Coventry Building Society

Confirmation of Funds API Specification V2.0

Contents

Version control

Version	Date	Updated by	Changes made
2.0	04 Mar 2018	Coventry Building Society	Baseline version

Release Note

This release note explains what's new in The Confirmation of Funds API Specifications between versions.

Our APIs follow OpenBanking Confirmation of Funds API Specification v3.1 with the exception that the funds confirmation endpoint for FundsAvailable returns Yes or No rather than true or false as defined by OpenBanking.

Overview

This specification describes the Confirmation of Funds API flows and payloads.

The API endpoints described here allow a Card Based Payment Instrument Issuer ('CBPII') to:

- Register an intent to confirm funds by creating a "funds confirmation consent" resource with CBS, for agreement between the PSU and CBS. This consent is for a maximum of 90 days and contains the length of time (expiration date) the customer (PSU) would like to provide to the CBPII; and
- Subsequently make a request to confirm funds are available.
  - Funds can only be confirmed against the currency of the account.

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 CBPIIs and PSUs to authenticate themselves and provide consent.

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/

Idempotency

The API endpoints for creating funds-confirmation-consents and funds-confirmations resources are not idempotent.

If a time-out error occurs, we would expect a CBPII to create a new resource, rather than try with the same resource. This is particularly relevant for the funds-confirmations resource, where the availability of funds may have changed between requests.

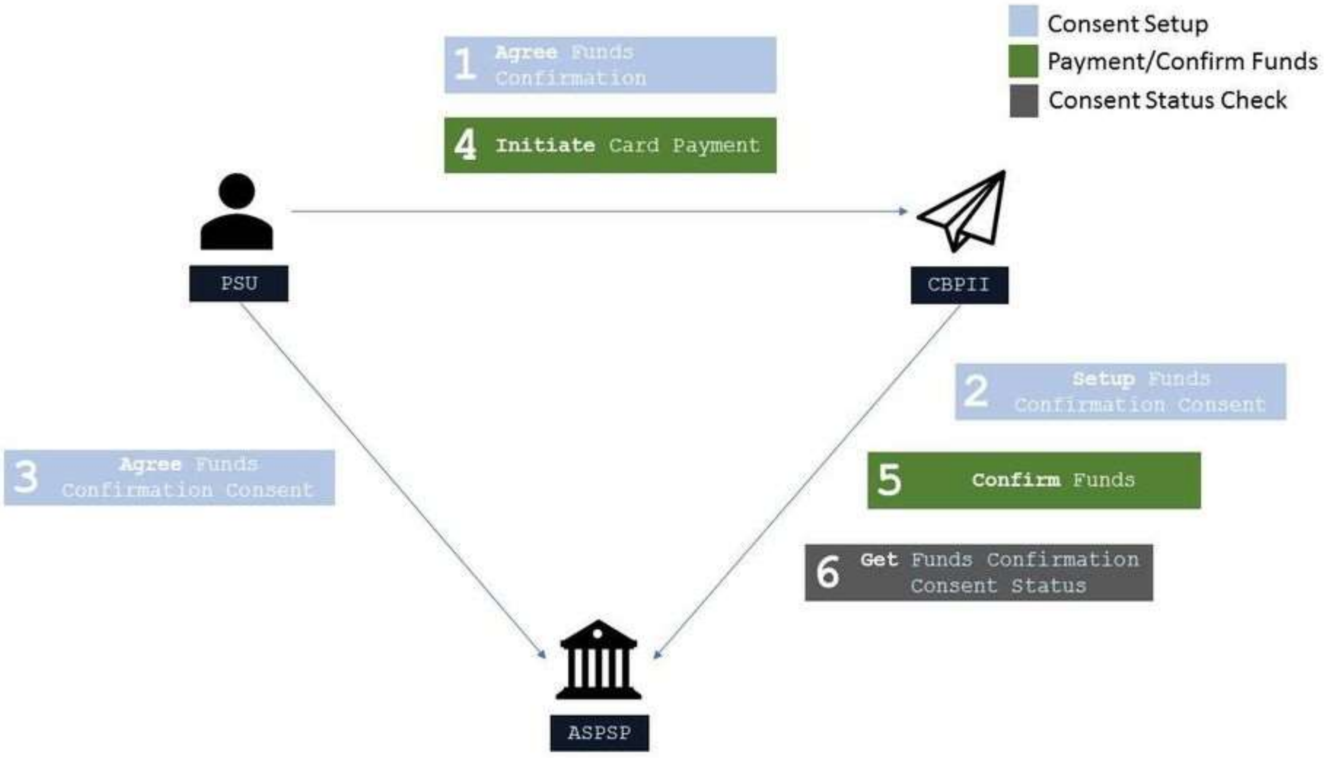
Scope

The APIs specified in this document provide the ability for CBPIIs to access a PSU's confirmation of funds for domestic PCA accounts.

