

공인전자문서중계자와 송신자 간 표준연계 기술규격

v1.0

2026년 5월

목 차

1. 개 요	3
1.1. 목적	3
1.2. 적용대상 및 범위	3
2. 용어 정리	4
3. 중계자와 송신시스템 간 연계 방안	6
3.1. 개요	6
3.2. 단계별 상세 연계 절차	7
3.2.1. 이용자(송신자) 인증 단계	7
3.2.2. 중계자 인증 단계	8
3.2.3. 전자문서 발송·열람 단계	10
3.2.5. 유통증명서 발급 단계	13
3.2.6. 원본증명서 발급 단계	15
4. 연계 인터페이스(API)	16
4.1. API 개요	16
4.2. API 공통사항	17
4.2.1. HTTP 헤더	17
4.2.2. 요청 및 응답 메시지	18
4.2.3. 데이터 표기방안	21
4.3. 중계자와 송신자 시스템 간 연계 API 상세설명	21
4.3.1. 이용자 인증토큰 발급 및 갱신	21
4.3.2. 중계자 인증토큰 발급 및 갱신	23
4.3.3. 전자문서 발송요청	24
4.3.4. 전자문서 처리상태 전달	29
4.3.5. 전자문서 열람요청	32
4.3.6. 열람인증토큰 검증 요청	33
4.3.7. 전자문서 열람정보 전달	34
4.3.8. 전자문서 처리상태 정보 요청	36
4.3.9. 유통증명서 발급요청	38
4.3.10. 유통증명서 발급결과 전달	40
4.3.11. 유통증명서 발급결과 요청	41
4.3.12. 원본증명서 발급요청	43

4.3.13. 원본증명서 발급결과 전달 44

4.3.14. 원본증명서 발급결과 요청 45

A. 부록 - 이용자 인증토큰 48

B. 부록 - 오류코드 정의 50

<그림 차례>

[그림 1] 송신자와 중계자 시스템 간 단계별 연계 절차 6

[그림 2] 이용자 인증단계 절차 7

[그림 3] 중계자 인증단계 절차 9

[그림 4] 전자문서 발송·열람 단계 절차 11

[그림 5] 유통증명서 발급단계 절차 14

1. 개요

1.1. 목적

“공인전자문서중계자와 송신자 간 연계 인터페이스” 기술규격(이하 ‘본 규격’)은 공인전자주소 등록자 중 송신자(발송기관)들이 송신시스템을 이용해 중계자 서비스를 이용하는 때, 중계자 변경에 따른 송신자의 부담을 최소화하기 위한 목적으로 작성되었다.

이를 위해 본 규격에서는 송신자가 자체 송신시스템으로 공인전자문서중계자(이하 ‘중계자’) 시스템 간 연계를 통해 전자문서를 발송하고자 할 때 중계자 시스템과의 연계를 위해 필요한 인터페이스 표준안을 제시한다.

중계자와 송신시스템 간 연계 인터페이스는 송신자가 특정 중계자 시스템에 기술적으로 종속되지 않도록 함으로써 송신자의 중계서비스 선택권을 넓히고 궁극적으로 전자문서의 유통을 활성화하는 것을 목적으로 한다.

1.2. 적용대상 및 범위

본 규격은 중계자와 송신자들을 주 대상으로 하며, 특히 송신자의 경우 자체 송신시스템을 가진 자만을 그 대상으로 한다. 중계자가 제공하는 송신시스템을 이용하는 경우는 본 규격의 범위에 포함하지 않는다.

본 규격은 다음과 같은 기능 범위를 대상으로 한다.

