: 400
Invalid Request	Invalid Headers	HTTP Status Code: 400
Invalid Request	Invalid Certificate	HTTP Status Code: 401
Invalid Request	Invalid Token Scope	HTTP Status Code: 403
Invalid Request	Invalid Method	HTTP Status Code: 405
Invalid Request	Throttling limit (> 500)	HTTP Status Code: 429
Invalid Request	Internal server error	HTTP Status Code: 500

Data Dictionary- Request

Data Item	Data Type	Occurrence	Notes
Data	Object	1..1	
ConsentId	String	1..1	Unique identification as assigned by the CBS to uniquely identify the funds confirmation consent resource.
Reference	String	1..1	Unique reference, as assigned by the CBPII, to unambiguously refer to the request related to the payment transaction
InstructedAmount	Object	1..1	Amount of money to be confirmed as available funds in the debtor account. Contains an Amount and a Currency.
Amount	Integer	1..1	A number of monetary units specified in an active currency.
Currency	String	1..1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds

Data Dictionary- Response

Data Item	Data Type	Cardinality	Comments
Data	Object	1..1	
FundsConfirmationId	String	1..1	Unique identification as assigned by the CBS to uniquely identify the funds confirmation resource.
ConsentId	String	1..1	Unique identification as assigned by the CBS to uniquely identify the funds confirmation consent resource.
CreationDateTime	DateTime	1..1	Date and time at which the resource was created.
FundsAvailable	String	1..1	Flag to indicate the result of a confirmation of funds check. (Yes/No)
Reference	String	1..1	Unique reference, as assigned by the CBPII, to unambiguously refer to the request related to the payment transaction
InstructedAmount	Object	1..1	Amount of money to be confirmed as available funds in the debtor account. Contains an Amount and a Currency.

Amount	Integer	1..1	A number of monetary units specified in an active currency.
Currency	String	1..1	A code allocated to a currency by a Maintenance Agency under an international identification scheme, as described in the latest edition of the international standard ISO 4217 "Codes for the representation of currencies and funds

Data Payload – Enumerations

This section gives the definitions for enumerations used in the Account Info APIs.

TransactionEntryCode

Name	Definition
Booked	<p>Booked means that the transfer of money has been completed between account servicer and account owner</p> <p>Usage: Status Booked does not necessarily imply finality of money as this depends on other factors such as the payment system used, the completion of the end- to-end transaction and the terms agreed between account servicer and owner.</p> <p>Status Booked is the only status that can be reversed.</p>
Pending	<p>Booking on the account owner's account in the account servicer's ledger has not been completed.</p> <p>Usage: this can be used for expected items, or for items for which some conditions still need to be fulfilled before they can be booked. If booking takes place, the entry will be included with status Booked in subsequent account report or statement.</p> <p>Status Pending cannot be reversed.</p>

CreditDebitCode

Name	Definition
Credit	Operation is a credit
Debit	Operation is a debit

BalanceTypeCode

Name	Definition
ClosingAvailable	Closing balance of amount of money that is at the disposal of the account owner on the date specified.
ClosingBooked	Balance of the account at the end of the pre-agreed account reporting period. It is the sum of the opening booked balance at the beginning of the period and all entries booked to the account during the pre-agreed account reporting period.
Expected	Balance, composed of booked entries and pending items known at the time of calculation , which projects the end of day balance if everything is booked on the account and no other entry is posted.
ForwardAvailable	Forward available balance of money that is at the disposal of the account owner on the date specified.
Information	Balance for informational purposes.
InterimAvailable	Available balance calculated in the course of the account servicer's business day, at the time specified, and subject to further changes during the business day. The interim balance is calculated on the basis of booked credit and debit items during the calculation time/period specified.
InterimBooked	Balance calculated in the course of the account servicer's business day, at the time specified, and subject to further changes during the business day. The interim balance is calculated on the basis of booked credit and debit items during the calculation time/period specified.
OpeningAvailable	Opening balance of amount of money that is at the disposal of the account owner on the date specified.
OpeningBooked	Book balance of the account at the beginning of the account reporting period. It always equals the closing book balance from the previous report.
PreviouslyClosedBooked	Balance of the account at the previously closed account reporting period. The opening booked balance for the new period has to be equal to this balance. Usage: the previously booked closing balance should equal (inclusive date) the booked closing balance of the date it references and equal the actual booked opening balance of the current date.

ExternalAccountIdentification2Code

Name	Definition
SortCodeAccountNumber	Sort Code and Account Number - identifier scheme used in the UK by financial institutions to identify the account of a customer. The identifier is the concatenation of the 6 digit UK sort code and 8 digit account number. The regular expression for this identifier is: <code>^[0-9]{6}[0-9]{8}\$</code>

ExternalDirectDebitStatus1Code

Name	Definition
Active	The direct debit mandate is active.
Inactive	The direct debit mandate is inactive.

ExternalPermissions1Code

Name	Definition
ReadAccountsBasic	Permission to read basic account information.
ReadAccountsDetail	Access to additional elements in the account payload.
ReadBalances	Permission to read all balance information.
ReadBeneficiariesBasic	Permission to read basic beneficiary details.
ReadBeneficiariesDetail	Access to additional elements in the beneficiaries payload.
ReadDirectDebits	Permission to read all direct debit information.
ReadStandingOrdersBasic	Permission to read standing order information.
ReadStandingOrdersDetail	Access to additional elements in the standing-orders payload.
ReadTransactionsBasic	Permission to read basic transaction information.
ReadTransactionsDetail	Access to additional elements in the transactions payload.
ReadTransactionsCredits	Access to only credit transactions.
ReadTransactionsDebits	Access to only debit transactions.
ReadProducts	Permission to read all product information.

ExternalProductType1Code

Name	Definition
PCA	Personal Current Account

ExternalRequestStatus1Code

Name	Definition
Authorised	The account request has been successfully authorised.
AwaitingAuthorisation	The account request is awaiting further authorisation.
Rejected	The account request has been rejected.
Revoked	The account request has been revoked via CBS interface.

TransactionEntryCode

Name	Definition
Booked	<p>Booked means that the transfer of money has been completed between account servicer and account owner</p> <p>Usage: Status Booked does not necessarily imply finality of money as this depends on other factors such as the payment system used, the completion of the end- to-end transaction and the terms agreed between account servicer and owner.</p> <p>Status Booked is the only status that can be reversed.</p>
Pending	<p>Booking on the account owner's account in the account servicer's ledger has not been completed.</p> <p>Usage: this can be used for expected items, or for items for which some conditions still need to be fulfilled before they can be booked. If booking takes place, the entry will be included with status Booked in subsequent account report or statement.</p> <p>Status Pending cannot be reversed.</p>

This section gives the definitions for enumerations used in the Payment APIs.

Mandate/Payment Status

Name	Status Description
Pending	Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.
Rejected	Payment initiation or individual transaction included in the payment initiation has been rejected.
AcceptedSettlementInProgress	All preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.
AcceptedSettlementCompleted	Settlement on the debtor's account has been completed.

Regular Payment Frequency [Mandates]

Name	Description
Weekly	The number of times something happens within a particular period. This Regular Payment is set up to happen on a weekly basis
Fortnightly	The number of times something happens within a particular period. This Regular Payment is set up to happen on a Fortnightly basis
Monthly	The number of times something happens within a particular period. This Regular Payment is set up to happen on a Monthlybasis
Quarterly	The number of times something happens within a particular period. This Regular Payment is set up to happen on a Quarterly basis
Half Yearly	The number of times something happens within a particular period. This Regular Payment is set up to happen on a Half Yearly basis
Annually	The number of times something happens within a particular period. This Regular Payment is set up to happen on a Annual basis

Payment Type [Mandate]

Name	Description
Future Dated	Categorises the different types of payment methods within CBS for processing purposes. This Payment Mandate is set up to happen at a Future Date and not Regular.
Regular	Categorises the different types of payment methods within CBS for processing purposes. This Mandate Payment is set up to happen Regularly at a specified period.

Mandate/Payment SchemeName

Name	Description
SortCodeAccountNumber	Name of the identification scheme, in a coded form as published in an external list.

Mandate/Payment PaymentContextCode

Name	Description
------	-------------

BillPayment	Specifies the payment context
EcommerceGoods	Specifies the payment context
EcommerceServices	Specifies the payment context
Other	Specifies the payment context
PartyToParty	Specifies the payment context

Funds Confirmation Status

Name	Definition
Authorised	The Funds Confirmation Consent has been rejected
AwaitingAuthorisation	The Funds Confirmation Consent is awaiting agreement.
Rejected	The Funds Confirmation Consent has been successfully agreed.
Revoked	The Funds Confirmation Consent has been successfully agreed.

Funds Confirmation Status

Name	Definition
Yes	Funds Confirmed as available
No	Funds Confirmation not available meaning insufficient funds in Account.

Usage Examples

All Permissions Granted

This set of payload examples is for an AISP:

- Setting up an account-request
- Getting the status of an account-request
- Retrieving data from each of the Account Info API endpoints
- Deleting the account-request

In this scenario:

- All permissions have been granted to access all Account Info API resources

Setup Account Request

Request

Post Account Requests Request

```
POST /account-requests HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Content-Type: application/json
Accept: application/json
{
  "Data": {
    "Permissions": [
      "ReadAccountsDetail",
      "ReadBalances",
      "ReadBeneficiariesDetail",
      "ReadDirectDebits",
      "ReadProducts",
      "ReadStandingOrdersDetail",
      "ReadTransactionsCredits",
      "ReadTransactionsDebits",
      "ReadTransactionsDetail"
    ],
    "ExpirationDateTime": "2017-05-
02T00:00:00+00:00",
    "TransactionFromDate": "2017-05-
03T00:00:00+00:00",
    "TransactionToDate": "2017-12-
03T00:00:00+00:00"
  }
}
```

Response

Post Account Requests Response

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "AccountRequestId": "88379",
    "Status": "AwaitingAuthorisation",
    "CreationDateTime": "2017-05-
02T00:00:00+00:00",
    "Permissions": [
      "ReadAccountsDetail",
      "ReadBalances",
      "ReadBeneficiariesDetail",
      "ReadDirectDebits",
      "ReadProducts",
      "ReadStandingOrdersDetail",
      "ReadTransactionsCredits",
      "ReadTransactionsDebits",
      "ReadTransactionsDetail"
    ],
    "ExpirationDateTime": "2017-08-
02T00:00:00+00:00",
    "TransactionFromDate": "2017-05-
03T00:00:00+00:00",
    "TransactionToDate": "2017-12-
03T00:00:00+00:00"
  },
  "Risk": {},
  "Links": {
    "Self": "/account-requests/88379"
  }
}
```

```
},  
"Risk": {}}}
```

```
}}
```

Status - AwaitingAuthorisation

This is an example of a GET request which is made **before** the account request resource is authorised.

Request

Get Account Requests Request

```
GET /account-requests/88379 HTTP/1.1  
Authorization: Bearer  
2YotnFZFEjrlzCsicMWpAA  
x-fapi-financial-id: OB/2017/001  
x-fapi-customer-last-logged-time: Sun, 10  
Sep 2017 19:43:31 UTC  
x-fapi-customer-ip-address: 104.25.212.99  
x-fapi-interaction-id: 93bac548-d2de-4546-  
b106-880a5018460d  
X-Client-Id: tppclientid  
Accept: application/json
```

Response

Get Account Requests Response