Basics

Overview

The figure below provides a general outline of a confirmation of funds request and flow using the Confirmation of Funds APIs.


Consent Setup
Payment/Confirm Funds
Consent Status Check
1 Agree Funds Confirmation
4 Initiate Card Payment
PSU
CBPII
2 Setup Funds Confirmation Consent
3 Agree Funds Confirmation Consent
5 Confirm Funds
6 Get Funds Confirmation Consent Status
ASPSP

Steps
The Consent model for the Confirmation of Funds API differs to the Payments API and the Account and Transactions API, as the consent is held between the PSU and CBS, rather than between the PSU and the TPP. Whilst the flow follows the same process, the context for each step has a different meaning and is detailed below.

Step 1: Agree Funds Confirmation

- This flow begins with a PSU committing to give explicit consent, to CBS to respond to confirmation of funds requests from the CBPII.

Step 2: Setup Funds Confirmation Consent

- The CBPII connects to CBS that services the PSU's account(s) and creates a **funds-confirmation-consent** resource. This informs CBS that one of its PSUs would like to grant access to confirm the availability of funds to a CBPII. CBS responds with an identifier for the resource (the ConsentId - which is the intent identifier).
- This step is carried out by making a **POST** request to the /funds-confirmation-consents endpoint, under a client credentials grant.
- The setup payload will include these fields:
  - Debtor Account - mandatory debtor account details to capture the account from which the availability of funds will be confirmed.
  - Expiration Date Time - an optional expiration for when the CBPII will no longer have access to confirm funds on a PSU's account.

Step 3: Agree Funds Confirmation Consent

- The CBPII requests the PSU to agree the consent. The ASPSP may carry this out by using a redirection flow or a decoupled flow.
- In a redirection flow, the CBPII redirects the PSU to CBS.
  - The redirect includes the ConsentId generated in the previous step.
  - This allows CBS to correlate the **funds-confirmation-consent** that was setup.
  - CBS authenticates the PSU.
  - The PSU gives explicit consent to CBS to respond to confirmation of funds requests from the CBPII.
  - CBS updates the state of the **funds-confirmation-consent** resource internally to indicate that the resource has been authorised.
  - Once the consent has been authorised, the PSU is redirected back to the CBPII.

Step 4: Initiate Card Payment

- A card payment is initiated by the PSU (directly or indirectly). This process is outside the scope of the Confirmation of Funds API.