- 송신자가 중계자에 송신자 정보를 등록, 인증, 갱신하는 기능
- 송신자가 중계자에게 전자문서 발송의 요청과 이에 대한 결과를 전달받는 기능
- 중계자가 송신자에게 열람시각을 전달하는 기능
- 송신자가 중계자에게 전자문서 발송 및 열람정보를 요청하고 이에 대한 결과를 전달받는 기능
- 송신자가 중계자에게 유통증명서 발급을 요청하고, 증명서를 전달받는 기능
- 송신자가 중계자에게 원본증명서 발급을 요청하고, 증명서를 전달받는 기능

2. 용어 정리

본 규격에 사용되는 용어는 다음과 같다. 아울러, 본 규격에서 정의되지 않은 용어는 “전자문서 및 전자거래기본법”에서 정의한 용어 정의를 따른다.

1. “연계”란 2개 이상의 프로그램이나 모듈을 상호 간 작용하도록 하는 것을 말한다.
2. “전자문서 유통허브시스템(이하 ‘유통허브시스템’)”이란 「전자문서 및 전자거래 기본법」(이하 ‘전자문서법’) 제22조에서 규정하고 있는 전담기관이 공인전자문서중계자 서비스 인프라를 위해 구축 및 운영하고 있는 시스템을 말한다.
3. “전담기관”이란 전자문서법 제22조에서 규정하고 있는 기관으로 공인전자문서중계자 제도를 관리하는 기관을 말한다.
4. “발송기관”이란 고지나 안내 등을 위해 전자고지서 등 전자문서를 발송하는 기관을 말한다.
5. “중계자 목록(Whitelist)”이란 전자문서법에 따라 인증받은 중계자들을 인식할 수 있는 정보들의 집합을 말한다.
6. “수신 동의”란 이용자가 특정 발송기관이 발송하는 전자문서를 이용약관 동의 등을 거쳐 수신하겠다는 전자적 의사표시를 말한다.
7. “송신중계자”란 전자문서의 송신자와 수신자가 각기 다른 중계자에 등록되어 있는 경우, 송신자가 등록되어 전자문서를 송신하는 역할을 하는 중계자를 말한다.
8. “수신중계자”란 전자문서의 송신자와 수신자가 각기 다른 중계자에 등록되어 있는 경우, 수신자가 등록되어 전자문서를 수신하는 역할을 하는 중계자를 말한다.
9. “인증토큰”이란 사용자가 자신의 신분을 증명한 뒤 서비스 제공자로부터 발급받은 토큰으로서 허가된 사용자임을 증명하기 위해 사용된다.
10. “열람인증토큰”이란 URL방식 전자문서 열람일 경우에 수신자가 수신중계자로부터 발급받는 인증토큰을 말하며 발송기관에 저장된 전자문서의 열람에 사용된다.
11. “중계자 정보”란 중계자 명칭, 중계자 인증 일시, 연락처 정보, 서비스 정보 등 이용자가 중계자를 파악할 수 있도록 하는 정보를 말한다.

12. “클라이언트 아이디”란 유통허브시스템에서 중계자에게 발급한 임의의 식별 값으로서 중계자 플랫폼 시스템의 인증을 위해 사용된다.
13. “클라이언트 비밀번호”란 유통허브시스템에서 클라이언트 아이디 발급 시 함께 발급하는 값으로서 중계자 플랫폼 인증을 위한 역할을 담당하는 임의의 문자열이다.
14. “전자문서 정보”란 전자문서 명칭, 전자문서 식별코드, 전자문서와 관련된 기관의 명칭, 전자문서의 주요 항목 등을 포괄하는 정보로, 이용자가 수신 동의할 때 참고할 수 있도록 제공되는 정보를 말한다.
15. “유통증명서”란 공인전자주소를 통하여 전자문서가 송신 또는 수신되거나 열람된 경우 전자문서법 제18조의5 제1항 각 호의 사항이 포함된 정보를 담은 증명서를 말한다.
16. “원본증명서”란 공인전자문서센터가 전자문서의 원본을 보관하고 있음을 증명하며, 해당 문서의 등록 시점, 내용의 무결성(해시값 등)을 보장하는 증명서를 말한다.
17. “전자증명서”란 전자문서의 유통 사실 및 원본 여부를 함께 증명하기 위하여 유통증명서와 원본증명서를 함께 제공하는 증명서를 말한다.

