PASS인증서 연동 규격서

- 이용기관 / 대행사 용 -

2023.03

목차

2. 인증서 발급 에러(1xxx)
3. 인증서 삭제 에러(2xxx)
4. 인증서 알림내용 등록 에러(3xxx)
5. 검증 요청 에러(4xxx)
6. 인증서 알림내용 상세 조회 에러(5xxx)
7. 인증 처리상태 조회 에러(6xxx)
8. 인증요청 거절(취소) 에러(7xxx)

4. 코드 정의

1. API 규격

PASS인증서 전자서명을 위해서는 TLS 1.2 이상 적용이 필수입니다.

1.1 인증요청 API

(이용기관/대행사/체험서비스 ▶ PASS인증서 서비스 중계서버)

사전에 담당자에게 접근 토큰(Access Token)을 전달받아야 합니다.

URL

URL	PROTOCOL	METHOD	Content-Type	DESCRIPTION
/v1/certification/notice	HTTPS	POST	Application/json;charset=utf8	

REQUEST HEADER

PARAMETER	TYPE	LENGTH	REQUIRED	DESCRIPTION
Authorization	String	20	M	- 전달받은 접근토큰(Access Token) 값을 적용 - ex) Authorization: Bearer **{Access Token}**

REQUEST BODY (REQUIRED - M : Mandatory / O : Optional)

PARAMETER	TYPE	LENGTH	REQUIRED	DESCRIPTION
companyCd	String	5	M	- 이용기관 코드
reqTxId	String	20	M	- 이용기관 또는 대행사에서 생성한 트랜잭션 ID **- 반드시 20자리의 값으로 요청 (특수문자 사용 불가)**
serviceTycd	String	5	M	- 인증서 서비스 유형 코드 - S1001 : 간편증빙 - S1002 : 간편날인 - S1003 : 간편통지 - S2001 : 출금이체동의 - S3001 : 간편로그인 - S3002 : 간편인증

telcoTycd	String	1	O	- 통신사 구분 코드 - S: SKT - K: KT - L: LGU+
phoneNo	String	40	M	- 휴대폰 번호 - AES128 또는 AES256으로 암호화
userNm	String	300	M	- 사용자의 이름 - AES128 또는 AES256으로 암호화
birthday	String	40	M	- 생년월일로 “YYMMDD” 형식을 사용 - AES128 또는 AES256으로 암호화
gender	String	40	M	- 성별로 0~9의 값을 가짐 - AES128 또는 AES256으로 암호화 - 9: 1800 ~ 1899년에 태어난 남성 - 0: 1800 ~ 1899년에 태어난 여성 - 1: 1900 ~ 1999년에 태어난 남성 - 2: 1900 ~ 1999년에 태어난 여성 - 3: 2000 ~ 2099년에 태어난 남성 - 4: 2000 ~ 2099년에 태어난 여성 - 5: 1900 ~ 1999년에 태어난 외국인 남성 - 6: 1900 ~ 1999년에 태어난 외국인 여성 - 7: 2000 ~ 2099년에 태어난 외국인 남성 - 8: 2000 ~ 2099년에 태어난 외국인 여성
reqTitle	String	50	M	- 인증요청 알림 제목

reqCSPhoneNo	String	12	M	- 인증을 요청하는 이용기관의 고객센터 연락처
reqEndDttm	String	20	M	- 인증요청의 유효 만료일시 “YYYY-MM-DD hh:mi:ss” 형식사용
signTargetTycd	String	1	M	- 서명대상 유형 코드 - 1 : 서명대상이 원문 PlainText 인 경우 - 2: 서명대상이 원본Hash인 경우 - 3: 서명대상이 원본URL인 경우 - 4: 서명대상이 nonce인 경우(serviceTycd값이 S3001, S3002인 경우 사용)
signTarget	String	500,000	M	- 서명대상 정보 - 최대 500,000자리(약 1MB)의 값을 가짐 - 서명대상 유형 코드(signTargetTycd)값이 1번(원문), 3번(원본URL)인 경우 AES128 또는 AES256으로 암호화 - 인증서 서비스 유형 코드(serviceTycd)값이 S3001(간편로그인 서비스), S3002(간편인증 서비스)인 경우 이용기관/대행사에서 1회용으로 생성한 nonce값(재사용 불가)을 사용 - 인증서 서비스 유형 코드(serviceTycd)값이 S2001(출금이체동의 서비스)인 경우, 서명대상정보에 다음과 같은 출금관련 정보가 포함되어 있어야함. **- 오픈뱅킹 출금동의인 경우 : 이름, 금융기관명, 계좌번호**