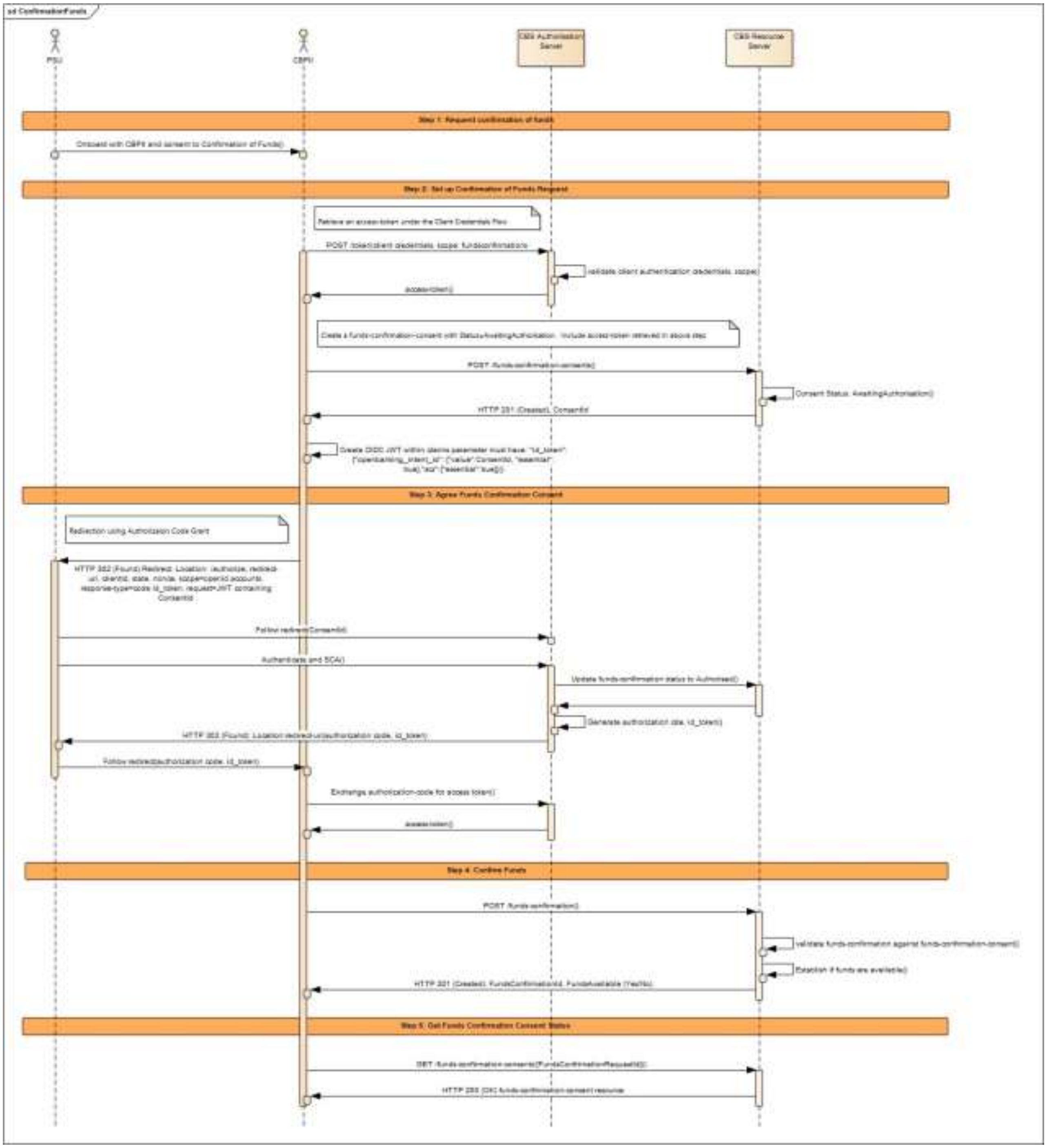
Step 5: Confirm Funds

- The CBPII connects to CBS that services the PSU's account(s) and creates a **funds-confirmation** resource. This informs CBS that the CBPII would like to confirm funds are available in the specific payment account.
- CBS responds with a yes/no (boolean) for the resource.
- This step is carried out by making a **POST** request to the /funds-confirmations endpoint, under an authorization code grant.
- The setup payload will include these fields - which describe the data that the PSU has consented with the CBPII:
  - Amount - the amount to be confirmed available.
  - ConsentId - an Id that relates the request to a **funds-confirmation-consent**, and specific account with CBS. This Id must match the intent identifier.

Step 6: Get Funds Confirmation Consent Status

- The CBPII may check the status of the **funds-confirmation-consent** resource (with the ConsentId).
- This step is carried out by making a **GET** request to the /funds-confirmation-consents endpoint, under a client credentials grant.

Sequence Diagram


sd ConfirmationFunds
PSU
CBPII
CBS Authorisation Server
CBS Resource Server
Step 1: Request confirmation of funds
Onboard with CBPII and consent to Confirmation of Funds()
Step 2: Set up Confirmation of Funds Request
Retrieve an access-token under the Client Credentials Flow
POST /token(client credentials, scope: fundsconfirmations
validate client authentication credentials, scope()
access-token()
Create a funds-confirmation-consent with Status=AwaitingAuthorisation. Include access-token retrieved in above step
POST /funds-confirmation-consents()
Consent Status: AwaitingAuthorisation()
HTTP 201 (Created), ConsentId
Create OIDC JWT within claims parameter must have: "id_token": {"openbanking_intent_id": {"value":ConsentId, "essential": true},"acr":{"essential":true}}
Step 3: Agree Funds Confirmation Consent
Redirection using Authorization Code Grant
HTTP 302 (Found) Redirect: Location: /authorize, redirect-url, clientId, state, nonce, scope=openid accounts, response-type=code id_token, request=JWT containing ConsentId
Follow redirect(ConsentId)
Authenticate and SCA()
Update funds-confirmation status to Authorised()
Generate authorization code, id_token()
HTTP 302 (Found): Location redirect-url(authorization code, id_token)
Follow redirect(authorization code, id_token)
Exchange authorization-code for access token()
access-token()
Step 4: Confirm Funds
POST /funds-confirmation()
validate funds-confirmation against funds-confirmation-consent()
Establish if funds are available()
HTTP 201 (Created), FundsConfirmationId, FundsAvailable (Yes/No)
Step 5: Get Funds Confirmation Consent Status
GET /funds-confirmation-consents/{FundsConfirmationRequestId}()
HTTP 200 (OK) funds-confirmation-consent resource

