

오픈뱅킹공동업무
API 명세서
Version 2.2.1

금융결제원
2020. 01.

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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 간편가입 회원의 입금이체 암호문구 관련 안내 보강 - 거래내역조회/송금인정보조회 시 다음페이지 처리 안내 추가 	금융결제원
1.8	2018.07.13	<ul style="list-style-type: none"> - 다국어 4개 추가(방글라데시/파키스탄/러시아/우즈베키스탄) 	금융결제원
1.8	2018.07.18	<ul style="list-style-type: none"> - 거래내역조회 API 페이지 처리 관련 설명 추가 - 송금인정보조회 API 페이지 처리 관련 설명 추가 	금융결제원
1.8	2018.07.25	<ul style="list-style-type: none"> - 계좌실명조회 기능 개선(주민등록번호, 외국인등록번호, 국내거소신고번호 등 실명번호 종류 추가) 	금융결제원
1.8	2018.08.07	<ul style="list-style-type: none"> - 다국어 2개 추가(몽골/스리랑카) 	금융결제원
1.8	2018.08.30	<ul style="list-style-type: none"> - 다국어 2개 추가(일본/중국) 	금융결제원
1.8	2019.01.17	<ul style="list-style-type: none"> - 거래내역조회 API 설명 추가 (1개월 이내 이체거래 조회 가능) - 오타 수정 	금융결제원
2.0.0 (초안)	2019.10.23	<ul style="list-style-type: none"> - 설계서 명 변경 (은행권 공동 오픈플랫폼 → 오픈뱅킹공동업무) - 도메인 변경 (open-platform.or.kr → openbanking.or.kr) - API URI 변경 (v1.0 → v2.0) - 사용자인증, 계좌등록확인, 잔액조회, 거래내역조회, 입금이체, 출금이체 API URI 변경 - 센터인증, 자체인증 이용기관 API URI 추가 - API URI 맵 안내사항 수정 - 버전관리정책(1.4) 삭제 - [deprecated] 구)사용자인증(OAuth 2.0) API 삭제 - 사용자인증 예외처리 안내 수정 - 자체인증 이용기관 토큰발급 API 추가 - 토큰 권한범위(sa) 추가 - 토큰 설명 변경 및 Scope 범위 추가 - 사용자정보조회 API 설명 및 응답 항목 추가 - 등록계좌조회 API 계좌상태코드 삭제 - 계좌해지 API 설명 추가 - 사용자탈퇴 API 추가 - 사용자계좌등록 (자체인증 이용기관용) API 추가 - 잔액조회(계좌번호 사용) API 추가 - 거래내역조회 기능 개선 (시간기준 조회 추가) - 거래내역조회(계좌번호 사용) API 추가 - 계좌실명조회 실명번호 구분코드(N) 추가 - 이체 시 복수 약정계좌 사용 관련 항목, 자금세탁방지 정보 수집 관련 항목, 하위가맹점 정보 관련 항목 추가 - 이체 시 예러처리 관련 안내 추가 - 출금이체(계좌번호 사용) API 추가 - 입금이체 API 입금이체용 암호문구 안내 수정 - 요청 고객 정보 안내 추가 - 이체용도 안내 추가 - 응답코드 추가 및 수정 - 상기사항 추가에 따른 페이지 조정 	금융결제원

2.0.0a	2019.10.23	<ul style="list-style-type: none"> - 은행거래고유번호 요청 항목 추가(계좌해지, 사용자계좌등록, 잔액조회, 거래내역조회, 계좌실명조회, 송금인정보조회, 출금이체, 입금이체) - 사용자계좌등록 에러처리 및 유의사항 추가 - 잔액조회(계좌번호 사용), 거래내역조회(계좌번호 사용), 출금이체(계좌번호 사용) 사용자일련번호 요청 항목 추가 - 거래내역조회 조회기준 처리 안내 변경 및 입출금구분 설명 추가 - 거래내역조회 총 조회건수 응답 항목 삭제 - 출금이체 출금계좌인자내역 설명 추가 - 출금이체 출금한도잔여금액 응답 항목 추가 - 요청고객 회원번호 개인정보 유의사항 추가 - 응답코드 추가 및 수정 - 자체 출금동의 방식의 안전한 구현을 위한 금융보안원 권고사항 추가 - 이체 시 에러처리 관련 안내 수정 - 이체결과조회 설명 변경 및 "입금 처리 중" 응답코드 확인방법 추가 - 이체 API 처리조회 [deprecated] 안내 - /transfer/result API와 /transfer/recheck API의 차이 설명 삭제 - 데이터 Type (B64) 추가 	금융결제원
2.0.0b	2019.10.23	<ul style="list-style-type: none"> - 사용자계좌등록 이메일주소 항목 추가 - 데이터 Type (E) 추가 	금융결제원
2.0.0c	2019.10.23	<ul style="list-style-type: none"> - 거래내역조회, 송금인정보조회 직전조회추적정보 요청 항목 필수여부 변경(Y → N) - 토큰발급 시 에러 응답 안내 추가 - 사용자 출금동의 만료 및 금융정보 제3자제공동의 만료 처리 추가 - 계좌등록확인 설명 수정 	금융결제원
2.0.1	2019.10.23	<ul style="list-style-type: none"> - 토큰 형태 안내 문구 추가 - client_id, client_secret, token의 최대 길이 추가 - 사용자인증 API, 계좌등록확인 API 요청 항목(state), 응답 항목(state) 추가 - 데이터 Type (aN, aNS) 추가 및 데이터 Type (A, AN, ANS) 수정 (a : 대소문자 모두 가능, A : 대문자만 가능) - 데이터 Type 관련하여 모든 요청, 응답 항목에 상기사항 반영 * 데이터 Type 변경 항목 목록 <ul style="list-style-type: none"> - 거래고유번호(API) : ANS → aNS (대소문자 가능) - 입금이체용 암호문구 : AN → aN (대소문자 가능) - 수취인성명 검증 여부 : AN → aN (대소문자 가능) - 사용자정보조회 URL 수정 (/user/me → /v2.0/user/me) - 응답메시지(API) 항목 길이 수정 (100 → 300) - 데이터 Type (AH) 설명 추가 및 데이터 Type 길이 설명 추가 <ul style="list-style-type: none"> ※ 데이터 Type의 길이는 Byte 길이를 의미함 - 응답메시지(참가은행) DATA Type 수정 (AN → AH) - 이체 시 약정 계좌/계정 구분 설명 문구 수정 - 은행코드 추가 (케이뱅크, 카카오뱅크) - 상기사항 추가에 따른 페이지 조정 	금융결제원

2.1.0	2019.12.07	<ul style="list-style-type: none"> - 사용자인증 API 계좌소유인증 사용여부 항목 추가 - 사용자탈퇴 API 설명 문구 추가 - 계좌해지 API 설명 추가 - 사용자계좌등록 API 요청 시 조회서비스 등록 시에만 이메일주소 입력하도록 수정 - 사용자계좌등록 API 요청 시 오류 응답코드(A0019) 추가 - 사용자계좌등록 응답 항목(출금등록거래고유번호, 출금등록거래일자) 및 설명 추가 - 사용자탈퇴 API 요청 후 사용자계좌등록 관련 처리 안내 추가 - 계좌실명조회 응답 항목(계좌종류) 추가 - 이체 시 요청고객 핀테크이용번호 요청 항목 추가 - 출금이체 출금계좌인자내역 항목 추가 - 출금이체 최종수취고객정보 항목 추가 - 인증구분 및 출금계좌인자내역 항목 설정여부에 따른 사용자(고객)통장 기재 내용 안내 추가 - 출금이체 API 사용자한도초과 오류 시에도 응답메시지에 출금한도잔여금액 항목 포함하도록 수정 - 입금이체 에러 처리 안내 (A0007 응답코드) 추가 - 수수료조회, 집계조회 API 추가 - 수수료조회, 집계조회 시 응답코드 (A0020) 추가 - 참가은행 응답코드 (555) 추가 - 출금이체한도조회 API 추가 - 별첨10) API 관리 ID 추가 	금융결제원
2.1.1	2020.02.19. 예정	<ul style="list-style-type: none"> - 사용자인증, 계좌등록확인 재인증 기능 및 요청 항목 삭제 - 이동통신사 코드표 삭제 	금융결제원
2.2.0	2020.03.26 예정	<ul style="list-style-type: none"> - [deprecated] 이체API처리조회 API 삭제 - 복수 계좌로의 입금이체 요청 [deprecated] 안내 추가 - 사용자 토큰 발급 API redirect_uri 설명문구 추가 - 등록계좌조회 정렬순서 관련 기준 추가 - 거래내역조회 일자기준조회 시 최소제공기간 명시 (6개월) - 수취조회 API 추가 - 이체서비스 설명 문구 변경 - 입금이체 API 요청, 응답 항목 (CMS 번호) 추가 - 수취인성명 검증 안내 문구 추가 - 이체처리결과조회 오류 응답 처리 안내 추가 - 자금반환 청구, 자금반환 결과조회 API 추가 - 수수료조회 이용기관수수료 설명 추가 - 출금이체한도조회 건당 출금이체한도금액 응답항목 오타 수정 (perc_wd_lmit_amt → perc_wd_limit_amt) - 이상금융거래 탐지내역 조회 API 추가 - 복수 기관코드 관련 내용 추가 - 오픈뱅킹센터 지정 에러메시지 응답코드 (A0021) 추가 - 자금반환 관련 참가은행 응답코드 추가 - API 관리 ID(008:수취조회) 추가 - 상기사항 추가에 따른 페이지 조정 	금융결제원
2.2.1	2020.03.26 예정	<ul style="list-style-type: none"> - 자금반환 청구 요청 항목 (이용기관 담당자 이메일주소) 추가 	금융결제원

Table of Contents

제 1 장 API 목록	1
1.1. 개 요	1
1.2. API URI 맵	1
1.3. OAuth 이용안내	4
1.4. 응답코드 처리	4
제 2 장 API 명세	5
2.1. 사용자인증(OAuth 2.0)	5
2.1.1. 사용자인증 API	5
2.1.2. 토큰발급 API	9
2.1.3. 계좌등록확인 API	15
2.1.4. 권한범위 및 토큰갱신 안내	17
2.1.5. 사용자인증 에러 처리 방식	20
2.1.6. 계좌 선거절 처리 안내	21
2.1.7. 다국어 설정 안내	22
2.2. 사용자/계좌 관리	23
2.2.1. 사용자정보조회 API	23
2.2.2. 사용자로그인연결동의해제 API	26
2.2.3. 등록계좌조회 API	27
2.2.4. 계좌정보변경 API	29
2.2.5. 계좌해지 API	30
2.2.6. 사용자탈퇴 API	32
2.2.7. 사용자계좌등록 API	33
2.3. 조회서비스(사용자)	36
2.3.1. 잔액조회 API	36
2.3.2. 거래내역조회 API	40
2.4. 조회서비스(이용기관)	47
2.4.1. 계좌실명조회 API	47
2.4.2. 송금인정보조회 API	51
2.4.3. 수취조회 API	55

2.5. 이체서비스	58
2.5.1. 출금이체 API	58
2.5.2. 입금이체 API	67
2.5.3. 이체결과조회 API	78
2.5.4. 자금반환 청구 API	82
2.5.5. 자금반환 결과조회 API	86
2.6. 관리 API	89
2.6.1. 참가은행상태조회 API	89
2.6.2. 수수료조회 API	91
2.6.3. 집계조회 API	93
2.6.4. 출금이체한도조회 API	95
2.6.5. 이상금융거래 탐지내역 조회 API	97
별첨	99
별첨1) 데이터 Type 설명	99
별첨2) 용어 설명	99
별첨3) 은행코드	100
별첨4) OAuth 응답코드	101
별첨5) API 인증요청 거부(O0001)에 대한 세부 응답코드(error_code)	101
별첨6) 사용자인증 Callback 에러코드 및 에러메시지	102
별첨7) API 응답코드	103
별첨8) 참가은행 응답코드	105
별첨9) 자체 출금동의 방식의 안전한 구현을 위한 금융보안원 권고사항	109
별첨10) API ID	109

제 1 장 API 목록

1.1. 개 요

본 문서는 "오픈뱅킹공동업무"(이하 "오픈뱅킹")에서 제공하는 API 명세를 기술합니다. 오픈뱅킹 이용승인을 받은 핀테크기업(이하 "이용기관")은 오픈뱅킹이 제공하는 API를 활용하여 이용기관용 응용프로그램(이하 "앱")을 개발할 수 있습니다.

1.2. API URI 맵

오픈뱅킹의 API는 REST(REpresentational State Transfer) 방식으로 구현되어, 인터넷 웹 서비스의 형태로 제공됩니다. 아래와 같은 API를 제공하며, 하나의 API는 하나의 웹 서비스 URI로 매핑됩니다.

기능구분	API 명	URI	HTTP Method
사용자인증 (OAuth 2.0)	사용자인증	https://openapi.openbanking.or.kr/oauth/2.0/authorize	GET
	토큰발급	https://openapi.openbanking.or.kr/oauth/2.0/token	POST
	계좌등록확인	https://openapi.openbanking.or.kr/oauth/2.0/authorize_account	GET
사용자/계좌 관리	사용자정보조회	https://openapi.openbanking.or.kr/v2.0/user/me	GET
	사용자로그인 연결동의해제	https://openapi.openbanking.or.kr/v2.0/user/unlink	POST
	등록계좌조회	https://openapi.openbanking.or.kr/v2.0/account/list	GET
	계좌정보변경	https://openapi.openbanking.or.kr/v2.0/account/update_info	POST
	계좌해지	https://openapi.openbanking.or.kr/v2.0/account/cancel	POST
	사용자탈퇴	https://openapi.openbanking.or.kr/v2.0/user/close	POST
	사용자계좌등록	https://openapi.openbanking.or.kr/v2.0/user/register	POST
조회서비스 (사용자)	잔액조회	https://openapi.openbanking.or.kr/v2.0/account/balance/fin_num	GET
		https://openapi.openbanking.or.kr/v2.0/account/balance/acnt_num	POST
	거래내역조회	https://openapi.openbanking.or.kr/v2.0/account/transaction_list/fin_num	GET
		https://openapi.openbanking.or.kr/v2.0/account/transaction_list/acnt_num	POST
조회서비스 (이용기관)	계좌실명조회	https://openapi.openbanking.or.kr/v2.0/inquiry/real_name	POST
	송금인정보조회	https://openapi.openbanking.or.kr/v2.0/inquiry/remittance_list	POST
	수취조회	https://openapi.openbanking.or.kr/v2.0/inquiry/receive	POST
이체서비스	출금이체	https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/fin_num	POST
		https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/acnt_num	POST
	입금이체	https://openapi.openbanking.or.kr/v2.0/transfer/deposit/fin_num	POST
		https://openapi.openbanking.or.kr/v2.0/transfer/deposit/acnt_num	POST
	이체결과조회	https://openapi.openbanking.or.kr/v2.0/transfer/result	POST
	자금반환청구	https://openapi.openbanking.or.kr/v2.0/return/claim	POST

	자금반환결과 조회	https://openapi.openbanking.or.kr/v2.0/return/result	POST
관리 API	참가은행상태 조회	https://openapi.openbanking.or.kr/v2.0/bank/status	GET
	수수료조회	https://openapi.openbanking.or.kr/v2.0/manage/fee	GET
	집계조회	https://openapi.openbanking.or.kr/v2.0/manage/count	GET
	출금이체한도 조회	https://openapi.openbanking.or.kr/v2.0/manage/wd_limit	GET
	이상금융거래 탐지내역조회	https://openapi.openbanking.or.kr/v2.0/inquiry/fds_detect	GET

■ 센터인증(자체인증 미사용) 이용기관 API URI 맵

기능구분	API 명	URI	HTTP Method
사용자인증 (OAuth 2.0)	사용자인증	https://openapi.openbanking.or.kr/oauth/2.0/authorize	GET
	토큰발급	https://openapi.openbanking.or.kr/oauth/2.0/token	POST
	계좌등록확인	https://openapi.openbanking.or.kr/oauth/2.0/authorize_account	GET
사용자/계좌 관리	사용자정보조회	https://openapi.openbanking.or.kr/v2.0/user/me	GET
	사용자로그인 연결동의해제	https://openapi.openbanking.or.kr/v2.0/user/unlink	POST
	등록계좌조회	https://openapi.openbanking.or.kr/v2.0/account/list	GET
	계좌정보변경	https://openapi.openbanking.or.kr/v2.0/account/update_info	POST
	계좌해지	https://openapi.openbanking.or.kr/v2.0/account/cancel	POST
	사용자탈퇴	https://openapi.openbanking.or.kr/v2.0/user/close	POST
조회서비스 (사용자)	잔액조회	https://openapi.openbanking.or.kr/v2.0/account/balance/fin_num	GET
	거래내역조회	https://openapi.openbanking.or.kr/v2.0/account/transaction_list/fin_num	GET
조회서비스 (이용기관)	계좌실명조회	https://openapi.openbanking.or.kr/v2.0/inquiry/real_name	POST
	송금인정보조회	https://openapi.openbanking.or.kr/v2.0/inquiry/remittance_list	POST
	수취조회	https://openapi.openbanking.or.kr/v2.0/inquiry/receive	POST
이체서비스	출금이체	https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/fin_num	POST
	입금이체	https://openapi.openbanking.or.kr/v2.0/transfer/deposit/fin_num	POST
		https://openapi.openbanking.or.kr/v2.0/transfer/deposit/acnt_num	POST
	이체결과조회	https://openapi.openbanking.or.kr/v2.0/transfer/result	POST
	자금반환청구	https://openapi.openbanking.or.kr/v2.0/return/claim	POST
관리 API	자금반환결과 조회	https://openapi.openbanking.or.kr/v2.0/return/result	POST
	참가은행상태 조회	https://openapi.openbanking.or.kr/v2.0/bank/status	GET
	수수료조회	https://openapi.openbanking.or.kr/v2.0/manage/fee	GET
	집계조회	https://openapi.openbanking.or.kr/v2.0/manage/count	GET
	출금이체한도 조회	https://openapi.openbanking.or.kr/v2.0/manage/wd_limit	GET

■ 자체인증 이용기관 API URI 맵

기능구분	API 명	URI	HTTP Method
사용자인증 (OAuth 2.0)	토큰발급	https://openapi.openbanking.or.kr/oauth/2.0/token	POST
사용자/계좌 관리	사용자정보조회	https://openapi.openbanking.or.kr/v2.0/user/me	GET
	등록계좌조회	https://openapi.openbanking.or.kr/v2.0/account/list	GET
	계좌해지	https://openapi.openbanking.or.kr/v2.0/account/cancel	POST
	사용자탈퇴	https://openapi.openbanking.or.kr/v2.0/user/close	POST
	사용자계좌등록	https://openapi.openbanking.or.kr/v2.0/user/register	POST
조회서비스 (사용자)	잔액조회	https://openapi.openbanking.or.kr/v2.0/account/balance/fin_num	GET
		https://openapi.openbanking.or.kr/v2.0/account/balance/acnt_num	POST
	거래내역조회	https://openapi.openbanking.or.kr/v2.0/account/transaction_list/fin_num	GET
		https://openapi.openbanking.or.kr/v2.0/account/transaction_list/acnt_num	POST
조회서비스 (이용기관)	계좌실명조회	https://openapi.openbanking.or.kr/v2.0/inquiry/real_name	POST
	송금인정보조회	https://openapi.openbanking.or.kr/v2.0/inquiry/remit_list	POST
	수취조회	https://openapi.openbanking.or.kr/v2.0/inquiry/receive	POST
이체서비스	출금이체	https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/fin_num	POST
		https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/acnt_num	POST
	입금이체	https://openapi.openbanking.or.kr/v2.0/transfer/deposit/fin_num	POST
		https://openapi.openbanking.or.kr/v2.0/transfer/deposit/acnt_num	POST
	이체결과조회	https://openapi.openbanking.or.kr/v2.0/transfer/result	POST
	자금반환청구	https://openapi.openbanking.or.kr/v2.0/return/claim	POST
	자금반환결과 조회	https://openapi.openbanking.or.kr/v2.0/return/result	POST
관리 API	참가은행상태 조회	https://openapi.openbanking.or.kr/v2.0/bank/status	GET
	수수료조회	https://openapi.openbanking.or.kr/v2.0/manage/fee	GET
	집계조회	https://openapi.openbanking.or.kr/v2.0/manage/count	GET
	출금이체한도 조회	https://openapi.openbanking.or.kr/v2.0/manage/wd_limit	GET
	이상금융거래 탐지내역조회	https://openapi.openbanking.or.kr/v2.0/inquiry/fds_detect	GET

[안내사항]

1) 인터넷 회선 기반 API 이용시

- 인터넷 회선 기반 API 호출 시에는 https://openapi.openbanking.or.kr 로 호출하셔야 합니다.
- 테스트 API 호출 시에는 https://testapi.openbanking.or.kr 로 호출하셔야 합니다.