```
HTTP/1.1 200 OK  
x-fapi-interaction-id: 93bac548-d2de-4546-  
b106-880a5018460d  
Content-Type: application/json  
  
{  
  "Data": {  
    "AccountRequestId": "88379",  
    "Status": "AwaitingAuthorisation",  
    "CreationDateTime": "2017-05-  
02T00:00:00+00:00",  
    "Permissions": [  
      "ReadAccountsDetail",  
      "ReadBalances",  
      "ReadBeneficiariesDetail",  
      "ReadDirectDebits",  
      "ReadProducts",  
      "ReadStandingOrdersDetail",  
      "ReadTransactionsCredits",  
      "ReadTransactionsDebits",  
      "ReadTransactionsDetail"  
    ],  
    "ExpirationDateTime": "2017-08-  
02T00:00:00+00:00",  
    "TransactionFromDate": "2017-05-  
03T00:00:00+00:00",  
    "TransactionToDate": "2017-12-  
03T00:00:00+00:00"  
  },  
  "Risk": {},  
  "Links": {  
    "Self": "/account-requests/88379"  
  },  
  "Meta": {  
    "TotalPages": 1  
  }  
}
```

Status - Authorised

This is an example of a GET request which is made **after** the account request resource is authorised.

Valid Request - The CBS customer has Authorized the Request
Account Request ID = 6495080e-cc91-4171-a560-4ec9522740ee

Request

Get Account Requests Request

```
GET /account-requests/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Requests Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "AccountRequestId": "6495080e-cc91-
4171-a560-4ec9522740ee",
    "CreationDateTime": "2019-02-
05T09:15:34.455Z",
    "Status": "Authorised",
    "Permissions": [
      "ReadTransactionsCredits",
      "ReadBalances",
      "ReadAccountsDetail",
      "ReadStandingOrdersBasic"
    ],
    "ExpirationDateTime": "2019-04-
05T08:15:34.455Z",
    "TransactionFromDate": "2018-08-
05T08:15:34.455Z",
    "TransactionToDate": "2019-08-
05T08:15:34.455Z"
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/account-requests/6495080e-
cc91-4171-a560-4ec9522740ee"},
  "Meta": {},
  "Risk": {}
}
```

This is an example of a GET request which is made **after** the account request resource is authorised.

Valid Request - The CBS customer has Authorized the Request
Account Request ID = 6495080e-cc91-4171-a560-4ec9522740ee

Request

Get Account Requests Request

```
GET /account-requests/88379 HTTP/1.1
```

Response

Get Account Requests Response

```
HTTP/1.1 200 OK
```

```

Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json

```

```

x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "AccountRequestId": "6495080e-cc91-
4171-a560-4ec9522740ee",
    "CreationDateTime": "2019-02-
05T09:15:34.455Z",
    "Status": "Authorised",
    "Permissions": [
      "ReadTransactionsCredits",
      "ReadBalances",
      "ReadAccountsDetail",
      "ReadStandingOrdersBasic"
    ],
    "ExpirationDateTime": "2019-04-
05T08:15:34.455Z",
    "TransactionFromDate": "2018-08-
05T08:15:34.455Z",
    "TransactionToDate": "2019-08-
05T08:15:34.455Z"
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/account-requests/6495080e-
cc91-4171-a560-4ec9522740ee"},
  "Meta": {},
  "Risk": {}
}

```

This is an example of a GET request which is made **when** the account request resource is yet to be Authorised.

Valid Request - The CBS customer is yet to Authorize the Request Account Request ID = d02d3f54-234a-45a9-a097-1ddb1a8d828e

Request

Response

Get Account Requests Request

Get Account Requests Response

```

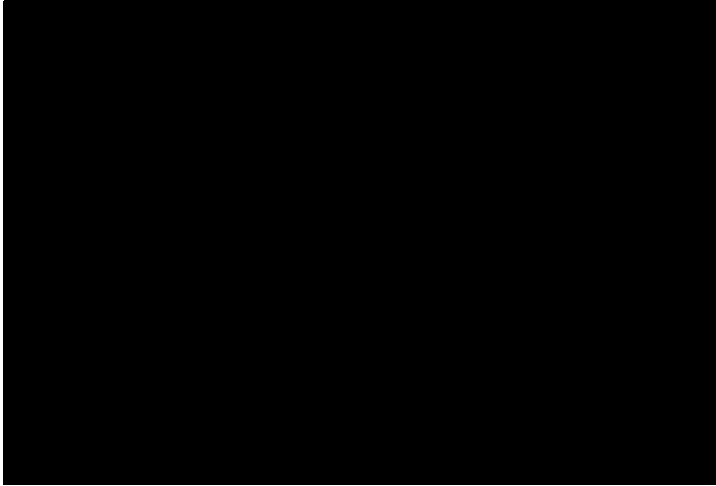
GET /account-requests/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json

```

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "AccountRequestId": "d02d3f54-234a-
45a9-a097-1ddb1a8d828e",
    "CreationDateTime": "2019-02-
05T09:15:34.841Z",
    "Status": "AwaitingAuthorisation",
    "Permissions": [
      "ReadTransactionsCredits",
      "ReadBalances",

```



```

    "ReadAccountsDetail",
    "ReadStandingOrdersBasic"
  ],
  "ExpirationDateTime": "2019-04-05T08:15:34.841Z",
  "TransactionFromDateTime": "2018-08-05T08:15:34.841Z",
  "TransactionToDateTime": "2019-08-05T08:15:34.841Z"
},
"Links": {"Self": "/open-banking-sandbox/v2.0/account-requests/d02d3f54-234a-45a9-a097-1ddb1a8d828e"},
"Meta": {},
"Risk": {}
}

```

This is an example of a GET request which is made **after** the account request resource is Rejected.

Valid Request - The CBS Customer Rejects the Request during the Authentication process
 Account Request ID = dd5447f7-1898-4855-86a6-fd50d37eb42f

Request	Response
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Get Account Requests Request</div> <pre> GET /account-requests/88379 HTTP/1.1 Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA x-fapi-financial-id: OB/2017/001 x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC x-fapi-customer-ip-address: 104.25.212.99 x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d X-Client-Id: tppclientid Accept: application/json </pre>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Get Account Requests Response</div> <pre> HTTP/1.1 200 OK x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d Content-Type: application/json { "Data": { "AccountRequestId": "dd5447f7-1898-4855-86a6-fd50d37eb42f", "CreationDateTime": "2019-02-05T09:15:35.204Z", "Status": "Rejected", "Permissions": ["ReadTransactionsCredits", "ReadBalances", "ReadAccountsDetail", "ReadStandingOrdersBasic"], "ExpirationDateTime": "2019-04-05T08:15:35.204Z", "TransactionFromDateTime": "2018-08-05T08:15:35.204Z", "TransactionToDateTime": "2019-08-05T08:15:35.204Z" }, "Links": {"Self": "/open-banking-sandbox/v2.0/account-requests/dd5447f7-1898-4855-86a6-fd50d37eb42f"}, "Meta": {} } </pre>


```
"Risk": {}
}
```

This is an example of a GET request which is made **after** the account request resource is Revoked.

Valid Request - The account request has been revoked via the ASPSP interface
Account Request ID = 397a699a-0c6a-486c-b11f-7249cebfe344

Request

Response

Get Account Requests Request

```
GET /account-requests/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Get Account Requests Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "AccountRequestId": "397a699a-0c6a-
486c-b11f-7249cebfe344",
    "CreationDateTime": "2019-02-
05T09:15:35.557Z",
    "Status": "Revoked",
    "Permissions": [
      "ReadTransactionsCredits",
      "ReadBalances",
      "ReadAccountsDetail",
      "ReadStandingOrdersBasic"
    ],
    "ExpirationDateTime": "2019-04-
05T08:15:35.557Z",
    "TransactionFromDate": "2018-08-
05T08:15:35.557Z",
    "TransactionToDate": "2019-08-
05T08:15:35.557Z"
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/account-requests/397a699a-
0c6a-486c-b11f-7249cebfe344"},
  "Meta": {},
  "Risk": {}
}
```

Accounts - Bulk

The call to GET /accounts is the first step after an account-request is authorised.

This will allow the AISP to discover which account (and AccountId value) are associated with the authorisation of consent.

Note: For Accounts Bulk, the Coventry Building Society is not using this feature. We always send only one account for the GET/Accounts Request.

Request

Get Accounts Request

```
GET /accounts HTTP/1.1
Authorization: Bearer Az90SA0Jklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Accounts Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "Account": [
      {
        "AccountId": "88379",
        "Currency": "GBP",
        "Nickname": "Bills",
        "Account": {
          "SchemeName":
"SortCodeAccountNumber",
          "Identification":
"40630112345678",
          "Name": "Mr Kevin"
        }
      }
    ]
  },
  "Links": {
    "Self": "/accounts/"
  }
}
```

Accounts - Specific Account

An AISP can also retrieve the account resource details specifically for AccountId 88379

Request

Get Accounts Request

```
GET /accounts/88379 HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Accounts Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "Account": [
      {
        "AccountId": "88379",
        "Currency": "GBP",
        "Nickname": "Bills",
        "Account": {
          "SchemeName":
"SortCodeAccountNumber",
          "Identification":
"40630112345678",
          "Name": "Mr Kevin
        }
      }
    ]
  },
  "Links": {
    "Self": "/accounts/88379"
  },
  "Meta": {
    "TotalPages": 1
  }
}
```

Balances - Specific Account

Request

Get Account Balances Request

```
GET /accounts/88379/balances HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Balances Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "Balance": [
      {
        "AccountId": "88379",
        "Amount": {
          "Amount": "1230.00",
          "Currency": "GBP"
        },
        "CreditDebitIndicator": "Credit",
        "Type": "InterimAvailable",
        "DateTime": "2017-04-
05T10:43:07+00:00"
      }
    ]
  },
  "Links": {
    "Self": "/accounts/88379/balances/"
  },
  "Meta": {
    "TotalPages": 1
  }
}
```

Beneficiaries - Specific Account Request

Get Account Beneficiaries Request

```
GET /accounts/88379/beneficiaries HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Beneficiaries Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "Beneficiary": [
      {
        "AccountId": "88379",
        "BeneficiaryId": "Ben1",
        "Reference": "Towbar Club",
        "CreditorAccount": {
          "SchemeName":
"SortCodeAccountNumber",
          "Identification":
"80200112345678",
          "Name": "Mrs Juniper"
        }
      }
    ]
  },
  "Links": {
    "Self":
"/accounts/88379/beneficiaries/"
  }
}
```

Direct Debits - Specific Account

Request

Get Account Direct Debits Request

```
GET /accounts/88379/direct-debits HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Direct Debits Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "DirectDebit": [
      {
        "AccountId": "88379",
        "MandateIdentification":
"Caravanners",
        "DirectDebitStatusCode": "Active",
        "Name": "Towbar Club 3 - We Love
Towbars",
        "PreviousPaymentDateTime": "2017-
04-05T10:43:07+00:00",
        "PreviousPaymentAmount": {
          "Amount": "0.57",
          "Currency": "GBP"
        }
      }
    ]
  },
  "Links": {
    "Self": "/accounts/88379/direct-
debits/"
  }
}
```