Actors

Actor	Abbreviation	Type	Specializes	Description
Payment Service User	PSU	Person	N/A	A natural or legal person making use of a payment service as a payee, payer or both (PSD2 Article 4(10))
Payment Service Provider	PSP	Legal Entity	N/A	A legal entity (and some natural persons) that provide payment services as defined by PSD2 Article 4(11)
Account Servicing Payment Service Provider	ASPSP	Legal Entity	PSP	An ASPSP is a PSP that provides and maintains a payment account for a payment services user (PSD 2 Article 4(15). CBS is an ASPSP.
Third Party Providers / Trusted Third Parties	TPP	Legal Entity	PSP	A party other than an ASPSP that provides payment related services. The term is not actually defined in PSD2, but is generally deemed to include all payment service providers that are 3rd parties (CBS and the PSU to whom the account belongs being the first two parties)
Payment Initiation Service Provider	PISP	Legal Entity	TPP	A TPP that provides Payment Initiation Services. PSD2 does not offer a formal definition. Article 4(18) quite circularly defines a PISP as a PSP that provides Payment Initiation Services.
Account Information Service Provider	AISP	Legal Entity	TPP	A TPP that provides Account Information Services. Again, PSD2 defines AISPs in Article 4(19) circularly as a PSP that provides account information services
Card Based Payment Instrument Issuer	CBPII	Legal Entity	TPP	A TPP that provides Card Based Payment

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/open-banking/v2.0/funds-confirmation-consents
- https://resourcema.coventrybuildingsociety.co.uk/open-banking/v2.0/funds-confirmations

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	Optional	Optional	Optional

	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.	

(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	Standard HTTP Header; Represents the format of the payload returned in the response. 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)	Conditionally Mandatory
x-fapi-interaction-id	An RFC4122 UID used as a correlation id. This **must** be the same value provided in the x-fapi-interaction-id request header. Mandatory if provided in the request.	Conditionally Mandatory
Retry-After	Header indicating the time (in seconds) that the TPP should wait before retrying an operation. CBS **will** include this header along with responses with the HTTP status code of 429 (Too many requests).	Optional

Return & Error Codes

The following are the HTTP response codes for the different HTTP methods - across all Confirmation of Funds 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
Confirmation of Funds Request has malformed, missing or non-compliant JSON body or URL	400 Bad Request	The requested operation will not be carried out.	Yes	No	No

parameters					
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 request contained an accept header that requested a content-type other than application/json and a character set other than UTF-8	406 Not Acceptable		Yes	Yes	Yes
Schema errors	422 Invalid	The requested operation will not be carried out due to an error with the schema The moreInformation section provides additional information as to where the schema validation has failed.	Yes	No	
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 /funds-confirmation-consents/22289 where 22289 is not a valid ConsentId, CBS **will** respond with a 400.

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 a funds confirmation consent with a ConsentId that does not exist	GET /funds-confirmation-consents/1001	400 (Bad Request)
TPP attempts to retrieve a resource that is not defined	GET /credit-cards	404 (Not Found)

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 attempted to access a resource with an Id that it does not have access to.
  E.g., an attempt to access GET /funds-confirmation-consents/1001 where an funds-confirmation-consents 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.

422 (Invalid)
When there are schema errors CBS will return a 422 (invalid) response.

The situation could arise when:

- The TPP includes an additional field in the request.
- The TPP uses an invalid data type
- The TPP has not included a required field

Example:

```
{

  "httpCode": "422",

  "httpMessage": "Invalid",

  "moreInformation": "Validate REST: xa35://tmp/temp_962878:1: [JSV0002] Invalid object: the property 'Code' is missing."

}
```

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 onboarding with CBS and have been issued with a ClientId and Secret
2. The software application registered with CBS must have "fundsconfirmations" as one of the required scopes.
3. The TPP must have valid network certificates issued by CBS or a valid PSD2 eIDAS certificate.