3. 중계자와 송신시스템 간 연계 방안

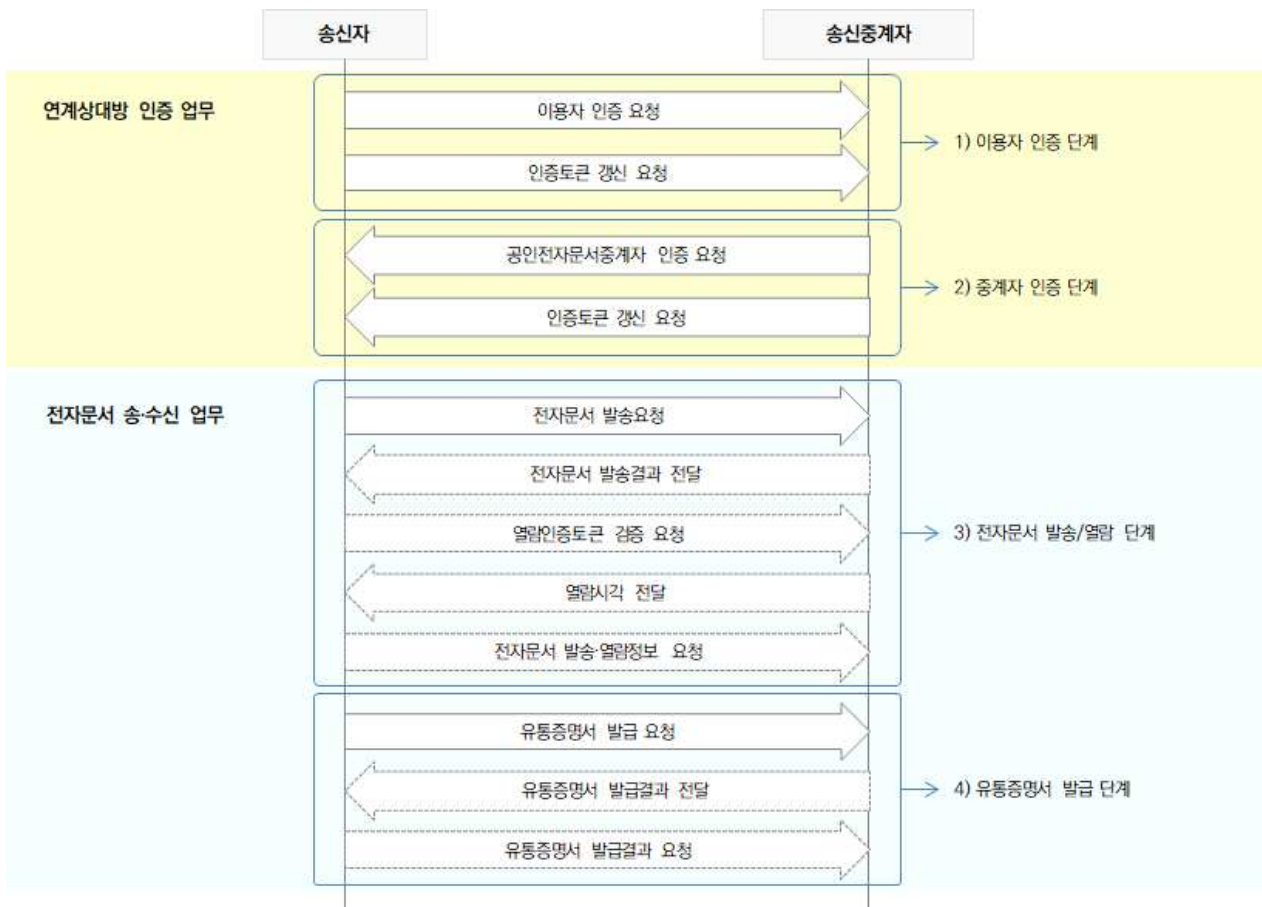
3.1. 개요

본 규격에서는 송신자가 중계자 시스템과 연계하여 기업·기관의 내부시스템에서 자동으로 전자문서를 발송하고자 하는 경우에 필요한 연계 인터페이스의 표준을 제시한다.

