

은행권 공동 오픈플랫폼
오픈 API 명세서
Version 1.8

금융결제원
2018. 8.

개정 이력

버전	개정일자	개정내용	작성자
1.0	2016.04.25	- 버전 1.0 작성	금융결제원
1.0	2016.05.13	- 잔액조회 결과 응답 시 출금가능금액 항목 추가 - Content-Type 일부 오타 수정	금융결제원
1.0	2016.07.04	- unlink API 추가 - API URI 개선	금융결제원
1.0	2016.07.25	- 계좌목록조회 API 추가 - 이체API처리조회 API 추가 - 참가은행 응답코드 추가	금융결제원
1.0	2016.07.29	- 입금이체2(계좌번호) API 요청 시 '수취인성명' 항목 추가	금융결제원
1.0	2016.08.10	- OAuth 인증 : Login Access Token 갱신 추가	금융결제원
1.0	2016.08.11	- 입금이체, 출금이체 요청전문 cms_no 항목 추가	금융결제원
1.0	2016.08.22	- API 응답코드 수정, 참가은행 응답코드 추가	금융결제원
1.0	2016.09.29	- 거래내역조회 API 응답 파라미터 추가 - API 응답코드, 참가은행 응답코드 추가	금융결제원
1.0	2016.10.21	- API별 요청메시지 access_token에 scope 내용 추가 - 별첨3) API 응답코드 : Gateway 응답코드 수정 및 추가	금융결제원
1.0	2016.11.01	- 오타 수정 및 명세서 게시	금융결제원
1.0	2017.01.24	- 거래내역조회 파라미터 추가 (최종거래내역순번, 직전조회추적정보) - 이체API처리조회 파라미터 수정(원거래요청일시)	금융결제원
1.0	2017.07.24	- 잔액조회 응답 파라미터 추가(계좌종류, 상품명) - 등록계좌조회 요청 파라미터 추가 (정렬순서, 해지계좌포함여부) - API 응답코드, 참가은행 응답코드 추가	금융결제원
1.1	2017.09.18	- 사용자 인증 개선버전 추가(/authorize2, /authorize_account2) - 개인정보 제공 관련 필드 추가 - 입금이체, 출금이체 파라미터 변경 - /result, /recheck 관련 설명 추가	금융결제원
1.5	2017.12.29	- 송금인정보조회 API 추가 - 사용자인증 등 API 설명 보완, 문구 수정 - 은행코드표 추가, 참가은행 응답코드 추가	금융결제원
1.6	2018.03.23	- 입금이체 수취인성명 검증 옵션 추가 - 사용자인증 에러 처리 방식 추가 - 지역농축협 계좌, 기업은행 평생계좌 선거절 처리 안내 추가 - 다국어 설정 안내 추가	금융결제원

1.7	2018.04.20	- 사용자인증 시 사용자정보 전달항목의 입력란 수정가능 여부 추가 - 사용자인증 시 사용자정보 전달항목에 이동통신사 추가 - 입금이체 암호문구 관련 안내 보강	금융결제원
1.7	2018.05.21	- 간편가입 회원의 입금이체 암호문구 관련 안내 보강 - 거래내역조회/송금인정보조회 시 다음페이지 처리 안내 추가	금융결제원
1.8	2018.07.13	- 다국어 4개 추가(방글라데시/파키스탄/러시아/우즈베키스탄)	금융결제원
1.8	2018.07.18	- 거래내역조회 API 페이지 처리 관련 설명 추가 - 송금인정보조회 API 페이지 처리 관련 설명 추가	금융결제원
1.8	2018.07.25	- 계좌실명조회 기능 개선(주민등록번호, 외국인등록번호, 국내거소신고번호 등 실명번호 종류 추가)	금융결제원
1.8	2018.08.07	- 다국어 2개 추가(몽골/스리랑카)	금융결제원
1.8	2018.08.30	- 다국어 2개 추가(일본/중국)	금융결제원
1.8	2019.01.17	- 이체결과조회 API 설명 추가 (1개월 이내 이체거래 조회 가능) - 오타 수정	금융결제원

Table of Contents

제 1 장 오픈 API 목록	1
1.1. 개 요	1
1.2. API URI 맵	1
1.3. OAuth 이용안내	2
1.4. 버전관리 정책	2
1.5. 응답코드 처리	2
제 2 장 오픈 API 명세	3
2.1. 사용자인증(OAuth 2.0)	3
2.1.1. 사용자인증 API (3-legged) : 개선버전	3
2.1.2. 토큰발급 API	6
2.1.3. 계좌등록확인 API (3-legged) : 개선버전	10
2.1.4. 권한범위 및 토큰갱신 안내	12
2.1.5. 사용자인증 에러 처리 방식	16
2.1.6. 계좌 선거절 처리 안내	16
2.1.7. 다국어 설정 안내	17
2.2. 사용자/계좌 관리	18
2.2.1. 사용자정보조회 API	18
2.2.2. 사용자로그인연결동의해제 API	21
2.2.3. 등록계좌조회 API	22
2.2.4. 계좌정보변경 API	24
2.2.5. 계좌해지 API	25
2.3. 조회서비스(사용자)	27
2.3.1. 잔액조회 API	27
2.3.2. 거래내역조회 API	29
2.4. 조회서비스(이용기관)	32
2.4.1. 계좌실명조회 API	32
2.4.2. 송금인정보조회 API	35
2.5. 이체서비스	39
2.5.1. 출금이체 API	39
2.5.2. 입금이체 API	42
2.5.3. 이체결과조회 API	49

2.5.4. 이체API처리조회 API	52
2.5.5. /transfer/result API와 /transfer/recheck API의 차이 설명	55
2.6. 관리 API	56
2.6.1. 참가은행상태조회 API	56
2.7. [deprecated] 구)사용자인증(OAuth 2.0)	58
2.7.1. 사용자로그인연결 API (3-legged) : 구버전	58
2.7.2. 계좌등록 API (3-legged) : 구버전	59
2.7.3. 계좌등록확인 API (3-legged) : 구버전	60
별첨	61
별첨1) 데이터 Type 설명	61
별첨2) 용어 설명	61
별첨3) 은행코드	61
별첨4) OAuth 응답코드	62
별첨5) API 인증요청 거부(O0001)에 대한 세부 응답코드(error_code)	62
별첨6) 사용자인증 Callback 에러코드 및 에러메시지	63
별첨7) API 응답코드	63
별첨8) 참가은행 응답코드	64

제 1 장 오픈 API 목록

1.1. 개 요

본 문서는 "은행권 공동 오픈플랫폼"(이하 "오픈플랫폼")에서 제공하는 오픈 API 명세를 기술합니다. 오픈플랫폼 이용승인을 받은 핀테크기업(이하 "이용기관")은 오픈플랫폼이 제공하는 오픈 API를 활용하여 이용기관용 응용프로그램(이하 "앱")을 개발할 수 있습니다.

1.2. API URI 맵

오픈플랫폼의 오픈 API는 REST(**RE**presentational **S**tate **T**ransfer) 방식으로 구현되어, 인터넷 웹 서비스의 형태로 제공됩니다. 아래와 같은 오픈 API를 제공하며, 하나의 API는 하나의 웹 서비스 URI로 매핑됩니다.

기능구분	API 명	URI	HTTP Method
사용자인증 (OAuth 2.0)	사용자인증 (개선버전)	https://openapi.open-platform.or.kr/oauth/2.0/authorize2	GET
	토큰발급	https://openapi.open-platform.or.kr/oauth/2.0/token	POST
	계좌등록확인 (개선버전)	https://openapi.open-platform.or.kr/oauth/2.0/authorize_account2	GET
사용자/계좌 관리	사용자정보조회	https://openapi.open-platform.or.kr/v1.0/user/me	GET
	사용자로그인연결 동의해제	https://openapi.open-platform.or.kr/v1.0/user/unlink	POST
	등록계좌조회	https://openapi.open-platform.or.kr/v1.0/account/list	GET
	계좌정보변경	https://openapi.open-platform.or.kr/v1.0/account/update_info	POST
	계좌해지	https://openapi.open-platform.or.kr/v1.0/account/cancel	POST
조회서비스 (사용자)	잔액조회	https://openapi.open-platform.or.kr/v1.0/account/balance	GET
	거래내역조회	https://openapi.open-platform.or.kr/v1.0/account/transaction_list	GET
조회서비스 (이용기관)	계좌실명조회	https://openapi.open-platform.or.kr/v1.0/inquiry/real_name	POST
	송금인정보조회	https://openapi.open-platform.or.kr/v1.0/inquiry/remit_list	POST
이체서비스	출금이체	https://openapi.open-platform.or.kr/v1.0/transfer/withdraw	POST
	입금이체 1 입금이체 2	https://openapi.open-platform.or.kr/v1.0/transfer/deposit https://openapi.open-platform.or.kr/v1.0/transfer/deposit2	POST
	이체결과조회	https://openapi.open-platform.or.kr/v1.0/transfer/result	POST
	이체 API 처리조회	https://openapi.open-platform.or.kr/v1.0/transfer/recheck	POST
관리 API	참가은행상태조회	https://openapi.open-platform.or.kr/v1.0/bank/status	GET
[deprecated]	사용자로그인연결	https://openapi.open-platform.or.kr/oauth/2.0/authorize	GET
구) 사용자인증 (OAuth 2.0)	계좌등록	https://openapi.open-platform.or.kr/oauth/2.0/register_account	GET
	계좌등록확인	https://openapi.open-platform.or.kr/oauth/2.0/authorize_account	GET

[안내사항]

- 1) 테스트 API 호출 시에는 <https://testapi.open-platform.or.kr> 로 호출하셔야 합니다.
 - 운영 API(<https://openapi.open-platform.or.kr>)는 최종 계약체결 이후에만 호출 가능합니다.
 - 운영 API의 경우, API에 따라 과금될 수 있으니 유의하시기 바랍니다.
- 2) **[deprecated]**로 표시된 API는 지원이 종료되는 API입니다.

1.3. OAuth 이용안내

오픈플랫폼은 이용기관 및 일반고객(이용기관 앱을 이용하는 개인을 의미하며, 이하 "사용자")과 오픈플랫폼의 3자 인증(이하 "3-legged"), 이용기관과 오픈플랫폼의 상호 인증(이하 "2-legged")을 위하여 OAuth 절차를 준용하며, 본 문서는 OAuth 2.0 기준으로 작성되었습니다.

오픈플랫폼은 사용자인증, 계좌등록확인 시 사용자의 동의를 획득하기 위하여 OAuth 3-legged 절차를 적용하며, 입금이체, 계좌실명조회 등과 같이 사용자가 개입되지 않는 경우에는 OAuth 2-legged 절차를 적용합니다.

1.4. 버전관리 정책

오픈 API 호출 시 URI에 API의 버전을 지정할 수 있습니다. 특별히 버전을 기술하지 않으면 최신버전이 자동으로 매핑됩니다. 현재 제공되는 최신 버전은 v1.0 입니다. 버전을 명시하여 사용하는 것을 권고합니다.

(사용예)

- <https://openapi.open-platform.or.kr/account/balance> (최신 버전으로 자동 매핑)
- <https://openapi.open-platform.or.kr/v1.0/account/balance> (버전 1.0 호출)

1.5. 응답코드 처리

오픈 API의 응답코드는 두 가지로 분류됩니다. 하나는 API 처리 시에 발생하는 "응답코드(API)"이며, 나머지 하나는 참가은행 처리 시에 발생하는 "응답코드(참가은행)"입니다. API 처리 시에 오류가 발생하여 "응답코드(API)"가 정상값이 아닌 경우에는 오픈플랫폼 고객센터로 오류에 대한 문의를 하여야 하며, 참가은행 처리 시에 오류가 발생하여 "응답코드(참가은행)"가 정상값이 아닌 경우에는 응답 시 전달되는 "응답코드를 부여한 참가은행.표준코드"를 참조하여 해당 참가은행에 오류를 문의하여야 합니다. 오류 문의 시에는 "거래고유번호(API)" 혹은 "거래고유번호(참가은행)"를 알려주시면 확인이 용이합니다.

제 2 장 오픈 API 명세

2.1. 사용자인증(OAuth 2.0)

□ 기 능

사용자인증(OAuth 2.0)은 3-legged 인증과 2-legged 인증으로 구분됩니다.

3-legged 인증은 사용자가 이용기관의 앱을 통하여 오픈플랫폼 서비스에 접근하기 위한 인증 기능을 제공합니다. 사용자는 오픈플랫폼(Service Provider)의 인증 페이지에서 휴대폰 본인인증 및 ARS 동의를 통하여 인증하고, 이용기관은 사용자에게 Access Token을 얻음으로써 사용자의 인증을 획득합니다.

2-legged 인증은 사용자의 개입 없이 단순히 이용기관이 자신에 대한 인증정보(Client ID, Client Secret)를 가지고 인증함으로써, 오픈플랫폼으로부터 이용기관에 대한 Access Token을 획득하는 것을 의미합니다.