All available Endpoints

	Resource	HTTP Operation	Endpoint
1	token	POST	POST /mga/sps/oauth/oauth20/token
2	funds-confirmation-consents	POST	POST /funds-confirmation-consents
3	funds-confirmation-consents	GET	GET /funds-confirmation-consents/{ConsentId}
4	funds-confirmation-consents	DELETE	DELETE /funds-confirmation-consents/{ConsentId}
5	funds-confirmations	POST	POST /funds-confirmations

POST /token

- This flow begins with a TPP requesting a Client Credentials Access Token from the token endpoint:

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

  Request must include:

  grant_type=”Client Credentials”

  scope=”openId fundsconfirmations”

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

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

POST /funds-confirmation-consents

The API allows the CBPII to ask CBS to create a new **funds-confirmation-consents** resource.

- This endpoint allows the CBPII to propose a consent to be agreed between CBS and PSU, to authorise the CBPII access to confirm funds are available.
- CBS creates the funds-confirmation-consent resource and responds with a unique ConsentId to refer to the resource.
- Prior to calling the operation, the CBPII must have an access token issued by CBS using a client credentials grant.

Data Model

Request

Data Dictionary

Name	Card-inality	DataType	Notes
Request			
Request/Data	1..1		
Request /Data/ExpirationDateTime	0..1	ISODateTime	Specified date and time the funds confirmation authorisation will expire. If this is not populated, the authorisation will be open ended.
Request/Data/DebtorAccount	1..1		Unambiguous identification of the account of the debtor to which a confirmation of funds consent will be applied.
Request/Data/DebtorAccount/SchemeName	1..1		Name of the identification scheme, in a coded form as published in an external list.
Request/Data/DebtorAccount/Identification	1..1	Max256Text	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Request/Data/DebtorAccount/Name	0..1	Max70Text	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.
Request/Data/DebtorAccount/SecondaryIdentification	0..1	Max34Text	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).

Response

Data Dictionary

Name	Card-inality	DataType	Notes
Response			
Response/Data	1..1		
Response/Data/ConsentId	1..1	Max128Text	Unique identification as assigned to identify the funds confirmation consent resource.
Response/Data/Status	1..1	ExternalRequestStatusCode Enumeration (see enumeration section below for details)	Specifies the status of the consent resource.
Response/Data/CreationDateTime	1..1	ISODateTime	Date and time at which the resource was created.
Response/Data/StatusUpdateDateTime	1..1	ISODateTime	Date and time at which the resource status was updated.
Response/Data/ExpirationDateTime	0..1	ISODateTime	Specified date and time the funds confirmation authorisation will expire.
Request/Data/DebtorAccount	1..1		Unambiguous identification of the account of the debtor to which a confirmation of funds consent will be applied.
Request/Data/DebtorAccount/SchemeName	1..1		Name of the identification scheme, in a coded form as published in an external list.
Request/Data/DebtorAccount/Identification	1..1	Max256Text	Identification assigned by an institution to identify an account. This identification is known by the account owner.
Request/Data/DebtorAccount/Name	0..1	Max70Text	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.

Request/Data/DebtorAccount/SecondaryIdentification	0..1	Max34Text	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).

Error Information

ErrorCode	Message	Notes
1002	We're unable to complete this request due to an issue with the consent details received	
1007	We're unable to complete this request due to an issue with the Organisation details provided	There is a problem with the details we hold about the organisation.
1014	We're unable to complete this request due to a technical issue	There was a technical problem which occurred within CBS.
1018	We're unable to complete this request due to an issue with the Software Application details provided	There is a problem with the software application.
99997	We're unable to complete this request due to an Invalid Header Check	The headers in the request were incomplete
99998 99999	We're unable to complete this request due to a technical issue	There was a technical problem which occurred within CBS.

Funds Confirmation Consent Status

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

	Status	Status Description
1	Authorised	The funds confirmation consent has been successfully agreed
2	AwaitingAuthorisation	The funds confirmation consent is awaiting agreement.
3	Rejected	The funds confirmation consent has been rejected.
4	Revoked	The funds confirmation consent has been revoked via CBS interface.

GET /funds-confirmation-consents/{ConsentId}

A CBPII may optionally retrieve a 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 CBS using a client credentials grant.