Product - Specific Account Request

Get Account Product Request

```
GET /accounts/88379/product HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Product Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "Product": [
      {
        "AccountId": "88379",
        "ProductIdentifier": "51B",
        "ProductType": "PCA"
      }
    ]
  },
  "Links": {
    "Self": "/accounts/88379/product"
  },
  "Meta": {
    "TotalPages": 1
  }
}
```

Standing Orders - Specific Account Request

Get Accounts Standing Orders Request

```
GET /accounts/88379/standing-orders
HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Accounts Standing Orders Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "StandingOrder": [
      {
        "AccountId": "88379",
        "Frequency":
        "IntrvlMnthDay:01:02",
        "Reference": "Towbar Club 2 - We
        Love Towbars",
        "FirstPaymentDateTime": "2017-08-
        12T00:00:00+00:00",
        "FirstPaymentAmount": {
          "Amount": "0.57",
          "Currency": "GBP"
        },
        "NextPaymentDateTime": "2017-09-
        12T00:00:00+00:00",
        "NextPaymentAmount": {
          "Amount": "0.56",
          "Currency": "GBP"
        },
        "FinalPaymentDateTime": "2027-01-
        12T00:00:00+00:00",
        "FinalPaymentAmount": {
          "Amount": "0.56",
          "Currency": "GBP"
        },
        "CreditorAccount": {
          "SchemeName":
          "SortCodeAccountNumber",
          "Identification":
          "80200112345678",
          "Name": "Mrs Juniper"
        }
      }
    ]
  },
  "Links": {
    "Self": "/accounts/88379/standing-
    orders/"
  }
}
```


Transactions - Specific Account Request

Get Account Transactions Request

```
GET /accounts/88379/transactions HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Transactions Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "Transaction": [
      {
        "AccountId": "88379",
        "TransactionId": "123",
        "TransactionReference": "Ref 1",
        "Amount": {
          "Amount": "10.00",
          "Currency": "GBP"
        },
        "CreditDebitIndicator": "Credit",
        "Status": "Booked",
        "BookingDateTime": "2017-04-
05T10:43:07+00:00",
        "ValueDateTime": "2017-04-
05T10:45:22+00:00",
        "TransactionInformation": "Cash
from Aubrey",
        "BankTransactionCode": {
          "Code":
"ReceivedCreditTransfer",
          "SubCode":
"DomesticCreditTransfer"
        },
        "ProprietaryBankTransactionCode":
{
          "Code": "Transfer",
          "Issuer": "AlphaBank"
        },
        "Balance": {
          "Amount": {
            "Amount": "230.00",
            "Currency": "GBP"
          },
          "CreditDebitIndicator":
"Credit",
          "Type": "InterimBooked"
        }
      }
    ]
  },
  "Links": {
    "Self":
"/accounts/88379/transactions/"
  }
}
```

Scheduled Payment - Specific Account Request

Get Account Product Request

```
GET /accounts/88379/product HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Product Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {"ScheduledPayment": [ {
    "AccountId": "3701d604-0dce-4509-8aa6-
d7898fc53d5e",
    "ScheduledPaymentDateTime": "2019-02-
06T09:56:21.387Z",
    "ScheduledType": "Execution",
    "InstructedAmount": {
      "Amount": "10.00",
      "Currency": "GBP"
    },
  },
  "CreditorAccount": {
    "SchemeName":
    "UK.OBIE.SortCodeAccountNumber",
    "Identification": "13705492179019"
  }
  }],
  "Links": {"Self": "/open-banking-
sandbox/v2.0/accounts/3701d604-0dce-4509-
8aa6-d7898fc53d5e/scheduled-payments"},
  "Meta": {"TotalPages": 1}
}
```

Delete Account Request

The DELETE /account-requests call allows an AISP to delete a previously created account-request (whether it is currently authorised or not). The PSU may want to remove their consent via the AISP instead of revoking authorisation with CBS.

This API call allows the PSU to revoke consent with the AISP - and for that consent to be reflected in authorisation with CBS.

Request

Delete Account Requests Request

Response

Delete Account Requests Response

```
DELETE /account-requests/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
```

```
HTTP/1.1 204 No Content
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
```

Pagination

The example below illustrates how CBS will return a paginated response.

Request

Paginated Transactions Request

```
GET /accounts/22289/transactions HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Paginated Resource Response

Paginated Transactions Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    ...
  },
  "Links": {
    "Self":
"/accounts/22289/transactions/",
    "Last":
"/accounts/22289/transactions?pg=20",
    "First":
"/accounts/22289/transactions/",
    "Next":
"/accounts/22289/transactions?pg=2"
  },
  "Meta": {
    "TotalPages": 20
  }
}
```

The AISP can follow the links provided in the Links section of the payload to navigate to the first, last, next and previous pages:

Request Next Page of Results

Paginated Resource Response

Paginated Transactions Request (Next)

```
GET /accounts/22289/transactions?pg=2
HTTP/1.1
Authorization: Bearer Az90SAOJklae
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Paginated Transactions Response

```
HTTP/1.1 200 OK
x-jws-signature:
V2hhdCB3ZSBnb3QgaGVyZQ0K..aXMgZmFpbHVyZSB0
byBjb21tdW5pY2F0ZQ0K
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    ...
  },
  "Links": {
    "Self":
"/accounts/22289/transactions?pg=2",
    "Last":
"/accounts/22289/transactions?pg=20",
    "First":
"/accounts/22289/transactions/",
    "Next":
"/accounts/22289/transactions?pg=3",
    "Prev":
"/accounts/22289/transactions?pg=1"
  },
  "Meta": {
    "TotalPages": 20
  }
}
```

Payments Examples

This set of payload examples is for an PISP:

- Setting up a Payment Request for a Single Immediate Payment
- Getting the status of a Payment Request for a Single Immediate Payment
- Setting up a Mandate Payment Request for a Future or Regular Dated Payment
- Getting the status of a Mandate Payment Request for a Regular or Future Dated Payment
- Setting up a Payment Submission Request for a Single Immediate Payment
- Getting the status of a Payment Submission Request for a Single Immediate Payment
- Setting up a Mandate Payment Submission Request for a Future or Regular Dated Payment
- Getting the status of a Mandate Payment Submission for a Regular or Future Dated Payment

In this scenario:

- All permissions have been granted to access all Payment API resources

Setup Payment Request

InstructionIdentification = ACME412

Request

Post Payment Requests Request

```
POST /payments HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-idempotency-key: FRESCO.21302.GFX.20
x-jws-signature:
TGlmZSdzIGegam91cm5leSBub3QgYSBkZXN0aW5hd
GlvbiA=..T2ggZ29vZCBldmVuaW5nIG1yIHR5bGVy
IGdvaW5nIGRvd24gPw==
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
```

Response

Post Payment Requests Response

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "PaymentId": "b58fe243-4189-45bf-
ac69-e071c78bf8ae",
    "CreationDateTime": "2019-03-
07T19:57:59.890Z",
    "Initiation": {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.22",
      "InstructedAmount": {
        "Amount": "0.1",
        "Currency": "GBP"
      },
      "CreditorAccount": {
```

```

Content-Type: application/json
Accept: application/json
{
  "Data": {
    "Initiation": {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.22",
      "InstructedAmount": {
        "Amount": "0.01",
        "Currency": "GBP"
      },
      "CreditorAccount": {
        "SchemeName":
"SortCodeAccountNumber",
        "Identification":
"98149034822195",
        "Name": "Amazon",
        "SecondaryIdentification": "0002"
      },
      "RemittanceInformation": {
        "Reference": "Immediate-Payment",
        "Unstructured": "Internal ops
code 5120101"
      }
    },
    "Risk": {
      "PaymentContextCode":
"EcommerceGoods",
      "MerchantCategoryCode": "5967",
      "MerchantCustomerIdentification":
"053598653254",
      "DeliveryAddress": {
        "AddressLine": [
          "Flat 7",
          "Acacia Lodge"
        ],
        "StreetName": "Acacia Avenue",
        "BuildingNumber": "27",
        "PostCode": "GU31 2ZZ",
        "TownName": "Sparsholt",
        "CountrySubDivision": [
          "Wessex"
        ],
        "Country": "UK"
      }
    }
  }
}

```

```

      "SchemeName":
"SortCodeAccountNumber",
      "Identification":
"98149034822195",
      "Name": "Amazon",
      "SecondaryIdentification":
"0002"
    },
    "RemittanceInformation":
{
      "Unstructured": "Internal ops
code 5120101",
      "Reference": "Immediate-
Payment"
    }
  },
  "Status":
"AcceptedTechnicalValidation"
},
  "Risk": {
    "PaymentContextCode":
"EcommerceGoods",
    "MerchantCategoryCode": "5967",
    "MerchantCustomerIdentification":
"053598653254",
    "DeliveryAddress": {
      "StreetName": "Acacia Avenue",
      "BuildingNumber": "27",
      "PostCode": "GU31 2ZZ",
      "TownName": "Sparsholt",
      "Country": "UK",
      "AddressLine": [
        "Flat 7",
        "Acacia Lodge"
      ],
      "CountrySubDivision": ["Wessex"]
    }
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/payments"},
  "Meta": {}
}

```

Retrieve Status of Previously Setup Payment Request for Single Immediate Payment

This is an example of a GET request which is made when Payment initiation or individual transaction included in the payment initiation is pending. In this Scenario, further checks and status update will be performed