□ 명 세

2.1.1. 사용자인증 API (3-legged) : 개선버전

■ 설 명

별도의 회원가입 없이 한번의 OAuth 인증으로 본인인증 및 계좌등록까지 처리할 수 있는 사용자 인증 API이며, 상황에 따른 3가지 호출 scheme을 제공합니다.

- (0)최초인증 : 처음 오픈플랫폼에 접속하는 사용자의 인증 처리 (본인인증+ARS동의)
- (1)재인증 : 최초 접속 이후에 추가로 계좌등록 필요시 인증 처리 (본인재인증+ARS동의)
선택항목을 파라미터로 받은 경우 인증화면 입력란에 자동으로 정보 셋팅
- (2)인증생략 : 유효한 접속토큰을 이용하여 본인인증 절차 생략 (본인인증생략+ARS동의)
ARS동의를 위한 휴대폰번호는 DB에 저장된 값을 활용
(전자금융업자 등 특정 자격요건을 갖춘 사업자에게만 선별적으로 허용)

API 호출 시에는, 상기 3가지 경우에 대하여 모두 동일한 Parameter를 지원하며, (1)재인증 및 (2)인증생략의 경우에는 HTTP Header에 항목(선택/필수)을 추가로 설정하여야만 합니다.

[예외 처리 안내]

- auth_type(사용자인증타입)을 지정하지 않은 경우에는, (0)최초인증으로 처리되며 HTTP Header를 처리(parsing)하지 않습니다.
- auth_type=1로 지정하여 (1)재인증을 요청하였으나 Kftc-Bfop-UserSeqNo, Kftc-Bfop-UserCI값을 지정하지 않은 경우에는, (0)최초인증으로 처리되며 입력란에 자동으로 정보를 셋팅합니다.
- auth_type=1로 지정하여 (1)재인증을 요청하였으나 유효한 사용자가 없는 경우에는, (0)최초인증으로 처리되며 입력란에 자동으로 정보를 셋팅합니다.
- auth_type=2로 지정하여 (2)인증생략을 요청하였으나, Kftc-Bfop-UserSeqNo, Kftc-Bfop-UserCI값을 지정하지 않았거나 유효한 사용자가 없는 경우에는 에러로 처리됩니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/authorize2
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	설명
Parameter [공통] (0)최초인증 (1)재인증 (2)인증생략	response_type	고정값: code	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	사용자인증이 성공하면 이용기관으로 연결되는 URL
	scope	[login inquiry transfer]	Access Token 권한 범위 (다중 scope 가능) [12p] 「2.1.4. 권한범위 및 토큰갱신 안내」 참조
	client_info	<client_info> (Max: 256 bytes)	이용기관이 셋팅하는 임의의 정보 (Callback 호출 시 그대로 전달됨)
	auth_type	[0 1 2] (필수) (최초인증 시 생략가능)	사용자인증타입 구분 (0:최초인증, 1:재인증, 2:인증생략)
	lang	[kor eng vnm idn khm phl npl bgd pak rus uzb mng lka jpn chn] (선택)	다국어 설정 [17p] 「2.1.7. 다국어 설정 안내」 참조
	edit_option	"on" or "off" (선택)	입력란의 수정가능 여부(on:수정가능) (미지정 시 기본값: "on")
	bg_color	#000000 (선택)	배경색 지정
	txt_color	#000000 (선택)	글자색 지정
	btn1_color	#000000 (선택)	확인버튼색 지정
	btn2_color	#000000 (선택)	취소버튼색 지정
Headers (1)재인증	Kftc-Bfop-UserSeqNo	<user_seq_no> (선택)	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci> (선택)	사용자 CI(Connect Info)
	Kftc-Bfop-UserName	<user_name> (선택)	사용자명
	Kftc-Bfop-UserInfo	<user_info> (선택)	생년월일(8 자리)+성별(1 자리)
	Kftc-Bfop-UserCellNo	<user_cell_no> (선택)	휴대폰번호
	Kftc-Bfop-PhoneCarrier	[skt ktf lgt skm ktm lgm] (선택)	이동통신사 [17p] 「이동통신사 코드표」 참조
	Kftc-Bfop-UserEmail	<user_email> (선택)	이메일주소
	Kftc-Bfop-BankCodeStd	<user_bank_code_std>(선택)	계좌번호 지정등록 시 셋팅
	Kftc-Bfop-AccountNum	<user_account_num> (선택)	
Headers (2)인증생략	Kftc-Bfop-UserSeqNo	<user_seq_no> (필수)	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci> (필수)	사용자 CI(Connect Info)
	Kftc-Bfop-AccessToken	<access_token> (필수)	"login" scope 을 포함한 토큰 (유효한 접속토큰이어야 함)

※ 오픈플랫폼 신규 사용자인 경우에도 auth_type=1(재인증)로 지정하여, 이용기관이 기 보유하고 있는 사용자 정보를 프로바이더 페이지에 전달할 수 있습니다. 이 경우에는 Kftc-Bfop-UserSeqNo, Kftc-Bfop-UserCI값을 지정하지 않으며, 나머지 전달된 항목들은 입력란(input box)에 자동으로 셋팅됩니다.

■ 응답 메시지 URL

HTTP URL	<callback_uri> [Callback URL 로 redirect]
HTTP Method	GET

■ 응답 메시지 명세

항목	값	설명
code	<authorization_code>	사용자인증 성공 시 반환되는 코드
scope	[login inquiry transfer]	Access Token 권한 범위 (다중 scope 가능)
client_info	<client_info>	요청 시 이용기관이 셋팅한 정보를 그대로 전달

■ 화면 예시

① 본인인증	② 계좌등록	③ ARS 요청
④ ARS 동의	⑤ 이메일 등록	⑥ 이용기관 앱 복귀

2.1.2. 토큰발급 API

□ 사용자 토큰발급 API (3-legged)

■ 설 명

사용자인증 API를 통하여 Authorization Code를 획득(사용자의 동의를 받았다는 의미)한 이후에 이 code값을 이용하여 토큰을 발급받는 API입니다. 3-legged 인증이 아닌 2-legged, 갱신(refresh)의 경우는 호출 scheme이 다소 상이합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/token
HTTP Method	POST
Content-Type	application/x-www-form-urlencoded; charset=UTF-8

■ 요청 메시지 명세

항목	값	설명
code	<authorization_code>	사용자인증 성공 후 획득한 Authorization Code
client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret>	오픈플랫폼에서 발급한 이용기관 앱의 Client Secret
redirect_uri	<callback_uri>	Access Token 을 전달받을 Callback URL
grant_type	고정값: authorization_code	3-legged 인증을 위한 권한부여 방식 지정

■ 응답 메시지 명세

Header	status: 200
Content-Type	application/json

HTTP	항목	값	TYPE (길이)	설명
Body	access_token	<access_token>		오픈플랫폼에서 발행된 Access Token
	token_type	고정값: Bearer		Access Token 유형
	expires_in	<expire_time>		Access Token 만료 기간(초)
	refresh_token	<refresh_token>		Access Token 갱신 시 사용하는 Refresh Token
	scope	[login inquiry transfer]		Access Token 권한 범위 (사용자인증 시 요청했던 권한 범위와 동일)
	user_seq_no	<user_seq_no>	AN(10)	사용자일련번호

■ 응답 메시지 형태

JSON
<pre>{ "access_token": <access_token>, "token_type": "Bearer", "expires_in": <expire_time>, "refresh_token": <refresh_token>, "scope": "login inquiry transfer", "user_seq_no": "U123456789" }</pre>

□ 이용기관 토큰발급 API (2-legged)

■ 설 명

사용자의 개입이 없는 2-legged 인증의 경우에는 scope=oob로 정의되며, 사용자인증 API의 호출 없이 바로 /token API를 호출하여 토큰을 획득한다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/token
HTTP Method	POST
Content-Type	application/x-www-form-urlencoded; charset=UTF-8

■ 요청 메시지 명세

항목	값	설명
client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret>	오픈플랫폼에서 발급한 이용기관 앱의 Client Secret
scope	고정값: oob	Access Token 권한 범위
grant_type	고정값: client_credentials	2-legged 인증을 위한 권한부여 방식 지정

■ 응답 메시지 명세

Header	status: 200
Content-Type	application/json

HTTP	항목	값	TYPE (길이)	설명
Body	access_token	<access_token>		오픈플랫폼에서 발행된 Access Token
	token_type	고정값: Bearer		Access Token 유형
	expires_in	<expire_time>		Access Token 만료 기간(초)
	scope	고정값: oob		Access Token 권한 범위
	client_use_code	<client_use_code>	AN(10)	이용기관코드

■ 응답 메시지 형태

JSON
<pre>{ "access_token": <access_token>, "token_type": "Bearer", "expires_in": <expire_time>, "scope": "oob", "client_use_code": "F001234560" }</pre>

□ 사용자 토크갱신 API (Refresh Token 사용)

■ 설 명

Access Token 발급 시 함께 받았던 Refresh Token을 사용하여 토크의 유효기간을 갱신한다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/token
HTTP Method	POST
Content-Type	application/x-www-form-urlencoded; charset=UTF-8

■ 요청 메시지 명세

항목	값	설명
client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret>	오픈플랫폼에서 발급한 이용기관 앱의 Client Secret
refresh_token	<refresh_token>	토크발급 시 받은 Access Token 갱신용 토크
scope	[login inquiry transfer]	Access Token 권한 범위
grant_type	고정값: refresh_token	토크 갱신을 위한 권한부여 방식 지정

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	access_token	<access_token>		오픈플랫폼에서 발행된 Access Token
	token_type	고정값: Bearer		Access Token 유형
	expires_in	<expire_time>		Access Token 만료 기간(초)
	scope	[login inquiry transfer]		Access Token 권한 범위
	user_seq_no	<user_seq_no>	AN(10)	사용자일련번호

■ 응답 메시지 형태

JSON
<pre>{ "access_token": <access_token>, "token_type": "Bearer", "expires_in": <expire_time>, "refresh_token": <refresh_token>, "scope": "login inquiry transfer", "user_seq_no": "1000000106" }</pre>

2.1.3. 계좌등록확인 API (3-legged) : 개선버전

■ 설 명

사용자인증을 통하여 이미 등록된 계좌에 대해서 재확인을 하고 접속토큰을 재발급받는 API이며, 상황에 따라 3가지 호출 scheme을 제공합니다.

- (0)최초인증 : 사용자정보 미보유 시 (본인인증+계좌내역확인)
- (1)재인증 : 사용자정보 기보유 시 (본인재인증+계좌내역확인)
선택항목을 파라미터로 받은 경우 인증화면 입력란에 자동으로 정보 셋팅
- (2)인증생략 : 사용자정보 및 유효한 접속토큰 보유 시 (본인인증생략+계좌내역확인)
(전자금융업자 등 특정 자격요건을 갖춘 사업자에게만 선별적으로 허용)

Refresh Token을 활용하여 3개월 마다 토큰을 재발급 받을 수 있습니다.

단, 토큰 재발급과는 별도로 매 1년 마다 사용자의 재확인을 받아야 합니다.

(조회/출금 등의 모두 해당됨)

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/authorize_account2
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	설명
Parameter [공통] (0)최초인증 (1)재인증 (2)인증생략	response_type	고정값: code	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	사용자인증이 성공하면 이용기관으로 연결되는 URL
	scope	[login inquiry transfer]	Access Token 권한 범위 (다중 scope 가능) [12p] 「2.1.4. 권한범위 및 토큰갱신 안내」 참조
	client_info	<client_info> (선택) (Max: 256 bytes)	이용기관이 셋팅하는 임의의 정보 (Callback 호출 시 그대로 전달됨)
	auth_type	[0 1 2] (필수) (최초인증 시 생략가능)	사용자인증타입 구분 (0:최초인증, 1:재인증, 2:인증생략)
	lang	[kor eng vnm idn khm phl np bgd pak rus uzb mng lka jpn chn] (선택)	다국어 설정 [17p] 「2.1.7. 다국어 설정 안내」 참조
	edit_option	"on" or "off" (선택)	입력란의 수정가능 여부(on:수정가능) (미지정 시 기본값: "on")
	bg_color	#000000 (선택)	배경색 지정
	txt_color	#000000 (선택)	글자색 지정
	btn1_color	#000000 (선택)	확인버튼색 지정
	btn2_color	#000000 (선택)	취소버튼색 지정

Headers (1)재인증	Kftc-Bfop-UserSeqNo	<user_seq_no> (선택)	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci> (선택)	사용자 CI(Connect Info)
	Kftc-Bfop-UserName	<user_name> (선택)	사용자명
	Kftc-Bfop-UserInfo	<user_info> (선택)	생년월일(8 자리)+ 성별(1 자리)
	Kftc-Bfop-UserCellNo	<user_cell_no> (선택)	휴대폰 번호
	Kftc-Bfop-PhoneCarrier	[skt ktf lgt skm ktm lgm] (선택)	이동통신사 [17p] 「이동통신사 코드표」 참조
	Kftc-Bfop-UserEmail	<user_email> (선택)	이메일주소
Headers (2)인증생략	Kftc-Bfop-UserSeqNo	<user_seq_no> (필수)	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci> (필수)	사용자 CI(Connect Info)
	Kftc-Bfop-AccessToken	<access_token> (필수)	"login" scope 을 포함한 토큰 (유효한 접속토큰이어야 함)

※ auth_type=1(재인증)인 경우에도 Kftc-Bfop-UserSeqNo, Kftc-Bfop-UserCI값을 지정하지 않을 수 있으며, 이때는 휴대폰 본인인증 후에 얻어진 CI값으로 사용자를 확인합니다.