Data Model

Response

Uses the same response model as the POST /funds-confirmation-consents detailed above.

Error Information

ErrorCode	**Message**	**Notes**
1000	We're unable to complete this request due to an issue with the consent details received	The funds confirmation consentId is not valid
99997	We're unable to complete this request due to an Invalid Header Check	The headers in the request were incomplete
99998 99999	We're unable to complete this request due to a technical issue	There was a technical problem which occurred within CBS.

DELETE /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 CBS using a client credentials grant.

Error Information

ErrorCode	Message	Notes
1000	We're unable to complete this request due to an issue with the consent details received	The funds confirmation consentId is not valid
99997	We're unable to complete this request due to an Invalid Header Check	The headers in the request were incomplete
99998 99999	We're unable to complete this request due to a technical issue	There was a technical problem which occurred within CBS.

POST /funds-confirmations

If the CBPII would like to confirm funds with CBS, 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 an authorization code grant and specify the ConsentId in the request payload.
- This CBPII must use a currency of the account.

Data Model

Request

Data Dictionary

Name	Card-inality	DataType	Notes
Request			
Request/Data	1..1		
Request/Data/ConsentId	1..1	Max128Text	Unique identification as assigned by CBS to uniquely identify the funds confirmation consent resource.
Request/Data/Reference	1..1	Max35Text	Unique reference, as assigned by the CBPII, to unambiguously refer to the request related to the payment transaction.
Request/Data/InstructedAmount	1..1		Amount of money to be confirmed as available funds in the debtor account. Contains an Amount and a

			Currency.
Request/Data/InstructedAmount/Amount	1..1	^\d{1,13}\.\d{1,5}$	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Request/Data/InstructedAmount/Currency	1..1	^[A-Z]{3,3}$	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". **Note: CBS only supports GBP**

Response

Data Dictionary

Name	Cardinality	DataType	Notes
Response			
Response/Data	1..1		
Response/Data/FundsConfirmationId	1..1	Max128Text	Unique identification as assigned to identify the funds confirmation resource.
Response/Data/ConsentId	1..1	Max128Text	Unique identification as assigned to identify the funds confirmation consent resource.
Response/Data/CreationDateTime	1..1	ISODateTime	Date and time at which the resource was created.
Response/Data/FundsAvailable	1..1	xs:Boolean (Yes or No)	Flag to indicate the result of a confirmation of funds check.
Response/Data/Reference	1..1	Max35Text	Unique reference, as assigned by the CBPII, to unambiguously refer to the request related to the payment transaction.
Response/Data/InstructedAmount	1..1		Amount of money to be confirmed as available funds in the debtor account. Contains an Amount and a Currency.
Response/Data/InstructedAmount/Amount	1..1	^\d{1,13}\.\d{1,5}$	A number of monetary units specified in an active currency where the unit of currency is explicit and compliant with ISO 4217.
Response/Data/InstructedAmount/Currency	1..1	^[A-Z]{3,3}$	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". **Note: CBS only supports GBP**

Error Information

ErrorCode	Message	Notes
1000	We're unable to complete this request due to a technical issue	There was a technical problem which occurred within CBS.
1001	We're unable to complete this request due to an issue with the consent details received	The ConsentId has not been authorized
1002	We're unable to complete this request due to an issue with the consent details received	The ConsentId has expired
1001	We're unable to complete this request due to an issue with the consent details received	The ConsentId is not valid
99997	We're unable to complete this request due to an Invalid Header Check	The headers in the request were incomplete
99998 99999	We're unable to complete this request due to a technical issue	There was a technical problem which occurred within CBS.

Sandbox API Endpoints

Sandbox APIs have been provided to assist with testing connectivity. The APIs all return mocked data; they are only accessible using the TLS 1.2 MA with the certificate provided to you by Coventry Building Society or a valid eIDAS QWAC/QSealC Certificate (uses Client Credentials Grant only)

Please refer to the separate Coventry SandBox API Specification for further details.

Security & Access Control

API Scopes

The access tokens required for accessing the Confirmation of Funds APIs must have at least the following scope:

Scopes
fundsconfirmations

Grants Types

CBPIIs must use a client credentials grant to obtain a token to make POST requests to the funds-confirmation-consent endpoint. In the specification, this grant type is referred to as "Client Credentials".