* 2019 년 12 월 이전에 서비스를 시작한 이용기관은 open-platform.or.kr 로 호출하셔야 합니다.
(운영: https://openapi.open-platform.or.kr, 테스트: https://testapi.open-platform.or.kr)

- 2) 전용 회선 기반 API 이용시
 - 전용 회선 기반 API 호출 시에는 http://<전용 회선 네트워크 IP 주소>로 호출하셔야 합니다.
 - 테스트 API 호출 시에는 http://<테스트 전용 회선 네트워크 IP 주소>로 호출하셔야 합니다.
 - IP 주소는 별도 안내
- 3) 운영 API 는 최종 계약체결 이후에만 호출 가능합니다.
- 4) 운영 API 의 경우, API 에 따라 과금될 수 있으니 유의하시기 바랍니다.
- 5) **[deprecated]**로 표시된 API 는 지원이 종료되는 API 입니다.

1.3. OAuth 이용안내

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

오픈뱅킹은 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. 응답코드 처리

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 오픈뱅킹 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 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 오픈뱅킹 서비스에 접근하기 위한 인증 기능을 제공합니다. 사용자는 오픈뱅킹(Service Provider)의 인증 페이지에서 **본인인증 및 조회/출금동의**를 통하여 인증하고, 이용기관은 사용자에게 대한 Access Token을 얻음으로써 사용자의 인증을 획득합니다.

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

□ 명 세

2.1.1. 사용자인증 API (3-legged)

■ 설 명

별도의 회원가입 없이 한번의 OAuth 인증으로 본인인증 및 계좌등록까지 처리할 수 있는 사용자 인증 API이며, 상황에 따른 **2가지** 호출 scheme을 제공합니다.

- (0)최초인증 : 처음 오픈뱅킹에 접속하는 사용자의 인증 처리 (본인인증+**조회/출금동의**)
- (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+**조회/출금동의**)
ARS로 조회/출금동의 시 휴대폰번호는 DB에 저장된 값을 활용
(전자금융업자 등 특정 자격요건을 갖춘 사업자에게만 선별적으로 허용)

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

[예외 처리 안내]

사용자인증 타입	예외 상황	처리 방법
(1)재인증 요청	Kftc-Bfop-UserSeqNo, Kftc-Bfop-UserCi값을 셋팅하지 않은 경우	(0)최초인증으로 처리되며 입력란에 자동으로 정보 셋팅
(1)재인증 요청	유효한 사용자가 없는 경우	(0)최초인증으로 처리되며 입력란에 자동으로 정보 셋팅
(2)인증생략 요청	Kftc-Bfop-UserSeqNo, Kftc-Bfop-UserCi값을 셋팅하지 않은 경우	에러 처리
(2)인증생략 요청	유효한 사용자가 없는 경우	에러 처리
지정 안함	-	(0)최초인증으로 처리되며 HTTP Header 처리(parsing) 안함

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/oauth/2.0/authorize
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	설명
Parameter [공통] (0)최초인증 (1)재인증 (2)인증생략	response_type	고정값: code	Y	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	Y	사용자인증이 성공하면 이용기관으로 연결되는 URL
	scope	[login inquiry transfer]	Y	Access Token 권한 범위 (다중 scope 가능) [17p] 「2.1.4. 권한범위 및 토큰갱신 안내」 참조
	client_info	<client_info> (Max: 256 bytes)	N	이용기관이 셋팅하는 임의의 정보 (Callback 호출 시 그대로 전달됨)
	state	<state> (32 bytes 고정)	Y	CSRF 보안위협에 대응하기 위해 이용기관이 셋팅하는 난수값 (Callback 호출 시 그대로 전달됨)
	auth_type	[0 2]	Y	사용자인증타입 구분 (0:최초인증, 1:재인증, 2:인증생략)
	lang	"kor", "eng" 등	N	다국어 설정 [22p] 「2.1.7. 다국어 설정 안내」 참조
	edit_option	"on" or "off"	N	입력란의 수정가능 여부(on:수정가능) (미지정 시 기본값: "on")
	bg_color	#000000	N	배경색 지정 (HEX 색상코드)
	txt_color	#000000	N	글자색 지정 (HEX 색상코드)
	btn1_color	#000000	N	확인버튼색 지정 (HEX 색상코드)
	btn2_color	#000000	N	취소버튼색 지정 (HEX 색상코드)
	account_hold_auth_yn	"Y" or "N"	N	계좌소유인증 사용여부**
Headers (1)재인증	Kftc-Bfop-UserSeqNo	<user_seq_no>	N	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci>	N	사용자 CI(Connect Info)
	Kftc-Bfop-UserName	<user_name>	N	사용자명
	Kftc-Bfop-UserInfo	"198001012"	N	생년월일(8 자리)+성별(1 자리) (주민등록번호 상의 숫자 입력)
	Kftc-Bfop-UserCellNo	<user_cell_no>	N	휴대폰번호
	Kftc-Bfop-PhoneCarrier	"skt", "ktf" 등	N	이동통신사 [22p] 「이동통신사 코드표」 참조
	Kftc-Bfop-UserEmail	<user_email>	N	이메일주소
	Kftc-Bfop-BankCodeStd	<user_bank_code_std>	N	계좌번호 지정등록 시 셋팅
	Kftc-Bfop-AccountNum	<user_account_num>	N	
Headers (2)인증생략	Kftc-Bfop-UserSeqNo	<user_seq_no>	Y	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci>	Y	사용자 CI(Connect Info)
	Kftc-Bfop-AccessToken	<access_token>	Y	"login" scope 을 포함한 토큰 (유효한 접속토큰이어야 함)

※ 오픈뱅킹 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.

* auth_type=1(재인증)으로 지정하여 요청할 경우 auth_type=0(최초인증)으로 처리합니다.

** 계좌소유인증은 SMS 본인인증, ARS 조회/출금동의 시 추가 인증방식으로 사용할 수 있습니다.

■ 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 client_info 값을 그대로 전달
state*	<state>	요청 시 이용기관이 셋팅한 state 값을 그대로 전달

* 이용기관은 CSRF 보안위협에 대응하기 위해 요청 시의 state 값과 응답 시의 state 값을 비교해야 함

■ 화면 예시

<p>① 본인인증</p> 	<p>② 계좌등록</p> 	<p>③ ARS 요청</p> 
<p>④ ARS 동의</p> 	<p>⑤ 이메일 등록</p> 	<p>⑥ 이용기관 앱 복귀</p> <p>(Callback URL 호출)</p>

■ 사용자 출금동의 만료 및 금융정보 제3자제공동의 만료 처리

사용자는 사용자인증 API를 통하여 출금동의 및 금융정보 제3자제공동의를 하게 되며, 이용기관은 토큰 재발급과는 별도로 사용자인증 후 매 1년 마다 계좌등록확인 API를 이용하여 사용자의 재동의를 받아야 합니다. 사용자가 출금동의 및 제3자정보제공동의를 한 시점부터 1년이 경과 후 API를 요청하면 조회거래의 경우 “금융(거래)정보 제3자제공동의 만료(A0316)”, 출금이체의 경우 “출금동의 만료(A0319)” 응답코드가 발생합니다.

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.openbanking.or.kr/oauth/2.0/token
HTTP Method	POST
Content-Type	application/x-www-form-urlencoded; charset=UTF-8

■ 요청 메시지 명세

항목	값	필수	설명
code	<authorization_code>	Y	사용자인증 성공 후 획득한 Authorization Code
client_id	<client_id> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client Secret
redirect_uri	<callback_uri>	Y	Access Token 을 전달받을 Callback URL (Authorization Code 획득 시 요청했던 Callback URL)
grant_type	고정값: authorization_code	Y	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> (Max: 400 bytes)		오픈뱅킹에서 발행된 Access Token
	token_type	고정값: Bearer		Access Token 유형
	expires_in	<expire_time>		Access Token 만료 기간(초)
	refresh_token	<refresh_token> (Max: 400 bytes)		Access Token 갱신 시 사용하는 Refresh Token
	scope	[login inquiry transfer]		Access Token 권한 범위 (사용자인증 시 요청했던 권한 범위와 동일)
	user_seq_no	"1000000106"	AN(10)	사용자일련번호

■ 응답 메시지 형태

JSON
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiIiMDAwMDAwMTA2Iiwic2NvcGUiOiIiG9naW4iLCJpbnF1aXJ5IiwidHJhbnNmZXIiOiIiXSwiaXNzIjoiaHR0cHM6Ly93d3cub3BlbmJhbmtpbmcub3lua3IiLCJleHAiOiIiNTc2OTgxNDkzIiwianRpIjoia2M5NTVjYzAtM2I4ZS00ZTJkLTg0YWYtZTcyNWl5YzFNTAwln0.DONr1R0Ae2fe4qN56VZkEGB364vo951rUToHkDTxvJ8", "token_type": "Bearer", "expires_in": 7776000, "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiIiMDAwMDAwMTA2Iiwic2NvcGUiOiIiG9naW4iLCJpbnF1aXJ5IiwidHJhbnNmZXIiOiIiXSwiaXNzIjoiaHR0cHM6Ly93d3cub3BlbmJhbmtpbmcub3lua3IiLCJleHAiOiIiNTc2OTgxNDkzIiwianRpIjoia2M5NTVjYzAtM2I4ZS00ZTJkLTg0YWYtZTcyNWl5YzFNTAwln0.DONr1R0Ae2fe4qN56VZkEGB364vo951rUToHkDTxvJ8", "scope": "login inquiry transfer", "user_seq_no": "1000000106" }</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.openbanking.or.kr/oauth/2.0/token
HTTP Method	POST
Content-Type	application/x-www-form-urlencoded; charset=UTF-8

■ 요청 메시지 명세

항목	값	필수	설명
client_id	<client_id> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client Secret
scope	고정값: oob	Y	Access Token 권한 범위
grant_type	고정값: client_credentials	Y	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> (Max: 400 bytes)		오픈뱅킹에서 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	"F001234560"		이용기관코드

■ 응답 메시지 형태

JSON
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJGMDAxMjM0NTYwIiwic2NvcGUiOiJib29iIl0sImZcyI6Imh0dHBzOi8vd3d3Lm9wZW5iYW5raW5nLm9yLmtyliwiZShwLjoiMTU3Njk3OTEzMjM0NTYwIiwiaWF0Ij0iMTU3Njk3OTEzMjM0NTYwIiwiaXNjaXNWEzMTc5LTFEwMjYtNzZjNC05YzExLWQxNTZlOTcxMmRiZiJ9.miMnaiKpzXHTJr3JPIUd2amPpeXiqOM8XnRyX6c62Vs", "token_type": "Bearer", "expires_in": 7776000, "scope": "oob", "client_use_code": "F001234560" }</pre>

□ 자체인증 이용기관 토큰발급 API (2-legged)

■ 설 명

자체인증 이용기관 2-legged 인증의 경우에는 scope=sa로 정의되며, 사용자인증 API의 호출 없이 바로 /token API를 호출하여 토큰을 획득한다.

■ 요청 메시지 URL

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

■ 요청 메시지 명세

항목	값	필수	설명
client_id	<client_id> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client Secret
scope	고정값: sa	Y	Access Token 권한 범위
grant_type	고정값: client_credentials	Y	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> (Max: 400 bytes)		오픈뱅킹에서 발행된 Access Token
	token_type	고정값: Bearer		Access Token 유형
	expires_in	<expire_time>		Access Token 만료 기간(초)
	scope	고정값: sa		Access Token 권한 범위
	client_use_code	"B001234560"	AN(10)	이용기관코드

■ 응답 메시지 형태

JSON
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJxMjM0NTY3ODkwIiwic2NvcGUiOiJlc2EiXSwiaXNzIjoiaHR0cHM6Ly93d3cub3BlbmJhbmtpbmMub3lua3liLCJleHAiOiIxNTc2OTc5MTMzIiwianRpIjoiaWYyZjZjMxODMtNjc5Ni00OGM0LTljMjgtZDI0OOWM5OTExYmRlIn0.a3c0fp_rX-foiyeyHSoS_rS_8y-XA48sRpJHpCnLqEc", "token_type": "Bearer", "expires_in": 7776000, "scope": "sa", "client_use_code": "B001234560" }</pre>

□ 사용자 토큰갱신 API (Refresh Token 사용)

■ 설 명

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

■ 요청 메시지 URL

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

■ 요청 메시지 명세

항목	값	필수	설명
client_id	<client_id> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client ID
client_secret	<client_secret> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client Secret
refresh_token	<refresh_token> (Max: 400 bytes)	Y	토큰발급 시 받은 Access Token 갱신용 토큰

scope	[login inquiry transfer]	Y	Access Token 권한 범위
grant_type	고정값: refresh_token	Y	토큰 갱신을 위한 권한부여 방식 지정

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	access_token	<access_token> (Max: 400 bytes)		오픈뱅킹에서 발행된 Access Token
	token_type	고정값: Bearer		Access Token 유형
	expires_in	<expire_time>		Access Token 만료 기간(초)
	refresh_token	<refresh_token> (Max: 400 bytes)		Access Token 갱신 시 사용하는 Refresh Token
	scope	[login inquiry transfer]		Access Token 권한 범위
	user_seq_no	"1000000106"	AN(10)	사용자일련번호

■ 응답 메시지 형태

JSON
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiIwMDAwMDAwMTA2Iiwic2NvcGUiOiI0sibG9naW4iLCJpbnF1aXJ5IiwidHJhbnNmZXliXSwiaXNzIjoiaHR0cHM6Ly93d3cub3BlbmJhbmtpbmcub3lua3liLCJleHAiOiIxNTc2OTgxNDkzIiwianRpIjoia2M5NTVjYzAtM2I4ZS00ZTJkLTg0YWYtZCYyNW15YzFNTAwln0.DONr1R0Ae2fe4qN56VZkEGB364vo951rUToHkDTxvJ8", "token_type": "Bearer", "expires_in": 7776000, "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiIwMDAwMDAwMTA2Iiwic2NvcGUiOiI0sibG9naW4iLCJpbnF1aXJ5IiwidHJhbnNmZXliXSwiaXNzIjoiaHR0cHM6Ly93d3cub3BlbmJhbmtpbmcub3lua3liLCJleHAiOiIxNTc2OTQ1NDkzIiwianRpIjoia2M5NTVjYzAtM2I4ZS00ZTJkLTg0YWYtZCYyNW15YzFNTAwln0.DONr1R0Ae2fe4qN56VZkEGB364vo951rUToHkDTxvJ8", "scope": "login inquiry transfer", "user_seq_no": "1000000106" }</pre>

[토큰 형태 안내]

- 오픈뱅킹에서는 Token으로 JWS*(JSON Web Signature)를 사용합니다.
 - * 「JSON Web Signature」(RFC 7515) 참조
- 형식 : Base64Encode(Header) + "." + Base64Encode(Payload) + "." + Signature
- Header값 예시


```
{
  "alg": "HS256",
  "typ": "JWT"
}
```
- Payload값 예시


```
{
  "aud": "1000000106",
```

토큰발급 시 에러가 발생할 경우 아래와 같이 JSON 형태로 응답코드와 응답메시지를 전송합니다.

Header		status: 200		
Content-Type		application/json		

HTTP	항목	값	TYPE (길이)	설명
Body	rsp_code	"O0001"	AN(5)	응답코드(API)
	rsp_message	"인증요청 거부-인증 파라미터 오류 ([3000201])"	AH(300)	응답메시지(API)

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

응답코드는 아래의 별첨자료를 참고하시기 바랍니다.