■ 응답 메시지 URL

HTTP URL	<callback_uri> [Callback URL로 redirect]
HTTP Method	GET

■ 응답 메시지 명세

항목	값	설명
code	<authorization_code>	사용자인증 성공 시 반환되는 코드
scope	[login inquiry transfer]	Access Token 권한 범위 (다중 scope 가능)
client_info	<client_info>	요청 시 이용기관이 셋팅한 정보를 그대로 전달

■ 화면 예시

① 본인인증	② 계좌내역확인	③ 이용기관 앱 복귀						
 <p>핀테크 스타트업의 모임 앱을 사용하기 위한 고객님의 이용동의 및 본인인증 절차를 진행합니다.</p> <p>제공되는 정보 : 성명, 성별, 휴대폰번호, 이메일주소, CI(ConnectInfo)</p> <p>성명 <input type="text"/> 내국인 외국인</p> <p>생년월일(8자리) <input type="text"/> 남 여</p> <p>휴대폰번호(숫자만) <input type="text"/></p> <p>이동통신사를 선택하세요 ▼</p> <p>인증코드 <input type="text"/> 인증코드 받기</p> <p>서비스 이용 및 개인정보처리 동의 ▼</p>	 <p>사용자인증 개선버전 (Case1)</p> <p>CASE1 CASE2 CASE3</p> <p>https://testapi.open-platform.or.kr/oa 필첨</p> <p>아래와 같이 홍길동 님의 계좌가 등록되어 있습니다. 아래 계좌에 대해서 핀테크 스타트업의 모임 앱이 출금/조회 서비스를 이용하는데 동의하시겠습니까?</p> <p>확인</p> <table border="1"> <thead> <tr> <th>은행/계좌번호(별명)</th> <th>등록일시(최신순)</th> <th>상태</th> </tr> </thead> <tbody> <tr> <td>오픈은행 12300654321*** (모임회비계좌)</td> <td>2018-03-30 14:11:30</td> <td>동의완료</td> </tr> </tbody> </table>	은행/계좌번호(별명)	등록일시(최신순)	상태	오픈은행 12300654321*** (모임회비계좌)	2018-03-30 14:11:30	동의완료	<p>(Callback URL 호출)</p>
은행/계좌번호(별명)	등록일시(최신순)	상태						
오픈은행 12300654321*** (모임회비계좌)	2018-03-30 14:11:30	동의완료						

2.1.4. 권한범위 및 토큰갱신 안내

■ 권한 범위(scope)

API 구분		권한범위* (scope)	유효기간** (expires_in)	Refresh 지원여부
개선버전*** (3-legged)	사용자인증(조회)	login inquiry	90 일 (+10)	O
	사용자인증(출금)	login transfer	90 일 (+10)	O
	사용자인증(조회/출금)	login inquiry transfer	90 일 (+10)	O
이용기관인증 (2-legged)		oob	90 일	X
구)버전 (3-legged)	사용자로그인연결	login	14 일 (+5)	O
	계좌등록(조회)	inquiry	90 일 (+10)	O
	계좌등록(출금)	transfer	90 일 (+10)	O

* 다중 scope의 경우에는 scope값 사이를 공백(space)으로 구분함

** () 안의 숫자는 Refresh Token의 만료 기간(초)을 의미함. +일 만큼 유효기간이 더 길게 책정됨

*** 개선버전은 본인인증, 로그인연결, 계좌등록을 하나의 흐름으로 처리하게 되어있으며, "사용자 인증"이란 용어로 통칭함

[호출예제 : Authorization Code를 획득 후 토큰 요청 시]

호출 Method	POST	호출 URI	/oauth/2.0/token
Content-type		application/x-www-form-urlencoded; charset=UTF-8	
요청(Request)		응답(Response)	
code=<authorization_code>& client_id=<client_id>& client_secret=<client_secret>& redirect_uri=<callback_uri>& grant_type=authorization_code		{ "access_token": <access_token>, "token_type": "Bearer", "expires_in": <expires_time>, "refresh_token": <refresh_token>, "scope": "login transfer", "user_seq_no": "1000210099" }	

■ 토큰 갱신

API 구분		갱신전 토큰		갱신후 토큰	
		권한범위 (scope)	유효기간 (expires_in)	권한범위 (scope)	유효기간 (expires_in)
개선버전 (3-legged)	사용자인증(조회)	login inquiry	90 일 (+10)	동일	동일
	사용자인증(출금)	login transfer	90 일 (+10)	동일	동일
	사용자인증(조회/출금)	login inquiry transfer	90 일 (+10)	동일	동일
이용기관인증 (2-legged)		oob	90 일	Refresh 불가	
구버전 (3-legged)	사용자로그인연결	login	14 일 (+5)	동일	동일
	계좌등록(조회)	inquiry	90 일 (+10)	동일	동일
	계좌등록(출금)	transfer	90 일 (+10)	동일	동일

[호출예제 : 토큰의 Refresh 요청 시]

호출 Method	POST	호출 URI	/oauth/2.0/token
Content-type		application/x-www-form-urlencoded; charset=UTF-8	
요청(Request)		응답(Response)	
client_id=<client_id>& client_secret=<client_secret>& refresh_token=<refresh_token>& scope=login+transfer& grant_type=refresh_token		{ "access_token": <access_token> "token_type": "Bearer" "expires_in": <expires_time> "scope": "login transfer" "user_seq_no": "1000210099" }	

■ API별 호출 Method, 요청토큰의 필요 Scope

기능구분	API 명	API URI	호출 Method	요청토큰 Scope*
사용자인증 (OAuth 2.0)	사용자인증(개선버전)	/oauth/2.0/authorize2	GET	해당없음
	토큰발급	/oauth/2.0/token	POST	해당없음
	계좌등록확인(개선버전)	/oauth/2.0/authorize_account2	GET	해당없음
사용자/계좌 관리	사용자정보조회	/v1.0/user/me	GET	login
	사용자로그인연결해제	/v1.0/user/unlink	POST	login
	등록계좌조회	/v1.0/account/list	GET	login
	계좌정보변경	/v1.0/account/update_info	POST	login
	계좌해지	/v1.0/account/cancel	POST	login
조회서비스 (사용자)	잔액조회	/v1.0/account/balance	GET	inquiry
	거래내역조회	/v1.0/account/transaction_list	GET	inquiry

조회서비스 (이용기관)	계좌실명조회	/v1.0/inquiry/real_name	POST	oob
	송금인정보조회	/v1.0/inquiry/remittance_list	POST	oob
출금서비스	출금이체	/v1.0/transfer/withdraw	POST	transfer
	입금이체 1 입금이체 2	/v1.0/transfer/deposit /v1.0/transfer/deposit2	POST	transfer
	이체결과조회	/v1.0/transfer/result	POST	transfer
	이체 API 처리조회	/v1.0/transfer/recheck	POST	transfer
관리 API	참가기관상태조회	/v1.0/bank/status	GET	oob
[deprecated] 구) 사용자인증 (OAuth 2.0)	사용자로그인연결	/oauth/2.0/authorize	GET	해당없음
	계좌등록	/oauth/2.0/register_account	GET	해당없음
	계좌등록확인	/oauth/2.0/authorize_account	GET	해당없음

* API 호출 시에 요청토큰을 필요로 하는 경우, 해당 요청토큰이 가져야하는 scope을 의미함
필요로 하는 scope을 가지고 있지 않을 경우에는 권한거부 처리됨

2.1.5. 사용자인증 에러 처리 방식

■ 설 명

사용자인증 API 호출 시의 에러발생은 두 가지 경우로 나눌 수 있습니다.

첫째는, 프로바이더 페이지가 호출되기 전에 에러 처리되는 경우입니다. 예를 들면 요청 파라미터가 누락되었거나 허용되지 않은 이용기관인 경우가 해당됩니다.

둘째는, 프로바이더 페이지는 호출되었으나 페이지 상에서 에러 처리되는 경우입니다. 예를 들면 사용자가 동의를 거부하였거나 계좌등록 처리 시 에러가 발생한 경우가 해당됩니다.

■ 프로바이더 페이지 호출전 에러 처리

아래와 같이 JSON 형태로 응답코드와 응답메시지를 전송합니다.

JSON
<pre>{ "rsp_code": "O0001", "rsp_message": "인증요청 거부-인증 파라미터 오류 ([3000201])" }</pre>

※ "[O0001] 인증요청 거부" 에러의 경우에는 [] 사이에 세부 응답코드가 표시됩니다.

응답코드는 아래의 별첨자료를 참고합니다.

<p60> 별첨4) OAuth 응답코드

<p60> 별첨5) API 인증요청 거부(O0001)에 대한 세부 응답코드(error_code)

■ 프로바이더 페이지 호출후 에러 처리

이미 프로바이더 페이지로 전환(redirect)된 상태이기 때문에 이용기관의 Callback URL을 호출하여 에러코드와 에러메시지를 전달합니다.

[응답 메시지 URL]

HTTP URL	<callback_uri> [Callback URL로 redirect]
HTTP Method	GET

[응답 메시지 명세]

항목	값	설명
error	(문자열)	에러코드
error_description	(문자열)	에러메시지
client_info	<client_info>	요청 시 이용기관이 셋팅한 정보를 그대로 전달

[에러 호출 예]

GET Method
<p>http://<callback_uri>?error=access_denied& error_description=사용자가 '취소' 버튼을 클릭한 경우& client_info=[test] whatever you want</p>

에러코드 및 에러메시지는 아래의 별첨자료를 참고합니다.

<p61> 별첨6) 사용자인증 Callback 에러코드 및 에러메시지

※ 사용자인증 시 Callback URL 호출을 통한 에러 처리는 국제표준인 「The OAuth 2.0 Authorization Framework」(RFC 6749)의 "4.2.2.1. Error Response"를 준용하였습니다.

2.1.6. 계좌 선거절 처리 안내

■ 설 명

오픈플랫폼에서는 사용자가 조회/출금 동의하고자 하는 계좌를 참가은행과 전문처리를 통하여 계좌등록 처리하고 있습니다. 그러나 계좌등록이 불가한 경우에 대하여 사용자에게 즉각적인 안내를 하기 위하여 다음의 경우에 대하여 선거절(참가은행에 등록요청전 거절) 처리하고 있습니다.

■ 선거절 기준

계좌구분	거절사유	기준		
지역농축협 계좌	지역농축협(은행코드:012)은 오픈플랫폼 미참가은행인데 농협은행(은행코드:011)으로 지역농축협 계좌의 등록을 요청하는 경우에 계좌등록이 불가함	농협은행(은행코드:011) + 13~14 자리이고, 아래의 조건에 해당하는 경우		
		계좌번호	자리수(위치)	
			1~3	13
		■■■-□□□□-□□□□-□-■	351,352,356,355	
			354,360,384,394,398	3,4,5
기업은행 평생계좌	기업은행 평생계좌는 휴대폰번호를 이용한 가상매핑계좌로서 계좌등록이 불가함	□□□□□□-■■-□□□□□□-□	51,52,56,55,66,67	
		■■■-□□□□-□□□□-□□-□	792	
		기업은행(은행코드:003) + 11 자리 + "010"으로 시작		

2.1.7. 다국어 설정 안내

■ 설 명

오픈플랫폼의 프로바이더 페이지는 다국어를 지원합니다. 사용자인증(/authorize2) 및 계좌등록확인(/authorize_account2) API 호출 시 언어 파라미터(**lang=[언어코드]**)를 설정하면, 페이지 상의 문구 및 ARS 안내음성이 해당 언어로 출력됩니다.

단, 다음의 언어는 ARS 안내음성이 **영어**로 안내됩니다.

- 파키스탄어

[언어코드]

코드	언어	코드	언어
(미설정)	한국어	kor	한국어
eng	영어	vnm	베트남어
idn	인도네시아어	khm	캄보디아어
phl	필리핀어	npl	네팔어
bgd	방글라데시아어	pak	파키스탄어
rus	러시아어	uzb	우즈베키스탄어
mng	몽골어	lka	스리랑카어
jpn	일본어	chn	중국어

[다국어 호출 예]

GET Method
https://testapi.open-platform.or.kr/oauth/2.0/authorize2?client_id=l7xx****63bd&response_type=code&lang=eng&scope=login+inquiry+transfer&redirect_uri=http://<callback_uri>&client_info=[test] whatever you want&auth_type=0

※ 이동통신사 코드표

코드	이동통신사	코드	이동통신사
skt	SKT (SKT line phone)	skm	SKT 알뜰폰 (SKT card phone)
ktf	KT (KT line phone)	ktm	KT 알뜰폰 (KT card phone)
lgt	LG U+ (LG U+ line phone)	lgm	LG U+ 알뜰폰 (LG U+ card phone)

2.2. 사용자/계좌 관리

□ 기 능

오픈플랫폼에 사용자인증을 통하여 등록된 사용자의 고객정보 및 계좌목록을 조회합니다. 계좌별명을 변경하거나 등록을 해지할 수 있습니다.