CBPIIs must use an authorization code grant using a redirect or decoupled flow to obtain a token to make POST requests to the funds-confirmation endpoint. When accessing, the intent-id in the token must match the ConsentId in the message payload. In the specification, this grant type is referred to as "Authorization Code".

CBPIIs must use a client credentials grant to obtain a token to make GET requests.

Consent Authorisation

The CBPII **must** create a **funds-confirmation-consent** resource through a **POST** operation. This resource outlines the *consent* that the CBPII claims the PSU has committed to agreeing with CBS, to retrieve confirmation of funds information. At this stage, the consent is not yet agreed between the PSU and CBS.

CBS responds with a ConsentId. This is the intent-id that is used when initiating the authorization code grant (as described in the Trust Framework).

As part of the authorization code grant:

- CBS authenticates the PSU.
- CBS plays back the consent (registered by the CBPII) back to the PSU - to agree the consent. The PSU may agree or decline the consent in its entirety (but not selectively).

Once these steps are complete, the consent is considered to have been agreed between CBS and the PSU.

Consent Elements

The funds-confirmation-consent resource consists of the following fields, which together form the elements of the consent provided by the PSU to the CBPII:

- **DebtorAccount:** The account to which the consent has been applied.
    - The field is mandatory, as the consent for CBPII access to a PSU's data must be for a specific account known to the PSU and the CBPII.
- **ExpirationDateTime:** The date-time up to which the consent is valid.
    - The field is optional, as the consent for CBPII access to a PSU's data may be indefinite. However, CBS will require customers to re authorize after 90 days. This will require a new consentId.

Funds Confirmation Consent Status

The funds-confirmation-consent resource may have one of the following status codes after authorisation has taken place:

	Status	Description
1	Authorised	The Funds Confirmation Consent has been successfully authorised.
2	Rejected	The Funds Confirmation Consent has been rejected.
3	Revoked	The Funds Confirmation Consent has been revoked via CBS interface.

Consent Re-authentication

The funds-confirmation-consent resource is a long lived consent up to a maximum of 90 days. A funds-confirmation-consent can be re-authenticated if:

- the funds-confirmation-consent resource has a status of Authorised and
- The ExpirationDateTime, if specified, has not elapsed or 90 days have elapsed.

Error Condition

If the PSU does not complete a successful consent authorisation (e.g. if the PSU is not authenticated successfully), the authorization code grant ends with a redirection to the TPP with an error response as described in RFC 6749 Section 4.1.2.1. The PSU is redirected to the TPP with an error parameter indicating the error that occurred.

Consent Revocation

A PSU can revoke consent for accessing confirmation of funds at any point in time.

The PSU can revoke authorisation directly with CBS via OnlineServices or the Contact Centre.

The PSU can request the CBPII to revoke consent that it has authorised. If consent is revoked with the CBPII:

- The CBPII **must** cease to access the APIs at that point (otherwise it may be in breach of GDPR).
- The CBPII **should** call the **DELETE** operation on the funds-confirmation-consents resource to indicate to CBS that the PSU has revoked consent.

Handling Expired Access Tokens

Access Token issued through Client Credentials Grant

When an access token issued through a Client Credentials Grant expires, the TPP must get a new access token by executing a client credential grant again.

Access Token issued through Authorization Code Grant

CBS will issue an access token for fundsconfirmations scope which will expire after 90 days after which a new access token will be required using a new consentId.

Data Model

High Level Payload Structure

This section gives an overview of the top level structure for the API payloads for the Confirmation of Funds APIs.

Request Structure

The top level request structure for Confirmation of Funds APIs:

Confirmation of Funds API Request

```
{
  "Data": {
    ...
  },
  "Risk": {
    ...
  }
}
```

The top level structure for the Confirmation of Funds API POST requests will be:

- Data

The Data section contains the request details.

A Risk section for the request structure has been separated out - so that this can evolve in isolation from request section of the payload.

Response Structure

The top level response structure for Confirmation of Funds APIs:

Confirmation of Funds API Response

```
{

  "Data": {

    ...

  },

  "Links": {

    ...

  ],

  "Meta": {

    ...

  }

}
```

In line with the principle on RESTful API practices - we are replaying the resource as part of the response.

Two additional top level sections are included for:

- Links
- Meta

Links

The Links section is mandatory and will always contain URIs to related resources, CBS only supports Self for this API

For example:

Example Links

```
  "Links": {

    "Self": "https://resourcema.coventrybuildingsociety.co.uk/open-banking/funds-confirmation-consents/123456

  }
```

Meta

The Meta section is mandatory, but can be empty.