공인전자문서중계자의 서비스 이용자가 전자문서 송신자로서 중계자 시스템과 연계가 필요한 2가지 업무분야에서 4가지 단계로 구성된다.

첫 번째 업무에는 송신자와 송신중계자가 서로 상대방이 제공하는 서비스 인터페이스를 호출하기 위해 사용자 인증을 하는 “1) 이용자 인증단계”, “2) 중계자 인증단계”가 있다.

두 번째 업무인 전자문서 송·수신 업무는 “3) 전자문서 발송/열람 단계”와 “4) 유통증명서 발급단계”로 구성되며, 전체 업무별 구상단계는 다음 그림과 같다.



[그림 1] 송신자와 중계자 시스템 간 단계별 연계 절차

첫 번째 이용자와 중계자 간 상호인증업무는 전자문서 송·수신 업무에서 서로 상대방의 I/F를 호출하기 전에 반드시 선행되어야 한다.

3.2. 단계별 상세 연계 절차

송신자와 중계자 간 연계를 위한 업무의 각 단계별로 상세 절차를 제시하고 이 절차에 따라 연계를 하기 위해 필요한 표준 인터페이스를 도출한다.

각 단계를 수행하기 위한 처리절차는 각 단계의 처리 흐름도에 표기된 번호 순으로 진행된다. 각 절차 중 송신자와 송신중계자가 표준화된 인터페이스가 필수인 절차는 “①, ② ...”와 같이 표기하였고 표준화된 인터페이스가 필수가 아닌 절차는 “①, ② ...”로 구분하여 표기하였으며, 본 규약에서는 표준화가 필수인 인터페이스에 대해서만 정의한다.

3.2.1. 이용자(송신자) 인증 단계

이 단계는 송신자가 중계자의 서비스 연계를 위해 중계자가 제공하는 I/F를 호출하기 전 이용자가 가입된 정당한 회원임을 인증하고 인증토큰을 발급받는 단계로 중계자 서비스 I/F 호출 전 반드시 수행되어야 하는 단계이다.

송신자는 중계자로부터 전달받은 ID와 비밀번호를 이용하여 인증을 수행하고 중계자로부터 인증토큰을 발급받으며, 발급된 인증토큰의 유효기간이 만료되어 갱신이 필요한 경우 갱신된 인증토큰의 발급을 요청할 수 있다.

송신자는 중계자의 서비스 I/F 호출 시점에 발급된 인증토큰을 같이 전달함으로써 중계자가 인증요청자를 검증할 수 있도록 한다.

□ 처리 절차에 따른 흐름도



[그림 2] 이용자 인증단계 절차

□ 흐름에 따른 처리 상세내역

- 1) 송신자가 중계자와 연계하여 서비스를 이용하기 위해서는 먼저 중계자의 전자문서유통 서비스에 회원으로 가입하여야 한다. 회원 가입 시, 송신자는 서비스 이용 및 연계를 위한 인증정보를 중계자에 등록하거나 중계자로부터 연계를 위한 인증정보를 발급받는다.
- 2) 송신자는 회원가입 시에 등록한(또는 발급 받은) 인증정보를 기반으로 중계자 시스템에 연계하여 인증토큰 발급을 요청(❶ 사용자 인증토큰 발급 및 갱신 API)한다.
- 3) 송신자가 인증토큰을 발급받은 후 지속적으로 중계자와 연계가 필요한 경우, 송신자로부터 발급 받은 갱신토큰이 유효한 상태에서 인증토큰의 유효기간을 연장하기 위해 갱신을 요청(❷ 사용자 인증토큰 발급 및 갱신 API)한다.