□ 명 세

2.2.1. 사용자정보조회 API

■ 설 명

사용자의 고객정보 및 계좌목록을 조회합니다. 해지된 계좌는 출력하지 않습니다. 고객정보의 제공 범위는 이용기관에 따라서 선별적으로 제공됩니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/user/me
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login]
Parameter	user_seq_no	"U123456789"	AN(10)	사용자일련번호

■ 응답 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	user_seq_no	"U123456789"	AN(10)	사용자일련번호
	user_ci	"Dqz4/7Rp.....UjVj34XFJTV=="	AN(4000)	CI(Connect Info)
	user_name	"오픈원"	AH(20)	고객명
	user_info (선택)	"19810101"	AN(8)	생년월일
	user_gender (선택)	"M" or "F"	A(1)	성별
	user_cell_no (선택)	"01012341234"	AN(11)	휴대폰번호
	user_email (선택)	"testbed@kftc.or.kr"	AN(305)	이메일주소

res_cnt	"5"	N(5)	등록된 계좌 개수
res_list	<object>		등록된 계좌 목록
--fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
--account_alias	"급여계좌"	AH(50)	계좌별명(Alias)
--bank_code_std	"097"	AN(3)	출금(개설)기관.표준코드
--bank_code_sub	"1230001"	AN(7)	출금(개설)기관.점별코드
--bank_name	"오픈은행"	AH(20)	출금(개설)기관명
--account_num (선택)	"0001230000123"	AN(16)	계좌번호
--account_num_masked	"000-1230000-***"	NS*(20)	마스킹된 출력용 계좌번호
--account_holder_name	"홍길동"	AH(20)	계좌예금주명
--account_type	"P"	A(1)	계좌구분(P:개인)
--inquiry_agree_yn	"Y"	A(1)	조회서비스 동의여부
--inquiry_agree_dtime	"20160310101921"	N(14)	조회서비스 동의일시
--transfer_agree_yn	"Y"	A(1)	출금서비스 동의여부
--transfer_agree_dtime	"20160310101921"	N(14)	출금서비스 동의일시

* Body 선택부의 경우, 특정 자격요건을 갖춘 이용기관에 선별적 제공(전자금융업자 등)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "user_seq_no": "U123456789", "user_ci": "Dqz4/7Rp.....UjVj34XFJTV==", "user_name": "홍길동", "res_cnt": "5", "res_list": [{ "fintech_use_num": "123456789012345678901234", "account_alias": "급여계좌", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num_masked": "000-1230000-***", "account_holder_name": "홍길동", "account_type": "P", "inquiry_agree_yn": "Y", "inquiry_agree_dtime ": "20160310101921", "transfer_agree_yn ": "Y", "transfer_agree_dtime ": "20160310101921" }, { ... }, ...] }</pre>

■ 에러 처리

"access token 거부" 혹은 "access token 만료"의 경우에는 OAuth 2.0 사용자인증 혹은 계좌등록 확인을 다시 시도하여 새로운 토큰을 획득합니다.

2.2.2. 사용자로그인연결동의해제 API

■ 설 명

이용기관과 사용자 간의 연결을 해제합니다. 연결을 복원하려면 OAuth 2.0 사용자인증을 다시 시도합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/user/unlink
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login]
Parameter	client_use_code	"F001234560"	AN(10)	이용기관코드
	user_seq_no	"U123456789"	AN(10)	사용자일련번호

* 이용기관코드 확인 : (계약전) "포털 > 마이페이지 > 회원정보 변경"
(계약후) "포털 > 마이페이지 > 계약신청 > 이용계약 조회"

■ 응답 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	user_seq_no	"U123456789"	AN(10)	사용자일련번호

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "user_seq_no": "U123456789" }</pre>

2.2.3. 등록계좌조회 API

■ 설 명

사용자의 등록된 계좌목록을 조회합니다. 해지계좌 포함여부 및 정렬순서를 지정할 수 있습니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/account/list
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login]
Parameter	user_seq_no	"U123456789"	AN(10)	사용자일련번호
	include_cancel_yn	"Y"	A(1)	해지계좌포함여부 Y:해지계좌포함 N:해지계좌불포함
	sort_order	"D"	A(1)	정렬순서 D:Descending, A:Ascending

■ 응답 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Body	api_tran_id	"AA123498HZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	user_name	"홍길동"	AH(20)	사용자명
	res_cnt	"5"	N(5)	사용자 등록계좌 개수
	res_list	<object>		사용자 등록계좌 목록
	--fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	--account_alias	"급여계좌"	AH(50)	계좌별명(Alias)
	--bank_code_std	"097"	N(3)	출금(개설)기관.대표코드
	--bank_code_sub	"1230001"	N(7)	출금(개설)기관.점별코드
	--bank_name	"오픈은행"	AN(20)	출금(개설)기관명
	--account_num (선택)	"0001230000123"	AN(16)	계좌번호
	--account_num_masked	"000-1230000-***"	NS*(20)	마스킹된 출력용 계좌번호
	--account_holder_name	"홍길동"	AH(20)	계좌예금주명

--account_type	"P"	A(1)	계좌구분(P:개인)
--inquiry_agree_yn	"Y"	A(1)	조회서비스 동의여부
--inquiry_agree_dtime	"20160310101921"	N(14)	조회서비스 동의일시
--transfer_agree_yn	"Y"	A(1)	출금서비스 동의여부
--transfer_agree_dtime	"20160310101921"	N(14)	출금서비스 동의일시
--account_state	"01"	A(2)	계좌상태 01:사용, 03:계좌등록요청, 05:해지요청, 09:해지

* Body 선택부의 경우, 특정 자격요건을 갖춘 이용기관에 선별적 제공(전자금융업자 등)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "user_name": "홍길동", "res_cnt": "5", "res_list": [{ "fintech_use_num": "123456789012345678901234", "account_alias": "급여계좌", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num_masked": "000-1230000-***", "account_holder_name": "홍길동", "account_type": "P", "inquiry_agree_yn": "Y", "inquiry_agree_dtime": "20160310101921", "transfer_agree_yn": "Y", "transfer_agree_dtime": "20160310101921", "account_state": "01", }, { ... }, ... { ... }] }</pre>

2.2.4. 계좌정보변경 API

■ 설 명

등록된 계좌의 별명을 변경합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/account/update_info
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login]
Body	fintech_use_num	<fintech_use_num>	AN(24)	핀테크이용번호
	account_alias	"보험료납부계좌"	AH(50)	변경코자하는 계좌별명

■ 응답 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AH(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	" "	AH(100)	응답메시지(API)
	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	account_alias	"보험료납부계좌"	AH(50)	변경된 계좌별명

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": " ", "fintech_use_num": "123456789012345678901234", "account_alias": "보험료납부계좌", }</pre>

2.2.5. 계좌해지 API

■ 설 명

오픈플랫폼에 등록된 사용자의 계좌를 해지(등록삭제 및 동의해지)합니다. 계좌해지는 참가은행에 등록된 내역까지 해지합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/account/cancel
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		계좌등록 시 오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login]
Body	scope	[inquiry transfer]		inquiry : 조회서비스 transfer : 출금서비스
	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호

■ 응답 메시지 형태

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AH(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	bank_rsp_message	""	AN(100)	응답메시지(참가은행)

■ 응답 메시지 명세

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "" }</pre>

2.3. 조회서비스(사용자)

□ 기 능

오픈플랫폼에 사용자인증(3-legged)을 통하여 등록된 사용자 계좌의 금융정보를 조회합니다.

□ 명 세

2.3.1. 잔액조회 API

■ 설 명

사용자 계좌의 잔액을 조회합니다. 계좌종류 및 상품명이 함께 전달됩니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/account/balance
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		조회서비스 계좌등록 시 오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = inquiry]
Parameter	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	tran_dtime	"20160310101921"	N(14)	요청일시

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AH(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호

	balance_amt	"1000000"	SN(13)	계좌잔액(-금액가능)
	available_amt	"1000000"	N(12)	출금가능금액
	account_type	"1"	N(1)	계좌종류 1:수시입출금 2:예적금 6:수익증권
	product_name	"내맘대로통장"	AH(40)	상품명

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "fintech_use_num": "123456789012345678901234", "balance_amt": "1000000", "available_amt": "1000000", "account_type": "2", "product_name": "알뜰살뜰적금" }</pre>

2.3.2. 거래내역조회 API

■ 설 명

사용자가 등록한 계좌의 거래내역을 조회합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/account/transaction_list
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		조회서비스 계좌등록 시 오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = inquiry]
Parameter	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	inquiry_type	"A"	A(1)	조회구분코드 A:All, I:입금, O:출금
	from_date	"20160404"	N(8)	조회시작일자
	to_date	"20160405"	N(8)	조회종료일자
	sort_order	"D"	A(1)	정렬순서 D:Descending A:Ascending
	page_index	"1"	N(5)	페이지번호
	tran_dtime	"20160310101921"	N(14)	요청일시
	befor_inquiry_trace_info	"123"	AN(20)	직전조회추적정보
	list_tran_seqno	"0"	N(10)	최종거래내역순번

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	bank_rsp_code	"000"	AN(3)	응답코드(참가은행)

bank_rsp_message	""	AN(100)	응답메시지(참가은행)
fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
balance_amt	"1000000"	SN(12)	계좌잔액(-금액가능)
page_index_use_yn	"Y"	A(1)	페이지 Index 사용여부 ^{주4)}
page_index	"1"	N(5)	페이지번호 ^{주4)} (API 요청 시 셋팅했던 값을 그대로 응답. 왼쪽 "0" 패딩 제거)
total_record_cnt	"100"	N(5)	총 조회건수 ^{주4)}
page_record_cnt	"25"	N(2)	현재페이지 레코드건수 ^{주1)}
next_page_yn	"Y"	A(1)	다음페이지 존재여부 ^{주3)}
befor_inquiry_trace_info	"123"	AN(20)	직전조회추적정보
list_tran_seqno	"0"	N(10)	최종거래내역순번
res_list	<object>		조회된 거래내역
--tran_date	"20160310"	N(8)	거래일자
--tran_time	"113000"	N(6)	거래시간
--inout_type	"입금"	AH(8)	입출금구분 (입금, 출금, 지급, 기타)
--tran_type	"현금"	AH(10)	거래구분 ^{주2)}
--print_content	"통장인자내용"	AH(20)	통장인자내용
--tran_amt	"450000"	N(12)	거래금액
--after_balance_amt	"-1000000"	SN(13)	거래후잔액(-금액가능)
--branch_name	"분당점"	AH(20)	거래점명

주1) 한 페이지는 최대 25건 가능

주2) 현금, 대체, 급여, 타행환, F/B출금 등 (참가은행에서 직접 셋팅)

주3) <다음페이지 처리 안내>

"다음페이지 존재여부"(next_page_yn)가 "Y"인 경우에는 다음페이지의 거래내역을 확인하기 위하여, 결과로 받은 "직전조회추적정보"(befor_inquiry_trace_info)와 함께 계속적으로 거래내역조회 API를 재호출해야 합니다. 다만, 조회기간이 너무 길거나 거래내역이 비정상적으로 많은 경우에는 거래내역조회 API가 과도하게 호출될 수 있으니, **일정 횟수를 초과하지 않도록 제한하는 방어로직의 삽입을 권장**합니다.

(ex) 거래내역조회 API 호출 시 최대 10페이지를 초과하지 않도록 제한 (10페이지 = 250건)

주4) <페이지 Index 사용 안내>

o 페이지 Index 기능을 지원하는 경우

- 광주(034), 전북(037), 신한(088) 은행이 지원

- "page_index_use_yn": "Y", "page_index": "[요청했던 페이지 번호]", "total_record_cnt": "[총 조회/건수]"로 응답 JSON에 셋팅되어 전달됩니다.

- 다음페이지가 존재하는 경우("next_page_yn": "Y")에 페이지 Index를 지정하여 API를 다시 호출하면 해당 페이지의 내역을 응답합니다. (단, 광주은행은 페이지를 순차적으로 요청해야 함)

o 페이지 Index 기능을 지원하지 않는 경우

- 상기 3개 은행을 제외한 나머지 참가은행

- "page_index_use_yn": "N", "page_index": "[요청했던 페이지 번호]", "total_record_cnt": "0"으로

응답 JSON에 셋팅되어 전달됩니다.