PaymentId = e738bd3b-985b-4c84-9e99-b06a8e8a0c2f

Request

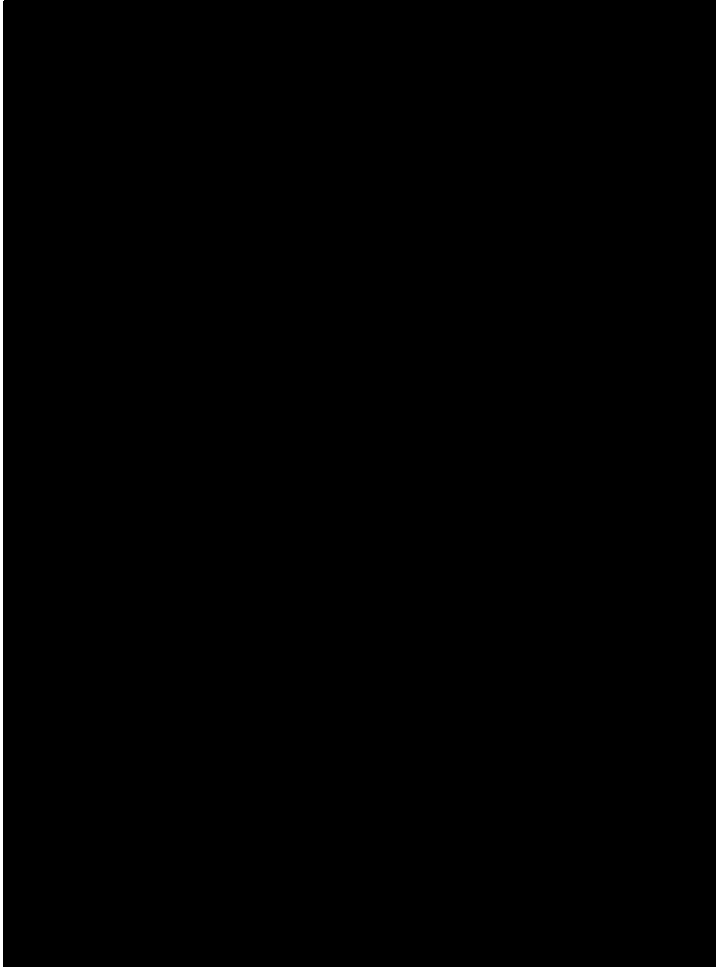
Get Payment Requests Request

```
GET /payments/e738bd3b-985b-4c84-9e99-
b06a8e8a0c2f HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun,
10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Account Requests Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "PaymentId": "e738bd3b-985b-4c84-
9e99-b06a8e8a0c2f",
    "Status": "Pending",
    "CreationDateTime": "2019-02-
04T21:24:58.351Z",
    "Initiation": {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.20",
      "InstructedAmount": {
        "Amount": "165.88",
        "Currency": "GBP"
      },
      "CreditorAgent": {
        "SchemeName": "BICFI",
        "Identification": "SC080800"
      },
      "CreditorAccount": {
        "Identification": "21325698",
        "Name": "ACME Inc",
        "SchemeName":
"SortCodeAccountNumber",
        "SecondaryIdentification":
"0002"
      },
      "DebtorAccount": {
        "Identification": "1254653",
        "Name": "Prince",
        "SchemeName":
"SortCodeAccountNumber",
        "SecondaryIdentification":
"2221"
      },
      "DebtorAgent": {
        "Identification": "54667",
        "SchemeName": "BICFI"
      },
      "RemittanceInformation":
{
        "Reference": "FRESCO-101",
```



```

    "Unstructured": "Internal ops
code 5120101"
  }
},
"Risk": {
  "PaymentContextCode":
"EcommerceGoods",
  "MerchantCategoryCode": "5967",
  "MerchantCustomerIdentification":
"053598653254",
  "DeliveryAddress": {
    "AddressLine": [
      "Flat 7",
      "Acacia Lodge"
    ],
    "StreetName": "Acacia Avenue",
    "BuildingNumber": "27",
    "PostCode": "GU31 2ZZ",
    "TownName": "Sparsholt",
    "CountySubDivision": [
      "Wessex",
      "kessel"
    ],
    "Country": "UK"
  }
},
"Links": {"Self": "/open-banking-
sandbox/v2.0/payments/e738bd3b-985b-4c84-
9e99-b06a8e8a0c2f"},
"Meta": {"TotalPages": 1}
}

```

This is an example of a GET request which is made when Authentication and syntactical and semantic validation are successful.

PaymentId = b58fe243-4189-45bf-ac69-e071c78bf8ae

Request

Response

Get Payment Requests Request

Get Payment Requests Response

```

GET /payments/b58fe243-4189-45bf-ac69-
e071c78bf8ae HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun,
10 Sep 2017 19:43:31 UTC

```

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

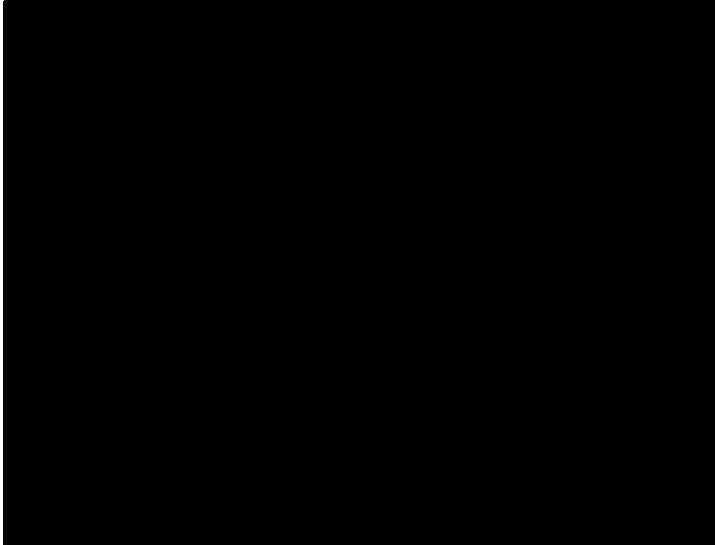
{
  "Data": {

```



```
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

```
"PaymentId": "b58fe243-4189-45bf-
ac69-e071c78bf8ae",
  "Status":
    "AcceptedTechnicalValidation",
    "CreationDateTime": "2019-02-
04T21:15:21.011Z",
    "Initiation": {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.20",
      "InstructedAmount": {
        "Amount": "165.88",
        "Currency": "GBP"
      },
      "CreditorAgent": {
        "SchemeName": "BICFI",
        "Identification": "SC080800"
      },
      "CreditorAccount": {
        "Identification": "21325698",
        "Name": "ACME Inc",
        "SchemeName":
"SortCodeAccountNumber",
        "SecondaryIdentification":
"0002"
      },
      "DebtorAccount": {
        "Identification": "1254653",
        "Name": "Prince",
        "SchemeName":
"SortCodeAccountNumber",
        "SecondaryIdentification":
"2221"
      },
      "DebtorAgent": {
        "Identification": "54667",
        "SchemeName": "BICFI"
      },
      "RemittanceInformation":
{
      "Reference": "FRESCO-101",
      "Unstructured": "Internal ops
code 5120101"
    }
    },
    "Risk": {
      "PaymentContextCode":
"EcommerceGoods",
      "MerchantCategoryCode": "5967",
      "MerchantCustomerIdentification":
"053598653254",
      "DeliveryAddress": {
        "AddressLine": [
          "Flat 7",
          "Acacia Lodge"
```



```

    ],
    "StreetName": "Acacia Avenue",
    "BuildingNumber": "27",
    "PostCode": "GU31 2ZZ",
    "TownName": "Sparsholt",
    "CountySubDivision": [
        "Wessex",
        "kessel"
    ],
    "Country": "UK"
  },
  "Links": {
    "Self": "/open-banking-sandbox/v2.0/payments/b58fe243-4189-45bf-ac69-e071c78bf8ae"
  },
  "Meta": {
    "TotalPages": 1
  }
}

```

This is an example of a GET request which is made when preceding check of technical validation was successful. Customer profile check was also successful.

PaymentId = 4fcbf613-a220-4c1a-ad55-02a12cb02734

Request

Response

Get Payment Requests Request

Get Payment Requests Response

```

GET /payments/4fcbf613-a220-4c1a-ad55-02a12cb02734 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json

```

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "PaymentId": "4fcbf613-a220-4c1a-ad55-02a12cb02734",
    "Status": "AcceptedCustomerProfile",
    "CreationDateTime": "2019-02-04T21:22:55.701Z",
    "Initiation": {
      "InstructionIdentification": "ACME412",
      "EndToEndIdentification": "FRESCO.21302.GFX.20",
      "InstructedAmount": {
        "Amount": "165.88",
        "Currency": "GBP"
      }
    }
  },

```

```
"CreditorAgent": {
  "SchemeName": "BICFI",
  "Identification": "SC080800"
},
"CreditorAccount": {
  "Identification": "21325698",
  "Name": "ACME Inc",
  "SchemeName":
"SortCodeAccountNumber",
  "SecondaryIdentification":
"0002"
},
"DebtorAccount": {
  "Identification": "1254653",
  "Name": "Prince",
  "SchemeName":
"SortCodeAccountNumber",
  "SecondaryIdentification":
"2221"
},
"DebtorAgent": {
  "Identification": "54667",
  "SchemeName": "BICFI"
},
"RemittanceInformation":
{
  "Reference": "FRESCO-101",
  "Unstructured": "Internal ops
code 5120101"
}
},
"Risk": {
  "PaymentContextCode":
"EcommerceGoods",
  "MerchantCategoryCode": "5967",
  "MerchantCustomerIdentification":
"053598653254",
  "DeliveryAddress": {
    "AddressLine": [
      "Flat 7",
      "Acacia Lodge"
    ],
    "StreetName": "Acacia Avenue",
    "BuildingNumber": "27",
    "PostCode": "GU31 2ZZ",
    "TownName": "Sparsholt",
    "CountySubDivision": [
      "Wessex",
      "kessel"
    ],
    "Country": "UK"
  }
},
"Links": {"Self": "/open-banking-
sandbox/v2.0/payments/4fcbf613-a220-4c1a-
ad55-02a12cb02734"},
```

```
"Meta": {}
}
```

This is an example of a GET request which is made when Payment initiation or individual transaction included in the payment initiation has been rejected.

```
PaymentId = ea5a0af4-f058-40c3-be52-f93391571222
```

Request

Get Payment Requests Request

```
GET /payments/ea5a0af4-f058-40c3-be52-
f93391571222 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun,
10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Payment Requests Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "PaymentId": "ea5a0af4-f058-40c3-
be52-f93391571222",
    "Status": "Rejected",
    "CreationDateTime": "2019-02-
04T21:26:37.273Z",
    "Initiation": {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.20",
      "InstructedAmount": {
        "Amount": "165.88",
        "Currency": "GBP"
      },
      "CreditorAgent": {
        "SchemeName": "BICFI",
        "Identification": "SC080800"
      },
      "CreditorAccount": {
        "Identification": "21325698",
        "Name": "ACME Inc",
        "SchemeName":
"SortCodeAccountNumber",
        "SecondaryIdentification":
"0002"
      },
      "DebtorAccount": {
        "Identification": "1254653",
        "Name": "Prince",
        "SchemeName":
"SortCodeAccountNumber",
```

```

        "SecondaryIdentification":
"2221"
    },
    "DebtorAgent": {
        "Identification": "54667",
        "SchemeName": "BICFI"
    },
    "RemittanceInformation":
{
    "Reference": "FRESCO-101",
    "Unstructured": "Internal ops
code 5120101"
}
},
    "Risk": {
        "PaymentContextCode":
"EcommerceGoods",
        "MerchantCategoryCode": "5967",
        "MerchantCustomerIdentification":
"053598653254",
        "DeliveryAddress": {
            "AddressLine": [
                "Flat 7",
                "Acacia Lodge"
            ],
            "StreetName": "Acacia Avenue",
            "BuildingNumber": "27",
            "PostCode": "GU31 2ZZ",
            "TownName": "Sparsholt",
            "CountySubDivision": [
                "Wessex",
                "kessel"
            ],
            "Country": "UK"
        }
    },
    "Links": {"Self": "/open-banking-
sandbox/v2.0/payments/ea5a0af4-f058-40c3-
be52-f93391571222"},
    "Meta": {}
}

```

Setup Payment Submission Request for Single or Immediate Payments

InstructionIdentification = ACME412

Request

Post Payment Submission Requests Request

Response

Post Payment Submission Requests Response