□ 처리 절차에 따른 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
❶ 사용자 인증토큰 발급 (및 갱신)	<ul style="list-style-type: none"> ▪ 송신자가 중계자의 서비스를 이용하기 위해 서비스 요청 전 본인인증을 받는 API ▪ 송신자는 중계서비스 API 호출 시 본인인증 성공의 결과로 받은 인증토큰을 전달하여야 함 	송신자	송신 중계자	발급받은 인증토큰은 중계자의 서비스 API 호출 시, 메시지 헤더에 인자로 추가하여 전달
❷ 사용자 인증토큰 (발급 및) 갱신(※ 필요 시)	<ul style="list-style-type: none"> ▪ 인증토큰 발급 시 같이 받은 갱신토큰이 유효한 상태에서 기발급 받은 인증토큰의 유효기간 갱신을 요청하는 API ▪ 갱신요청을 받은 중계자는 요청 중계자의 인증토큰 검증 후 갱신된 인증토큰을 발급하여 이용자에게 전달함 	송신자	송신 중계자	“❶ 사용자 인증토큰 발급” 기능과 통합하여 “사용자 인증토큰 발급 및 갱신” API 하나로 제공함

3.2.2. 중계자 인증 단계

이 단계는 중계자가 이용자 수신동의에 대한 정보 관리 등 송신자가 제공하는 서비스를 연계하고자 할 때 인증토큰을 발급받는 단계로 송신자의 서비스 I/F 호출 전 반드시 수행되어야 한다.

중계자가 송신자로부터 서비스연계를 위한 인증정보를 먼저 전달 받은 후, 전달받은 인증정보를 기반으로 송신자로부터 인증토큰을 발급받는다. 또한 발급된 인증토큰의 유효기간이 만료되어 갱신이 필요한 경우 갱신된 인증토큰의 발급을 요청할 수 있다.

중계자는 송신자의 서비스 I/F 호출 시점에 발급된 인증토큰을 같이 전달함으로써 송신자가 인증요청자인 중계자를 검증할 수 있도록 한다.

□ 처리 절차에 따른 흐름도



[그림 3] 중계자 인증단계 절차

□ 흐름에 따른 처리 상세내역

- 1) 중계자는 송신자와 서비스 연계를 위한 협약을 진행한 후, 연계를 위한 인증정보를 발급받는다. 이 때 연계 인증정보를 발급받는 방법은 송신자 시스템의 연계 API 인증 관리 프로세스에 따라 오프라인 또는 온라인 방식을 활용한다.
- 2) 중계자는 송신자로부터 받은 인증정보를 기반으로 송신자 시스템에 연계하여 인증토큰 발급을 요청(① 중계자 인증토큰 발급 및 갱신 API)한다.
- 3) 중계자가 인증토큰을 받은 후 지속적으로 송신자와 연계가 필요한 경우에는 송신자로부터 발급 받은 갱신토큰이 유효한 상태에서 인증토큰의 유효기간을 연장하기 위해 갱신을 요청(② 중계자 인증토큰 발급 및 갱신 API)한다.

□ 처리 절차에 따른 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
① 중계자 인증토큰 발급 (및 갱신)	<ul style="list-style-type: none"> 중계자가 송신자의 서비스를 이용하기 위해 서비스 요청 전 본인인증을 받는 API 중계자는 송신자가 제공하는 서비스 API 호출 시 중계자 인증 성공의 결과로 받은 인증토큰을 전달하여야 함 	송신 중계자	송신자	발급받은 인증토큰은 송신자의 서비스 API 호출 시, 메시지 헤더에 인자로 추가하여 전달
② 중계자 인증토큰 (발급 및) 갱신(※ 필요 시)	<ul style="list-style-type: none"> 인증토큰 발급 시 같이 받은 갱신크인이 유효한 상태에서 기발급 받은 인증토큰의 유효기간 갱신을 요청하는 API 갱신요청을 받은 송신자는 요청 중계자의 인증토큰 검증 후 갱신된 인증토큰을 발급하여 이용자에게 전달함 	송신 중계자	송신자	“① 중계자 인증토큰 발급” 기능과 통합하여 “중계자 인증토큰 발급 및 갱신” API 하나로 제공함