Data Payload – Error Response Structure

Example

```
{
  "Code": "...",
  "Id": "...",
  "Message": "...",
  "Errors": [
    {
      "ErrorCode": "...",
      "Message": "...",
      "Path": "...",
      "Url": "..."
    }
  ]
}
```

Data Dictionary

Name	Cardinality	DataType	Notes
ErrorResponse			An array of detail error codes, and messages, and URLs to documentation to help remediation.
ErrorResponse /Code	1..1		High level textual error code, to help categorize the errors.
ErrorResponse /Message	1..1	Max128Text	Brief Error message, e.g., 'There is something wrong with the request parameters provided'
ErrorResponse /Errors	0..1		
ErrorResponse /Errors/ErrorCode	1..1	ISODateTime	Low level textual error code
ErrorResponse /Errors/Message	1..n		A description of the error that occurred

Usage Examples

POST Funds Confirmation Consent

Request

POST Funds Confirmation Consent Request

```
POST /funds-confirmation-consents HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id: CBSOpenBanking
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
Accept: application/json
{
  "Data": {
    "DebtorAccount": {
      "SchemeName":
"SortCodeAccountNumber",
      "Identification": "11280001234567",
      "SecondaryIdentification": "Roll
12345"
    },
    "ExpirationDateTime": "2017-05-
02T00:00:00+00:00"
  }
}
```

Response

POST Funds Confirmation Consent Response

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "ConsentId": "test12345",
    "CreationDateTime": "2017-05-
02T00:00:00+00:00",
    "Status": "AwaitingAuthorisation",
    "ExpirationDateTime": "2017-05-
02T00:00:00+00:00",
    "DebtorAccount": {
      "SchemeName":
"SortCodeAccountNumber",
      "Identification": "11280001234567",
      "SecondaryIdentification": "Roll
12345"
    }
  }
  "Links": {
    "Self": "/open-banking/v2.0/funds-
confirmation-consents/test12345"
  },
  "Meta": {}
}
```

GET Funds Confirmation Consent

Request

GET Funds Confirmation Consent Request

```
GET /funds-confirmation-consents/test12345
HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id: CBSOpenBanking
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
Accept: application/json
```

Response

GET Funds Confirmation Consent Response

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "ConsentId": "test12345",
    "CreationDateTime": "2017-05-
02T00:00:00+00:00",
    "Status": "AwaitingAuthorisation",
    "ExpirationDateTime": "2017-05-
02T00:00:00+00:00",
    "DebtorAccount": {
      "SchemeName":
"SortCodeAccountNumber",
      "Identification": "11280001234567",
      "SecondaryIdentification": "Roll
12345"
    }
  }
  "Links": {
    "Self": "/open-banking/v2.0/funds-
confirmation-consents/test12345"
  },
  "Meta": {}
}
```

POST Funds Confirmation

Request

`POST Funds Confirmations Request`

```
POST /funds-confirmations HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id: CBSOpenBanking
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
Accept: application/json
{
  "Data": {
    "ConsentId": "test12345",
    "Reference": "TPP Reference",
    "InstructedAmount": {
      "Amount": "20.00",
      "Currency": "GBP"
    }
  }
}
```

Response

`POST Funds Confirmations Response`

```
HTTP/1.1 201 Created
x-fapi-interaction-id: 93bac548-d2de-4546-
b106-880a5018460d
Content-Type: application/json
{
  "Data": {
    "FundsConfirmationId":
"testfundsconfid12345",
    "ConsentId": "test12345",
    "CreationDateTime": "2017-05-
02T00:00:00+00:00",
    "FundsAvailable": "Yes",
    "Reference": "TPP Reference",
    "InstructedAmount": {
      "Amount": "20.00",
      "Currency": "GBP"
    }
  }
  "Links": {
    "Self": "/open-banking/v2.0/funds-
confirmations/testfundsconfid12345"
  },
  "Meta": {}
}
```

DELETE Funds Confirmation Consent

Request

DELETE Funds Confirmation Consents Request

```
POST /funds-confirmation-consents HTTP/1.1
Authorization: Bearer
2YotnFZFEjr1zCsicMWpAA
x-fapi-financial-id: CBSOpenBanking
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
Accept: application/json
```

Response

DELETE Funds Confirmation Consents Response

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