```
POST /payment-submissions HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-idempotency-key: FRESCO.21302.GFX.20
x-jws-signature:
TGlmZSdzIGEGam91cm5leSBub3QgYSBkZXN0aW5hd
GlvbiA=..T2ggZ29vZCBldmVuaW5nIG1yIHR5bGVy
IGdvaW5nIGRvd24gPw==
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
X-Client-Id: tppclientid
Accept: application/json
{
  "Data": {
    "PaymentId":
    "b58fe243-4189-45bf-ac69-e071c78bf8ae",
    "Initiation":
    {
      "InstructionIdentification":
      "ACME412",
      "EndToEndIdentification":
      "FRESCO.21302.GFX.22",
      "InstructedAmount": {
        "Amount":
        "0.01",
        "Currency": "GBP"
      },
      "CreditorAccount"
      : {
        "SchemeName":
        "SortCodeAccountNumber",
        "Identification":
        "98149034822195",
        "Name":
        "Amazon",
        "SecondaryIdentification":
        "0002"
      },
      "RemittanceInformation": {
        "Reference":
        "Immediate-Payment",
```

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "PaymentSubmissionId": "58923-001",
    "PaymentId": "b58fe243-4189-45bf-
ac69-e071c78bf8ae",
    "Status": "AcceptedSettlementInProces
s",
    "CreationDateTime": "2017-06-
05T15:15:22+00:00"
  },
  "Links": {
    "Self": "open-banking-
sandbox/v2.0/payment-submissions/58923-
001"
  },
  "Meta": {}
}
```

```

    "Unstructured":
    "Internal ops code 5120101"
      }
    }
  },
  "Risk": {
    "PaymentContextCode":
    "EcommerceGoods",
    "MerchantCategoryCode": "5967",
    "MerchantCustomerIdentification":
    "053598653254",

    "DeliveryAddress"
    : {

    "AddressLine": [
      "Flat 7",
      "Acacia
Lodge"
    ],
    "StreetName": "Acacia Avenue",
    "BuildingNumber": "27",
    "PostCode": "GU31 2ZZ",
    "TownName": "Sparsholt",
    "CountrySubDivision": [
      "Wessex"
    ],
    "Country":
    "UK"
  }
}
}
}

```

Retrieve Status of Previously Setup Payment Request for Single Immediate Payment

This is an example of a GET request which is made when Payment initiation or individual transaction included in the payment initiation is pending. Further checks and status update will be performed.

PaymentSubmissionId = e0c20b12-3ff6-4aad-90dd-30a70fe22ac5
--

Request

Get Payment Submissions Requests Request

```

GET /payment-submissions/e0c20b12-3ff6-
4aad-90dd-30a70fe22ac5 HTTP/1.1
Authorization: Bearer Jhingapulaav

```

Response

Get Payment Submissions Status Response

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{

```

```
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun,
10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

```
"Data": {
  "PaymentSubmissionId": "e0c20b12-
3ff6-4aad-90dd-30a70fe22ac5",
  "PaymentId": "b58fe243-4189-45bf-
ac69-e071c78bf8ae",
  "CreationDateTime": "2019-03-
08T10:18:51Z",
  "Status": "Pending"
},
"Links": {"Self": "/open-banking-
sandbox/v2.0/payment-submissions/e0c20b12-
3ff6-4aad-90dd-30a70fe22ac5"},
"Meta": {}
}
```

This is an example of a GET request which is made when Payment initiation or individual transaction included in the payment initiation has been rejected.

PaymentSubmissionId = cf66b703-05d9-40fc-9b19-67460b1f1166

Request

Get Payment Submissions Requests Request

```
GET /payment-submissions/cf66b703-05d9-
40fc-9b19-67460b1f1166 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun,
10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Payment Submissions Status Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "PaymentSubmissionId":
"cf66b703-05d9-40fc-9b19-67460b1f1166",
    "PaymentId": "b58fe243-4189-45bf-
ac69-e071c78bf8ae",
    "CreationDateTime": "2019-03-
08T10:18:51Z",
    "Status": "Rejected"
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/payment-submissions/
cf66b703-05d9-40fc-9b19-67460b1f1166"},
  "Meta": {}
}
```


This is an example of a GET request which is made when all preceding checks such as technical validation and customer profile were successful and therefore the payment initiation has been accepted for execution.

PaymentSubmissionId = 2a087a2d-47cd-4054-90b2-2f320cc654f2

Request

Get Payment Submissions Requests Request

```
GET /payment-submissions/2a087a2d-47cd-4054-90b2-2f320cc654f2 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Payment Submissions Status Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "PaymentSubmissionId": "2a087a2d-47cd-4054-90b2-2f320cc654f2",
    "PaymentId": "b58fe243-4189-45bf-ac69-e071c78bf8ae",
    "CreationDateTime": "2019-03-08T10:18:51Z",
    "Status": "AcceptedSettlementInProgress"
  },
  "Links": {"Self": "/open-banking-sandbox/v2.0/payment-submissions/2a087a2d-47cd-4054-90b2-2f320cc654f2"},
  "Meta": {}
}
```

This is an example of a GET request which is made when settlement on the debtor's account has been completed.

PaymentSubmissionId = 6c20f296-77a8-4bbc-98da-b8c130c421a7

Request

Get Payment Submissions Requests Request

```
GET /payment-submissions/6c20f296-77a8-4bbc-98da-b8c130c421a7 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC
```

Response

Get Payment Submissions Status Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "PaymentSubmissionId": "6c20f296-77a8-4bbc-98da-b8c130c421a7",
    "PaymentId": "b58fe243-4189-45bf-
```

```
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

```
ac69-e071c78bf8ae",
  "CreationDateTime": "2019-03-
08T10:18:51Z",
  "Status": "AcceptedSettlement
Completed"
},
  "Links": {"Self": "/open-banking-
sandbox/v2.0/payment-submissions/
6c20f296-77a8-4bbc-98da-b8c130c421a7"},
  "Meta": {}
}
```

Mandates Examples

Setup Mandate Payment Request for Future or Regular Dated Payment

InstructionIdentification = ACME412

Future Dated Payments

Request

Post Mandate Requests Request

```
POST /open-banking-extensions-
Sandbox/v2.0/mandates
HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-idempotency-key: FRESCO.21302.GFX.20
x-jws-signature:
TGlmZSdzIGegam91cm5leSBub3QgYSBkZXN0aW5hd
GlvbIA=..T2ggZ29vZCBldmVuaW5nIG1yIHR5bGVy
IGdvaW5nIGRvd24gPw==
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
X-Client-Id: tppclientid
Accept: application/json
```

Response

Post Mandate Requests Response

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "PaymentId": "172cf16c-5819-40cf-aa3
fae1b16b8f4",
    "CreationDateTime": "2019-03-
07T21:05:22.223Z",
    "Initiation":
    {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.20",
      "InstructedAmount":
    {
      "Amount":
"165.88",
      "Currency": "GBP"
    }
    }
  }
}
```

```

{
  Data: {
    PaymentType: "FutureDated",
    Initiation: {
      InstructionIdentification: "ACME412",
      EndToEndIdentification:
        "FRESCO.21302.GFX.20",
      FuturePaymentDateTime: "2019-12-25",
      InstructedAmount: {
        Amount: "165.88",
        Currency: "GBP"
      },
      CreditorAccount: {
        SchemeName: "SortCodeAccountNumber",
        Identification: "08080021325698",
        Name: "ACME Inc",
        SecondaryIdentification: "0002"
      },
      RemittanceInformation: {
        Reference: "FRESCO-101",
        Unstructured: "Internal ops code 5120101"
      }
    },
    Risk: {
      PaymentContextCode: "EcommerceGoods",
      MerchantCategoryCode: "5967",
      MerchantCustomerIdentification:
        "053598653254",
      DeliveryAddress: {
        AddressLine: [
          Flat 7,
          Acacia Lodge
        ],
        StreetName: "Acacia Avenue",
        BuildingNumber: "27",
        PostCode: "GU31 2ZZ",
        TownName: "Sparsholt",
        CountrySubDivision: [
          Wessex
        ],
        Country: "UK"
      }
    }
  },
  "CreditorAccount":
  {
    "SchemeName":
      "SortCodeAccountNumber",
    "SortCodeAccountNumber":
      "08080021325698",
    "Identification":
      "08080021325698",
    "Name":
      "ACME Inc",
    "SecondaryIdentification":
      "0002"
  },
  "FuturePaymentDateTime":
    "2019-03-08",
  "RemittanceInformation":
  {
    "Reference":
      "FRESCO-101",
    "Unstructured": "Internal
ops code 5120101"
  },
  "Status":
    "AcceptedTechnicalValidation",
  "PaymentType":
    "FutureDated"
  },
  "Risk":
  {
    "PaymentContextCode":
      "EcommerceGoods",
    "MerchantCategoryCode":
      "5967",
    "MerchantCustomerIdentification":
      "053598653254",
    "DeliveryAddress":
    {
      "AddressLine":
      [
        "Flat 7",
        "Acacia
Lodge"
      ],
      "StreetName":
        "Acacia Avenue",
      "BuildingNumber":
        "27",
      "PostCode":
        "GU31 2ZZ",
      "TownName":
        "Sparsholt",
    }
  }
}

```

```

"CountrySubDivision":
  ["Wessex"],
    "Country":
      "UK"
    }
  },
  "Links": {"Self": "/open-banking-
extensions-sandbox/v2.0/mandates"},
  "Meta": {}
}

```

Regular Dated Payments

Request

Post Mandate Requests Request

```

POST/open-banking-extensions-Sandbox/
/mandates
HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-idempotency-key: FRESCO.21302.GFX.20
x-jws-signature:
TGlmZSdzIGegam91cm5leSBub3QgYSBkZXN0aW5hd
GlvbiA=..T2ggZ29vZCBldmVuaW5nIG1yIHR5bGVy
IGdvaW5nIGRvd24gPw==
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
X-Client-Id: tppclientid
Content-Type: application/json
Accept: application/json
{
  Data: {
    PaymentType: "Regular",
    Initiation: {
      InstructionIdentification: "ACME412",
      Frequency: "Monthly",
      EndToEndIdentification:
        "FRESCO.21302.GFX.20",

```

Response

Post Mandate Requests Response

```

HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "PaymentId": "219e279c-a1dd-4f80-9d5
822853dff653",
    "CreationDateTime": "2019-03-
07T21:05:37.599Z",
    "Initiation":
    {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.20",
      "InstructedAmount":
    {
      "Amount":
"165.88",
      "Currency":
"GBP"
    },
    "CreditorAccount":
    {
      "SchemeName":
"SortCodeAccountNumber",
      "Identification":
"08080021325698",
      "Name":
"ACME Inc",
      "SecondaryIdentification": "0002"

```