<101p> 별첨5) API 인증요청 거부(O0001)에 대한 세부 응답코드(error code)

2.1.3. 계좌등록확인 API (3-legged)

■ 설 명

사용자인증을 통하여 이미 등록된 계좌에 대해서 재확인 및 출금 재동의/금융정보 제3자제공 재동의를 하고 접속토큰을 재발급 받는 API이며, 상황에 따라 **2가지** 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)

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/oauth/2.0/authorize_account
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	설명
Parameter [공통] (0)최초인증 (1)재인증 (2)인증생략	response_type	고정값: code	Y	OAuth 2.0 인증 요청 시 반환되는 형태
	client_id	<client_id> (Max: 40 bytes)	Y	오픈뱅킹에서 발급한 이용기관 앱의 Client ID
	redirect_uri	<callback_uri>	Y	사용자인증이 성공하면 이용기관으로 연결되는 URL
	scope	[login inquiry transfer]	Y	Access Token 권한 범위 (다중 scope 가능) [17p] 「2.1.4. 권한범위 및 토큰갱신 안내」 참조
	client_info	<client_info> (Max: 256 bytes)	N	이용기관이 셋팅하는 임의의 정보 (Callback 호출 시 그대로 전달됨)
	state	<state> (32 bytes 고정)	Y	CSRF 보안위협에 대응하기 위해 이용기관이 셋팅하는 난수값 (Callback 호출 시 그대로 전달됨)
	auth_type	[0 2]	Y	사용자인증타입 구분* (0:최초인증, 1:재인증, 2:인증생략)
	lang	"kor", "eng" 등	N	다국어 설정 [22p] 「2.1.7. 다국어 설정 안내」 참조
	edit_option	"on" or "off"	N	입력란의 수정가능 여부(on:수정가능) (미지정 시 기본값: "on")
	bg_color	#000000	N	배경색 지정 (HEX 색상코드)
	txt_color	#000000	N	글자색 지정 (HEX 색상코드)
	btn1_color	#000000	N	확인버튼색 지정 (HEX 색상코드)
	btn2_color	#000000	N	취소버튼색 지정 (HEX 색상코드)
Headers (1)재인증	Kftc-Bfop-UserSeqNo	<user_seq_no>	N	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci>	N	사용자 CI(Connect-Info)
	Kftc-Bfop-UserName	<user_name>	N	사용자명
	Kftc-Bfop-UserInfo	"198001012"	N	생년월일(8 자리)+성별(1 자리) (주민등록번호 상의 숫자 입력)
	Kftc-Bfop-UserCellNo	<user_cell_no>	N	휴대폰번호
	Kftc-Bfop-PhoneCarrier	"skt", "kft" 등	N	이동통신사

	Kftc-Bfop-UserEmail	<user_email>	N	[22p] 「이동통신사 코드표」 참조 이메일주소
	Kftc-Bfop-BankCodeStd	<user_bank_code_std>	N	계좌번호 지정등록 시 셋팅
	Kftc-Bfop-AccountNum	<user_account_num>	N	
Headers (2)인증생략	Kftc-Bfop-UserSeqNo	<user_seq_no>	Y	기존 고객의 사용자일련번호
	Kftc-Bfop-UserCI	<user_ci>	Y	사용자 CI(Connect Info)
	Kftc-Bfop-AccessToken	<access_token>	Y	"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.

* auth_type=1(재인증)으로 지정하여 요청할 경우 auth_type=0(최초인증)으로 처리합니다.

■ 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
state*	<state>	요청 시 이용기관이 셋팅한 state 값을 그대로 전달

* 이용기관은 CSRF 보안위협에 대응하기 위해 요청 시의 state 값과 응답 시의 state 값을 비교해야 함

■ 화면 예시

① 본인인증	② 계좌내역확인	③ 이용기관 앱 복귀
		<p>(Callback URL 호출)</p>

2.1.4. 권한범위 및 토큰갱신 안내

■ 권한 범위(scope)

API 구분		권한범위* (scope)	유효기간** (expires_in)	Refresh 지원여부
사용자인증*** (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
		sa****	90 일	X

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

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

*** 본인인증, 로그인연결, 계좌등록을 하나의 흐름으로 처리하게 되어있음

**** sa(self-authentication) scope의 경우에는 자체인증이 가능한 이용기관만 발급 가능

[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)
사용자인증 (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	
		sa	90 일	Refresh 불가	

[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 (자체인증 미사용 이용기관)

기능구분	API 명	API URI	호출 Method	요청토큰 Scope*
사용자인증 (OAuth 2.0)	사용자인증	/oauth/2.0/authorize	GET	해당없음
	토큰발급	/oauth/2.0/token	POST	해당없음
	계좌등록확인	/oauth/2.0/authorize_account	GET	해당없음
사용자/계좌 관리	사용자정보조회	/v2.0/user/me	GET	login
	사용자로그인연결해제	/v2.0/user/unlink	POST	login
	등록계좌조회	/v2.0/account/list	GET	login
	계좌정보변경	/v2.0/account/update_info	POST	login
	계좌해지	/v2.0/account/cancel	POST	login
	사용자탈퇴	/v2.0/user/close	POST	login
조회서비스 (사용자)	잔액조회	/v2.0/account/balance/fin_num	GET	inquiry
	거래내역조회	/v2.0/account/transaction_list/fin_num	GET	inquiry
조회서비스 (이용기관)	계좌실명조회	/v2.0/inquiry/real_name	POST	oob
	송금인정보조회	/v2.0/inquiry/remit_list	POST	oob
	수취조회	/v2.0/inquiry/receive	POST	oob
출금서비스	출금이체	/v2.0/transfer/withdraw/fin_num	POST	transfer
	입금이체	/v2.0/transfer/deposit/fin_num /v2.0/transfer/deposit/acnt_num	POST	oob
	이체결과조회	/v2.0/transfer/result	POST	oob
	자금반환청구	/v2.0/return/claim	POST	oob
	자금반환결과조회	/v2.0/return/result	POST	oob
관리 API	참가기관상태조회	/v2.0/bank/status	GET	oob
	수수료조회	/v2.0/manage/fee	GET	oob
	집계조회	/v2.0/manage/count	GET	oob
	출금이체한도조회	/v2.0/manage/wd_limit	GET	oob

* 요청토큰 Scope

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

■ API별 호출 Method, 요청토큰의 필요 Scope (자체인증 사용 이용기관)

기능구분	API 명	API URI	호출 Method	요청토큰 Scope
사용자인증 (OAuth 2.0)	토큰발급	/oauth/2.0/token	POST	해당없음
사용자/계좌 관리	사용자정보조회	/v2.0/user/me	GET	sa
	등록계좌조회	/v2.0/account/list	GET	
	계좌해지	/v2.0/account/cancel	POST	
	사용자탈퇴	/v2.0/user/close	POST	
	사용자계좌등록	/v2.0/user/register	POST	
조회서비스 (사용자)	잔액조회	/v2.0/account/balance/fin_num /v2.0/account/balance/acnt_num	GET POST	
	거래내역조회	/v2.0/account/transaction_list/fin_num /v2.0/account/transaction_list/acnt_num	GET POST	
조회서비스 (이용기관)	계좌실명조회	/v2.0/inquiry/real_name	POST	
	송금인정보조회	/v2.0/inquiry/remit_list	POST	
	수취조회	/v2.0/inquiry/receive	POST	
출금서비스	출금이체	/v2.0/transfer/withdraw/fin_num /v2.0/transfer/withdraw/acnt_num	POST	
	입금이체	/v2.0/transfer/deposit/fin_num /v2.0/transfer/deposit/acnt_num	POST	
	이체결과조회	/v2.0/transfer/result	POST	
	자금반환청구	/v2.0/return/claim	POST	
	자금반환결과조회	/v2.0/return/result	POST	
관리 API	참가기관상태조회	/v2.0/bank/status	GET	
	수수료조회	/v2.0/manage/fee	GET	
	집계조회	/v2.0/manage/count	GET	
	출금이체한도조회	/v2.0/manage/wd_limit	GET	
	이상금융거래탐지내역조회	/v2.0/inquiry/fds_detect	GET	

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 참고하시기 바랍니다.

<101p> 별첨4) OAuth 응답코드

<101p> 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
http://<callback_uri>?error=access_denied& error_description =사용자가 '취소' 버튼을 클릭한 경우& client_info=[test] whatever you want

에러코드 및 에러메시지는 아래의 별첨자료를 참고합니다.

<102p> 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

오픈뱅킹에서는 사용자가 조회/출금 동의하고자 하는 계좌를 참가은행과 전문처리를 통하여 계좌 등록 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 오픈뱅킹 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
		□□□□□□-■■-□□□□□□-□	51,52,56,55,66,67	
		■■■-□□□□-□□□□-□□-□	792	
기업은행 평생계좌	기업은행 평생계좌는 휴대폰번호를 이용한 가상매핑계좌로서 계좌등록이 불가함	기업은행(은행코드:003) + 11 자리 + "010"으로 시작		

2.1.7. 다국어 설정 안내

■ 설 명

오픈뱅킹의 프로바이더 페이지는 다국어를 지원합니다. 사용자인증(/authorize) 및 계좌등록확인(/authorize_account) API 호출 시 언어 파라미터(**lang=[언어코드]**)를 설정하면, 페이지(**SMS본인인증 페이지 및 공인인증서 인증페이지 제외**) 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

* ISO 3166-1 표준 코드

[다국어 호출 예]

GET Method
https://testapi.openbanking.or.kr/oauth/2.0/authorize?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

오픈뱅킹에 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. 이용기관은 사용자정보조회 API를 호출하여 사용자의 최신 고객정보 및 계좌목록을 가져올 수 있습니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/user/me
HTTP Method	GET

■ 요청 메시지 명세

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

■ 응답 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	user_seq_no	"U123456789"	AN(10)	사용자일련번호
	user_ci	"Dqz4/7Rp.....UjVj34XFJTV=="	B64(100)	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"	E(100)	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	"20190910101921"	N(14)	조회서비스 동의일시
--transfer_agree_yn	"Y"	A(1)	출금서비스 동의여부
--transfer_agree_dtime	"20190910101921"	N(14)	출금서비스 동의일시
--payer_num	"01234567890123456789"	AN(30)	납부자번호

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

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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": "20190910101921", "transfer_agree_yn": "Y", "transfer_agree_dtime": "20190910101921", "payer_num": "01234567890123456789" }, { ... }, ...] }</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.openbanking.or.kr/v2.0/user/unlink
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

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

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

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	user_seq_no	"U123456789"	AN(10)	사용자일련번호

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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.openbanking.or.kr/v2.0/account/list
HTTP Method	GET

■ 요청 메시지 명세

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

* 정렬 기준 : 조회서비스 동의일시, 출금서비스 동의일시 중 최근 값

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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"	AN(3)	출금(개설)기관.대표코드
	--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	"20190910101921"	N(14)	조회서비스 동의일시
--transfer_agree_yn	"Y"	A(1)	출금서비스 동의여부
--transfer_agree_dtime	"20190910101921"	N(14)	출금서비스 동의일시
--account_state	"01"	AN(2)	계좌상태 01:사용, 09:해지

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

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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": "20190910101921", "transfer_agree_yn": "Y", "transfer_agree_dtime": "20190910101921", "account_state": "01", }, { ... }, ... { ... }] }</pre>

2.2.4. 계좌정보변경 API

■ 설 명

등록된 계좌의 별명을 변경합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/account/update_info
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

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

■ 응답 메시지 명세

HTTP	항목	값(예제포함)	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	account_alias	"보험료납부계좌"	AH(50)	변경된 계좌별명

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "fintech_use_num": "123456789012345678901234", "account_alias": "보험료납부계좌", }</pre>

2.2.5. 계좌해지 API

■ 설 명

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

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/account/cancel
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login, sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	scope	[inquiry transfer]	Y		서비스구분 inquiry : 조회서비스 transfer : 출금서비스
	fintech_use_num	"123456789012345678901234"	Y	AN(24)	핀테크이용번호

주1) [30p] [은행거래고유번호 생성 안내] 참조

[은행거래고유번호 생성 안내]

- 은행거래고유번호는 이용기관에서 생성하여 셋팅하는 값입니다.
- 하루동안 유일성을 보장해야 합니다. (날짜가 다를 경우에는 중복발생이 가능)
- "하루 동안의 유일성 보장" 이외의 그 어떤 규칙(순차증가 등)도 보장하지 않습니다.
- 형식 : 이용기관코드(10자리) + 생성주체구분코드("U")* + 이용기관 부여번호(9자리)
- * U: 이용기관에서 생성한 은행거래고유번호, O: 오픈뱅킹에서 생성한 은행거래고유번호
- 예) F123456789U4BC34239Z
- 생성한 은행거래고유번호는 오픈뱅킹과 참가은행과의 송수신 시 거래고유번호로 사용되며, 응답 메시지의 '거래고유번호(참가은행)'의 값과 동일합니다.
- 하루 내에 중복 거래고유번호가 다시 요청되거나, 형식에 맞지 않을 경우 에러 처리

※ 조회서비스와 출금서비스는 분리된 API로 처리해야 합니다. 예를 들어 특정 고객이 조회서비스와 출금서비스를 모두 해지하는 경우, 이용기관은 '조회서비스' 계좌해지 API와 '출금서비스' 계좌해지 API를 각각 호출해야 합니다.

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(100)	응답메시지(참가은행)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "" }</pre>

※ 해지 API 응답 시 응답코드(참가은행)가 "기 해지 사용자(551)", "해당 사용자 없음(555)"인 경우, 해지 완료된 것으로 처리합니다.

2.2.6. 사용자탈퇴 API

■ 설 명

이용기관에 등록된 모든 계좌를 해지(등록해지 및 동의해지)하고 오픈뱅킹에 등록된 사용자를 탈퇴합니다. 계좌해지는 참가은행에 등록된 내역까지 해지합니다. 이용기관은 장기 미사용 등을 이유로 고객정보 파기 시에는 사용자를 탈퇴해야 합니다. 사용자탈퇴 API를 요청하여 계좌를 해지하는 경우, 은행은 사용자탈퇴 API 요청을 받은 날의 익 영업일 중에 계좌 해지처리를 합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/user/close
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = login, sa]
Body	client_use_code	"F001234560"	Y	AN(10)	이용기관코드
	user_seq_no	"U123456789"	Y	AN(10)	사용자일련번호

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "" }</pre>

2.2.7. 사용자계좌등록 API (자체인증 이용기관만 이용가능)

■ 설 명

자체인증 이용기관이 사용자의 계좌를 등록합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/user/register
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	bank_code_std	"097"	Y	AN(3)	등록계좌개설기관.표준코드
	register_account_num	"1001234567890123"	Y	AN(16)	등록계좌번호
	user_info	"19810101"	Y	N(8)	생년월일 (8 자리) (주민등록번호 등록 기준)
	user_name	"홍길동"	Y	AH(20)	사용자명
	user_ci	"Dqz4/p.....j34XFJTV=="	Y	B64(100)	CI(Connect Info)
	user_email	"testbed@kftc.or.kr"	N	E(100)	이메일주소 ^{주 2)}
	scope	[inquiry transfer]	Y		서비스구분 inquiry : 조회서비스 transfer : 출금서비스
	info_prvd_agmt_yn	"Y"	N	A(1)	제 3 자정보제공동의여부 ^{주 2)}
	wd_agmt_yn	"Y"	N	A(1)	출금동의여부 ^{주 2)}
	agmt_data_type	"1"	N	N(1)	동의자료 구분 ^{주 2)} 1:서면, 2:공인인증, 3: 일반인증 4:녹취, 5:ARS, 6:기타

주1) [30p] [은행거래고유번호 생성 안내] 참조

주2) 서비스구분에 따른 동의여부 및 동의자료 구분 셋팅 안내

필드명 등록 구분	서비스구분	제 3 자정보제공 동의여부	출금동의여부	동의자료구분	이메일주소
조회서비스 사용자 등록 시	inquiry	"Y"	미설정	미설정	설정*
출금서비스 사용자 등록 시	transfer	미설정	"Y"	설정**	미설정

* 금융실명법 제4조의2제1항에 따라 금융거래정보 등의 제공사실 통보를 위한 정보 수집 목적으로 조회서비스 사용자 등록 시에만 설정한다.

** 출금이체 동의자료 제출 면제기관인 경우, 동의자료 구분을 설정하지 않습니다.

※ 조회서비스와 출금서비스는 분리된 API로 처리해야 합니다. 예를 들어 특정 고객이 조회서비스와 출금서비스를 이용하는 경우, 이용기관은 '조회서비스' 사용자계좌등록 API와 '출금서비스' 사용자계좌등록 API를 각각 호출해야 합니다.

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(100)	응답메시지(참가은행)
	user_seq_no	"U123456789"	AN(10)	사용자일련번호
	fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
	payer_num	"16000101100000002570"	AN(30)	납부자번호
	transfer_bank_tran_id	"F123456789U4BC34239Z"	AN(20)	출금등록거래고유번호(참가은행)*
	transfer_bank_tran_date	"20190910"	N(8)	출금등록거래일자(참가은행)*

* 출금서비스 정상 등록 시 사용된 거래고유번호 및 거래일자로 출금서비스 사용자 등록 시에만 셋팅하여 응답합니다.

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "user_seq_no": "U123456789", "fintech_use_num": "123456789012345678901234", "payer_num": "16000101100000002570", "transfer_bank_tran_id": "F123456789U4BC34239Z", "transfer_bank_tran_date": "20190910" }</pre>

■ 에러 처리

사용자계좌등록 API를 호출 후 응답을 받지 못했을 경우에는 API를 재호출하여 사용자계좌등록 상태를 확인할 수 있습니다. 오픈뱅킹에 기등록이 되어 있는 계좌라면 "응답코드(API)"가 "기등록된 조회서비스용 사용자 계좌(A0324)", "기등록된 출금서비스용 사용자 계좌(A0325)"로 응답하며 사용자일련번호, 핀테크이용번호 및 납부자번호를 셋팅하여 응답하며, 출금서비스 사용자 등록 시에는 추가적으로 출금등록거래고유번호 및 출금등록거래일자를 셋팅하여 응답합니다.

■ 사용자탈퇴 API 요청 후 사용자계좌등록 관련 처리

사용자탈퇴 API를 요청하여 계좌를 해지하는 경우, 은행은 사용자탈퇴 API 요청을 받은 날의 익영업일 중에 계좌 해지처리를 합니다. 은행에서 계좌 해지처리하기 전에 동일한 계좌에 대해서 다시 계좌등록을 요청하게 되면 "사용자탈퇴 처리중인 계좌(A0019)"로 거부 응답하며, 은행에서 계좌 해지 처리 후에 정상적으로 계좌등록을 할 수 있습니다.