3.2.3. 전자문서 발송·열람 단계

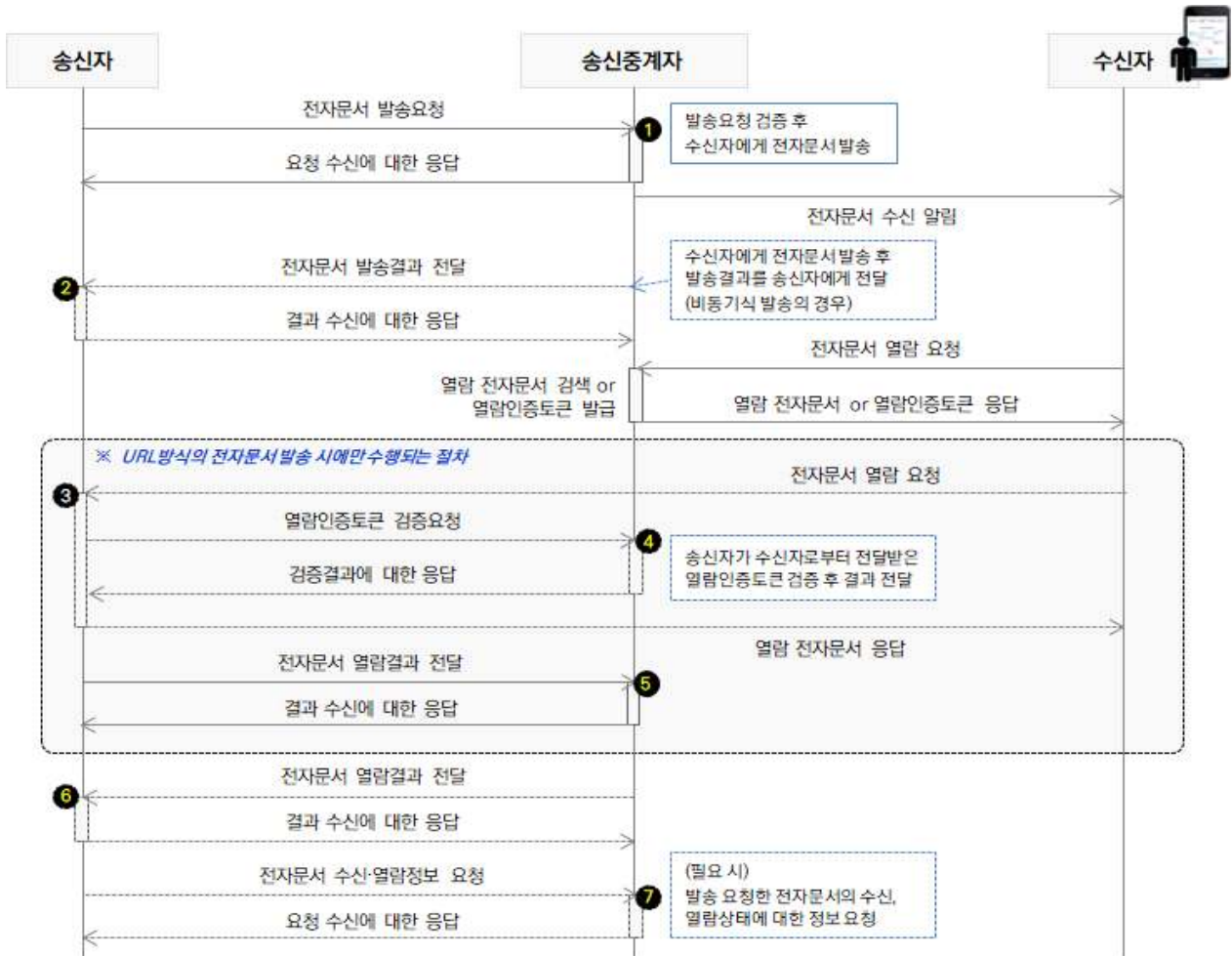
이 단계는 송신자가 내부 업무시스템과 중계자 시스템을 직접 연계하여 전자문서를 발송하고, 발송 상태를 확인하며, 수신자가 전자문서를 열람하도록 지원하는 단계이다.