```

ValidFromDateTime: "2019-03-10T00:00:00+00:00",
ValidToDateTime: "2019-04-08T00:00:00+00:00",
InstructedAmount: {
  Amount: "165.88",
  Currency: "GBP"
},
CreditorAccount: {
  SchemeName: "SortCodeAccountNumber",
  Identification: "08080021325698",
  Name: "ACME Inc",
  SecondaryIdentification: "0002"
},
RemittanceInformation: {
  Reference: "FRESCO-101",
  Unstructured: "Internal ops code 5120101"
}
},
Risk: {
  PaymentContextCode: "EcommerceGoods",
  MerchantCategoryCode: "5967",
  MerchantCustomerIdentification: "053598653254",
  DeliveryAddress: {
    AddressLine: [
      Flat 7,
      Acacia Lodge
    ],
    StreetName: "Acacia Avenue",
    BuildingNumber: "27",
    PostCode: "GU31 2ZZ",
    TownName: "Sparsholt",
    CountrySubDivision: [
      Wessex
    ],
    Country: "UK"
  }
}
},
ValidFromDateTime": "2019-03-08T09:23:39.527Z",
"Frequency": "Monthly",
"ValidToDateTime": "2019-04-06T09:23:39.527Z",
"RemittanceInformation": {
  "Reference": "FRESCO-101",
  "Unstructured": "Internal ops code 5120101"
},
"Status": "AcceptedTechnicalValidation",
"PaymentType": "Regular"
},
"Risk": {
  "PaymentContextCode": "EcommerceGoods",
  "MerchantCategoryCode": "5967",
  "MerchantCustomerIdentification": "053598653254",
  "DeliveryAddress": {
    "AddressLine": [
      "Flat 7",
      "Acacia Lodge"
    ],
    "StreetName": "Acacia Avenue",
    "BuildingNumber": "27",
    "PostCode": "GU31 2ZZ",
    "TownName": "Sparsholt",
    "CountrySubDivision": ["Wessex"],
    "Country": "UK"
  }
},
"Links": {"Self": "/open-banking-extensions-sandbox/v2.0/mandates"},
"Meta": {}
}

```

Retrieve Status of Previously Setup Mandate Request for Future or Regular Dated Payments

This is an example of a GET request which is made when authentication and syntactical and semantic validation are successful

```
PaymentId = 219e279c-a1dd-4f80-9d52-822853dff653
```

Request

Response

Get Mandate Status Request

```
GET /mandates/219e279c-a1dd-4f80-9d52-822853dff653
HTTP/1.1
Authorization: Bearer
Jhingapulaav
x-fapi-financial-id:
OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017
19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id:
93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Get Mandate Status Request Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "CreationDateTime": "2019-02-04T19:39:58.578Z",
    "Initiation": {
      "CreditorAccount": {
        "Identification": "08080021325698",
        "Name": "Bob Smith",
        "SchemeName": "SortCodeAccountNumber",
        "SecondaryIdentification": "DMBM511"
      },
      "EndToEndIdentification": "FRDFYH.21302.GFX.44",
      "InstructedAmount": {
        "Amount": "50.55",
        "Currency": "GBP"
      },
      "InstructionIdentification": "ANSM023",
      "CreditorAgent": {
        "Identification": "ANM345",
        "SchemeName": "BICFI"
      },
      "DebtorAccount": {
        "Name": "Andrea Hughes",
        "SchemeName": "SortCodeAccountNumber",
        "SecondaryIdentification": "BMDM542",
```

```

        "Identification":
"11280001234567"
    },
    "DebtorAgent": {
        "Identification": "JHK267",
        "SchemeName": "BICFI"
    },
    "RemittanceInformation": {
        "Reference": "FRDFYH-044",
        "Unstructured":
"Internal code 2345612"
    },
    "Frequency": "Monthly"
    },
    "PaymentId": "20dc7be4-5c28-
48a9-a1eb-670e76e741f5",
    "Status": "AcceptedTechnicalValidation",
    "PaymentType": "Regular"
    },
    "Links": {"Self": "/open-banking-
extensions-sandbox/v2.0/mandates/
219e279c-a1dd-4f80-9d52-822853dff653"},
    "Meta": {},
    "Risk": {
        "DeliveryAddress": {
            "BuildingNumber": "27",
            "Country": "UK",
            "CountrySubDivision": ["Wessex"],
            "PostCode": "GU31 2ZZ",
            "StreetName": "Acacia Avenue",
            "TownName": "Sparsholt",
            "AddressLine":
["\Flat 7\", \"Acacia Lodge\"]
        },
        "MerchantCategoryCode": "5967",
        "MerchantCustomerIdentification":
"053598653254",
        "PaymentContextCode":
"EcommerceGoods"
    }
}

```

This is an example of a GET request which is made when Authentication and syntactical and semantic validation are successful.

PaymentId = 20dc7be4-5c28-48a9-a1eb-670e76e741f5

Request

Get Mandates Requests Request

```
GET /mandates/20dc7be4-5c28-48a9-a1eb-670e76e741f5 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Mandates Requests Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "CreationDateTime": "2019-02-04T19:39:58.578Z",
    "Initiation": {
      "CreditorAccount": {
        "Identification": "08080021325698",
        "Name": "Bob Smith",
        "SchemeName": "SortCodeAccountNumber",
        "SecondaryIdentification": "DMBM511"
      },
      "EndToEndIdentification": "FRDFYH.21302.GFX.44",
      "InstructedAmount": {
        "Amount": "50.55",
        "Currency": "GBP"
      },
      "InstructionIdentification": "ANSM023",
      "CreditorAgent": {
        "Identification": "ANM345",
        "SchemeName": "BICFI"
      },
      "DebtorAccount": {
        "Name": "Andrea Hughes",
        "SchemeName": "SortCodeAccountNumber",
        "SecondaryIdentification": "BMDM542",
        "Identification": "11280001234567"
      },
      "DebtorAgent": {
        "Identification": "JHK267",
        "SchemeName": "BICFI"
      }
    }
  }
}
```



```

    },
    "RemittanceInformation": {
      "Reference": "FRDFYH-044",
      "Unstructured":
"Internal code 2345612"
    },
    "Frequency": "Monthly"
  },
  "PaymentId": "20dc7be4-5c28-48a9-a1eb-670e76e741f5",
  "Status": "AcceptedCustomerProfile",
  "PaymentType": "Regular"
},
"Links": {"Self": "/open-banking-extensions-sandbox/v2.0/mandates/20dc7be4-5c28-48a9-a1eb-670e76e741f5"},
"Meta": {},
"Risk": {
  "DeliveryAddress": {
    "BuildingNumber": "27",
    "Country": "UK",
    "CountrySubDivision": ["Wessex"],
    "PostCode": "GU31 2ZZ",
    "StreetName": "Acacia Avenue",
    "TownName": "Sparsholt",
    "AddressLine":
["\"Flat 7\", \"Acacia Lodge\""]
  },
  "MerchantCategoryCode": "5967",
  "MerchantCustomerIdentification":
"053598653254",
  "PaymentContextCode":
"EcommerceGoods"
}
}

```

Setup Mandate Submission Request for Regular or Future Dated Payments

InstructionIdentification = ACME412

Request

Post Mandate Submission Requests Request

```

POST /mandate-submissions HTTP/1.1
Authorization: Bearer Jhingapulaav

```

Response

Post Mandate Submission Requests Response

```

HTTP/1.1 201 Created

```

```
x-idempotency-key: FRESNO.1317.GFX.22
x-jws-signature:
TGlmZSdzIGEgam91cm5leSBub3QgYSBkZXN0aW5hd
GlvbiA=..T2ggZ29vZCBldmVuaW5nIG1yIHR5bGVy
IGdvaW5nIGRvd24gPw==
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
X-Client-Id: tppclientid
Accept: application/json
```

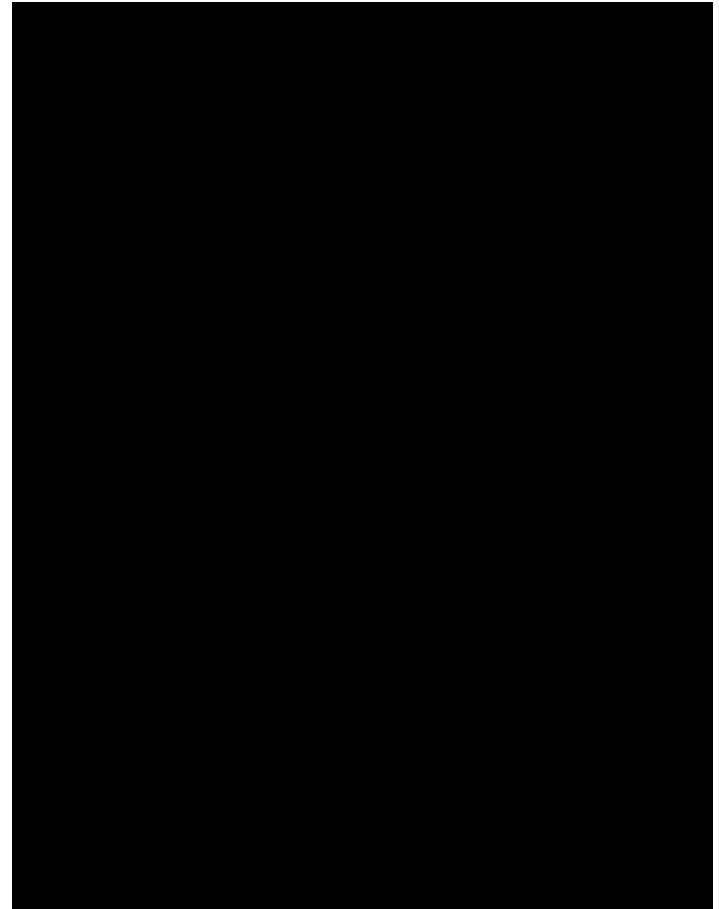
```
{
  "Data": {
    "PaymentId": "172cf16c-5819-40cf-
aa3f-faeelb16b8f4",
    "PaymentType": "FutureDated",
    "Initiation": {
      "InstructionIdentification":
"ACME412",
      "EndToEndIdentification":
"FRESCO.21302.GFX.20",
      "InstructedAmount": {
        "Amount": "165.88",
        "Currency": "GBP"
      },
      "CreditorAccount": {
        "SchemeName":
"SortCodeAccountNumber",
        "Identification":
"08080021325698",
        "Name": "ACME Inc",
        "SecondaryIdentification": "0002"
      },
      "RemittanceInformation": {
        "Reference": "FRESCO-101",
        "Unstructured": "Internal ops
code 5120101"
      }
    }
  }
}
```

```
x-fapi-interaction-id: 93bac548-d2de-
4546-b106-880a5018460d
Content-Type: application/json
{ "Data": { "CreationDateTime": "2017-08-
08T15:23:02Z", "PaymentId": "172cf16c-
5819-40cf-aa3f-faeelb16b8f4",
"PaymentSubmissionId": "13b27b4a-78b1-
487c-9c08-522167d2eba7", "Status":
"AcceptedSettlementInProgress" }, "Links":
{ "Self": "/open-banking-
sandbox/v2.0/mandate-submissions/" },
"Meta": {}}
```

```

},
"Risk": {
  "PaymentContextCode":
"EcommerceGoods",
  "MerchantCategoryCode": "5967",
  "MerchantCustomerIdentification":
"053598653254",
  "DeliveryAddress": {
    "AddressLine": [
      "Flat 7",
      "Acacia Lodge"
    ],
    "StreetName": "Acacia Avenue",
    "BuildingNumber": "27",
    "PostCode": "GU31 2ZZ",
    "TownName": "Sparsholt",
    "CountySubDivision": [
      "Wessex"
    ],
    "Country": "UK"
  }
}
}

```



Retrieve Status of Previously Setup Mandate Request for Future Dated Mandate

This is an example of a GET request which is made when Mandate initiation or individual transaction included in the mandate initiation is pending. Further checks and status update will be performed.

PaymentSubmissionId = 20a95b03-8bd7-489d-ae13-68b3676553ce
--

Request

Response

Get Payment Requests Request