■ 자체인증 이용기관 자체 출금동의 구현을 위한 금융보안원 권고사항

권고사항은 아래의 별첨자료를 참고하시기 바랍니다.

<109p> 별첨9) 자체 출금동의 방식의 안전한 구현을 위한 금융보안원 권고사항

2.3. 조회서비스(사용자)

□ 기 능

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

□ 명 세

2.3.1. 잔액조회 API

사용자 계좌의 잔액을 조회합니다. 계좌종류 및 상품명이 함께 전달됩니다. 등록된 핀테크이용번호로 요청하는 경우(/fin_num)와 실 계좌번호로 요청하는 경우(/acct_num)의 두 가지 기능을 각각 제공합니다.

□ 잔액조회 API : 핀테크이용번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/account/balance/fin_num
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = inquiry, sa]
Parameter	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	fintech_use_num	"123456789012345678901234"	Y	AN(24)	핀테크이용번호
	tran_dtime	"20190910101921"	Y	N(14)	요청일시

주1) [30p] [은행거래고유번호 생성 안내] 참조

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	" "	AH(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"	AN(1)	계좌종류 1:수시입출금, 2:예적금 6:수익증권	
product_name	"내맘대로통장"	AH(40)	상품명	

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": " ", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "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>

□ 잔액조회 API : 계좌번호 사용 (자체인증 이용기관만 이용가능)

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/account/balance/acnt_num
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	bank_code_std	"097"	Y	AN(3)	개설기관.표준코드
	account_num	"1001234567890123"	Y	AN(16)	계좌번호
	user_seq_no	"1123456789"	Y	AN(10)	사용자일련번호
	tran_dtime	"20190910101921"	Y	N(14)	요청일시

주1) [30p] [은행거래고유번호 생성 안내] 참조

■ 요청 메시지 형태

JSON	
<pre>{ "bank_tran_id": "F123456789U4BC34239Z", "bank_code_std": "097", "account_num": "1001234567890123", "user_seq_no": "1123456789", "tran_dtime": "20190910101921" }</pre>	

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	"	AH(300)	응답메시지(API)
	bank_tran_id	"12345678901234567890"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(100)	응답메시지(참가은행)
account_num	"1001234567890123"	AN(16)	계좌번호
balance_amt	"1000000"	SN(13)	계좌잔액(-금액가능)
available_amt	"1000000"	N(12)	출금가능금액
account_type	"1"	AN(1)	계좌종류 1:수시입출금, 2:예적금 6:수익증권
product_name	"내맘대로통장"	AH(40)	상품명

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "account_num": "1001234567890123", "balance_amt": "1000000", "available_amt": "1000000", "account_type": "2", "product_name": "알뜰살뜰적금" }</pre>

2.3.2. 거래내역조회 API

사용자가 등록한 계좌의 거래내역을 조회합니다. 등록된 핀테크이용번호로 요청하는 경우(/fin_num)와 실
계좌번호로 요청하는 경우(/acct_num)의 두 가지 기능을 각각 제공합니다.

□ 거래내역조회 API : 핀테크이용번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/account/transaction_list/fin_num
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = inquiry, sa]
Parameter	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	fintech_use_num	"123456789012345678901234"	Y	AN(24)	핀테크이용번호
	inquiry_type	"A"	Y	A(1)	조회구분코드 A:All, I:입금, O:출금
	inquiry_base	"D"	Y	A(1)	조회기준코드 ^{주 2)} D:일자, T:시간
	from_date	"20160404"	Y	N(8)	조회시작일자
	from_time	"100000"	N	N(6)	조회시작시간 ^{주 2)}
	to_date	"20160405"	Y	N(8)	조회종료일자
	to_time	"110000"	N	N(6)	조회종료시간 ^{주 2)}
	sort_order	"D"	Y	A(1)	정렬순서 D:Descending A:Ascending
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	befor_inquiry_trace_info	"123"	N	AN(20)	직전조회추적정보 ^{주 3)}

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)

bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
bank_tran_date	"20190910"	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	""	AH(100)	응답메시지(참가은행)
fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호
balance_amt	"1000000"	SN(12)	계좌잔액(-금액가능)
page_record_cnt	"25"	N(2)	현재페이지 레코드건수 ^{주 4)}
next_page_yn	"Y"	A(1)	다음페이지 존재여부 ^{주 5)}
befor_inquiry_trace_info	"123"	AN(20)	직전조회추적정보
res_list	<object>		조회된 거래내역
--tran_date	"20190910"	N(8)	거래일자
--tran_time	"113000"	N(6)	거래시간
--inout_type	"입금"	AH(8)	입출금구분 ^{주 6)} (입금, 출금, 지급, 기타)
--tran_type	"현금"	AH(10)	거래구분 ^{주 7)}
--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) [30p] [은행거래고유번호 생성 안내] 참조

주2) <조회기준 처리 안내>

- 일자 기준으로 조회를 하는 경우 ("inquiry_base" : "D")
 - 요청시 조회시작일자, 조회종료일자를 셋팅하여 요청합니다.

- 일자기준 조회의 경우 은행은 조회 요청일자 기준으로 최소 6개월 범위에서 거래내역을 제공한다. (은행에 따라 더 넓은 범위의 조회결과 제공 가능)

예) 12월 20일에 조회를 요청하는 경우 6월 21일부터의 거래 내역은 항상 제공하나, 6월 20일을 포함한 이전 거래 내역 제공 여부는 은행별 상이

- 시간 기준으로 조회를 하는 경우 ("inquiry_base" : "T")
 - 요청시 조회시작일자, 조회종료일자, 조회시작시간, 조회종료시간을 셋팅하여 요청합니다.
 - 시간기준 조회의 경우 은행은 조회 요청일자 기준으로 최소 3일 이하의 범위에서 거래내역을 제공한다. (은행에 따라 더 넓은 범위의 조회결과 제공 가능)

예) 1월 4일에 조회를 요청하는 경우 최소 1월 2일 00:00:00부터의 거래 내역은 항상 제공하나, 1월 1일을 포함한 이전 거래 내역 제공 여부는 은행별 상이

주3) 다음 페이지 요청 시에 직전 조회의 응답에서 얻은 값을 그대로 셋팅

다음 페이지 요청이 아닌 경우에는 파라미터 자체를 설정하지 않음

주4) 한 페이지는 최대 25건 가능

주5) <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)

주6) 입출금구분이 '기타'인 경우에는 '거래금액'이 0원으로 셋팅됨

주7) 현금, 대체, 급여, 타행환, F/B출금 등 (참가은행에서 직접 셋팅)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "fintech_use_num": "123456789012345678901234", "balance_amt": "1000000", "page_record_cnt": "25", "next_page_yn": "Y", "befor_inquiry_trace_info": "1T201806171", "res_list": [{ "tran_date": "20190910", "tran_time": "113000", "inout_type": "입금", "tran_type": "현금", "printed_content": "통장인자내용", "tran_amt": "450000", "after_balance_amt": "-1000000", "branch_name": "분당점" }, { ... }, ... { ... }] }</pre>

□ 거래내역조회 API : 계좌번호 사용 (자체인증 이용기관만 이용가능)

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/account/transaction_list/acnt_num
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	bank_code_std	"097"	Y	AN(3)	개설기관.표준코드
	account_num	"0001234567890123"	Y	AN(16)	계좌번호
	user_seq_no	"1123456789"	Y	AN(10)	사용자일련번호
	inquiry_type	"A"	Y	A(1)	조회구분코드 A:All, I:입금, O:출금
	inquiry_base	"D"	Y	A(1)	조회기준코드 ^{주 2)} D:일자, T:시간
	from_date	"20160404"	Y	N(8)	조회시작일자
	from_time	"100000"	N	N(6)	조회시작시간 ^{주 2)}
	to_date	"20160405"	Y	N(8)	조회종료일자
	to_time	"110000"	N	N(6)	조회종료시간 ^{주 2)}
	sort_order	"D"	Y	A(1)	정렬순서 D:Descending, A:Ascending
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	befor_inquiry_trace_info	"123"	N	AN(20)	직전조회추적정보 ^{주 3)}

■ 요청 메시지 형태

JSON
<pre>{ "bank_tran_id": "F123456789U4BC34239Z", "bank_code_std": "097", "account_num": "1001234567890123", "user_seq_no": "1123456789", "inquiry_type": "A", "inquiry_base": "T", "from_date": "20160404", "from_time": "100000", "to_date": "20160405", "to_time": "110000", "sort_order": "D",</pre>

```
"tran_dtime": "20190910101921",
"befor_inquiry_trace_info": "123"
}
```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(100)	응답메시지(참가은행)
	account_num	"0001234567890123"	AN(16)	계좌번호
	balance_amt	"1000000"	SN(12)	계좌잔액(-금액가능)
	page_record_cnt	"25"	N(2)	현재페이지 레코드건수 ^{주 4)}
	next_page_yn	"Y"	A(1)	다음페이지 존재여부 ^{주 5)}
	befor_inquiry_trace_info	"123"	AN(20)	직전조회추적정보
	res_list	<object>		조회된 거래내역
	--tran_date	"20190910"	N(8)	거래일자
	--tran_time	"113000"	N(6)	거래시간
	--inout_type	"입금"	AH(8)	입출금구분 ^{주 6)} (입금, 출금, 지급, 기타)
	--tran_type	"현금"	AH(10)	거래구분 ^{주 7)}
	--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) [30p] [은행거래고유번호 생성 안내] 참조

주2) <조회기준 처리 안내>

- o 일자 기준으로 조회를 하는 경우 ("inquiry_base" : "D")
 - 요청시 조회시작일자, 조회종료일자를 셋팅하여 요청합니다.

- 일자기준 조회의 경우 은행은 조회 요청일자 기준으로 최소 6개월 범위에서 거래내역을 제공
한다. (은행에 따라 더 넓은 범위의 조회결과 제공 가능)

예) 12월 20일에 조회를 요청하는 경우 6월 21일부터의 거래 내역은 항상 제공하나, 6월 20일
을 포함한 이전 거래 내역 제공 여부는 은행별 상이

- o 시간 기준으로 조회를 하는 경우 ("inquiry_base" : "T")
 - 요청시 조회시작일자, 조회종료일자, 조회시작시간, 조회종료시간을 셋팅하여 요청합니다.
 - 시간기준 조회의 경우 은행은 조회 요청일자 기준으로 최소 3일 이하의 범위에서 거래내역을 제공한다. (은행에 따라 더 넓은 범위의 조회결과 제공 가능)
- 예) 1월 4일에 조회를 요청하는 경우 최소 1월 2일 00:00:00부터의 거래 내역은 항상 제공하나, 1월 1일을 포함한 이전 거래 내역 제공 여부는 은행별 상이
- 주3) 다음 페이지 요청 시에 직전 조회의 응답에서 얻은 값을 그대로 셋팅
다음 페이지 요청이 아닌 경우에는 파라미터 자체를 설정하지 않음
- 주4) 한 페이지는 최대 25건 가능
- 주5) <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)
- 주6) 입출금구분이 '기타'인 경우에는 '거래금액'이 0원으로 셋팅됨
- 주7) 현금, 대체, 급여, 타행환, F/B출금 등 (참가은행에서 직접 셋팅)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "account_num": "0001234567890123", "balance_amt": "1000000", "page_index_use_yn": "N", "page_index": "1", "page_record_cnt": "25", "next_page_yn": "Y", "befor_inquiry_trace_info": "1T201806171", "list_tran_seqno": "0", "res_list": [{ "tran_date": "20190910", "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

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

□ 명 세

2.4.1. 계좌실명조회 API

■ 설 명

이용기관이 특정 계좌의 계좌번호와 예금주 실명번호를 보유하고 있는 경우 해당 계좌의 유효성 및 예금주성명을 확인합니다.

□ 계좌실명조회 API : 실명조회

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/inquiry/real_name
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	bank_code_std	"097"	Y	AN(3)	개설기관.표준코드
	account_num	"1101230000678"	Y	AN(16)	계좌번호
	account_holder_info_type	"1"	Y	AN(1)	예금주 실명번호 구분코드 ^{주 2)}
	account_holder_info	"880101"	N	AN(13)	예금주 실명번호 ^{주 2)}
	tran_dtime	"20190910101921"	Y	N(14)	요청일시

주 1) [30p] [은행거래고유번호 생성 안내] 참조

주 2) [47p] [실명번호 구분코드] 참조

[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)
"N"	실명번호 구분코드 및 실명번호 검사 생략 (예금주 실명번호 셋팅 안함)		

- ※ 생년월일이 아닌 실명번호(전체)를 통한 계좌실명조화는 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 응답에
는 처리된 6자리만 전달됩니다. 추후에는 6자리만 허용될 예정입니다.
- ※ 실명번호 구분코드 및 실명번호 검사 생략을 통한 계좌실명조화는 법적근거가 있는 이용기관에게만
허용됩니다. (ex:전자금융업자)

[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 체
류할 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_tran_id": "F123456789U4BC34239Z", "bank_code_std": "097", "account_num": "1101230000678", "account_holder_info_type": " ", "account_holder_info": "880101", "tran_dtime": "20190910101921" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(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	"1101230000678"	AN(16)	계좌번호
	account_holder_info_type	[" 1 2 3 4 5 6 E N"]	AN(1)	예금주 실명번호 구분코드
	account_holder_info	"880101"	AN(13)	예금주 실명번호
	account_holder_name	"홍길동"	AH(20)	예금주성명
	account_type	"1"	AN(1)	계좌종류 1:수시입출금, 2:예적금 6:수익증권

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num": "1101230000678", "account_holder_info_type": " ", "account_holder_info": "880101", "account_holder_name": "홍길동", "account_type": "1" }</pre>

2.4.2. 송금인정보조회 API

■ 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.openbanking.or.kr/v2.0/inquiry/remit_list
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	bank_code_std	"097"	Y	AN(3)	조회하고자 하는 이용기관수취계좌의 개설기관.표준코드
	account_num	"3001230000678"	Y	AN(16)	조회하고자 하는 이용기관수취계좌의 계좌번호
	from_date	"20171001"	Y	N(8)	조회시작일자
	from_time	"143000"	Y	N(6)	조회시작시간
	to_date	"20171002"	Y	N(8)	조회종료일자
	to_time	"223000"	Y	N(6)	조회종료시간
	sort_order	"D"	Y	A(1)	정렬순서 D:Descending, A:Ascending
	befor_inquiry_trace_info	"12345678901234567890"	N	AN(20)	직전조회추적정보 ^{주 2)}
	tran_dtime	"20171001143000"	Y	N(14)	요청일시

■ 요청 메시지 형태

JSON
{ "bank_tran_id": "F123456789U4BC34239Z", "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"
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(은행)
	bank_tran_date	"20190910"	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 3)}
	page_record_cnt	"18"	N(2)	현재페이지 레코드건수 ^{주 4)}
	next_page_yn	"Y"	A(1)	다음페이지 존재여부 ^{주 5)}
	befor_inquiry_trace_info	"12345678901234567890"	AN(20)	직전조회 추적정보
	res_list	<object>		조회된 송금내역
	--tran_date	"20190910"	N(8)	거래일자
	--tran_time	"113000"	N(6)	거래시간
	--tran_type	"현금"	AH(10)	거래구분 ^{주 6)}
	--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) [30p] [은행거래고유번호 생성 안내] 참조

주2) 다음 페이지 요청 시에 직전 조회의 응답에서 얻은 값을 그대로 셋팅

다음 페이지 요청이 아닌 경우에는 파라미터 자체를 설정하지 않음

주3) "총 조회건수"는 **광주(034), 전북(037), 신한(088)** 은행의 경우에만 건수가 셋팅되며, 나머지 참가은행의 경우에는 "total_record_cnt": "0"으로 셋팅됩니다.

주4) 한 페이지는 최대 18건 가능

주5) <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 송금인정보조회 API가 과도하게 호출될 수 있으니, **일정 횟수를 초과하지 않도록 제한하는 방어로직의 삽입을 권장**합니다.