- 다음페이지가 존재하는 경우("next_page_yn": "Y")에 다음페이지를 요청하면서 지정하는 페이지 번호(page_index)는 1씩 순차적으로 증가시켜 주시면 됩니다. (2, 3, 4, ...)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "fintech_use_num": "123456789012345678901234", "balance_amt": "1000000", "page_index_use_yn": "N", "page_index": "1", "total_record_cnt": "0", "page_record_cnt": "25", "next_page_yn": "Y", "befor_inquiry_trace_info": "1T201806171", "list_tran_seqno": "0", "res_list": [{ "tran_date": "20160310", "tran_time": "113000", "inout_type": "입금", "tran_type": "현금", "printed_content": "통장인자내용", "tran_amt": "450000", "after_balance_amt": "-1000000", "branch_name": "분당점" }, { ... }, ... { ... }] }</pre>

2.4. 조회서비스(이용기관)

□ 기 능

오픈플랫폼의 이용기관인증(2-legged)을 통하여 획득한 토큰으로 금융정보를 조회합니다.

□ 명 세

2.4.1. 계좌실명조회 API

■ 설 명

이용기관이 특정 계좌의 계좌번호와 **예금주 실명번호**를 보유하고 있는 경우 해당 계좌의 유효성 및 예금주성명을 확인합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/inquiry/real_name
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 이용기관 Access Token 을 HTTP Header 에 추가 [scope = oob]
Body	bank_code_std	"097"	AN(3)	개설기관.표준코드
	account_num	"0001230000678"	AN(16)	계좌번호
	account_holder_info_type	[" " "1" "2" "3" "4" "5" "6" "E"]	AN(1)	예금주 실명번호 구분코드 [32p] 「실명번호 구분코드」 참조
	account_holder_info	"880101"	AN(13)	예금주 실명번호 [32p] 「실명번호 구분코드」 참조
	tran_dtime	"20160310101921"	N(14)	요청일시

[실명번호 구분코드]

코드	실명번호	코드	실명번호
" "(SPACE)	생년월일(실명번호 앞 6 자리)	"1"	주민등록번호
"2"	외국인등록번호	"3"	국내거소신고번호
"4"	조합주민번호	"5"	여권번호
"6"	사업자등록번호	"E"	기타 (외국인투자등록증번호, 외국인투자신고수리서(인가서)번호 등)

※ 생년월일이 아닌 실명번호(전체)를 통한 계좌실명조회는 법적근거가 있는 이용기관에게만 허용됩니다. (ex:소액해외송금업자)

※ 구분코드가 " "(SPACE)인데 예금주 실명번호를 6자리 초과하여 입력한 경우에는 거절처리(응답코드:A0321) 됩니다.

단, 기존에 7자리 실명번호(생년월일+성별)를 입력하는 경우에 대해서는 당분간 앞 6자리만 잘라서 처리하며 응답에는 처리된 6자리만 전달됩니다. 추후에는 6자리만 허용될 예정입니다.

[외국인 계좌 관련 실명번호 체계 안내]

○ 외국인등록번호 : 외국국적을 가진 자가 우리나라에 들어와 90일 이상 체류할 경우 출입국관리 사무소에 신고하여 부여받는 번호

○ 국내거소신고번호 : 해외에 거주하는 우리나라 동포(외국국적동포 포함)가 국내에 90일 이상 체류할 경우 부여받는 번호

<p>외국인등록번호 및 국내거소신고번호</p>	<ul style="list-style-type: none"> ○ 앞자리 1~6 자리 숫자 : 생년월일 ○ 뒷자리 1 자리 숫자 : 출생세기와 성별 <ul style="list-style-type: none"> - 5, 6 : 1900년대 태어난 남자, 여자 - 7, 8 : 2000년대 태어난 남자, 여자 ○ 뒷자리 2~3 자리 숫자 : 등록기관 ○ 뒷자리 4~5 자리 숫자 : 일련번호 ○ 뒷자리 6 자리 숫자는 등록자구분번호 <ul style="list-style-type: none"> - 7 : 외국국적동포 - 8 : 재외국민 - 9 : 외국인 ○ 뒷자리 7 자리 숫자는 오류검증번호
---------------------------	--

○ 조합주민번호(실명번호) : 외국인등록증 및 재외국민(외국국적동포) 국내 거소신고증이 없을 경우 여권번호로 만드는 번호

<p>실명번호</p>	<ul style="list-style-type: none"> ○ 앞자리 1~6 자리 숫자 : 생년월일 ○ 뒷자리 1 자리 숫자 : 성별 <ul style="list-style-type: none"> - 7, 8 : 남자, 여자 ○ 뒷자리 2 자리 숫자 : 국적 <ul style="list-style-type: none"> - 1 : 미국 - 2 : 일본 - 3 : 중국 - 4 : 기타(대한민국 포함) ○ 뒷자리 3~7 자리 숫자 : 여권번호의 마지막 숫자부터 5 자리 <p>※ 여권번호 예) J 0 2 4 6 8 9 0 첫 번째 자리가 국적(A:미국, J:일본, E:기타)</p>
-------------	--

※ 「금융실명거래 업무해설」(은행연합회, 2016. 8.) 발췌 (p.20)

■ 요청 메시지 형태

JSON
<pre>{ "bank_code_std": "097", "account_num": "0001230000678", "account_holder_info_type": " ", "account_holder_info": "880101", "tran_dtime": "20160310101921" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	bank_code_std	"097"	AN(3)	개설기관.표준코드
	bank_code_sub	"1230001"	AN(7)	개설기관.점별코드
	bank_name	"오픈은행"	AH(20)	개설기관명
	account_num	"0001230000678"	AN(16)	계좌번호
	account_holder_info_type	[" " "1" "2" "3" "4" "5" "6" "E"]	AN(1)	예금주 실명번호 구분코드 [32p] 「실명번호 구분코드」 참조
	account_holder_info	"880101"	AN(10)	예금주 실명번호 [32p] 「실명번호 구분코드」 참조
	account_holder_name	"홍길동"	AH(20)	예금주성명

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num": "0001230000678", "account_holder_info_type": " ", "account_holder_info": "880101", "account_holder_name": "홍길동" }</pre>

2.4.2. 송금인정보조회 API (2018.1월 추가)

■ 설 명

소액해외송금업자의 경우에 외화송금을 위하여 이용기관의 수취계좌로 입금한 송금인의 성명과 계좌번호를 확인하여 송금인의 신분을 확인할 의무가 있습니다. 기 등록된 수취계좌에 대하여 특정 기간 동안 입금된 송금인 성명과 계좌번호 내역을 전달합니다. 소액해외송금업자인 경우에만 이용신청이 가능합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/inquiry/remit_list
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		조회서비스 계좌등록 시 오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob]
Parameter	bank_code_std	"097"	AN(3)	조회하고자 하는 이용기관수취계좌의 개설기관.표준코드
	account_num	"3001230000678"	AN(16)	조회하고자 하는 이용기관수취계좌의 계좌번호
	from_date	"20171001"	N(8)	조회시작일자
	from_time	"143000"	N(6)	조회시작시간
	to_date	"20171002"	N(8)	조회종료일자
	to_time	"223000"	N(6)	조회종료시간
	sort_order	"D"	A(1)	정렬순서 D:Descending, A:Ascending
	befor_inquiry_trace_info	"12345678901234567890"	AN(20)	직전조회추적정보*
	tran_dtime	"20171001143000"	N(14)	요청일시

* 다음 페이지 요청 시에 직전 조회의 응답에서 얻은 값을 그대로 셋팅
다음 페이지 요청이 아닌 경우에는 파라미터 자체를 설정하지 않음

■ 요청 메시지 형태

JSON
<pre>{ bank_code_std: "097", account_num: "3001230000678", from_date: "20171001", from_time: "143000", to_date: "20171002", to_time: "223000", sort_order: "D", tran_dtime: "20171001143000" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(은행)
	bank_tran_date	"20160310"	N(8)	거래일자(은행)
	bank_code_tran	"097"	AN(3)	응답코드를 부여한 은행표준코드
	bank_rsp_code	"000"	AN(3)	응답코드(은행)
	bank_rsp_message	""	AN(100)	응답메시지(은행)
	bank_code_std	"097"	AN(3)	조회된 이용기관수취계좌의 개설기관.표준코드
	account_num	"3001230000678"	AN(16)	조회된 이용기관수취계좌의 계좌번호
	balance_amt	"1000000"	SN(12)	계좌잔액(-금액가능)
	total_record_cnt	"0"	N(5)	총 조회건수 ^{주 4)}
	page_record_cnt	"18"	N(2)	현재페이지 레코드건수 ^{주 1)}
	next_page_yn	"Y"	A(1)	다음페이지 존재여부 ^{주 3)}
	befor_inquiry_trace_info	"12345678901234567890"	AN(20)	직전조회 추적정보
	res_list	<object>		조회된 송금내역
	--tran_date	"20160310"	N(8)	거래일자
	--tran_time	"113000"	N(6)	거래시간

--tran_type	"현금"	AH(10)	거래구분 주2)
--print_content	"통장인자내용"	AH(20)	통장인자내용
--tran_amt	"450000"	N(12)	거래금액
--after_balance_amt	"-1000000"	SN(13)	거래 후 잔액(-금액가능)
--branch_name	"분당점"	AH(20)	거래점명
--remitter_name	"송금인"	AH(20)	송금인명
--remitter_bank_code	"097"	AN(3)	송금한 계좌의 은행코드
--remitter_account_num	"5009870000321"	AN(16)	송금한 계좌번호

주1) 한 페이지는 최대 18건 가능

주2) 현금, 대체, 급여, 타행환, F/B출금 등 (참가은행에서 직접 셋팅)

주3) <다음페이지 처리 안내>

"다음페이지 존재여부"(next_page_yn)가 "Y"인 경우에는 다음페이지의 거래내역을 확인하기 위하여, 결과로 받은 "직전조회추적정보"(befor_inquiry_trace_info)와 함께 계속적으로 송금인정보조회 API를 재호출해야 합니다. 다만, 조회기간이 너무 길거나 거래내역이 비정상적으로 많은 경우에는 송금인 정보조회 API가 과도하게 호출될 수 있으니, **일정 횟수를 초과하지 않도록 제한하는 방어로직의 삽입을 권장합니다.**

(ex) 송금인정보조회 API 호출 시 최대 10페이지를 초과하지 않도록 제한 (10페이지 = 180건)

주4) "총 조회건수"는 **광주(034), 전북(037), 신한(088)** 은행의 경우에만 건수가 셋팅되며, 나머지 참가은행의 경우에는 "total_record_cnt": "0"으로 셋팅됩니다.

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "account_num": "3001230000678", "balance_amt": "1000000", "total_record_cnt": "0", "page_record_cnt": "18", "next_page_yn": "Y", "befor_inquiry_trace_info": "12345678901234567890", "res_list": [{ "tran_date": "20160310", "tran_time": "113000", "tran_type": "현금", "printed_content": "통장인자내용",</pre>

```
    "tran_amt": "450000",  
    "after_balance_amt": "-1000000",  
    "branch_name": "분당점",  
    "remitter_name": "송금인",  
    "remitter_bank_code": "097",  
    "remitter_account_num": "5009870000321" },  
    { ... },  
    ...  
    { ... }  
  ]  
}
```

2.5. 이체서비스

□ 기 능

사용자의 계좌로부터 이용기관의 계좌로 자금을 입금하거나, 이용기관의 계좌에서 사용자의 계좌로 자금을 송금합니다. 또한 이체결과를 확인하거나 API요청에 대한 결과를 확인합니다.

□ 명 세

2.5.1. 출금이체 API

■ 설 명

이용기관이 등록된 한 개의 사용자 계좌로부터 대금을 출금합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/transfer/withdraw
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		출금서비스 계좌등록 시 오픈플랫폼으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = transfer]
Body	dps_print_content	"쇼핑물환불"	AH(20)	입금계좌인자내역
	fintech_use_num	"123456789012345678901234"	AN(24)	출금계좌핀테크이용번호
	tran_amt	"10000"	N(12)	거래금액
	tran_dtime	"20160310101921"	N(14)	요청일시

■ 요청 메시지 형태

JSON
<pre>{ "dps_print_content": "쇼핑물환불", "fintech_use_num": "123456789012345678901234", "tran_amt": "10000", "tran_dtime": "20160310101921" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	dps_bank_code_std	"097"	AN(3)	입금기관.표준코드
	dps_bank_code_sub	"1230001"	AN(7)	입금기관.점별코드
	dps_bank_name	"오픈은행"	AH(20)	입금기관명
	dps_account_num_masked	"000-1230000-***"	NS*(20)	입금계좌번호(출력용)
	dps_print_content	"입금계좌인자내역"	AH(20)	입금계좌인자내역
	dps_account_holder_name	"허균"	AH(20)	수취인성명
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	fintech_use_num	"123456789012345678901234"	AN(24)	출금계좌핀테크이용번호
	account_alias	"급여계좌"	AH(50)	출금계좌별명(Alias)
	bank_code_std	"097"	AN(3)	출금(개설)기관.표준코드
	bank_code_sub	"1230001"	AN(7)	출금(개설)기관.점별코드
	bank_name	"오픈은행"	AH(20)	출금(개설)기관명
	account_num_masked	"000-1230000-***"	NS*(20)	출금계좌번호(출력용)
	print_content	"출금계좌인자내역"	AH(20)	출금계좌인자내역
	account_holder_name	"홍길동"	AH(20)	송금인성명
	tran_amt	"10000"	N(12)	거래금액

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "dps_bank_code_std": "097", "dps_bank_code_sub": "1230001", "dps_bank_name": "오픈은행", "dps_account_num_masked": "000-1230000-***", "dps_print_content": "입금계좌인자내역", "dps_account_holder_name": "허균", "bank_tran_id": "12345678901234567890",</pre>

```
{
  "bank_tran_date": "20160310",
  "bank_code_tran": "097",
  "bank_rsp_code": "000",
  "bank_rsp_message": "",
  "fintech_use_num": "123456789012345678901234",
  "account_alias": "급여계좌",
  "bank_code_std": "097",
  "bank_code_sub": "1230001",
  "bank_name": "오픈은행",
  "account_num_masked": "000-1230000-***",
  "print_content": "출금계좌인자내역",
  "account_holder_name": "홍길동",
  "tran_amt": "10000"
}
```

■ 에러 처리

"응답코드(참가은행)"가 정상이 아닌 경우에는 미수납으로 간주하되, 응답코드가 "입금 처리 중"인 경우에는 반드시 이체결과조회 API를 통하여 최종결과를 확인토록 합니다.