				- 자동이체출금동의(CMS출금동의)인 경우 : 이름, 금융기관명, 계좌번호, 금액
isUserAgreement	String	1	O	- 사용자 동의 필요 여부 **- 기본값 : "N"** - Y : 통신사 PASS 앱에 노출하여 사용자 동의를 받고자 하는 경우 - N : 통신사 PASS 앱에 노출할 필요 없는 경우.
originalInfo	jsonObject		O	- 서명대상 원본 정보 **- serviceTycd값 또는 signTargetTycd에 따라 필수, 옵션 구분** **- 필수값인 경우 :** serviceTycd가 S1001, S1003, S2001 이고 signTargetTycd 2 또는 3 인 경우 **- 아래의 경우 사용되지 않음** - serviceTycd값 : S1002, S3001, S3002
isDigitalSign	String	1	O	- 인증서 서비스 결과 알림 수신 시 전자서명값 포함 여부 **- 기본값: “Y”** - Y : 서명 완료 후 전자서명값 수신 - N : 서명 완료 후 전자서명값 수신하지 않음

> **참고**: jsonObject 의 세부항목은 아래를 참고하시기 바랍니다.

```
originalInfo : {
    originalTycd : {
        "type": "string",
        "length" : 2,
        "description":  AG: Agreement(동의서)
                        AP: Application(신청서)
                        CT: Contract(계약서)
                        GD: Guide(안내서)
                        NT: Notice(통지서)
                        TR: Terms(약관)
    },
    originalURL : {
        "type": "string",
        "length" : 100,
        "description": 반드시 SSL이 적용된 https여야함(안드로이드, iOS 정책임)
    },
    originalFormatCd : {
        "type": "string",
        "length" : 1,
        "description":  1: Plain Text
                        2: HTML
                        3: Download Image
                        4: Download Document
    }
}
```

REQUEST SAMPLE (JSON)

```
Authorization: Bearer {Access Token}
{
    “companyCd”: “이용기관코드”,
    “serviceTycd”: “S1001”,
    "telcoTycd": "S",
    "phoneNo": “01012345678”,
    “userNm”: “홍길동”,
    “birthday”: “801031”,
    “gender”: “1”,
    "reqTitle": "인증요청 알림 제목",
    “reqCSPhoneNo”: “1833-1234”,
    “reqEndDttm”: “2018-12-31 23:59:59”,
    “isNotification”: “Y”,
    “isPASSVerify”: “Y”,
    “signTargetTycd”: “2”,
    “signTarget”: “1728FB69B522C169EAFE27E49B…",
    “isUserAgreement”: “N”,
    "originalInfo": {
        “originalTycd”: “AG”,
        “originalURL”: “https://example.com/example/agreement”,
        “originalFormatCd”: “4
    },
    “reqTxId”: “abcdefghij0123456789”,
    “isDigitalSign”:”Y”
}
```

RESPONSE BODY

PARAMETER	TYPE	LENGTH	REQUIRED	DESCRIPTION
reqTxId	String	20	M	- 요청시 전달받은 값을 사용
certTxId	String	20	M	- PASS인증서서비스 중계서버에서 생성한 트랜잭션 ID - PASS 앱 실행 시 전달할 경우 해당하는 트랜잭션의 인증 알림내용을 조회할 수 있으며, 전달하지 않을 경우 모든 인증 알림내용을 조회
telcoTxId	String	50	O	- 통신사 PASS 시스템에서 생성한 트랜잭션 ID - 통신사 암호화 키로 암호화된 값 - 요청 시 isNotification = “N”으로 보낼경우 전달

RESPONSE SAMPLE (JSON)

```
HTTP 200 OK
{
    “reqTxId”: “abcdefghij0123456789”,
    “certTxId”: “1234567890klmnopqrst”
    “telcoTxId”: “암호화된 데이터”
}
```

3.2 검증 결과 요청

(이용기관/대행사 ▶ PASS인증서 서비스 중계서버)

URL

URL	PROTOCOL	METHOD	Content-Type	DESCRIPTION
/certification/result	HTTPS	POST	Application/json;charset=utf8	

REQUEST BODY

PARAMETER	TYPE	LENGTH	REQUIRED	DESCRIPTION
companyCd	String	5	M	- 인증 알림내용 등록을 요청하는 이용기관 코드로 PASS인증서서비스 중계서버에서 발급한 이용기관 코드를 사용함
reqTxId	String	20	M	- 인증 알림내용 등록 요청시 전달했던 이용기관 또는 대행사에서 생성한 트랜잭션 ID
certTxId	String	20	M	- 인증 알림내용 등록 요청의 응답으로 전달받은 PASS인증서서비스 중계서버에서 생성한 트랜잭션 ID
phoneNo	String	40	M	- 사용자 휴대폰번호 - AES128 또는 AES256으로 암호화
userNm	String	300	M	- 사용자 이름 - AES128 또는 AES256으로 암호화