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

주6) 현금, 대체, 급여, 타행환, F/B출금 등 (참가은행에서 직접 셋팅)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "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": "20190910", "tran_time": "113000", "tran_type": "현금", "printed_content": "통장인자내용", "tran_amt": "450000", "after_balance_amt": "-1000000", "branch_name": "분당점", "remitter_name": "송금인", </pre>

```
    "remitter_bank_code": "097",  
    "remitter_account_num": "5009870000321" },  
    { ... },  
    ...  
    { ... }  
  ]  
}
```

2.4.3. 수취조회 API

■ 설 명

이용기관이 입금이체 요청 전 수취계좌의 입금가능여부 및 수취인성명을 사전에 조회합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/inquiry/receive
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	cntr_account_type	"N"	Y	A(1)	약정 계좌/계정구분 ^{주 2)} N:계좌, C:계정
	cntr_account_num	"1101230000678"	Y	AN(16)	약정 계좌/계정 번호 ^{주 2)}
	bank_code_std	"097"	N	AN(3)	입금은행.표준코드 ^{주 3)}
	account_num	"3001230000678"	N	AN(16)	계좌번호 ^{주 3)}
	fintech_use_num	"123456789012345678901234"	N	AN(24)	핀테크이용번호 ^{주 3)}
	print_content	"홍길동송금"	Y	AH(20)	입금계좌인자내역
	tran_amt	"10000"	Y	N(12)	거래금액
	req_client_name	"홍길동"	Y	AH(20)	요청고객성명 ^{주 4)}
	req_client_bank_code	"097"	N	AN(3)	요청고객계좌 개설기관.표준코드 ^{주 4)}
	req_client_account_num	"1221230000678"	N	AN(16)	요청고객계좌번호 ^{주 4)}
	req_client_fintech_use_num	"123456789012345678901234"	N	AN(24)	요청고객핀테크이용번호 ^{주 4)}
	req_client_num	"HONGGILDONG1234"	Y	AN(20)	요청고객회원번호 ^{주 4)}
	transfer_purpose	"TR"	Y	AN(2)	이체용도 ^{주 5)} TR:송금, ST:결제, AU:인증
	sub_frnc_name	"하위가맹점"	N	AH(40)	하위가맹점명 ^{주 6)}
	sub_frnc_num	"123456789012"	N	AN(20)	하위가맹점번호 ^{주 6)}
	sub_frnc_business_num	"1234567890"	N	N(10)	하위가맹점 사업자등록번호 ^{주 6)}
	cms_num	"93848103221"	N	AN(32)	CMS 번호 ^{주 7)}

주1) [30p] [은행거래고유번호 생성 안내] 참조

주2) 계약신청 시 오픈뱅킹에 제출한 이용기관 출금 계좌번호/계정과목코드 중 이체 시 사용할 출금
계좌번호/계정과목코드 정보를 셋팅

* 기본적으로 계좌번호를 사용하나 은행 이용기관에서 필요한 경우 계좌과목코드 사용

주3) 조회방식에 따라 핀테크이용번호 혹은 계좌정보(계좌번호, 계좌개설기관코드)를 셋팅하여야 하
며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

주4) 자금세탁방지 정보 수집을 위해 해당 거래를 요청한 고객의 정보를 셋팅

요청고객의 계좌번호 및 계좌개설기관코드를 알지 못하는 이용기관의 경우 요청고객 핀테크
이용번호를 셋팅하여야 하며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

주5) 출금이체 용도에 따라 구분코드를 셋팅

* [77p] 「이체 용도 안내」 참조

주6) 이용기관의 하위가맹점을 통해 거래가 발생한 경우 셋팅

주7) 사용자(고객)가 필요한 경우 셋팅

■ 요청 메시지 형태

JSON	
<pre>{ "bank_tran_id": "F123456789U4BC34239Z", "cntr_account_type": "N", "cntr_account_num": "1101230000678", "bank_code_std": "097", "account_num": "3001230000678", "print_content": "홍길동송금", "tran_amt": "10000", "req_client_name": "홍길동", "req_client_bank_code": "097", "req_client_account_num": "1221230000678", "req_client_num": "HONGGILDONG1234", "transfer_purpose": "TR", "sub_frnc_name": "하위가맹점", "sub_frnc_num": "123456789012", "sub_frnc_business_num": "1234567890", "cms_num": "93848103221" }</pre>	

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	bank_code_std	"097"	AN(3)	입금기관.표준코드
	bank_code_sub	"1230001"	AN(7)	입금기관.점별코드
	bank_name	"오픈은행"	AH(20)	입금기관명
	account_num	"3001230000678"	AN(16)	입금계좌번호*
	account_num_masked	"300123-0000-***"	NS*(20)	입금계좌번호(출력용)

print_content	"홍길동송금"	AH(20)	입금계좌인자내역
account_holder_name	"허균(개인)"	AH(20)	수취인성명
bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
bank_tran_date	"20190910"	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	"	AH(100)	응답메시지(참가은행)
wd_bank_code_std	"097"	AN(3)	출금(개설)기관.표준코드
wd_bank_name	"오픈은행"	AH(20)	출금(개설)기관명
wd_account_num	"1101230000678"	AN(16)	출금계좌번호
tran_amt	"10000"	N(12)	거래금액
cms_num	"93848103221"	AN(32)	CMS 번호

* 요청 시 계좌번호 및 계좌개설기관코드로 조회한 경우에만 입금계좌번호 항목 셋팅하여 응답

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num": "3001230000678", "account_num_masked": "300123-0000-***", "print_content": "홍길동송금", "account_holder_name": "허균", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "wd_bank_code_std": "097", "wd_bank_name": "오픈은행", "wd_account_num": "1101230000678", "tran_amt": "10000", "cms_num": "93848103221" }</pre>

2.5. 이체서비스

□ 기 능

사용자의 계좌로부터 이용기관의 계좌로 자금을 입금하거나, 이용기관의 계좌에서 사용자의 계좌로 자금을 송금합니다. 또한 이체결과를 확인하거나 **자금반환 청구 및 자금반환 결과를 조회합니다.**

□ 명 세

2.5.1. 출금이체 API

이용기관이 등록된 한 개의 사용자 계좌로부터 대금을 출금합니다. 등록된 핀테크이용번호로 요청하는 경우(/fin_num)와 실 계좌번호로 요청하는 경우(/acct_num)의 두 가지 기능을 각각 제공합니다.

□ 출금이체 API : 핀테크이용번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/fin_num
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = transfer, sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	cntr_account_type	"N"	Y	A(1)	약정 계좌/계정 구분 ^{주 2)} N:계좌, C:계정
	cntr_account_num	"1101230000678"	Y	AN(16)	약정 계좌/계정 번호 ^{주 2)}
	dps_print_content	"쇼핑몰환불"	Y	AH(20)	입금계좌인자내역
	fintech_use_num	"123456789012345678901234"	Y	AN(24)	출금계좌핀테크이용번호
	wd_print_content	"오픈뱅킹출금"	N	AH(14)	출금계좌인자내역 ^{주 3)}
	tran_amt	"10000"	Y	N(12)	거래금액
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	req_client_name	"홍길동"	Y	AH(20)	요청고객성명 ^{주 4)}
	req_client_bank_code	"097"	N	AN(3)	요청고객계좌 개설기관.표준코드 ^{주 4)}
	req_client_account_num	"1101230000678"	N	AN(16)	요청고객계좌번호 ^{주 4)}
	req_client_fintech_use_num	"123456789012345678901234"	N	AN(24)	요청고객핀테크이용번호 ^{주 4)}

req_client_num	"HONGGILDONG1234"	Y	AN(20)	요청고객회원번호 ^{주 4)}
transfer_purpose	"TR"	Y	AN(2)	이체용도 ^{주 5)} TR:송금, ST:결제, RC:충전
sub_frnc_name	"하위가맹점"	N	AH(40)	하위가맹점명 ^{주 6)}
sub_frnc_num	"123456789012"	N	AN(20)	하위가맹점번호 ^{주 6)}
sub_frnc_business_num	"1234567890"	N	N(10)	하위가맹점 사업자등록번호 ^{주 6)}
recv_client_name	"김오픈"	N	AH(20)	최종수취고객성명 ^{주 7)}
recv_client_bank_code	"097"	N	AN(3)	최종수취고객계좌 개설기관.표준코드 ^{주 7)}
recv_client_account_num	"232000067812"	N	AN(16)	최종수취고객계좌번호 ^{주 7)}

주1) [30p] [은행거래고유번호 생성 안내] 참조

주2) 계약신청 시 오픈뱅킹에 제출한 이용기관 입금 계좌번호/계정과목코드 중 이체 시 사용할 입금
계좌번호/계정과목코드 정보를 셋팅

* 기본적으로 계좌번호를 사용하나 은행 이용기관에서 필요한 경우 계정과목코드 사용

주3) 인증구분 및 출금계좌인자내역 항목 설정여부에 따른 사용자(고객)통장 기재 내용 안내

인증구분	출금계좌인자내역	사용자(고객)통장 기재 내용
자체인증	설정	설정된 출금계좌 인자내역 (사용자가 이용기관을 정확히 인지할 수 있는 값을 포함)
	미설정	이용기관에서 이용계약 시 설정하는 통장기재내용
센터인증	설정	
	미설정	

주4) 자금세탁방지 정보 수집을 위해 해당 거래를 요청한 고객의 정보를 셋팅

요청고객의 계좌번호 및 계좌개설기관코드를 알지 못하는 이용기관의 경우 요청고객 핀테크
이용번호를 셋팅하여야 하며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

* [77p] 「요청 고객 정보 안내」 참조

주5) 출금이체 용도에 따라 구분코드를 셋팅

* [77p] 「이체 용도 안내」 참조

주6) 이용기관의 하위가맹점을 통해 거래가 발생한 경우 셋팅

주7) 피싱 등 금융사고 발생 시 출금기관에서 최종 수취기관으로 지급정지 등 신속한 대응을 하기
위한 정보로써 이체용도 필드값이 송금("TR") 및 결제("ST")인 경우 해당 필드 값을 설정한다.
(단, 모든 이용기관에서 해당 정보 설정이 가능하도록 조치될 때 까지 동 정보의 검증은 하지
않는다.)

■ 요청 메시지 형태

JSON
<pre>{ "bank_tran_id": "F123456789U4BC34239Z", "cntr_account_type": "N", "cntr_account_num": "00012345678901234", "dps_print_content": "쇼핑몰환불", </pre>

```

"fintech_use_num": "123456789012345678901234",
"wd_print_content": "오픈뱅킹출금",
"tran_amt": "10000",
"tran_dtime": "20190910101921",
"req_client_name": "홍길동",
"req_client_bank_code": "097",
"req_client_account_num": "1101230000678",
"req_client_num": "HONGGILDONG1234",
"transfer_purpose": "TR",
"sub_frnc_name": "하위가맹점",
"sub_frnc_num": "123456789012",
"sub_frnc_business_num": "1234567890",
"recv_client_name": "김오픈",
"recv_client_bank_code": "097",
"recv_client_account_num": "232000067812"
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(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

	wd_limit_remain_amt	"9990000"	N(12)	출금한도잔여금액*
--	---------------------	-----------	-------	-----------

* 거래일 당일 남아있는 사용자(고객)의 출금이체 가능금액 (사용자 단위)

- 출금한도잔여금액이 남아있더라도, 은행에 설정된 사용자의 출금한도에 따라 거부될 수 있음
- "응답코드(API)"가 "사용자 출금이체 한도 초과(일 한도)(A0112)"인 경우에도 출금한도잔여금액을 셋팅하여 응답합니다.

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "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", "wd_limit_remain_amt": "9990000" }</pre>

■ 에러 처리

[\[62p\] \[출금이체 에러 처리 안내\] 참조](#)

[출금이체 에러 처리 안내]

오픈뱅킹으로부터 수신한 "응답코드(참가은행)"가 정상(000)이 아닌 경우에는 '이체 불능'으로 간주 하되, 오픈뱅킹으로부터 응답을 받지 못했을 경우나 아래 응답코드를 수신한 경우에는 처리 방법에 따라 처리해야 합니다.

응답코드(참가은행)	설명	처리 방법
입금 처리 중(400)	출금은 정상적으로 처리되었으나, 입금이 완료되지 않은 거래	이체결과조회 API를 통하여 최종결과를 확인

응답코드(API)	설명	처리 방법
내부 처리 에러(A0003)*	오픈뱅킹 내부 처리 에러	요청거래가 이체인 경우, 반드시 이체결과조회 API를 통하여 최종결과를 확인
처리시간 초과 에러(A0007)*	오픈뱅킹 내부 처리시간 초과 에러	
-	오픈뱅킹으로부터 응답을 받지 못했을 경우	
은행거래고유번호 중복(A0326)	동일 은행거래고유번호로 기 요청한 거래 존재	처리현황에 따라 조치 필요**

* 내부 처리과정에서 에러가 발생했으나, 이체 요청이 은행에서 정상 처리되었을 가능성이 있습니다.

** 동일한 은행거래고유번호로 기 요청한 거래가 존재하여 "은행거래고유번호 중복(A0326)" 응답코드를 수신한 것으로 이용기관은 처리현황에 따라 적절한 조치가 필요합니다.

요청	응답
① 최초 요청	최초 처리결과
② 중복 요청 (은행거래고유번호 중복)	은행거래고유번호 중복(A0326)

CASE 1) 최초 요청과 중복 요청이 동일한 거래인 경우 (하나의 거래를 두 번 송신한 경우)

→ 이체결과조회 API를 통하여 최초 처리결과 확인 필요

CASE 2) 최초 이체요청과 중복 이체요청이 별도 거래인 경우

(서로 다른 두 개의 거래를 처리해야 하나 은행거래고유번호가 동일하게 생성된 경우)

→ 최초 요청 거래의 처리결과를 확인하지 못했을 경우, 이체결과조회 API를 통하여 최초 처리 결과 확인 필요

→ 중복 요청된 두 번째 거래는 은행거래고유번호를 다시 생성하여 이체 요청 필요

□ 출금이체 API : 계좌번호 사용 (자체인증 이용기관만 이용가능)

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/transfer/withdraw/acnt_num
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	cntr_account_type	"N"	Y	A(1)	약정 계좌/계정구분 ^{주 2)} N:계좌, C:계정
	cntr_account_num	"1101230000678"	Y	AN(16)	약정 계좌/계정 번호 ^{주 2)}
	dps_print_content	"쇼핑물환불"	Y	AH(20)	입금계좌인자내역
	wd_bank_code_std	"097"	Y	AN(3)	출금은행.표준코드
	wd_account_num	"1101230000678"	Y	AN(16)	출금계좌번호
	tran_amt	"10000"	Y	N(12)	거래금액
	wd_print_content	"오픈뱅킹출금"	N	AH(14)	출금계좌인자내역 ^{주 3)}
	user_seq_no	"1123456789"	Y	AN(10)	사용자일련번호
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	req_client_name	"홍길동"	Y	AH(20)	요청고객성명 ^{주 4)}
	req_client_bank_code	"097"	N	AN(3)	요청고객계좌 개설기관.표준코드 ^{주 4)}
	req_client_account_num	"1101230000678"	N	AN(16)	요청고객계좌번호 ^{주 4)}
	req_client_fintech_use_num	"123456789012345678901234"	N	AN(24)	요청고객핀테크이용번호 ^{주 4)}
	req_client_num	"HONGGILDONG1234"	Y	AN(20)	요청고객회원번호 ^{주 4)}
	transfer_purpose	"TR"	Y	AN(2)	이체용도 ^{주 5)} TR:송금, ST:결제, RC:충전
	sub_frnc_name	"하위가맹점"	N	AH(40)	하위가맹점명 ^{주 6)}
	sub_frnc_num	"123456789012"	N	AN(20)	하위가맹점번호 ^{주 6)}
	sub_frnc_business_num	"1234567890"	N	N(10)	하위가맹점 사업자등록번호 ^{주 6)}
	recv_client_name	"김오폴"	N	AH(20)	최종수취고객성명 ^{주 7)}
	recv_client_bank_code	"097"	N	AN(3)	최종수취고객계좌 개설기관.표준코드 ^{주 7)}

recv_client_account_num	"232000067812"	N	AN(16)	최종수취고객계좌번호 ^{주7)}
-------------------------	----------------	---	--------	---------------------------

주1) [30p] [은행거래고유번호 생성 안내] 참조

주2) 계약신청 시 오픈뱅킹에 제출한 이용기관 입금 계좌번호/계정과목코드 중 이체 시 사용할 입금 계좌번호/계정과목코드 정보를 셋팅

* 기본적으로 계좌번호를 사용하나 은행 이용기관에서 필요한 경우 계좌과목코드 사용

주3) 인증구분 및 출금계좌인자내역 항목 설정여부에 따른 사용자(고객)통장 기재 내용 안내

인증구분	출금계좌인자내역	사용자(고객)통장 기재 내용
자체인증	설정	설정한 출금계좌 인자내역 (사용자가 이용기관을 정확히 인지할 수 있는 값을 포함)
	미설정	이용기관에서 이용계약 시 설정하는 통장기재내용
센터인증	설정	
	미설정	

주4) 자금세탁방지 정보 수집을 위해 해당 거래를 요청한 고객의 정보를 셋팅

요청고객의 계좌번호 및 계좌개설기관코드를 알지 못하는 이용기관의 경우 요청고객 핀테크 이용번호를 셋팅하여야 하며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

* [77p] 「요청 고객 정보 안내」 참조

주5) 출금이체 용도에 따라 구분코드를 셋팅

* [77p] 「이체 용도 안내」 참조

주6) 이용기관의 하위가맹점을 통해 거래가 발생한 경우 셋팅

주7) 피싱 등 금융사고 발생 시 출금기관에서 최종 수취기관으로 지급정지 등 신속한 대응을 하기 위한 정보로써 이체용도 필드값이 송금("TR") 및 결제("ST")인 경우 해당 필드 값을 설정한다.
(단, 모든 이용기관에서 해당 정보 설정이 가능하도록 조치될 때 까지 동 정보의 검증을 하지 않는다.)

■ 요청 메시지 형태

JSON
<pre>{ "bank_tran_id": "F123456789U4BC34239Z", "cntr_account_type": "N", "cntr_account_num": "00012345678901234", "dps_print_content": "쇼핑몰환불", "wd_bank_code_std": "097", "wd_account_num": "1101230000678", "wd_print_content": "오픈뱅킹출금", "tran_amt": "10000", "user_seq_no": "1123456789", "tran_dtime": "20190910101921", "req_client_name": "홍길동", "req_client_bank_code": "097", "req_client_account_num": "1101230000678", "req_client_num": "HONGGILDONG1234", "transfer_purpose": "TR", "sub_frnc_name": "하위가맹점", "sub_frnc_num": "123456789012", "sub_frnc_business_num": "1234567890",</pre>

```

"recv_client_name": "김오픈",
"recv_client_bank_code": "097",
"recv_client_account_num": "232000067812"
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20190910"	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	""	AH(100)	응답메시지(참가은행)
	account_num	"1101230000678"	AN(16)	출금계좌번호
	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
	wd_limit_remain_amt	"9990000"	N(12)	출금한도잔여금액*

* 거래일 당일 남아있는 사용자(고객)의 출금이체 가능금액 (사용자 단위)

- 출금한도잔여금액이 남아있더라도, 은행에 설정된 사용자의 출금한도에 따라 거부될 수 있음
- "응답코드(API)"가 "사용자 출금이체 한도 초과(일 한도)(A0112)"인 경우에도 출금한도잔여금액을 셋팅하여 응답합니다.