2.5.2. 입금이체 API

이용기관이 복수개의 계좌로 대금을 송금합니다. 등록된 핀테크이용번호로 요청하는 경우(/deposit)와 실 계좌번호로 요청하는 경우(/deposit2)의 두 가지 기능을 각각 제공합니다.

□ 입금이체 API : 핀테크이용번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/transfer/deposit
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 이용기관 Access Token 을 HTTP Header 에 추가 [scope = oob]
Body	wd_pass_phrase	"790d56ed.....6b821a69"	AN(128)	입금이체용 암호문구*
	wd_print_content	"환불금액"	AH(20)	출금계좌인자내역
	name_check_option	"on" or "off" (선택)	AN(3)	수취인성명 검증 여부(on:검증함) (미지정 시 기본값: "on")
	req_cnt	"25"	N(5)	입금요청건수**
	req_list	<object>		입금요청목록
	--tran_no	"1"	N(5)	거래순번
	--fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	--print_content	"쇼핑몰환불"	AH(20)	입금계좌인자내역
	--tran_amt	"10000"	N(12)	거래금액
	tran_dtime	"20160310101921"	N(14)	요청일시

* p46 <입금이체용 암호문구 안내> 참조

** 한번의 API 호출로 최대 25건 입금 가능(거래수수료는 입금건별 과금)

■ 요청 메시지 형태

JSON
<pre>{ "wd_pass_phrase": "790d56ed.....6b821a69", "wd_print_content": "환불금액", "name_check_option": "on", "req_cnt": "25", "req_list": [{ "tran_no": "1",</pre>

```

    "fintech_use_num": "123456789012345678901234",
    "print_content": "쇼핑물환불",
    "tran_amt": "10000"
  },
  { ... },
  ...
  { ... }
],
"tran_dtime": "20160310101921"
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	wd_bank_code_std	"097"	AN(3)	출금기관.표준코드
	wd_bank_code_sub	"1230001"	AN(7)	출금기관.점별코드
	wd_bank_name	"오픈은행"	AH(20)	출금기관명
	wd_account_num_masked	"000-1230000-****"	NS*(20)	출금계좌번호(출력용)
	wd_print_content	"환불금액"	AH(20)	출금계좌인자내역
	wd_account_holder_name	"허균"	AH(20)	송금인성명
	res_cnt	"25"	N(5)	입금건수
	res_list	<object>		입금목록
	--tran_no	"1"	N(5)	거래순번
	--bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	--bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	--bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	--bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	--bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	--fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	--account_alias	"급여계좌"	AH(50)	계좌별명(Alias)
	--bank_code_std	"097"	AN(3)	입금(개설)기관.표준코드
	--bank_code_sub	"1230001"	AN(7)	입금(개설)기관.점별코드
	--bank_name	"오픈은행"	AH(20)	입금(개설)기관명
	--account_num_masked	"000-1230000-****"	NS*(20)	입금계좌번호(출력용)
	--print_content	"쇼핑물환불"	AH(20)	입금계좌인자내역
	--account_holder_name	"홍길동"	AH(20)	수취인성명
	--tran_amt	"10000"	N(12)	거래금액

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "wd_bank_code_std": "097", "wd_bank_code_sub": "1230001", "wd_bank_name": "오픈은행", "wd_account_num_masked": "000-1230000-***", "wd_print_content": "환불금액", "wd_account_holder_name": "허균", "res_cnt": "25", "res_list": [{ "tran_no": "1", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "fintech_use_num": "123456789012345678901234", "account_alias": "급여계좌", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "fintech_use_num": "123456789012345678901234", "account_num_masked": "000-1230000-***", "print_content": "쇼핑몰환불", "account_holder_name": "홍길동", "tran_amt": "10000" }, { ... }, ... { ... }] }</pre>

■ 에러 처리

"응답코드(참가은행)"가 정상이 아닌 경우에는 미수납으로 간주하되, 응답코드가 "입금 처리 중"인 경우에는 반드시 이체결과조회 API를 통하여 최종결과를 확인토록 합니다.

□ 입금이체 API : 계좌번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/transfer/deposit2
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 이용기관 Access Token 을 HTTP Header 에 추가 [scope = oob]
Body	wd_pass_phrase	"790d56ed.....6b821a69"	AN(128)	입금이체용 암호문구*
	wd_print_content	"출금계좌인자내역"	AH(20)	출금계좌인자내역
	name_check_option	"on" or "off" (선택)	AN(3)	수취인성명 검증 여부(on:검증함) (미지정 시 기본값: "on")
	req_cnt	"25"	N(5)	입금요청건수**
	req_list	<object>		입금요청목록
	--tran_no	"1"	N(5)	거래순번
	--bank_code_std	"097"	AN(3)	입금은행.표준코드
	--account_num	"0001230000678"	AN(16)	계좌번호
	--account_holder_name	"홍길동"	AH(20)	입금계좌예금주명***
	--print_content	"입금계좌인자내역"	AH(20)	입금계좌인자내역
	--tran_amt	"10000"	N(12)	거래금액
	tran_dtime	"20160310101921"	N(14)	요청일시

* p46 <입금이체용 암호문구 안내> 참조

** 한번의 API 호출로 최대 25건 입금 가능(거래수수료는 입금건별 과금)

*** 참가은행에 등록되어 있는 입금계좌의 예금주명과 정확히 일치해야 함

■ 요청 메시지 형태

JSON
<pre>{ "wd_pass_phrase": "790d56ed.....6b821a69", "wd_print_content": "환불금액", "req_cnt": "25", "req_list": [{ "tran_no": "1", "bank_code_std": "097", "account_num": "0001230000678", "account_hloder_name": "홍길동", "print_content": "쇼핑몰환불", </pre>

```

    "tran_amt": "10000"
  },
  { ... },
  ...
  { ... }
],
"tran_dtime": "20160310101921"
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	wd_bank_code_std	"097"	AN(3)	출금기관.표준코드
	wd_bank_code_sub	"1230001"	AN(7)	출금기관.점별코드
	wd_bank_name	"오픈은행"	AH(20)	출금기관명
	wd_account_num_masked	"000-1230000-***"	NS*(20)	출금계좌번호(출력용)
	wd_print_content	"환불금액"	AH(20)	출금계좌인자내역
	wd_account_holder_name	"허균"	AH(20)	송금인성명
	res_cnt	"25"	N(5)	입금건수
	res_list	<object>		입금목록
	--tran_no	"1"	N(5)	거래순번
	--bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	--bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	--bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	--bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	--bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	--bank_code_std	"097"	AN(3)	입금(개설)기관.표준코드
	--bank_code_sub	"1230001"	AN(7)	입금(개설)기관.점별코드
	--bank_name	"오픈은행"	AH(20)	입금(개설)기관명
	--account_num	"00012300000678"	AN(16)	입금계좌번호
	--account_num_masked	"000-1230000-***"	NS*(20)	입금계좌번호(출력용)
	--print_content	"쇼핑물환불"	AH(20)	입금계좌인자내역
	--account_holder_name	"10000"	N(12)	거래금액
	--tran_amt	"홍길동"	AH(20)	수취인성명

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "wd_bank_code_std": "097", "wd_bank_code_sub": "1230001", "wd_bank_name": "오픈은행", "wd_account_num_masked": "000-1230000-***", "wd_print_content": "출금계좌인자내역", "wd_account_holder_name": "허균", "res_cnt": "25", "res_list": [{ "tran_no": "1", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num": "0001230000678", "account_num_masked": "000-1230000-***", "print_content": "입금계좌인자내역", "account_holder_name": "홍길동", "tran_amt": "10000" }, { ... }, ... { ... }] }</pre>

■ 에러 처리

"응답코드(참가은행)"가 정상인 아닌 경우에는 미수납으로 간주하되, 응답코드가 "입금 처리 중"인 경우에는 반드시 이체결과조회 API를 통하여 최종결과를 확인토록 합니다.

<입금이체용 암호문구 안내>

- 간편가입 회원인 경우 : 기본값 "NONE"를 셋팅함
- 이용기관인 경우 :
(테스트 환경) 기본값 "NONE"를 셋팅
(운영 환경) 오픈플랫폼 포털의 "마이페이지 > 서비스 이용설정" 메뉴에서 설정하며, 이용기관이 암호(9~20자리 영문 대소문자, 숫자, 특수문자)를 입력한 후에 alert창에 표시된 **해시값(128 bytes)**을 복사하여, 입금이체 시에 "입금이체용 암호문구(wd_pass_phrase)" 항목에 그대로 셋팅함

<수취인성명 검증 안내>

- 입금이체 시에 자금을 수취하는 사람이 입금요청한 계좌의 예금주와 동일인인지 비교

API 구분	검증대상	검증값	검증기준
/transfer/deposit	입금요청한 핀테크일련번호에 매핑되어 있는 예금주명 (DB 조회)	이체전 수취조회결과의 예금주명 (참가은행 조회)	문자열에서 모든 공백을 제거후 검증값의 글자수(최대 10 자) 만큼 비교
/transfer/deposit2	인자로 받은 예금주명 (account_holder_name)		

- 수취인성명 검증을 원하지 않는 경우 입금이체 요청 시 **name_check_option="off"**로 셋팅합니다.
(단, 오입금에 대한 책임은 이용기관에게 있습니다.)

※ 영문 이름에 대한 수취인성명 검증 예

	검증대상	검증값	유효글자수	검증결과
사례1)	JUSTIN LEE	JUSTINLEE	9	O
사례2)	JUSTINLEE	JUSTIN LEE	9	O
사례3)	JUSTINLE	JUSTIN LEE	9	X
사례4)	JUSTIN LEE	JUSTIN LE	8	O

- 문자열 비교 시에는 대소문자를 구분하며, 참가은행의 원장에 대문자로 등록되어 있는 경우가 대부분이므로 예금주명 입력 시 **대문자로 입력을 권장**합니다.

2.5.3. 이체결과조회 API

■ 설 명

이용기관이 출금이체 혹은 입금이체 후 이체결과를 다시 확인합니다. 이체결과조회(/result) API는 은행으로부터 이체응답을 받았으나, "처리 중"일 경우 확인하는 용도입니다. 1개월 이내의 이체건에 대해서만 확인이 가능합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/transfer/result
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 이용기관 Access Token 을 HTTP Header 에 추가 [scope = oob]
Body	check_type	"1"	AN(1)	1:출금이체, 2:입금이체
	req_cnt	"25"	N(5)	요청건수*
	req_list	<object>		요청목록
	--tran_no	"1"	N(5)	거래순번
	--org_bank_tran_id	"12345678901234567890"	AN(20)	원거래 거래고유번호(참가은행)
	--org_bank_tran_date	"20160310"	N(8)	원거래 거래일자(참가은행)**
	--org_tran_amt	"10000"	N(12)	원거래 거래금액
	tran_dtime	"20160310101921"	N(14)	요청일시

* 한번의 API 호출로 최대 25건 확인 가능

** 원칙적으로는 이체 원거래 응답에 있는 거래일자를 셋팅해야 하나
시뮬레이터 모드로 테스트시에는 이체 원거래의 실제 테스트일자를 셋팅함

■ 요청 메시지 형태

JSON
<pre>{ "check_type": "1" "req_cnt": "25" "req_list": [{ "tran_no": "1", "org_bank_tran_id": "12345678901234567890", "org_bank_tran_date": "20160310", "org_tran_amt": "10000" }, { ... }, ... { ... }], "tran_dtime": "20160310101921" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	res_cnt	"25"	N(5)	이체건수
	res_list	<object>		이체목록
	--tran_no	"1"	N(5)	거래순번
	--bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	--bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	--bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	--bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	--bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	--wd_bank_code_std	"097"	AN(3)	출금기관.표준코드
	--wd_bank_code_sub	"1230001"	AN(7)	출금기관.점별코드
	--wd_bank_name	"오픈은행"	AH(20)	출금기관명
	--wd_fintech_use_num	"123456789012345678901234"	AN(24)	출금계좌 핀테크이용번호*
	--wd_account_num_masked	"000-1230000-***"	NS*(20)	출금계좌번호(출력용)
	--wd_print_content	"환불금액"	AH(20)	출금계좌인자내역
	--wd_account_holder_name	"허균"	AH(20)	송금인성명
	--dps_bank_code_std	"097"	AN(3)	입금(개설)기관.표준코드
	--dps_bank_code_sub	"1230001"	AN(7)	입금(개설)기관.점별코드
	--dps_bank_name	"오픈은행"	AH(20)	입금(개설)기관명
	--dps_fintech_use_num	"123456789012345678901234"	AN(24)	입금계좌 핀테크이용번호*
	--dps_account_num_masked	"000-1230000-***"	NS*(20)	입금계좌번호(출력용)
	--dps_print_content	"쇼핑몰환불"	AH(20)	입금계좌인자내역
	--dps_account_holder_name	"홍길동"	AH(20)	수취인성명
	--tran_amt	"10000"	N(12)	거래금액