```

GET /mandate-submissions/75e48600-7236-4e97-95c2-25f24a08f478 HTTP/1.1
Authorization: Bearer Jhingapulaav
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun,
10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99

```

Get Account Requests Response

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "PaymentSubmissionId":
"75e48600-7236-4e97-95c2-25f24a08f478",
    "PaymentId":
"b58fe243-4189-45bf-ac69-e071c78bf8ae",
    "Status": "Pending",

```

```
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

```
"CreationDateTime":
"2019-03-07T16:56:22.726Z"
},
"Links": {"Self": "/open-banking-
extensions-sandbox/v2.0/
mandate-submissions/
75e48600-7236-4e97-95c2-25f24a08f478"}
}
```

Confirmation of Funds Examples

This set of payload examples is for an CBPII:

- Setting up a Funds Confirmation Consent Request Resource
- Getting the status of a Funds Confirmation Request Resource
- Get Funds Confirmation on specific Account

Setup Funds Confirmation Request

Request

Post Funds Confirmation Requests Request

```
POST / funds-confirmation-consents
HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json
X-Client-Id: tppclientid
Accept: application/json

{
  "Data": {
    "DebtorAccount": {
      "SchemeName":
"SortCodeAccountNumber",
      "Identification":
"01010112345678",
```

Response

Post Funds Confirmation Requests Response

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "ConsentId":
"514c144a-3288-4d04-b467-61772f80db27",
    "CreationDateTime":
"2019-03-08T12:57:55Z",
    "Status": "AwaitingAuthorisation",
    "StatusUpdateDateTime":
"2019-03-08T12:57:55Z",
    "DebtorAccount": {
      "SchemeName":
"SortCodeAccountNumber",
      "Identification":
"01010112345678",
      "SecondaryIdentification":
"Roll 56988"
    },
    "ExpirationDateTime": "2019-03-09T00:00:00+00:00"
```

```

    "SecondaryIdentification":
    "Roll 56988"
  },
  "ExpirationDateTime":
  "2019-03-09T00:00:00+00:00"
}
},
"Links": {"Self":
"/open-banking/v2.0/funds-confirmation-
consents"},
"Meta": {}
}
}

```

Status - Awaiting Authorisation

This is an example of a GET request which is made **before** the Funds Confirmation Consent request resource is yet to be Authorised.

ConsentId	514c144a-3288-4d04-b467-61772f80db27
-----------	--------------------------------------

Request

Response

Get Funds Confirmation Consent Requests

```

GET / funds-confirmation-consents
/514c144a-3288-4d04-b467-61772f80db27
HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10
Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json

```

Get Funds Confirmation Consent Requests

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "ConsentId": " 514c144a-3288-4d04-b46
61772f80db27",
    "CreationDateTime":
    "2019-03-08T12:57:55Z",
    "Status": "AwaitingAuthorisation",
    "StatusUpdateDateTime": "2019-03-
08T12:57:55Z",
    "DebtorAccount": {
      "SchemeName":
      "SortCodeAccountNumber",
      "Identification":
      "01010112345678",
      "SecondaryIdentification":
      "Roll 56988"
    },
    "ExpirationDateTime": "2019-03-
09T00:00:00+00:00"
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/funds-confirmation-
consents
/514c144a-3288-4d04-b467-61772f80db27"},
  "Meta": {}
}

```

Status - Authorised

This is an example of a GET request which is made **before** the Funds Confirmation Consent request resource is yet to be Authorised.

Valid Request - The CBS customer has Authorized the Request
Funds Confirmation Consent ID = 625d255b-3288-4d04-b467-61772f80db27

Request

Get Funds Confirmation Consent Requests

```
GET / funds-confirmation-consents
/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time:
Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-
d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Response

Get Funds Confirmation Consent Requests

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-
880a5018460d
Content-Type: application/json
{
  "Data": {
    "ConsentId": "625d255b-3288-4d04-b467-61772f80
    "CreationDateTime": "2019-03-08T13:19:54Z",
    "Status": "Authorised",
    "StatusUpdateDateTime": "2019-03-08T13:19:54Z",
    "DebtorAccount": {
      "SchemeName": "UK.OBIE.IBAN",
      "Identification": "GB76LOYD30949301273801",
      "SecondaryIdentification": "Roll 56988"
    },
    "ExpirationDateTime": "2019-03-08T13:19:54Z"
  },
  "Links": {"Self": "/open-banking-sandbox/v2.0/fund
consents/625d255b-3288-4d04-b467-61772f80db27"},
  "Meta": {}
}
```

Post Funds Request

This is an example of a Funds Confirmation request which is made **after** the Funds Confirmation Consent request resource has been authorized. This responds with a Yes or No enumeration..

Valid Request - The CBS customer has Authorized the Request
Amount is= £200.00
Funds Confirmation Consent Request ID = fad4af2f-0389-4895-9e1c-ca470c5d2940

Request

Post Funds Confirmation Requests

Response

Post Funds Confirmation Response

```

POST /funds-confirmation/ HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time:
Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id: 93bac548-
d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json

```

```

{
  "Data": {
    "ConsentId":
"514c144a-3288-4d04-b467-
61772f80db27",
    "Reference": "Purchase01",
    "InstructedAmount": {
      "Amount": "200.00",
      "Currency":
"GBP"
    }
  }
}

```

```

HTTP/1.1 201 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-
880a5018460d
Content-Type: application/json
{
  "Data": {
    "FundsConfirmationId": "fad4af2f-0389-4895-
9e1c-ca470c5d2940",
    "ConsentId": "514c144a-3288-4d04-b467-
61772f80db27",
    "CreationDateTime": "2019-03-08T12:41:07Z",
    "FundsAvailable": "Yes",
    "Reference": "Purchase01",
    "InstructedAmount": {
      "Amount": "200.00",
      "Currency": "GBP"
    }
  },
  "Links": {"Self": "/open-banking-
sandbox/v2.0/funds-confirmations"},
  "Meta": {}
}

```

This is an example of a POST request which is made **when** the Funds Confirmation request resource has been Authorised.

Valid Request - The CBS customer has Authorized the Request Amount is < £200.00 Funds Confirmation Consent Request ID = fad4af2f-0389-4895-9e1c-ca470c5d2940
--

Request

Response

Post Funds Confirmation Requests

```

POST/ funds-confirmation
/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id:
OB/2017/001
x-fapi-customer-last-logged-
time: Sun, 10 Sep 2017
19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id:
93bac548-d2de-4546-b106-
880a5018460d

```

Post Funds Confirmation Response

```

HTTP/1.1 201 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-
880a5018460d
Content-Type: application/json
{
  "Data": {
    "FundsConfirmationId": "fad4af2f-0389-4895-
9e1c-ca470c5d2940",
    "ConsentId": "514c144a-3288-4d04-b467-
61772f80db27",
    "CreationDateTime": "2019-03-08T12:47:04Z",
    "FundsAvailable": "Yes",

```

```
X-Client-Id: tppclientid
Accept: application/json

{
  "Data": {
    "ConsentId": "514c144a-361772f80db27",
    "Reference": "Purchase01"
  }
}
```

```
"Reference": "Purchase01",
  "InstructedAmount": {
    "Amount": "150.00",
    "Currency": "GBP"
  }
},
"Links": {"Self": "/open-banking-sandbox/v2.0/funds-confirmations"},
"Meta": {}
}
```

This is an example of a POST request which is made **when** the Funds Confirmation request resource has been Authorised.

Valid Request - The CBS customer has Authorized the Request
 Amount is > £200.00
 Funds Confirmation Consent Request ID = fad4af2f-0389-4895-9e1c-ca470c5d2940

Request

Response

Post Funds Confirmation Requests

```
POST/ funds-confirmation
/88379 HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id:
OB/2017/001
x-fapi-customer-last-logged-
time: Sun, 10 Sep 2017
19:43:31 UTC
x-fapi-customer-ip-address:
104.25.212.99
x-fapi-interaction-id:
93bac548-d2de-4546-b106-
880a5018460d
X-Client-Id: tppclientid
Accept: application/json

{
  "Data": {
    "ConsentId": "514c144a-361772f80db27",
    "Reference": "Purchase01"
  }
}
```

Post Funds Confirmation Response

```
HTTP/1.1 201 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {
    "FundsConfirmationId": "fad4af2f-0389-4895-9e1c-ca470c5d2940",
    "ConsentId": "514c144a-3288-4d04-b467-61772f80db27",
    "CreationDateTime": "2019-03-08T12:42:46Z",
    "FundsAvailable": "No",
    "Reference": "Purchase01",
    "InstructedAmount": {
      "Amount": "300.00",
      "Currency": "GBP"
    }
  }
},
"Links": {"Self": "/open-banking-sandbox/v2.0/funds-confirmations"},
"Meta": {}
}
```



```

"Currency":
"GBP"
  }
}
}

```

Payment Funds Confirmation

This is an example of GET request which is made **when** the PISP requests Funds Confirmation request resource with a Authorized Payment ID as the Consent ID.

Valid Request - The CBS customer has Authorized the Request and has sufficient funds Amount is > £200.00 Funds Confirmation Consent Request ID = c1e09335-ad8b-440f-8557-c38163984cca

Request

Response

Post Funds Confirmation Requests

```

GET /Payments/c1e09335-ad8b-440f-8557-c38163984cca/funds-confirmation/ HTTP/1.1
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json

```

Post Funds Confirmation Response

```

HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json

{
  "Data": {"FundsAvailableResult": {
    "FundsAvailableDateTime": "2019-03-07T16:04:25.4"
    "FundsAvailable": "Yes"
  }},
  "Links": {"Self": "/open-banking-sandbox/v2.0/payments/c1e09335-ad8b-440f-8557-c38163984cca/funds-confirmation"},
  "Meta": {}
}

```

This is an example of GET request which is made **when** the PISP requests Funds Confirmation request resource with a Authorized Payment ID as the Consent ID.

Valid Request - The CBS customer has Authorized the Request but does not have sufficient funds Amount is < £200.00 Funds Confirmation Consent Request ID = 50a23b84-9675-458d-b327-5e05fba1bda7

Request

Response

Post Funds Confirmation Requests

```
GET/Payments/50a23b84-9675-458d-b327-5e05fba1bda7funds-confirmation HTTP/1.1
Authorization: Bearer 2YotnFZFEjrlzCsicMWpAA
x-fapi-financial-id: OB/2017/001
x-fapi-customer-last-logged-time: Sun, 10 Sep 2017 19:43:31 UTC
x-fapi-customer-ip-address: 104.25.212.99
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
X-Client-Id: tppclientid
Accept: application/json
```

Post Funds Confirmation Response