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "account_num": "1101230000678", "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", "wd_limit_remain_amt": "9990000" }</pre>

■ 에러 처리

[\[62p\] \[출금이체 에러 처리 안내\] 참조](#)

2.5.2. 입금이체 API

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

* 복수개의 계좌로의 입금이체는 앞으로 지원이 중단될(deprecated) 예정입니다. (2020.06 중단예정)

□ 입금이체 API : 핀테크이용번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/transfer/deposit/fin_num
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	cntr_account_type	"N"	Y	A(1)	약정 계좌/계정구분 ^{주 1)} N:계좌, C:계정
	cntr_account_num	"1101230000678"	Y	AN(16)	약정 계좌/계정 번호 ^{주 1)}
	wd_pass_phrase	"790d56ed.....6b821a69"	Y	aN(128)	입금이체용 암호문구*
	wd_print_content	"환불금액"	Y	AH(20)	출금계좌인자내역
	name_check_option	"on" or "off"	Y	aN(3)	수취인성명 검증 여부(on:검증함) (미지정 시 기본값: "on")
	sub_frnc_name	"하위가맹점"	N	AH(40)	하위가맹점명 ^{주 2)}
	sub_frnc_num	"123456789012"	N	AN(20)	하위가맹점번호 ^{주 2)}
	sub_frnc_business_num	"1234567890"	N	N(10)	하위가맹점 사업자등록번호 ^{주 2)}
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	req_cnt	"1"	Y	N(5)	입금요청건수
	req_list	<object>	Y		입금요청목록
	--tran_no	"1"	Y	N(5)	거래순번
	--bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 3)}
	--fintech_use_num	"123456789012345678901234"	Y	AN(24)	핀테크이용번호
	--print_content	"쇼핑몰환불"	Y	AH(20)	입금계좌인자내역
	--tran_amt	"10000"	Y	N(12)	거래금액
	--req_client_name	"홍길동"	Y	AH(20)	요청고객성명 ^{주 4)}

--req_client_bank_code	"097"	N	AN(3)	요청고객계좌 개설기관.표준코드 주 4)
--req_client_account_num	"1101230000678"	N	AN(16)	요청고객계좌번호 주 4)
--req_client_fintech_use_num	"123456789012345678901234"	N	AN(24)	요청고객핀테크이용번호 주 4)
--req_client_num	"HONGGILDONG1234"	Y	AN(20)	요청고객회원번호 주 4)
--transfer_purpose	"TR"	Y	AN(2)	이체용도 주 5) TR:송금, ST:결제, AU:인증
--recv_bank_tran_id	"F123456789U4BCT4112Z"	N	AN(20)	수취조회 은행거래고유번호 주 6)
--cms_num	"93848103221"	N	AN(32)	CMS 번호 주 7)

* [76p] 「입금이체용 암호문구 안내」 참조

** 한번의 API 호출로 최대 25건 입금 가능(거래수수료는 입금건별 과금)하지만 다건 입금이체는 앞으로 지원이 중단될(deprecated) 예정입니다. (2020.06 중단예정)

주1) 계약신청 시 오픈뱅킹에 제출한 이용기관 출금 계좌번호/계정과목코드 중 이체 시 사용할 출금 계좌번호/계정과목코드 정보를 셋팅

* 기본적으로 계좌번호를 사용하나 은행 이용기관에서 필요한 경우 계정과목코드 사용

주2) 이용기관의 하위가맹점을 통해 거래가 발생한 경우 셋팅

주3) [30p] 「은행거래고유번호 생성 안내」 참조

주4) 자금세탁방지 정보 수집을 위해 해당 거래를 요청한 고객의 정보를 셋팅

요청고객의 계좌번호 및 계좌개설기관코드를 알지 못하는 이용기관의 경우 요청고객 핀테크 이용번호를 셋팅하여야 하며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

* [77p] 「요청 고객 정보 안내」 참조

주5) 입금이체 용도에 따라 구분코드를 셋팅

* [77p] 「이체 용도 안내」 참조

주6) 이체 요청 전 수취조회를 요청한 경우, 이용기관은 수취조회 은행거래고유번호를 셋팅해야 하며, 수취조회 시 입력한 정보(입금계좌번호, 거래금액, CMS번호)와 동일한 정보를 입력하여야 합니다. 또한, 이 경우에는 "수취인성명 검증여부" 항목의 값과 상관없이 수취인성명 검증을 하지 않습니다.

주7) 사용자(고객)가 필요한 경우 셋팅

■ 요청 메시지 형태

JSON
<pre>{ "cntr_account_type": "N", "cntr_account_num": "00012345678901234", "wd_pass_phrase": "790d56ed.....6b821a69", "wd_print_content": "환불금액", "name_check_option": "on", "sub_frnc_name": "하위가맹점", "sub_frnc_num": "123456789012", "sub_frnc_business_num": "1234567890", "tran_dtime": "20190910101921", "req_cnt": "25",</pre>

```

"req_list": [
{
  "tran_no": "1",
  "bank_tran_id": "F123456789U4BC34239Z",
  "fintech_use_num": "123456789012345678901234",
  "print_content": "쇼핑물환불",
  "tran_amt": "10000",
  "req_client_name": "홍길동",
  "req_client_bank_code": "097",
  "req_client_account_num": "00012300000678",
  "req_client_num": "HONGGILDONG1234",
  "transfer_purpose": "TR",
  "recv_bank_tran_id": "F123456789U4BCT4112Z",
  "cms_num": "93848103221"
},
{ ... },
...
{ ... }
]
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	--bank_tran_date	"20190910"	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	""	AH(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
--cms_num	"93848103221"	AN(32)	CMS 번호

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "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": "쇼핑물환불", "account_holder_name": "홍길동", "tran_amt": "10000", "cms_num": "93848103221" }, { ... }, ...] }</pre>

■ 에러 처리

[\[71p\] \[입금이체 에러 처리 안내\] 참조](#)

[입금이체 에러 처리 안내]

오픈뱅킹으로부터 수신한 "응답코드(참가은행)"가 정상(000)이 아닌 경우에는 '이체 불능'으로 간주 하되, 오픈뱅킹으로부터 응답을 받지 못했을 경우나 아래 응답코드를 수신한 경우에는 처리 방법에 따라 처리해야 합니다.

응답코드(참가은행)	설명	처리 방법
입금 처리 중 (400)	출금은 정상적으로 처리되었으나, 입금이 완료되지 않은 거래	요청거래가 이체인 경우, 반드시 이체결과조회 API를 통하여 최종결과를 확인
내부 처리 에러 (803)*	오픈뱅킹 내부 처리 에러	
처리시간 초과 에러 (804)*	오픈뱅킹 내부 처리시간 초과 에러	
-	오픈뱅킹으로부터 응답을 받지 못했을 경우	
은행거래고유번호 중복 (822)	동일 은행거래고유번호로 기 요청한 거래 존재	처리현황에 따라 조치 필요**

응답코드(API)	설명	처리 방법
처리시간 초과 에러(A0007)*	오픈뱅킹 내부 처리시간 초과 에러	요청거래가 이체인 경우, 반드시 이체결과조회 API를 통하여 최종결과를 확인

* 내부 처리과정에서 에러가 발생했으나, 이체 요청이 은행에서 정상 처리되었을 가능성이 있습니다.

** 동일한 은행거래고유번호로 기 요청한 거래가 존재하여 "은행거래고유번호 중복(822)" 응답코드를 수신한 것으로 이용기관은 처리현황에 따라 적절한 조치가 필요합니다.

요청	응답
① 최초 요청	최초 처리결과
② 중복 요청 (은행거래고유번호 중복)	은행거래고유번호 중복(822)

CASE 1) 최초 요청과 중복 요청이 동일한 거래인 경우 (하나의 거래를 두 번 송신한 경우)

→ 이체결과조회 API를 통하여 최초 처리결과 확인 필요

CASE 2) 최초 이체요청과 중복 이체요청이 별도 거래인 경우

(서로 다른 두 개의 거래를 처리해야 하나 은행거래고유번호가 동일하게 생성된 경우)

→ 최초 요청 거래의 처리결과를 확인하지 못했을 경우, 이체결과조회 API를 통하여 최초 처리 결과 확인 필요

→ 중복 요청된 두 번째 거래는 은행거래고유번호를 다시 생성하여 이체 요청 필요

□ 입금이체 API : 계좌번호 사용

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/transfer/deposit/acnt_num
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	cntr_account_type	"N"	Y	A(1)	약정 계좌/계정구분 ^{주 1)} N:계좌, C:계정
	cntr_account_num	"1101230000678"	Y	AN(16)	약정 계좌/계정 번호 ^{주 1)}
	wd_pass_phrase	"790d56ed.....821a69"	Y	aN(128)	입금이체용 암호문구*
	wd_print_content	"출금계좌인자내역"	Y	AH(20)	출금계좌인자내역
	name_check_option	"on" or "off"	Y	aN(3)	수취인성명 검증 여부(on:검증함) (미지정 시 기본값: "on")
	sub_frnc_name	"하위가맹점"	N	AH(40)	하위가맹점명 ^{주 2)}
	sub_frnc_num	"123456789012"	N	AN(20)	하위가맹점번호 ^{주 2)}
	sub_frnc_business_num	"1234567890"	N	N(10)	하위가맹점 사업자등록번호 ^{주 2)}
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	req_cnt	"25"	Y	N(5)	입금요청건수**
	req_list	<object>	Y		입금요청목록
	--tran_no	"1"	Y	N(5)	거래순번
	--bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 3)}
	--bank_code_std	"097"	Y	AN(3)	입금은행.표준코드
	--account_num	"1101230000678"	Y	AN(16)	계좌번호
	--account_holder_name	"홍길동"	Y	AH(20)	입금계좌예금주명***
	--print_content	"입금계좌인자내역"	Y	AH(20)	입금계좌인자내역
	--tran_amt	"10000"	Y	N(12)	거래금액
	--req_client_name	"홍길동"	Y	AH(20)	요청고객성명 ^{주 4)}
	--req_client_bank_code	"097"	N	AN(3)	요청고객계좌 개설기관.표준코드 ^{주 4)}
	--req_client_account_num	"1101230000678"	N	AN(16)	요청고객계좌번호 ^{주 4)}
	--req_client_fintech_use_num	"123456789012345678901234"	N	AN(24)	요청고객핀테크이용번호 ^{주 4)}
	--req_client_num	"HONGGILDONG1234"	Y	AN(20)	요청고객회원번호 ^{주 4)}

--transfer_purpose	"TR"	Y	AN(2)	이체용도 ^{주 5)} TR:송금, ST:결제, AU:인증
--recv_bank_tran_id	"F123456789U4BC4112Z"	N	AN(20)	수취조회 은행거래고유번호 ^{주 6)}
--cms_num	"93848103221"	N	AN(32)	CMS 번호 ^{주 7)}

* [76p] 「입금이체용 암호문구 안내」 참조

** 한번의 API 호출로 최대 25건 입금 가능(거래수수료는 입금건별 과금)하지만 다건 입금이체는 앞으로 지원이 중단될(deprecated) 예정입니다. (2020.06 중단예정)

*** 참가은행에 등록되어 있는 입금계좌의 예금주명과 정확히 일치해야 함

주1) 계약신청 시 오픈뱅킹에 제출한 이용기관 출금 계좌번호/계정과목코드 중 이체 시 사용할 출금 계좌번호/계정과목코드 정보를 셋팅

* 기본적으로 계좌번호를 사용하나 은행 이용기관에서 필요한 경우 계정과목코드 사용

주2) 이용기관의 하위가맹점을 통해 거래가 발생한 경우 셋팅

주3) [30p] 「은행거래고유번호 생성 안내」 참조

주4) 자금세탁방지 정보 수집을 위해 해당 거래를 요청한 고객의 정보를 셋팅

요청고객의 계좌번호 및 계좌개설기관코드를 알지 못하는 이용기관의 경우 요청고객 핀테크 이용번호를 셋팅하여야 하며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

* [77p] 「요청 고객 정보 안내」 참조

주5) 입금이체 용도에 따라 구분코드를 셋팅

* [77p] 「이체 용도 안내」 참조

주6) 이체 요청 전 수취조회를 요청한 경우, 이용기관은 수취조회 은행거래고유번호를 셋팅해야 하며, 수취조회 시 입력한 정보(입금계좌번호, 거래금액, CMS번호)와 동일한 정보를 입력하여야 합니다. 또한, 이 경우에는 "수취인성명 검증여부" 항목의 값과 상관없이 수취인성명 검증을 하지 않습니다.

주7) 사용자(고객)가 필요한 경우 셋팅

■ 요청 메시지 형태

JSON
<pre>{ "cntr_account_type": "N", "cntr_account_num": "00012345678901234", "wd_pass_phrase": "790d56ed.....6b821a69", "wd_print_content": "환불금액", "name_check_option": "on", "sub_frc_name": "하위가맹점", "sub_frc_num": "123456789012", "sub_frc_business_num": "1234567890", "tran_dtime": "20190910101921", "req_cnt": "25", "req_list": [{ "tran_no": "1", "bank_tran_id": "F123456789U4BC34239Z", "bank_code_std": "097", "account_num": "1101230000678", "account_hloder_name": "홍길동", "print_content": "쇼핑몰환불", </pre>


```

    "tran_amt": "10000",
    "req_client_name": "홍길동",
    "req_client_bank_code": "097",
    "req_client_account_num": "00012300000678",
    "req_client_num": "HONGGILDONG1234",
    "transfer_purpose": "TR",
    "recv_bank_tran_id": "F123456789U4BCT4112Z",
    "cms_num": "93848103221"
  },
  { ... },
  ...
  { ... }
]
}

```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	--bank_tran_date	"20190910"	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	""	AH(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	"1101230000678"	AN(16)	입금계좌번호
	--account_num_masked	"000-1230000-***"	NS*(20)	입금계좌번호(출력용)
	--print_content	"쇼핑몰환불"	AH(20)	입금계좌인자내역
	--account_holder_name	"홍길동"	AH(20)	수취인성명

--tran_amt	"10000"	N(12)	거래금액
--cms_num	"93848103221"	AN(32)	CMS 번호

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "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": "F123456789U4BC34239Z", "bank_tran_date": "20190910", "bank_code_tran": "097", "bank_rsp_code": "000", "bank_rsp_message": "", "bank_code_std": "097", "bank_code_sub": "1230001", "bank_name": "오픈은행", "account_num": "1101230000678", "account_num_masked": "000-1230000-***", "print_content": "입금계좌인자내역", "account_holder_name": "홍길동", "tran_amt": "10000", "cms_num": "93848103221" }, { ... }, ... { ... }] }</pre>

■ 에러 처리

[\[71p\] \[입금이체 에러 처리 안내\] 참조](#)

<입금이체용 암호문구 안내>

- 간편가입 회원인 경우 : 기본값 "NONE"를 셋팅함
- 이용기관인 경우 :
(테스트 환경) 기본값 "NONE"를 셋팅
(운영 환경) 오픈뱅킹 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
- 「전자금융감독규정」상 전용 회선을 사용해야 하는 이용기관인 경우 : 기본값 "NONE"를 셋팅함

<수취인성명 검증 안내>

- 입금이체 시에 자금을 수취하는 사람이 입금요청한 계좌의 예금주와 동일인인지 비교

API 구분	검증대상	검증값	검증기준
/v2.0/transfer/deposit/fin_num	입금요청한 핀테크이용번호에 매핑되어 있는 예금주명 (DB 조회)	이체전 수취조회결과의 예금주명 (참가은행 조회)	문자열에서 모든 공백을 제거후 검증값의 글자수(최대 10 자) 만큼 비교
/v2.0/transfer/deposit/acnt_num	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.)
- **이용기관이 사전에 수취조회를 통해 수취인성명을 확인한 후, 입금이체 시 수취조회 은행거래고유번호를 셋팅한 경우 수취인성명 검증을 하지 않습니다.**

※ 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.

<요청 고객 정보 안내>

- 이용기관은 참가기관의 자금세탁방지 정보 수집을 위한 요청 고객 정보(성명, 계좌, 회원번호)를 다음과 같이 설정한다.

예1) A고객이 B고객에게 이체를 요청하는 경우

처리내용	송금인 실명	수취인 성명	요청 고객 정보
① 출금이체 요청(A 고객 → 이용기관)	A	이용기관명	A (요청 고객)
② 입금이체 요청(이용기관 → B 고객)	이용기관명	B	A (요청 고객)

예2) A고객이 C업체의 재화를 구매하고, 결제를 요청하는 경우

처리내용	송금인 실명	수취인 성명	요청 고객 정보
① 출금이체 요청(A 고객 → 이용기관)	A	이용기관명	A (요청 고객)
② 입금이체 요청(이용기관 → C 업체)	이용기관명	C	A (요청 고객)

예3) A고객이 이용기관의 Point를 충전하는 경우

처리내용	송금인 실명	수취인 성명	요청 고객 정보
① 출금이체 요청(A 고객 → 이용기관)	A	이용기관명	A (요청 고객)

예4) 이용기관이 A고객 계좌소유인증을 위해 이체하는 경우

처리내용	송금인 실명	수취인 성명	요청 고객 정보
① 입금이체 요청(이용기관 → A 고객)	이용기관명	A	A (요청 고객)

- 요청고객 회원번호 : 해당 거래를 요청한 고객의 회원번호를 설정한다. 이용기관은 고객에 1:1로 대응하도록 회원번호를 부여해야 한다. 이용기관은 '요청고객 회원번호' 부여 시, 전화번호 등 개인 정보가 포함되지 않도록 유의하여야 한다.

<이체 용도 안내>

- 출금이체 및 입금이체 용도에 따라 다음과 같이 구분코드를 설정해야 합니다.

이체용도	구분코드	설명
송금	TR	고객이 다른 계좌에 송금을 요청하는 경우
결제	ST	고객이 재화 등을 구입하고, 대금을 결제하는 경우
충전	RC	고객이 이용기관 Point 등의 충전을 요청하는 경우 ※ 송금(결제)을 위해 충전하는 경우에는 송금(결제)으로 설정
인증	AU	계좌소유인증을 위해 입금이체* 하는 경우 * 이용기관계좌에서 출금해서 인증 대상 사용자 계좌에 입금

2.5.3. 이체결과조회 API

■ 설 명

이용기관이 출금이체 혹은 입금이체 후 이체결과를 다시 확인합니다. 이체 시 비정상적인 응답코드를 받았을 경우나 응답을 받지 못했을 경우 등 이체결과 확인이 필요한 경우 사용하는 용도입니다. 단, 1개월 이내의 이체 건에 대해서만 확인이 가능합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/transfer/result
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	check_type	"1"	Y	AN(1)	1:출금이체, 2:입금이체
	tran_dtime	"20190910101921"	Y	N(14)	요청일시
	req_cnt	"25"	Y	N(5)	요청건수*
	req_list	<object>	Y		요청목록
	--tran_no	"1"	Y	N(5)	거래순번
	--org_bank_tran_id	"F123456789U123A4239C"	Y	AN(20)	원거래 거래고유번호(참가은행)**
	--org_bank_tran_date	"20190910"	Y	N(8)	원거래 거래일자(참가은행)***
	--org_tran_amt	"10000"	Y	N(12)	원거래 거래금액

* 한번의 API 호출로 최대 25건 확인 가능

** 이체 시 이용기관에서 생성하여 사용했던 은행거래고유번호 셋팅

*** 원칙적으로는 이체 원거래 응답에 있는 거래일자를 셋팅해야 하나

시뮬레이터 모드로 테스트시에는 이체 원거래의 실제 테스트일자를 셋팅함

■ 요청 메시지 형태

JSON
<pre>{ "check_type": "1", "tran_dtime": "20190910101921", "req_cnt": "25", "req_list": [{ "tran_no": "1", "org_bank_tran_id": "F123456789U123A4239C", "org_bank_tran_date": "20190910", "org_tran_amt": "10000" }, { ... }, ...] }</pre>