* 해당 계좌에 대한 핀테크이용번호가 존재할 경우만 셋팅됨

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "req_cnt": "25", "req_list": [{ "tran_no": "1", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "wd_bank_code_std": "097", "wd_bank_code_sub": "1230001", "wd_bank_name": "오픈은행", "wd_fintech_use_num": "123456789012345678901234", "wd_account_num_masked": "000-1230000-***", "wd_print_content": "환불금액", "wd_account_holder_name": "허균", "dps_bank_code_std": "097", "dps_bank_code_sub": "1230001", "dps_bank_name": "오픈은행", "dps_fintech_use_num": "123456789012345678901234", "dps_account_num_masked": "000-1230000-***", "dps_print_content": "쇼핑몰환불", "dps_account_holder_name": "홍길동", "tran_amt": "10000" }, { ... }, ... { ... }] }</pre>

2.5.4. 이체API처리조회 API

■ 설 명

이용기관이 출금이체 혹은 입금이체 요청 후 이용기관의 시스템 장애 등으로 API 응답을 받지 못했을 경우에 이체결과를 확인하기 위하여 요청합니다. 이체API처리조회(/recheck) API는 API 요청에 대한 응답 자체를 받지 못한 경우 확인하는 용도입니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/transfer/recheck
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 이용기관 Access Token 을 HTTP Header 에 추가 [scope = oob]
Body	check_type	"1"	AN(1)	1:출금이체, 2:입금이체
	req_cnt	"25"	N(5)	요청건수*
	req_list	<object>		요청목록
	--tran_no	"1"	N(5)	거래순번
	--org_tran_dtime	"20160725100010"	N(14)	원거래요청일시**
	--org_req_gubun	"1"	AN(1)	원거래요청계좌지정방식구분*** 1:핀테크이용번호, 2:계좌번호
	--bank_code_std	"097"	AN(3)	원거래 은행 대표코드
	--account_num	"0001230000678"	N(16)	원거래 입출금계좌번호
	--print_content	"쇼핑몰환불"	AH(20)	원거래 입출금계좌인자내역
	--fintech_use_num	"123456789012345678901234"	AN(24)	원거래 핀테크이용번호
	--org_tran_amt	"10000"	N(12)	원거래 거래금액
	tran_dtime	"20160310101921"	N(14)	요청일시

* 한번의 API 호출로 최대 25건 확인 가능

** 이용기관이 입/출금이체 요청 시 송신한 tran_dtime 항목의 값을 그대로 셋

*** 구분값에 따라 bank_code_std, account_num 혹은 fintech_use_num을 셋팅

■ 요청 메시지 형태

JSON
<pre>{ "check_type": "1" "req_cnt": "25" "req_list": [{ "tran_no": "1", "org_tran_dtime": "20160725100010", "org_req_gubun": "1", "bank_code_std": "097", "account_num": "0001230000678", "print_content": "쇼핑물환불", "fintech_use_num": "", "org_tran_amt": "10000" }, { ... }, ... { ... }], "tran_dtime": "20160310101921" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	res_cnt	"25"	N(5)	이체건수
	res_list	<object>		이체목록
	--tran_no	"1"	N(5)	거래순번
	--bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	--bank_tran_date	"20160310"	N(8)	거래일자(참가은행)
	--bank_code_tran	"097"	AN(3)	응답코드를 부여한 참가은행.표준코드
	--bank_rsp_code	"000"	AN(3)	응답코드(참가은행)
	--bank_rsp_message	""	AN(100)	응답메시지(참가은행)
	--wd_bank_code_std	"097"	AN(3)	출금기관.표준코드
	--wd_bank_code_sub	"1230001"	AN(7)	출금기관.점별코드
	--wd_bank_name	"오픈은행"	AH(20)	출금기관명
	--wd_fintech_use_num	"123456789012345678901234"	AN(24)	출금계좌 핀테크이용번호*

--wd_account_num_masked	"000-1230000-***"	NS*(20)	출금계좌번호(출력용)
--wd_print_content	"출금계좌인자내역"	AH(20)	출금계좌인자내역
--wd_account_holder_name	"허균"	AH(20)	송금인성명
--dps_bank_code_std	"097"	AN(3)	입금(개설)기관.표준코드
--dps_bank_code_sub	"1230001"	AN(7)	입금(개설)기관.점별코드
--dps_bank_name	"오픈은행"	AH(20)	입금(개설)기관명
--dps_fintech_use_num	"123456789012345678901234"	AN(24)	입금계좌 핀테크이용번호*
--dps_account_num_masked	"000-1230000-***"	NS*(20)	입금계좌번호(출력용)
--dps_print_content	"입금계좌인자내역"	AH(20)	입금계좌인자내역
--dps_account_holder_name	"홍길동"	AH(20)	수취인성명
--tran_amt	"10000"	N(12)	거래금액

* 해당 계좌에 대한 핀테크이용번호가 존재할 경우만 셋팅됨

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "req_cnt": "25", "req_list": [{ "tran_no": "1", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20160310", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "wd_bank_code_std": "097", "wd_bank_code_sub": "1230001", "wd_bank_name": "오픈은행", "wd_fintech_use_num": "123456789012345678901234", "wd_account_num_masked": "000-1230000-***", "wd_print_content": "출금계좌인자내역", "wd_account_holder_name": "허균", "dps_bank_code_std": "097", "dps_bank_code_sub": "1230001", "dps_bank_name": "오픈은행", "dps_fintech_use_num": "123456789012345678901234", "dps_account_num_masked": "000-1230000-***", "dps_print_content": "입금계좌인자내역", "dps_account_holder_name": "홍길동", "tran_amt": "10000" }, { ... }, ... { ... }] }</pre>

2.5.5. /transfer/result API와 /transfer/recheck API의 차이 설명

/result API는 은행으로부터 이체응답을 받았으나, "처리 중"일 경우 확인하는 용도입니다.

처리 중일 경우 소정의 시간 간격을 주신 후 retry 시도를 해야 하며,

(예 : 1분, 2분, 4분 간격으로 3번까지 retry)

계속 "처리 중"인 경우 시간 간격을 크게 하여(1시간 후 등) 재질의해야 합니다.

수차례 시도해도 끝까지 확인이 안된 경우, 해당일자 업무종료 후 날짜가 넘어가게 되면 미완료 건으로 분류되고 최종적으로 입금완료 처리됩니다. (즉, 최대한 입금이 되도록 진행합니다. 이때는 개별 건에 대하여 참가은행에 이용기관이 직접 문의 필요)

/recheck API는 API 요청에 대한 응답 자체를 받지 못한 경우에 확인하는 용도입니다.

2.6. 관리 API

□ 기 능

이용기관의 비즈니스 관리를 도와주는 성격의 관리용 API 입니다.

□ 명 세

2.6.1. 참가은행상태조회 API

■ 설 명

이용기관이 참가은행의 서비스 상태(거래가능, 장애, 개시이전, 종료 등)를 확인합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/v1.0/bank/status
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>		오픈플랫폼으로부터 전송받은 이용기관 Access Token 을 HTTP Header 에 추가 [scope = oob]

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"AA12349BHZ1324K82AL3"	AN(20)	거래고유번호(API)
	api_tran_dtm	"20160310101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(100)	응답메시지(API)
	res_cnt	"16"	N(5)	참가은행개수
	res_list	<object>		참가은행목록
	bank_code_std	"097"	AN(3)	참가은행.표준코드
	bank_name	"오픈은행"	AH(20)	참가은행명
	bank_status	"Y"	AN(1)	참가은행상태 Y:거래가능, D:장애 L:개시이전, F:종료예고상태 A:집계상태, E:종료

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "AA12349BHZ1324K82AL3", "api_tran_dtm": "20160310101921567", "rsp_code": "A0000", "rsp_message": "", "res_cnt": "16", "res_list": [{ "bank_code_std": "097", "bank_name": "오픈은행", "bank_status": "Y" }, { ... }, ... { ... }] }</pre>

2.7. [deprecated] 구)사용자인증(OAuth 2.0)

□ 기 능

오픈플랫폼 실시 초기에 제공하던 ID/Password 기반의 사용자인증 API 입니다. 앞으로는 API 지원이 중단될(deprecated) 예정입니다.

□ 명 세

2.7.1. 사용자로그인연결 API (3-legged) : 구버전

■ 설 명

이용기관의 앱과 오픈플랫폼 사용자 ID를 연결합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/authorize
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	설명
Parameter	response_type	고정값: code	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	사용자로그인연결이 성공되면 이용기관으로 연결되는 URL
	scope	고정값: login [login inquiry transfer]	Access Token 권한 범위
	client_info	<client_info> (Max: 256 bytes)	이용기관이 셋팅하는 임의의 정보. Callback 호출 시 그대로 전달됨

■ 응답 메시지 URL

HTTP URL	<callback_uri> [Callback URL 로 redirect]
HTTP Method	GET

■ 응답 메시지 명세

항목	값	설명
code	<authorization_code>	사용자인증 성공 시 반환되는 코드
scope	고정값	Access Token 권한 범위
client_info	<client_info>	요청 시 이용기관이 셋팅한 정보를 그대로 전달

2.7.2. 계좌등록 API (3-legged) : 구버전

■ 설 명

사용자로그인연결이 완료된 이후에 오픈플랫폼에서 제공하는 서비스 이용을 위한 사용자 계좌를 ARS 동의를 통하여 등록합니다. 조회서비스(잔액조회, 거래내역조회)를 위한 계좌등록과 출금서비스(출금이체)를 위한 계좌등록으로 구분됩니다. (multi scope도 가능)

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/register_account
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	설명
Parameter	response_type	고정값: code	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	계좌등록이 완료되면 이용기관으로 연결되는 URL
	scope	[inquiry transfer]	inquiry : 조회서비스용 계좌등록 transfer : 출금서비스용 계좌등록
	client_info	<client_info> (Max: 256 bytes)	이용기관이 셋팅하는 임의의 정보 Callback 호출 시 그대로 전달됨

■ 응답 메시지 URL

HTTP URL	<callback_uri> [Callback URL로 redirect]
HTTP Method	GET

■ 응답 메시지 명세

항목	값	설명
code	<authorization_code>	사용자인증 성공 시 반환되는 코드
scope	[inquiry transfer]	Access Token 권한 범위
client_info	<client_info>	요청 시 이용기관이 셋팅한 정보를 그대로 전달

2.7.3. 계좌등록확인 API (3-legged) : 구버전

■ 설 명

사용자인증을 통하여 이미 등록된 계좌에 대해서 재확인 후 접속토큰을 재발급 받는 API입니다.
토큰발급(/token) API를 통하여 Refresh Token으로 토큰을 재발급(갱신)할 수 있지만, 토큰 재발급
과는 별도로 매 1년 마다 사용자의 재확인을 받아야 합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.open-platform.or.kr/oauth/2.0/authorize_account
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	설명
Parameter	response_type	고정값: code	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id>	오픈플랫폼에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	계좌등록확인이 성공되면 이용기관으로 연결되는 URL
	scope	[inquiry transfer]	Access Token 권한 범위
	client_info	<client_info> (Max: 256 bytes)	이용기관이 셋팅하는 임의의 정보 Callback 호출 시 그대로 전달됨

■ 응답 메시지 URL

HTTP URL	<callback_uri> [Callback URL로 redirect]
HTTP Method	GET

■ 응답 메시지 명세

항목	값	설명
code	<authorization_code>	사용자인증 성공 시 반환되는 코드
scope	[inquiry transfer]	인증된 사용자의 권한
client_info	<client_info>	요청 시 이용기관이 셋팅한 정보를 그대로 전달

※ 구버전의 토큰발급은 「2.1. 사용자인증(OAuth 2.0)」 [개선버전] 부분의 토큰발급 부분을 참조합니다.