REQUEST SAMPLE (JSON)

```
Authorization: Bearer {Access Token}
{
    “companyCd”: “이용기관코드”,
    “reqTxId”: “abcdefghij1234567890”,
    “certTxId”: “1234567890klmnopqrst”,
    "phoneNo": “01012345678”,
    “userNm”: “홍길동”
}
```

RESPONSE BODY

PARAMETER	TYPE	LENGTH	REQUIRED	DESCRIPTION
reqTxId	String	20	M	- 요청시 전달받은 값을 사용
telcoTxId	String	20	O	- 통신사 PASS 시스템에서 생성한 트랜잭션 ID - 서명 대기 시 전송되지 않음
certTxId	String	20	M	- PASS인증서서비스 중계서버에서 생성한 트랜잭션 ID
resultTycd	String	1	M	- 서비스 알림 유형 코드 (PASS인증서서비스에 등록된 PASS 인증서 가입 정보 기준으로 응답) 1: 인증 완료
resultDttm	String	20	O	- 결과 일시로 resultTycd에 해당하는 일시정보가 전달되며 “YYYY-MM-DD hh:mi:ss” 형식을 사용 - 서명 대기 시 전송되지 않음
digitalSign	String	가변	O	- 전자서명 값 - 최대값은 DB상의 CLOB 용량 (1GB) - 인증 알림내용 등록 요청 시 isDigitalSign값이 “Y” 인 경우에만 값이 전달됨 - resultTycd값이 “1: 인증 완료”인 경우에만 값이 전달됨
CI	String	200	M	- 사용자 연계정보 식별값 (Connectiong Information) - resultTycd값이 “1: 인증 완료”인 경우 전달됨 - 이용기관의 암호화 키로 암호화된 CI (AES256)
userNm	String	300	M	- 사용자 이름 - AES128 또는 AES256으로 암호화
birthday	String	40	M	- 사용자의 생년월일로 “YYMMDD” 형식을 사용 - AES128 또는 AES256으로 암호화

gender	String	40	M	- 사용자의 성별로 0~9의 값을 가짐 - AES128 또는 AES256으로 암호화 - 9: 1800 ~ 1899년에 태어난 남성 - 0: 1800 ~ 1899년에 태어난 여성 - 1: 1900 ~ 1999년에 태어난 남성 - 2: 1900 ~ 1999년에 태어난 여성 - 3: 2000 ~ 2099년에 태어난 남성 - 4: 2000 ~ 2099년에 태어난 여성 - 5: 1900 ~ 1999년에 태어난 외국인 남성 - 6: 1900 ~ 1999년에 태어난 외국인 여성 - 7: 2000 ~ 2099년에 태어난 외국인 남성 - 8: 2000 ~ 2099년에 태어난 외국인 여성
phoneNo	String	40	M	- 요청자 휴대폰 번호 AES128 또는 AES256으로 암호화
telcoTycd	String	1	O	- 통신사 구분 코드 - resultTycd값이 “1: 인증 완료”인 경우에만 값이 전달됨 - S: SKT - K: KT - L: LGU+

RESPONSE SAMPLE (JSON)

```
{
    “reqTxId”: “abcdefghij1234567890”,
    “telcoTxId”: “abcdefghij0123456789”,
    “certTxId”: “1234567890klmnopqrst”,
    “resultTycd: “1”,
    "digitalSign": "전자서명 값",
    "CI": "ALKDJF17KHJ11AC1T...",
    “userNm”: “홍길동”,
    “birthday”: “801031”,
    “gender”: “1”
}
```

3.3 PASS인증서서비스 에러 응답 및 에러 코드

RESPONSE BODY

PARAMETER	TYPE	LENGTH	REQUIRED	DESCRIPTION
errorCd	Integer	4	M	- 에러 코드
errorMessage	String	100	M	- 에러 메세지
errorPointCd	String	5	M	- 에러 발생 지점 코드 - TLPAS: 통신사 PASS 서버 (TeLco PASS Server) - PACPR: PASS인증서서비스 요청 관리(PASS Certification Platform Request manamgent) - UOSYS: 이용기관 시스템(Using Organization SYStem) - AGSYS: 대행사 시스템(AGency SYStem)
telcoTxId	String	20	O	- 요청시 전달받은 값을 사용
reqTxid	String	20	O	- 요청시 전달받은 값을 사용
certTxId	String	20	O	- 요청시 전달받은 값을 사용

RESPONSE SAMPLE (JSON)

```
HTTP 400/500 Bad Request/Internal Server Error
{
    “errorCd”: xxxx,
    “errorMessage”: “필수 요청함목이 누락되었습니다.”,
    “errorPointCd”: “PACPM”,
    “telcoTxId”: “abcdefghij0123456789”,
    “reqTxId”: “abcdefghij1234567890”,
    “certTxId”: “1234567890klmnopqrst”
}
```

o 공통 에러(9xxx)