```
{ ... }  
]  
}
```

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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	"20190910"	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	""	AH(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	"쇼핑물환불"	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": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "req_cnt": "25", "req_list": [{ "tran_no": "1", "bank_tran_id": "12345678901234567890", "bank_tran_date": "20190910", "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": "쇼핑물환불", "dps_account_holder_name": "홍길동", "tran_amt": "10000" }, { ... }, ... { ... }] }</pre>

■ “입금 처리 중 (400)” 응답코드 확인방법

이체결과조회 API의 응답이 “입금 처리 중(400)”일 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)

[이체결과조회 오류 응답 처리 안내]

오픈뱅킹으로부터 수신한 "응답코드(참가은행)"가 정상(000)이 아닌 경우에는 '이체 불능'으로 간주 하되, 오픈뱅킹으로부터 응답을 받지 못했을 경우나 아래 응답코드를 수신한 경우에는 처리 방법에 따라 처리해야 합니다.

응답코드(참가은행)	설명	처리 방법
입금 처리 중 (400)	출금은 정상적으로 처리되었으나, 입금이 완료되지 않은 거래	이체결과조회 API를 다시 호출하여 최종결과를 확인
전문 변환 에러 (802)*	오픈뱅킹 전문 변환 에러	
내부 처리 에러 (803)*	오픈뱅킹 내부 처리 에러	
처리시간 초과 에러 (804)*	오픈뱅킹 내부 처리시간 초과 에러	
내부 전문 송신 실패 (819)*	오픈뱅킹 내부 전문 송신 실패 에러	
-	오픈뱅킹으로부터 응답을 받지 못했을 경우	이체를 다시 요청하거나, 이체 실패로 처리
조회 대상거래 없음 (701)	오픈뱅킹으로 이체 요청이 들어 오지 않은 경우	
이체 내역 없음 (813)		

* 내부 처리과정에서 에러가 발생하여 이체결과를 조회하지 못한 경우입니다.

2.5.4. 자금반환 청구 API

■ 설 명

거래가 완료된 입금이체에 대해서 사용자의 착오송금, 이용기관의 장애 등으로 자금반환이 필요한 경우 이용기관은 자금반환을 청구합니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/return/claim
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>			오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	bank_tran_id	"F123456789U4BC34239Z"	Y	AN(20)	은행거래고유번호 ^{주 1)}
	org_bank_tran_date	"20200101"	Y	N(8)	원거래 거래일자 ^{주 2)}
	org_bank_tran_id	"F123456789U4BC312345"	Y	AN(20)	원거래 거래고유번호 ^{주 2)}
	org_dps_bank_code_std	"097"	N	AN(3)	원거래 입금기관.표준코드 ^{주 2)주 3)}
	org_dps_account_num	"221892175525"	N	AN(16)	원거래 입금계좌번호 ^{주 2)}
	org_dps_fintech_use_num	"123456789012345678901234"	N	AN(24)	원거래 입금핀테크이용번호 ^{주 2)}
	org_tran_amt	"10000"	Y	N(12)	거래금액(청구금액) ^{주 2)주 4)}
	org_wd_bank_code_std	"097"	Y	AN(3)	원거래 출금기관.표준코드 ^{주 2)주 3)}
	claim_code	"02"	Y	N(2)	청구사유코드 ^{주 5)}
	claim_message	"착오송금"	N	AH(100)	청구사유 ^{주 5)}
	total_return_yn	"Y"	Y	A(1)	전액반환여부 ^{주 4)}
	return_account_num	"102482575881"	Y	AN(16)	반환금액 입금계좌 ^{주 6)}
	use_org_contact	"010-1234-5678"	Y	ANS(15)	이용기관 담당자 연락처
	use_org_email	"testbed@kftc.or.kr"	Y	E(100)	이용기관 담당자 이메일주소 ^{주 7)}

주1) [30p] [은행거래고유번호 생성 안내] 참조

주2) 자금반환을 청구하려는 원거래(거래가 완료된 입금이체) 정보를 셋팅. 원거래 입금이체 방식에 따라 입금핀테크이용번호 혹은 입금계좌정보(입금계좌번호 및 입금기관.표준코드)를 셋팅하여야 하며, 둘 다 셋팅할 경우 포맷오류로 응답합니다.

주3) 원거래의 입금기관 대표코드, 출금기관 대표코드를 셋팅한다. 전산통합 등으로 대표코드에 변경이 발생한 경우 변경된 대표코드를 셋팅한다.

주4) 전액반환(전액청구)을 요청하는 경우 원거래금액 및 전액반환("Y")으로 셋팅하며, 부분 반환 등 청구금액이 원거래와 일치하지 않는 경우 청구금액 및 부분반환("N")으로 셋팅한다.

주5) 청구사유에 따라 다음과 같이 코드 및 사유를 셋팅

청구사유코드	내용	청구사유 설정방법
02	계좌입력오류	<ul style="list-style-type: none"> - 청구사유코드를 '기타'(05)로 선택한 경우, '청구사유' 필드를 반드시 선택하여야 함 - 청구사유코드가 '기타'(05)가 아닌 경우에도 '청구사유' 설정이 가능합니다.
03	금액입력오류	
04	이중입금	
05	기타	
08	고객 반환의사 표시로 재청구	

주6) 금액을 반환 받을 이용기관의 계좌번호(원거래 출금기관에 개설된 계좌)를 선택

주7) 오픈뱅킹센터에서 자금반환 처리 결과를 수신하였을 때, 입력받은 이용기관 담당자 이메일주소로 결과를 통보합니다.

■ 요청 메시지 형태

JSON
<pre>{ "bank_tran_id": "F123456789U4BC34239Z", "org_bank_tran_date": "20200101", "org_bank_tran_id": "F123456789U4BC312345", "org_dps_bank_code_std": "097", "org_dps_account_num": "221892175525", "org_tran_amt": "10000", "org_wd_bank_code_std": "097", "claim_code": "02", "claim_message": "착오송금", "total_return_yn": "Y", "return_account_num": "102482575881", "use_org_contact": "010-1234-5678", "use_org_email": "testbed@kftc.or.kr" }</pre>

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	dps_bank_code_std	"097"	AN(3)	입금기관.표준코드
	bank_tran_id	"F123456789U4BC34239Z"	AN(20)	거래고유번호(참가은행)
	bank_tran_date	"20200110"	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	""	AH(100)	응답메시지(참가은행)
	wd_bank_code_std	"097"	AN(3)	출금(개설)기관.표준코드
	wd_bank_code_sub	"1230001"	AN(7)	출금(개설)기관.점별코드
	wd_bank_name	"오픈은행"	AH(20)	출금(개설)기관명

org_bank_tran_date	"20200101"	AN(20)	원거래 거래일자
org_bank_tran_id	"F123456789U4BC312345"	N(8)	원거래 거래고유번호
org_dps_bank_code_std	"097"	AN(3)	원거래 입금기관.표준코드
org_dps_account_num	"221892175525"	AN(16)	원거래 입금계좌번호*
org_dps_fintech_use_num	"123456789012345678901234"	AN(24)	원거래 입금핀테크이용번호*
org_tran_amt	"10000"	N(12)	원거래 거래금액
return_account_num	"102482575881"	AN(16)	반환금액 입금계좌

* 요청 시 입력한 정보에 따라 셋팅됨 (원거래 입금계좌번호/원거래 입금핀테크이용번호)

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "dps_bank_code_std": "097", "bank_tran_id": "F123456789U4BC34239Z", "bank_tran_date": "20200110", "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", "org_bank_tran_date": "20200101", "org_bank_tran_id": "F123456789U4BC312345", "org_dps_bank_code_std": "097", "org_dps_account_num": "221892175525", "org_dps_fintech_use_num": "", "org_tran_amt": "10000", "return_account_num": "102482575881" }</pre>

■ 자금반환 청구 검증 안내

- 청구 요청일 기준으로 30일 이내 청구 내역이 없는지 확인
 - 원거래 거래일자 및 원거래 거래고유번호 기준
 - 반환기관에서 반환 불가로 응답한 내역은 재청구 가능
- 청구 요청일 기준으로 1년 이내 기 반환 내역이 존재하지 않는지 검증(원거래 일자/원거래 고유번호 기준) 단, 기 반환 내역과 신규 청구 요청 내역이 모두 부분 반환인 경우에는 처리 허용
- 원거래의 정보와 자금반환 청구 요청시 입력한 정보와 맞는지 검증 (원거래 입금계좌번호/핀테크이용번호, 원거래 거래금액, 원거래 출금기관.표준코드) 단, 부분반환은 거래금액 검증 안함
- 원거래 거래일자가 자금반환 청구 요청일 기준으로 1년 이내의 거래가 아닐 경우 원거래의 정보 입력값 검증 안함

■ 에러 처리

[85p] [자금반환 청구 에러 처리 안내] 참조

[자금반환 청구 에러 처리 안내]

오픈뱅킹으로부터 수신한 "응답코드(참가은행)"가 정상(000)이 아닌 경우에는 '자금반환 청구 불능'으로 간주하되, 오픈뱅킹으로부터 응답을 받지 못했을 경우나 아래 응답코드를 수신한 경우에는 처리 방법에 따라 처리해야 합니다.

응답코드(참가은행)	설명	처리 방법
청구 접수 처리중(900)	원거래 출금은행에서 자금반환 청구 접수 처리중인 경우	자금반환 결과조회 API를 통하여 최종결과를 확인

응답코드(API)	설명	처리 방법
내부 처리 에러(A0003)*	오픈뱅킹 내부 처리 에러	자금반환 결과조회 API를 통하여 최종결과를 확인
처리시간 초과 에러(A0007)*	오픈뱅킹 내부 처리시간 초과 에러	
-	오픈뱅킹으로부터 응답을 받지 못했을 경우	
은행거래고유번호 중복(A0326)	동일 은행거래고유번호로 기 요청한 거래 존재	처리현황에 따라 조치 필요**

* 내부 처리과정에서 에러가 발생했으나, 이체 요청이 은행에서 정상 처리되었을 가능성이 있습니다.

** 동일한 은행거래고유번호로 기 요청한 거래가 존재하여 "은행거래고유번호 중복(A0326)" 응답코드를 수신한 것으로 이용기관은 처리현황에 따라 적절한 조치가 필요합니다.

요청	응답
① 최초 요청	최초 처리결과
② 중복 요청 (은행거래고유번호 중복)	은행거래고유번호 중복(A0326)

CASE 1) 최초 요청과 중복 요청이 동일한 거래인 경우 (하나의 거래를 두 번 송신한 경우)

→ 자금반환 결과조회 API를 통하여 최초 처리결과 확인 필요

CASE 2) 최초 자금반환 청구요청과 중복 청구요청이 별도 거래인 경우

(서로 다른 두 개의 거래를 처리해야 하나 은행거래고유번호가 동일하게 생성된 경우)

→ 최초 요청 거래의 처리결과를 확인하지 못했을 경우, 자금반환 결과조회 API를 통하여 최초 자금반환 요청 결과 확인 필요

→ 중복 요청된 두 번째 거래는 은행거래고유번호를 다시 생성하여 자금반환 청구 요청 필요

2.5.5. 자금반환 결과조회 API

■ 설 명

이용기관이 자금반환 청구 요청 후 자금반환 결과를 확인합니다. 자금반환 청구 요청결과 확인이 필요한 경우나 자금반환 청구가 정상적으로 요청된 후 반환 결과를 확인하는 용도입니다. 단, 자금반환 청구 요청 시점 기준으로 1년 동안만 반환 결과 확인이 가능합니다. 자금반환 결과의 상태는 하루에 한번만 변경됩니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/return/result
HTTP Method	POST
Content-Type	application/json; charset=UTF-8

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>			오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Body	claim_bank_tran_date	"20200110"	Y	N(8)	청구 요청 거래일자 ^{주 1)}
	claim_bank_tran_id	"F123456789U4BC312345"	Y	AN(20)	청구 요청 거래고유번호 ^{주 1)}

주1) 자금반환 청구 요청일자 및 청구 요청 시 이용기관에서 사용했던 은행거래고유번호 셋팅

■ 요청 메시지 형태

JSON
{ "claim_bank_tran_date": "20200110", "claim_bank_tran_id": "F123456789U4BC312345" }

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	claim_bank_tran_date	"20200110"	N(8)	청구 요청 거래일자 ^{주 1)}
	claim_bank_tran_id	"F123456789U4BC312345"	AN(20)	청구 요청 거래고유번호 ^{주 1)}
	claim_normal_yn	"Y"	A(1)	청구 요청 정상여부 ^{주 1)}
	claim_bank_rsp_code	"000"	AN(3)	청구 요청 응답코드(참가은행) ^{주 1)}

return_yn	"Y"	A(1)	자금반환여부 ^{주 2)}
return_fail_code	""	N(2)	자금반환불가사유코드 ^{주 2)}
return_fail_message	""	AH(100)	자금반환불가사유 ^{주 2)}
total_return_yn	"Y"	A(1)	전액반환여부 ^{주 3)}
return_amt	"10000"	N(12)	반환금액

주1) 자금반환 청구 요청에 대한 결과 항목이며, 청구 요청이 정상적으로 접수되지 않았을 경우 자금반환 결과 항목은 셋팅되지 않음

주2) 자금반환 청구 요청이 정상적으로 접수 된 후 자금반환이 불가할 시 셋팅되며, 사유코드는 다음과 같음

코드	사 유	처리구분 (금융감독원 분류기준)
50	원거래상이	청구정보오류
51	고객거부	반환거부
52	법적제한계좌	반환불가
54	제휴 및 연계계좌	청구정보오류
55	해약계좌	반환불가
56	고객연락불가(연락처 상이)	
58	고객 직접반환	개별반환
59	이용기관 반환(가상계좌)	
60	지급정지계좌(기타채무)	반환불가
61	특이계좌(청약, 사망자)	
62	전자기기통신 사기거래계좌(대포통장 등)	
63	개인 간 분쟁	반환불가
99	기타	

주3) 자금반환 청구요청 시의 이용기관 셋팅값과 다를 수 있음

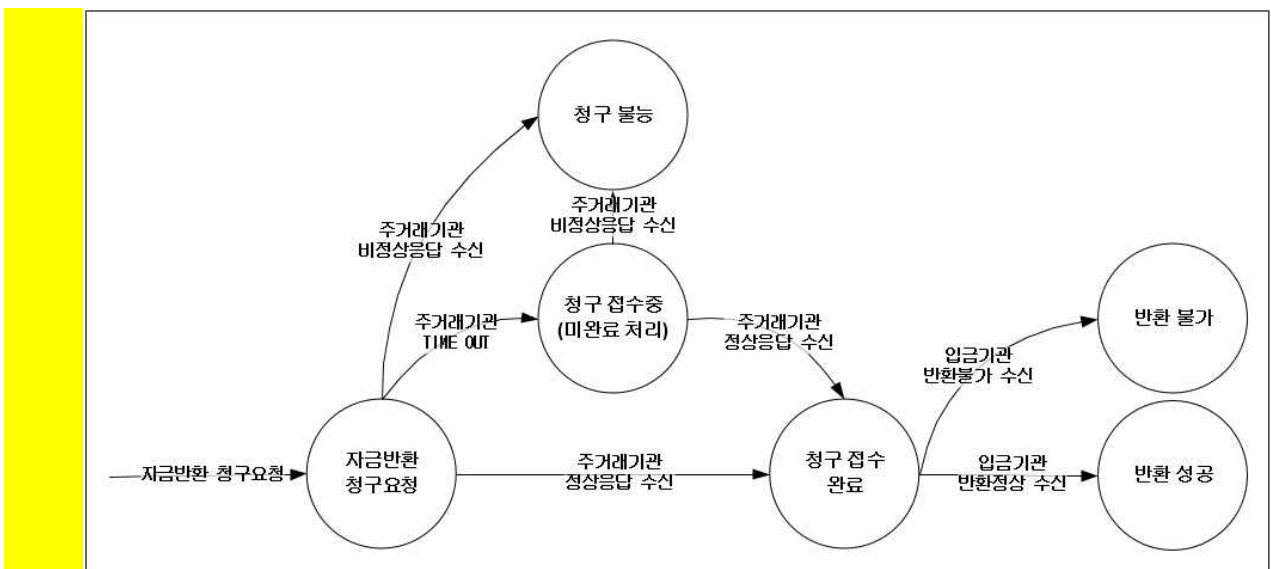
■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "claim_bank_tran_date": "20200110", "claim_bank_tran_id": "F123456789U4BC312345" "claim_normal_yn": "Y", "claim_bank_rsp_code": "000", "return_yn": "Y", "return_fail_code": "", "return_fail_message": "", "total_return_yn": "Y", "return_amt": "10000" }</pre>

■ 자금반환 결과 상태에 따른 응답 메시지 안내

자금반환 처리상태	claim_normal_yn	claim_bank_ rsp_code	return_yn	return_fail_ code	return_fail_ message	total_return_yn
청구 접수중	"N"	"900"	-	-	-	-
청구 불능	"N"	예) "903"	-	-	-	-
청구 접수 완료	"Y"	"000"	-	-	-	-
반환 불가	"Y"	"000"	"N"	예) "51"	예) "고객거부"	-
반환 성공	"Y"	"000"	"Y"	-	-	"Y" / "N"

■ 자금반환 결과 상태전이도



■ 자금반환 결과조회 오류 응답 처리 안내

오픈뱅킹으로부터 수신한 "청구 요청 응답코드(참가은행)"가 정상(000)이 아닌 경우에는 '청구 요청 불능'으로 간주하되, 오픈뱅킹으로부터 응답을 받지 못했을 경우나 아래 응답코드를 수신한 경우에는 처리 방법에 따라 처리해야 합니다.

청구 요청 응답코드(참가은행)	설명	처리 방법
청구 접수 처리중(900)	원거래 출금은행에서 자금반환 청구 접수 처리중인 경우	자금반환 결과조회 API를 통하여 최종결과를 확인
조회 대상거래 없음 (701)	오픈뱅킹으로 자금반환 청구 요청이 들어오지 않은 경우	자금반환 청구를 다시 요청

응답코드(API)	설명	처리 방법
내부 처리 에러(A0003)*	오픈뱅킹 내부 처리 에러	자금반환 결과조회 API를 통하여 최종결과를 확인
-	오픈뱅킹으로부터 응답을 받지 못했을 경우	

* 내부 처리과정에서 에러가 발생했으나, 이제 요청이 은행에서 정상 처리되었을 가능성이 있습니다.

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.openbanking.or.kr/v2.0/bank/status
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>	Y		오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	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": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "res_cnt": "16", "res_list": [{ "bank_code_std": "097", "bank_name": "오픈은행", "bank_status": "Y" }, { ... }, ... { ... }] }</pre>

2.6.2. 수수료조회 API

■ 설 명

이용기관이 사용한 API에 대한 수수료를 조회합니다. 단, 요청 당일 기준으로 1년 이내의 수수료에 대해서만 확인이 가능하며 요청 당일은 조회할 수 없습니다. 최대 조회기간은 1개월이며 API가 적용되는 시점 이후로 조회 가능합니다.(2019.12.07.)

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/manage/fee