별첨

별첨1) 데이터 Type 설명

약자	설명
N	NUMERIC
S	SPECIAL
A	ALPHABETIC
AH	한글 (KSC-5601 2 Byte 완성형)
AN	ALPHA-NUMERIC
SN	SIGNED-NUMERIC (Sign 부호 + Numeric)
ANS	ALPHA-NUMERIC SPECIAL (AN + '-'(Hyphen) + Space)
NS*	NUMERIC SPECIAL ASTERISK (N + '-'(Hyphen) + '*'(Asterisk) + Space)
Hx	HEXA-DECIMAL

별첨2) 용어 설명

약자	설명	비고
이용기관코드	이용기관이 오픈플랫폼에 이용기관 등록을 통하여 발급받은 이용기관 식별코드 [이용기관코드 확인] (계약전) "포털 > 마이페이지 > 회원정보 변경" (계약후) "포털 > 마이페이지 > 계약신청 > 이용계약 조회"	오픈플랫폼에서 이용기관에 발급
핀테크이용번호	사용자가 오픈플랫폼에 계좌등록 시 발급되는 가상의 계좌식별번호	오픈플랫폼에서 이용기관에 발급

별첨3) 은행코드

은행코드(3 자리)	은행명(축약)	은행명(전체)	은행코드(3 자리)	은행명(축약)	은행명(전체)
002	산업	KDB 산업은행	031	대구	대구은행
003	기업	IBK 기업은행	032	부산	부산은행
004	국민	KB 국민은행	034	광주	광주은행
007	수협	수협중앙회	035	제주	제주은행
011	농협	NH 농협은행	037	전북	전북은행
020	우리	우리은행	039	경남	경남은행
023	SC 제일	SC 제일은행	081	KEB 하나	KEB 하나은행
027	씨티*	한국씨티은행	088	신한	신한은행
			097	오픈**	오픈은행
※ 전체은행코드 참조 : 금융결제원 홈페이지(www.kftc.or.kr) 접속 > "금융회사 코드조회"(우측 하단) 클릭 > "금융회사 공동코드" 파일(HWP/PDF) 다운로드					

* 씨티는 오픈플랫폼 참가은행이나 서비스는 미실시 상태임

** 오픈은행은 테스트를 위한 가상의 은행임(운영 시스템에는 미존재)

별첨4) OAuth 응답코드

API 응답코드	API 응답메시지 (RESTful API 응답)
O0001	인증요청 거부-인증 파라미터 오류([error_code])
O0002	Access Token 거부
O0003	Access Token 만료
O0004	API 접근권한이 없음 (API 서비스를 신청하지 않았을 때)
O0005	허용되지 않은 API 접근 입니다. (API 서비스가 없을 때)
O0006	허용되지 않은 IP 입니다.
O0007	API 를 요청 또는 처리할 수 없습니다. (API 업무처리 Routing 실패 시)
O0008	Account Plan 사용량 한도 초과
O0009	API Plan 사용량 한도 초과
O0010	허용되지 않은 HTTP method 입니다.

별첨5) API 인증요청 거부(O0001)에 대한 세부 응답코드(error_code)

error_code	에러내용(영문)	에러내용(한글)
119	"error": "unsupported_grant_type", "error_description": "The given grant_type is not supported"	Gateway 가 제공하지 않는 grant_type 값이 들어왔을 때
992	"error": "invalid_request" "error_description": "Validation error"	Http Header 에 Authorization Bearer access token 이 없을 때
3000103	"error": "invalid_request" "error_description": "Missing or duplicate parameters"	인증에 필요한 필수 파라미터 값이 존재하지 않을 때 or 특정 파라미터 값이 중복으로 들어왔을 때
xxxx201	"error": "invalid_client" "error_description": "The given client credentials were not valid"	client_id 가 존재하지 않을 때 or key/secret 이 맞지 않을 때 or 비활성화되어 사용할 수 없을 때
xxxx113	"error": "invalid_grant", "error_description": "The given grant is invalid"	유효하지 않은 authorization code (만료 or 기사용)
3000114	"error": "invalid_redirect_uri" "error_description": "Mismatching redirect_uri"	등록한 callback url 과 요청한 callback url 이 맞지 않을 때
3000115	"error": "invalid_scope" "error_description": "No registered scope value for this client has been requested"	scope 이 맞지 않을 때(기본 제공하는 scope 명[oob login inquiry transfer]이 맞지 않을 때)
3000116	"error": "unsupported_response_type" "error_description": "None of the supported response_types were used"	response_type 유형이 맞지 않을 때
3002110	"error": "invalid_request" "error_description": "The session has expired or already been granted. The login process has to be repeated to be successful"	만료된 session 값을 사용하였을 때

별첨6) 사용자인증 Callback 에러코드 및 에러메시지

error	error_description	비고
invalid_request	파라미터에 오류가 있는 경우	요청 파라미터의 값 검증 시 오류
unauthorized_client	client_id 가 유효하지 않은 경우	사용자로그인연결동의가 안되어 있는 경우, 이용기관 앱 정보가 유효하지 않은 경우
access_denied	사용자가 '취소' 버튼을 클릭한 경우	휴대폰 본인인증에 대한 동의 거부, 계좌등록 ARS 녹취 동의에 대한 거부
unsupported_response_type	response_type 이 'code'가 아닌 경우	
invalid_scope	scope 이 지정된 scope 이 아닌 경우	login, inquiry, transfer 에 해당되지 않는 경우
server_error	HTTP 500 에러에 해당함 (Internal Server Error)	계좌등록 시 참가은행과의 전문처리 오류, 참가은행으로부터 오류응답을 받은 경우
temporarily_unavailable	HTTP 503 에러에 해당함 (Service Unavailable)	

별첨7) API 응답코드

API 응답코드	API 응답메시지 (RESTful API 응답)
A0000	처리 성공
A0001	처리 중(처리결과조회 요망, 이체 시)
A0002	참가은행 에러
A0003	내부 처리 에러
A0004	요청전문 포맷 에러
A0005	등록 데이터 에러
A0006	전문 변환 에러
A0007	처리시간 초과 에러
A0008	중복거래 에러
A0009	API 세부업무 처리실패(리스트 건별 처리결과 확인)
A0010	이용기관 APP 정보 확인 실패
A0011	이용기관 API 사용권한 없음
A0012	API 정보 확인 실패
A0013	이용기관 서비스 사용 불가
A0014	탈퇴 회원
A0015	시뮬레이터 응답전문 존재하지 않음
A0016	FLAT 전문 송신 실패

A0017	참가은행 응답전문 TIMEOUT
A0018	거래내역 없음
A0101	이용기관 입금이체 한도 초과(일 한도)
A0102	이용기관 입금이체 한도 초과(월 한도)
A0103	이용기관 출금이체 한도 초과(일 한도)
A0104	이용기관 출금이체 한도 초과(월 한도)
A0105	이용기관 출금이체 한도 초과(건당 한도)
A0106	이용기관 입금이체 한도 초과(건당 한도)
A0111	사용자 출금이체 한도 초과(건당 한도)
A0112	사용자 출금이체 한도 초과(일 한도)
A0301	접근 권한 없음
A0302	참가은행 API 이용권한 없음
A0303	등록된 이용기관 수수료 정책 없음
A0304	핀테크이용번호 정보 불일치
A0305	제 3 자정보제공동의 미완료
A0306	출금동의 미완료
A0307	이체암호문구 불일치
A0308	처리대행비용 할인대상 여부 없음
A0309	등록된 이용기관 수수료 정보 없음
A0310	이체 내역 없음
A0311	등록된 처리대행 수수료 정보 없음
A0312	예금주명 불일치
A0313	사용자 불일치
A0314	계좌이동 중
A0315	농협중앙회(단위농축협) 계좌번호를 입력했을 경우 오류 처리
A0316	금융(거래)정보 제 3 자제공동의 만료
A0317	미등록 수취계좌
A0318	수취계좌 사업자등록번호 상이
A0319	출금동의 만료
A0320	실명번호(전체) 조회 권한 없음
A0321	예금주 실명번호 구분코드와 실명번호의 형식(자리수) 불일치

별첨8) 참가은행 응답코드

유 형	코드	내 용	기관사용 가능여부
거래사항	000	정상	○
시 스템 관련사항	111	출금(개설)기관 SYSTEM 장애	×
	112	출금(개설)기관 개시 이전	×

	113	출금(개설)기관 업무 종료	×
	114	출금(개설)기관 서비스 시간 아님	×
	115	출금(개설)기관 처리지연으로 거부(잠시 후 거래 요망)	○
	121	중계센터 SYSTEM 장애	×
	122	중계센터 종료	×
	141	입금기관 SYSTEM 장애	×
	142	입금기관 개시 이전	×
	143	입금기관 업무 종료	×
	145	입금기관 처리지연으로 거부(잠시 후 거래요망)	○
	146	처리지연으로 거부(잠시 후 거래요망)	○
	150	미참가 기관	×
	152	개설기관 수신전문 오류	×
공통업무	311	TIME OVER(응답대기시간 경과)	×
	312	기관지정 에러메시지	○
	313	요청전문 포맷 에러	×
이체 처리결과조회	400	입금 처리 중	×
	401	이체 전 처리결과 조회 수신으로 거부	×
	402	수취 조회 거래고유번호 검색 실패	×
	403	수취 조회 정보 불일치	×
	999	순이체한도 초과	×
사용자등록 수취조회 이체 잔액조회 거래내역조회 송금인정보조회 계좌실명조회 처리결과조회	411	과목코드 오류	○
	412	해당계좌 없음(전출, 잡좌통합, 특별계좌 포함)	○
	413	통장 분실 재발행계좌	○
	414	이관 계좌	○
	415	해약 계좌	○
	416	잡좌	○
	417	비실명계좌	○
	418	보안계좌	○
	419	사고신고 계좌	○
	420	거래중지 계좌	○
	421	법인계좌 사용불가	○
	422	타행처리 불가계좌	○
	423	투자자예탁금 계좌 아님	○
	424	연체계좌	○
	425	법적등록계좌(압류,가압류,예금주사망등 기타)	○
	426	압류금지 전용 계좌로 입금불가	○
	427	기타 출금불가 계좌 (해당 지점 연락요망)	○
	428	동의서 미징구 계좌	○

	429	가상계좌 입금시간 아님	○
	430	예약금융상품 온라인 입금불가	○
	431	잔액 및 부채증명 발급계좌	○
	432	통장정리후 거래(무통건수 초과)	○
	433	인감분실	○
	434	총금액 상위	○
	435	불입금 상위(1 회 불입단위가 있는 경우)	○
	436	불입횟수 초과(해당 지점 연락요망)	○
	437	입금한도 초과	○
	451	지급횟수 초과	○
	452	비밀번호 입력횟수 초과	○
	453	예금잔액 부족	○
	454	출금가능잔액 부족	○
	455	건별 이체한도 초과	○
	456	일일 이체한도 초과	○
	457	대월한도 초과(대월잔액 부족)	○
	458	30 분간 지연인출(이체) 대상	○
	459	CMS 번호 기재 요망	○
	460	CMS 번호 상위	○
	461	의뢰인 성명 오류(해당 지점 연락요망)	○
	462	고객 앞 통지 반송 계좌(해당 지점 연락요망)	○
	463	실명번호 상위	○
	464	사용자 등록 정보 이상(기 해지, 동의정보 상위 등)	○
	465	실명번호 구분코드가 계좌에 등록된 실명번호와 불일치 ※ 본 응답코드는 '계좌실명 조회' 전문에만 사용 가능	○
	466	계좌 등록 실명번호가 생년월일로 시작하지 않음 ※ 본 응답코드는 '사용자 등록' 및 '계좌실명 조회' 전문에만 사용 가능	○
	467	장기 미사용 계좌	○
	480	이용기관 정보 주거래기관 등록 이상	○
	499	기타 처리불가(해당 지점 연락요망)	○
거래내역조회 송금인정보조회	501	조회 가능기간 경과	○
	502	조회 대상 거래내역 없음	○
사용자등록	551	기 해지 사용자	○
	552	계좌명의 구분 상위	○
	553	생년월일 상위	○
	554	기타 사용자 등록 불가 계좌(해당 지점 연락요망)	○
이체취소	601	원거래 없음	○
	602	기 취소된 거래임(정상취소)	○

	603	원거래 불명(원거래 비정상)	○
	607	원거래 계좌번호 상위	○
	608	원거래 금액 상위	○
	609	원거래 기타 사항 상위	○
처리결과조회	701	조회 대상거래 없음	×
내부전문 변환처리	801	등록 데이터 에러	×
	802	전문 변환 에러	×
	803	내부 처리 에러	×
	804	처리시간 초과 에러	×
	805	중복거래 에러	×
	806	참가은행 API 이용권한 없음	×
	807	핀테크이용번호 정보 불일치	×
	808	제 3 자정보제공동의 미완료	×
	809	출금동의 미완료	×
	810	이체비밀번호 불일치	×
	811	처리대행비용 할인 대상 여부 정보 없음	×
	812	등록된 이용기관 수수료 정보 없음	×
	813	이체 내역 없음	×
	814	등록된 처리대행 수수료 정보 없음	×
	815	예금주명 불일치	×
	816	탈퇴 회원	×
	817	사용자 불일치	×
	818	시뮬레이터 응답전문 존재하지 않음	×
	819	FLAT 전문 송신 실패	×
	820	농협중앙회(단위농축협) 계좌번호를 입력했을 경우 거절처리	×
	821	금융(거래)정보 제 3 자제공동의 만료	×