에러 코드	에러 메시지	HTTP Status Cd
9000	권한이 없습니다	401 Unauthorized
9001	요청 Body가 없습니다.	400 Bad Request
9002	요청 Body 형식이 잘못되었습니다.	400 Bad Request
9003	지원하지 않는 HTTP Method입니다.	400 Bad Request
9004	일시적인 오류가 발생했습니다. 담당자에게 문의하세요.	400 Bad Request
9099	일시적인 오류가 발생했습니다. 잠시 후 다시 요청해 주세요	500 Internal Server Error

o 인증 알림내용 등록 에러(3xxx)

에러 코드	에러 메시지	HTTP Status Cd
3101	필수항목 {요청항목}이 누락되었습니다.	400 Bad Request
3102	{요청항목} 값이 유효하지 않습니다.	400 Bad Request
3103	존재하지 않는 사용자입니다.	400 Bad Request
3199	일시적인 오류가 발생했습니다. 잠시 후 다시 요청해 주세요.	500 Internal Server Error

o 인증 처리상태 조회 에러(4xxx)

에러 코드	에러 메시지	HTTP Status Cd
4101	필수항목 {요청항목}이 누락되었습니다.	400 Bad Request
4102	{요청항목} 값이 유효하지 않습니다.	400 Bad Request
4103	인증 요청 유효기간이 만료되었거나, 일치하는 데이터가 없습니다.	400 Bad Request
4199	일시적인 오류가 발생했습니다. 잠시 후 다시 요청해 주세요.	500 Internal Server Error

3.4 코드 정의

CODE TYPE	PARAMETER	LENGTH	DESCRIPTION
운영 체제(OS) 유형 코드	deviceOsTycd	1	A - Android I - IOS
통신사 구분 코드	telcoTycd	1	S - SKT K - KT L - LGT+
인증서 서비스 유형 코드	serviceTycd	5	S1001 - 간편증빙 S1002 - 간편날인 S1003 - 간편통지 S2001 - 출금이체동의 S3001 - 간편로그인 S3002 - 간편인증
서명대상 유형 코드	signTargetTycd	1	1 - 서명대상이 원문 Plain Text인 경우 2 - 서명대상이 원본Hash인 경우 3 - 서명대상이 원본URL인 경우 4 - 서명대상이 nonce인 경우 (serviceTycd값이 S3001, S3002인 경우 사용) 5 - 서명대상이 원문 HTML인 경우
원본 유형 코드	originalTycd	2	AG - Agreement(동의서) AP - Application(신청서) CT - Contract(계약서) GD - Guide(안내서) NT - Notice(통지서) TR - Terms(약관)
원본 형태 유형 코드	originalFormatCd	1	1 - Plain Text 2 - HTML 3 - Download Image 4 - Download Document
인증 처리 상태 코드	statusCd	1	W - Waiting(대기중) V - Viewed(조회완료) C - Complete(인증처리 완료) R - Reject(인증요청 거절(취소)) F - 서명검증 실패 F01 - 인증서 유효성 검증 실패 F02 - 폐기된 인증서

			F03 - 만료된 인증서 F04 - 인증내역이 존재하지 않음
이용기관 코드	companyCd	5	담당자에게 문의 요망

암호화 샘플 소스 (AES-128 or AES-256)

```java
import org.apache.commons.codec.binary.Base64;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AESCipher {
    // 알고리즘/모드/패딩
    private static final String algorithm = "AES/CBC/PKCS5Padding";
    // 암호화 키
    private SecretKey secretKey;
    // 초기화 벡터
    private IvParameterSpec iv;
    // 문자인코딩 방식
    private final String charset = "UTF-8";

    public AESCipher(String aesKey) {
        if(aesKey == null){
            throw new NoSecretKeyException("No SecretKey, Please Set Se
        }

        if(aesKey.length() > 16) {
            this.iv = new IvParameterSpec(aesKey.substring(0, 16).getBy
        } else {
            this.iv = new IvParameterSpec(aesKey.getBytes());
        }

        this.secretKey = new SecretKeySpec(aesKey.getBytes(), "AES");
    }

    // 암호화
    public String encrypt(String str) throws Exception {
        Cipher c = Cipher.getInstance(algorithm);
        c.init(Cipher.ENCRYPT_MODE, this.secretKey, this.iv);
        return new String(Base64.encodeBase64(c.doFinal(str.getBytes(ch
    }

    // 복호화
    public String decrypt(String str) throws Exception {
        Cipher c = Cipher.getInstance(algorithm);
        c.init(Cipher.DECRYPT_MODE, this.secretKey, this.iv);
        return new String(c.doFinal(Base64.decodeBase64(str.getBytes())
    }
}
class NoSecretKeyException extends RuntimeException {
    public NoSecretKeyException(String msg) {
```

```
47            super(msg);
48        }
49    }
```