```
HTTP/1.1 200 OK
x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460d
Content-Type: application/json
{
  "Data": {"FundsAvailableResult": {
    "FundsAvailableDateTime": "2019-03-07T16:07:53.654Z",
    "FundsAvailable": "No"
  }},
  "Links": {"Self": "/open-banking-sandbox/v2.0/payments/50a23b84-9675-458d-b327-5e05fba1bda7/funds-confirmation"},
  "Meta": {}
}
```

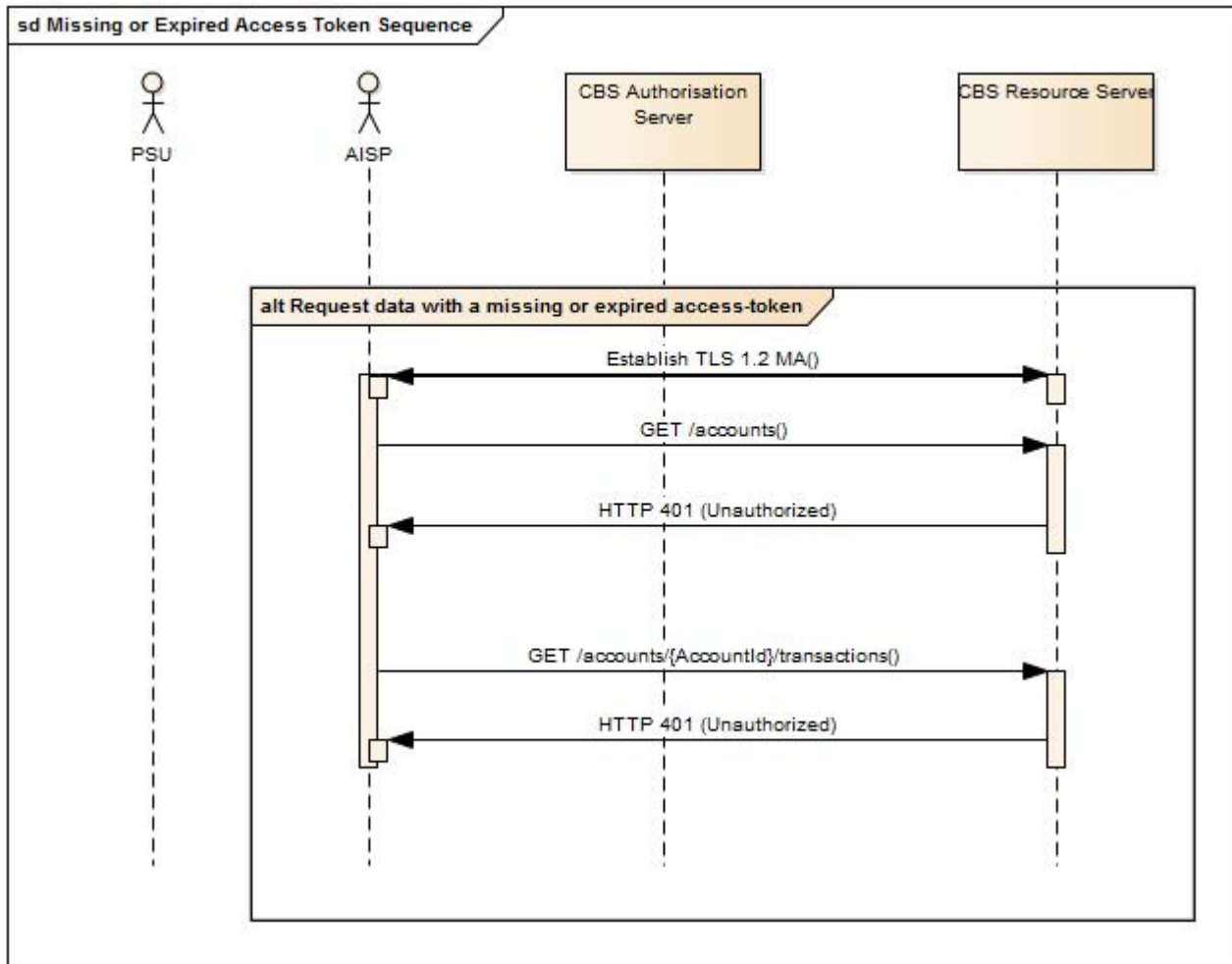
Alternate and Error Flows

Missing or Expired Access Token

This flow assumes that the following Steps have been completed successfully:

- Step 1: Request Account Information
- Step 2: Setup Account Request
- Step 3: Authorize Consent

The AISP attempts to provide an expired or missing access token to CBS in an attempt to Request Data

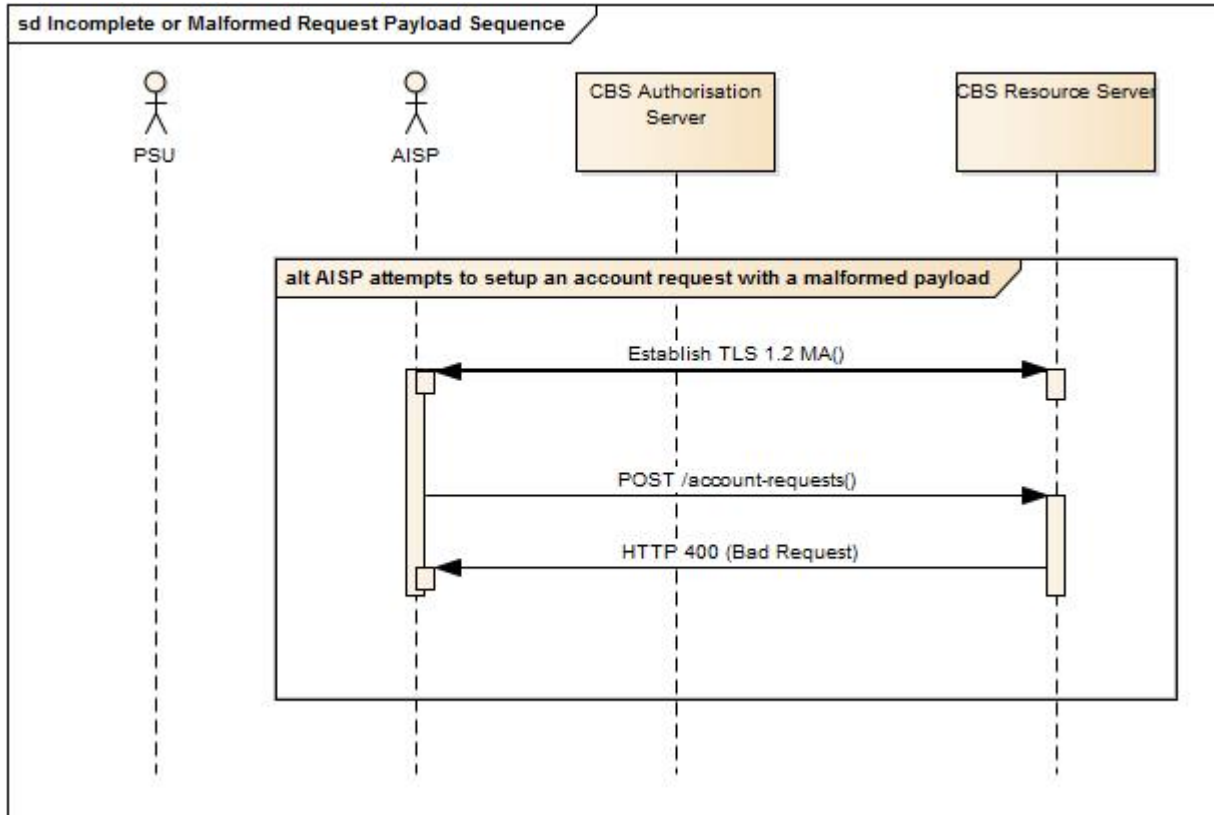


Incomplete or Malformed Request Payload

This flow assumes that the following Steps have been completed successfully:

- Step 1: Request Account Information
- Step 2: Setup Account Request
- Step 3: Authorize Consent

The AISP provides a malformed request to CBS in an attempt to setup an Account Request.

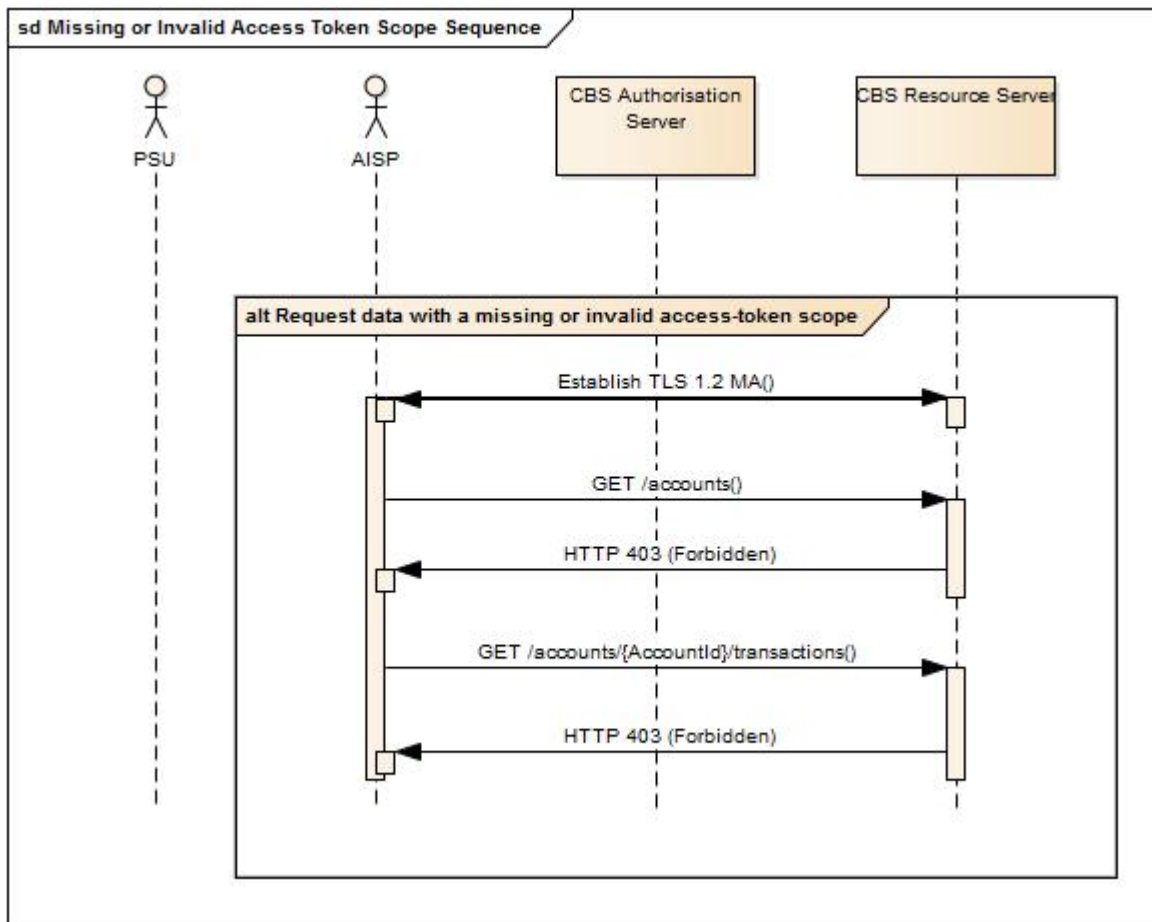


Missing or Invalid Access Token Scope

This flow assumes that the following Steps have been completed successfully:

- Step 1: Request Account Information
- Step 2: Setup Account Request
- Step 3: Authorize Consent

The AISP provides a (valid) access token which does not have a valid scope (or link to the correct Permissions) to Request Data



Failed Authorisation Consent

This flow assumes that the following Steps have been completed successfully:

- Step 1: Request Account Information
- Step 2: Setup Account Request

The Step 3: Authorize Consent Flow fails to succeed due to the PSU providing invalid credentials to CBS, resulting in no Authorization Code being generated.

sd Failed Authorisation Consent Sequence