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>			오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Parameter	from_date	"20191001"	Y	N(8)	조회시작일자
	to_date	"20191031"	Y	N(8)	조회종료일자

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	res_cnt	"8"	N(5)	조회대상 API 개수
	res_list	<object>		API 목록
	--api_manage_id	"001"	AN(3)	API 관리 ID 001:잔액조회 002:거래내역조회 003:입금이체* 004:입금이체(계좌소유인증) 005:출금이체 006:계좌실명조회 007:송금인정보조회 008:수취조회
	--use_org_fee	"10000000"	N(12)	이용기관수수료(조회기간 합계)**
	--hdl_agt_fee	"50000000"	N(12)	처리대행비용(조회기간 합계)

* 다건 입금이체의 경우, 수수료는 입금 건별로 과금됨

** 이용기관수수료 = 주거래은행 수수료 + 처리대행비용

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "res_cnt": "7", "res_list": [{ "api_manage_id": "001", "use_org_fee": "10000000", "hdl_agt_fee": "5000000" }, { ... }, ... { ... }] }</pre>

■ 조회 가능시간 아님 오류 안내

요청 전일을 포함한 수수료 조회 시 전일의 수수료집계 처리가 완료되지 않았을 경우, "조회 가능 시간 아님(A0020)"으로 거부 응답하며 수수료집계 처리 완료 후 정상적으로 조회할 수 있습니다.

2.6.3. 집계조회 API

■ 설 명

이용기관이 사용한 API에 대한 집계를 조회합니다. 단, 요청 당일 기준으로 1년 이내의 거래에 대해서만 확인이 가능하며 요청 당일은 조회할 수 없습니다. API가 적용되는 시점 이후로 조회 가능합니다.(2019.12.07.)

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/manage/count
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>			오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = oob, sa]
Parameter	inquiry_date	"20171001"	Y	N(8)	조회일자

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	res_cnt	"8"	N(5)	조회대상 API 개수
	res_list	<object>		API 목록
	--api_manage_id	"001"	AN(3)	API 관리 ID 001:잔액조회 002:거래내역조회 003:입금이체** 004:입금이체(계좌소유인증) 005:출금이체 006:계좌실명조회 007:송금인정보조회 008:수취조회
	--normal_count	"10000000"	N(12)	API 정상건수
	--normal_tran_amt	"0"	N(18)	API 정상이체금액

* 입금이체(계좌소유인증) 외 일반적인 입금이체금액의 합계

** 다건 입금이체의 경우, 정상 건수는 입금 건별로 집계됨

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "res_cnt": "7", "res_list": [{ "api_manage_id": "001", "normal_count": "10000000", "normal_tran_amt": "0" }, { ... }, ... { ... }] }</pre>

■ 조회 가능시간 아님 오류 안내

요청 전일의 집계 조회 시 전일의 집계 처리가 완료되지 않았을 경우, "조회 가능시간 아님 (A0020)"로 거부 응답하며 집계 처리 완료 후 정상적으로 조회할 수 있습니다.

2.6.4. 출금이체한도조회 API

■ 설 명

이용기관의 출금이체한도를 실시간으로 조회합니다. 요청 당일의 출금이체한도만 조회할 수 있습니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/manage/wd_limit
HTTP Method	GET

■ 요청 메시지 명세

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

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	res_cnt	"18"	N(5)	참가은행개수
	res_list	<object>		출금이체한도 목록
	--bank_code_std	"097"	AN(3)	참가은행.표준코드*
	--day_wd_limit_amt	"1000000000"	N(18)	일간 출금이체한도금액
	--day_wd_amt	"850000"	N(18)	당일 누적 출금이체금액
	--perc_wd_limit_amt	"2500000"	N(18)	건당 출금이체한도금액

* 은행별로 출금한도를 계약하지 않은 이용기관은 "000" (통합)으로 셋팅되어 응답함

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "res_cnt": "18", "res_list": [{ "bank_code_std": "097", "day_wd_limit_amt": "1000000000", "day_wd_amt": "850000", "perc_wd_limit_amt": "2500000" }, { ... }, ... { ... }] }</pre>

2.6.5. 이상금융거래 탐지내역 조회 API (자체인증 이용기관만 이용가능)

■ 설 명

이용기관과 관련된 이상금융거래 탐지내역을 조회합니다. 단, 요청 당일 기준으로 1개월 이내의 내역에 대해서만 확인이 가능하며 요청 일시 기준으로 1분 이전까지만 탐지내역 조회가 가능합니다. 최대 조회기간은 10분입니다. 개인정보보호 목적으로 자체인증 이용기관만 이용가능하며, 센터인증기관은 보안 이메일로 이상금융거래 탐지내역이 통지됩니다.

■ 요청 메시지 URL

HTTP URL	https://openapi.openbanking.or.kr/v2.0/inquiry/fds_detect
HTTP Method	GET

■ 요청 메시지 명세

HTTP	항목	값	필수	TYPE (길이)	설명
Header	Authorization	Bearer <access_token>			오픈뱅킹으로부터 전송받은 Access Token 을 HTTP Header 에 추가 [scope = sa]
Parameter	inquiry_date	"20200101"	Y	N(8)	조회일자
	from_time	"100000"	Y	N(6)	조회시작시간
	to_time	"100059"	Y	N(6)	조회종료시간

■ 응답 메시지 명세

HTTP	항목	값	TYPE (길이)	설명
Body	api_tran_id	"2ffd133a-d17a-431d-a6a5"	aNS(40)	거래고유번호(API)
	api_tran_dtm	"20190910101921567"	N(17)	거래일시(밀리세컨드)
	rsp_code	"A0000"	AN(5)	응답코드(API)
	rsp_message	""	AH(300)	응답메시지(API)
	cnt_exceed_yn	"N"	A(1)	탐지내역 건수 초과여부 ^{주 1)}
	res_cnt	"3"	N(3)	이상금융거래 탐지 건수 ^{주 2)}
	res_list	<object>		탐지 목록
	--detect_time	"20200101100130"	N(14)	탐지 시간
	--policy_rule	"동일 사용자의 지속적인 출금이체 요청 발생"	AH(300)	이상금융거래 탐지 정책을
	--user_seq_no	"1123456789"	AN(10)	사용자일련번호 ^{주 3)}
	--bank_code	"097"	AN(3)	참가은행.표준코드 ^{주 3)}
	--account_num	"152136512388"	AN(16)	계좌번호 ^{주 3)}
	--fintech_use_num	"123456789012345678901234"	AN(24)	핀테크이용번호 ^{주 3)}

	--tran_amt	"100000"	N(12)	거래금액 ^{주 3)}
--	------------	----------	-------	----------------------

주1) 조회범위 내에 탐지내역 건수가 API 응답건수(현재 100건)를 초과할 경우 셋팅되며, 이 경우 이용기관은 조회범위를 작게 하여 다시 요청하여야 합니다.

주2) 탐지내역은 탐지시간 기준으로 오름차순으로 정렬하여 응답하며, 탐지 건수가 100건이 초과 되면 정렬순으로 처음 100건만 응답합니다.

주3) 탐지된 내용의 정책룰에 따라 항목이 셋팅됨

■ 응답 메시지 형태

JSON
<pre>{ "api_tran_id": "2ffd133a-d17a-431d-a6a5", "api_tran_dtm": "20190910101921567", "rsp_code": "A0000", "rsp_message": "", "cnt_exceed_yn": "N", "res_cnt": "3", "res_list": [{ "detect_time": "20200101100130", "policy_rule": "동일 사용자의 지속적인 출금이체 요청 발생", "user_seq_no": "1123456789", "bank_code": "097", "account_num": "152136512388", "fintech_use_num": "123456789012345678901234", "tran_amt": "100000" }, { ... }, ... { ... }] }</pre>

별첨

별첨1) 데이터 Type 설명

약자	설명
N	NUMERIC
S	SPECIAL
AH	한글 (KSC-5601 코드 범위 내의 UTF-8 인코딩 문자열) 예) '각', '샷' 등 KSC-5601 코드 범위 외의 한글 입력시 오류 응답 * KSC-5601 코드 범위 (0xA1A1 ~ 0xFEFE) ** AH의 길이는 KSC-5601 2 Byte 완성형 코드로 변환 후의 Byte 길이를 의미함
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
B64	BASE-64 인코딩 포맷
E	이메일 주소(local-part@domain) - 대문자 : A-Z - 소문자 : a-z - 숫자 : 0-9 - @ - 특수문자 • !#\$%&'*+,-/=/?^_`{ }~;. • "(),;,<>@[W] (local-part에서는 인용부호("") 안에서만 유효) • SPACE(공백) (local-part에서는 인용부호("") 안에서만 유효)
A	ALPHABETIC * 단, 반드시 대문자로 설정
AN	ALPHA-NUMERIC * 단, 알파벳(ALPHABET)은 반드시 대문자로 설정
ANS	ALPHA-NUMERIC SPECIAL (AN + '-'(Hyphen) + Space) * 단, 알파벳(ALPHABET)은 반드시 대문자로 설정

※ 데이터 Type의 길이는 Byte 길이를 의미함

별첨2) 용어 설명

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

별첨3) 은행코드

은행코드(3 자리)	은행명(축약)	은행명(전체)	은행코드(3 자리)	은행명(축약)	은행명(전체)
002	산업	KDB 산업은행	032	부산	부산은행
003	기업	IBK 기업은행	034	광주	광주은행
004	국민	KB 국민은행	035	제주	제주은행
007	수협	수협은행	037	전북	전북은행
011	농협	NH 농협은행	039	경남	경남은행
020	우리	우리은행	081	KEB 하나	KEB 하나은행
023	SC 제일	SC 제일은행	088	신한	신한은행
027	씨티	한국씨티은행	089	K BANK	케이뱅크
031	대구	대구은행	090	카뱅	카카오뱅크
			097	오픈 ^{주 1)}	오픈은행
※ 전체은행코드 참조 : 금융결제원 홈페이지(www.kftc.or.kr) 접속 > "금융회사 코드조회"(우측 하단) 클릭 > "금융회사 공동코드" 파일(HWP/PDF) 다운로드					

주1) 오픈은행은 테스트를 위한 가상의 은행임(운영 시스템에는 미존재)

■ 복수기관코드 사용기관 및 코드

기관명	복수기관 코드	대표기관 코드
기업은행	003, 043	003
국민은행	004, 006, 019, 078, 079	004
수협은행	007, 009	007
농협은행	010, 011, 016	011
우리은행	020, 022, 024, 083, 084	020
한국씨티은행	027, 053	027
KEB하나은행	005, 025, 033, 080, 081, 082	081
신한은행	088, 021, 026	088

* 점별코드에는 복수기관코드가 들어갈 수 있습니다.

점별코드 형식 : 기관코드(3) + 지점코드(4)

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.
O0011	허용되지 않은 Scope 입니다.
O0012	중계센터 처리지연 (잠시 후 재거래 요망)
O0013	이용기관에서 과도하게 API 를 호출하였습니다.
O0014	Refresh Token 거부
O0015	Refresh Token 만료

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

error_code	에러내용
119	인증에 필요한 헤더 값이 존재하지 않을 때
801	사용자일련번호와 token 의 사용자가 맞지 않을 때
803	인증요청 시 사용자 세션 값과 인증코드발급 시 사용자 세션 값이 맞지 않을 때
804	사용자가 인증 페이지에서 동의하지 않을 때
992	Http Header 에 Authorization Bearer access token 이 없을 때
3000103	인증에 필요한 필수 파라미터 값이 존재하지 않을 때 or 특정 파라미터 값이 중복으로 들어왔을 때
xxxx201	client_id 가 존재하지 않을 때 or key/secret 이 맞지 않을 때 or 비활성화되어 사용할 수 없을 때
xxxx113	유효하지 않은 authorization code (만료 or 기사용)
3000114	등록한 callback url 과 요청한 callback url 이 맞지 않을 때
3000115	scope 이 맞지 않을 때(기본 제공하는 scope 명[oob login inquiry transfer sa]이 맞지 않을 때)
3000116	response_type 유형이 맞지 않을 때
3000117	grant_type 이 맞지 않을 때
3002110	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, oob, sa 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(이체결과조회 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	내부 전문 송신 실패
A0017	참가은행 응답전문 TIMEOUT
A0018	거래내역 없음
A0019	사용자탈퇴 처리중인 계좌
A0020	조회 가능시간 아님
A0021	오픈뱅킹센터 지정 에러메시지 ※ "응답메시지(API)" 항목에 자세한 내용이 설정되며, 이용기관은 반드시 해당 내용을 사용자에게 표시해야함)
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
A0322	미등록된 이용기관 약정 계좌/계정
A0323	이용기관에 등록된 사용자 계좌 아님
A0324	기등록된 조회서비스용 사용자 계좌
A0325	기등록된 출금서비스용 사용자 계좌
A0326	은행거래고유번호 중복 ※ 본 응답코드는 이용기관에서 잘못하여 은행거래고유번호를 중복 생성한 경우에 발생하므로 이용기관은 반드시 원인을 파악하여 조치 필요
A0327	출금한도 미협의 은행 (이용기관 문의)

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, 잡좌통할, 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	○
481	가상계좌 처리 불가	○
482	예적금계좌 처리 불가	○
483	수익증권계좌 처리 불가	○

	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)	○
	555	해당 사용자 없음	○
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 자금반환결과조회	701	조회 대상거래 없음	×
자금반환청구 자금반환결과조회	900	청구 접수 처리 중	×
	901	기 반환 거래	×
	902	원거래 없음	○
	903	거래금액 상위	○
	904	처리대행비용 상위	○
	905	거래구분코드 상위	○
	906	수취 계좌번호 상위	○
	907	기타사항 상위	○
	908	자금청구 접수 내역 존재	×
	909	자금청구 접수 전 자금청구 확인 수신으로 거부	×
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	내부 전문 송신 실패	×
	820	농협중앙회(단위농축협) 계좌번호를 입력했을 경우 거절처리	×
	821	금융(거래)정보 제 3 자제공동의 만료	×
	822	은행거래고유번호 중복 ※ 본 응답코드는 이용기관에서 잘못하여 은행거래고유번호를 중복 생성한 경우에 발생하므로 이용기관은 반드시 원인을 파악하여 조치 필요	×

별첨9) 자체 출금동의 방식의 안전한 구현을 위한 금융보안원 권고사항

■ 유의사항

자체 출금동의 방식* 구현 시, 공격자가 고객 개인정보 및 단말을 획득하여 고객 대신 서비스 가입 및 출금동의를 시도하는 공격에 적절히 대응할 수 있도록 안전한 출금동의 방식 구현 권고

* 이용기관이 고객으로부터 직접 동의를 획득하여 고객의 자금을 출금

공격 시나리오 1	지인의 단말을 빌린 후 무단으로 이체
① (사전 요건) 공격자 A씨는 지인인 B씨의 생년월일/성별과 은행 계좌번호를 사전에 획득 ② (서비스 가입) A씨는 B씨의 단말(스마트폰)을 빌려 간편송금 애플리케이션을 설치하고 B씨의 명의로 간편송금 서비스 가입 ③ (무단 출금동의) A씨는 B씨의 단말을 점유하고 있으므로, 통신사 본인확인 및 ARS 인증을 통과하고, B씨의 계좌에 대하여 무단으로 출금동의 ④ (부정 이체) A씨는 B씨의 단말을 점유하고 있는 동안 B씨의 계좌에서 A씨 등의 계좌로 무단으로 부정 이체	

공격 시나리오 2	대포폰을 통한 무단 출금
① (사전 요건) 공격자 C씨는 공격대상인 D씨의 신분증 정보(주민등록번호, 발급일자), 은행 계좌번호를 사전에 획득 ② (서비스 가입) C씨는 D씨의 명의로 대포폰을 만들어 간편송금 애플리케이션을 설치하고 D씨의 명의로 간편송금 서비스 가입 ③ (무단 출금동의) C씨는 D씨 명의로 개설된 단말을 보유하고 있으므로, 통신사 본인확인 및 ARS 인증을 통과하고, D씨의 계좌에 대하여 무단으로 출금동의 ④ (부정 이체) C씨는 D씨 명의로 개설된 단말을 보유하고 있으므로, 지속적으로 D씨의 계좌에서 C씨 등의 계좌로 무단으로 부정 이체	

■ 대책 예시

예시	추가 인증 적용
고객 개인정보 및 단말을 획득하더라도 무단 출금이체 등록이 가능하지 않도록, 통신사 본인확인 및 ARS인증 외 추가적인 인증 절차* 적용 * 예) 출금계좌에 소액이체로 전송한 일회용 인증번호 확인, 단말기 화면 잠금 해제 인증(패턴, 홍채, 지문 등), 단말기 운영체제에 등록된 계정 정보 (google계정 등)를 이용한 인증, 출금계좌 은행에 직접 인증 등	

별첨10) API 관리 ID

API 관리 ID(3 자리)	API 명	API 관리 ID(3 자리)	API 명
001	잔액조회	005	출금이체
002	거래내역조회	006	계좌실명조회
003	입금이체	007	송금인정보조회
004	입금이체(계좌소유인증)	008	수취조회