송신자는 API를 통해 중계자에게 전자문서의 발송을 요청하고, 중계자를 통해 전자문서가 발송된 결과정보를 전달받는다. 또한 송신자가 전자문서를 “첨부방식”이 아닌 “URL방식”으로 송신하는 경우 수신자가 발급받은 열람인증토큰을 검증한 후 열람할 수 있도록 지원한다.

“URL방식”의 송신인 경우 수신자의 전자문서 열람 요청에 대해 송신자(전자문서 열람시스템)가 정상적으로 응답하고 나면, 송신자는 열람시각 정보를 포함한 열람정보를 반드시 송신중계자에게 전달하여야 한다. “첨부방식”의 경우에는 수신자가 전자문서를 열람하고 나면 송신중계자는 송신자에게 열람시각 정보를 포함한 열람정보를 전달할 수 있는데, 이 경우는 송신자와 송신중계자 간 협의에 의해 결정한다.

송신자 및 중계자 시스템의 상황에 따라 송신자 시스템이 중계자 시스템으로부터 전자문서 송·수신 및 열람과 관련된 정보를 수신하지 못하는 경우에는 송신자가 필요한 시점에 중계자에게 해당정보를 요청하여 확인할 수 있다.

□ 처리 절차에 따른 흐름도



[그림 4] 전자문서 발송·열람 단계 절차

□ 흐름에 따른 처리 상세내역

- 1) 송신자는 수신자 정보를 기반으로 송신증계자에게 전자문서의 발송(*)을 (“① 전자문서 발송요청 API” 호출)을 요청한다.

* 전자문서 발송 시 송신자는 전자문서 원문을 직접 첨부하여 증계자에게 전달하는 “첨부방식”과 원문을 열람할 수 있는 URL과 전자문서의 진본성을 확인할 수 있는 정보(전자문서의 hash값)를 증계자에게 전달하는 “URL방식” 중 하나의 방식을 선택하여 발송을 요청함

- 2) 송신증계자는 수신한 발송요청 메시지를 처리한 후, 이에 대한 처리결과를 송신자에게 응답메시지로 전달한다. 이 때 송신증계자는 수신자에게 전자문서를 전달하는 단계까지 완료한 후 그 결과를 송신자에게 응답으로 전달하거나, 발송요청을 검증하여 접수 처리한 후 접수 결과만을 송신자에게 응답으로 전달할 수 있다. 송신증계자가 어느 단계까지 처리하고 응답할지는 전자문서 발송요청 건수 및 증계자의 처리 능력에 따라 송신증계자가 결정한다.

- 3) "2)"절차 중 송신중계자가 수신자에게 전자문서를 전달한 후 수신에 대한 결과까지 송신자에게 돌려준 경우가 아니라면 송신자는 다음 2 방안 중 하나의 방법으로 송신결과를 획득하여야 한다.
 - 3-1 방안) 송신중계자가 수신자에게 전자문서를 송신하는 절차까지 완료한 후 그 처리결과를 수신자에게 전달("❷ 전자문서 처리상태 전달 API" 호출)
 - 3-2 방안) 송신자가 송신중계자에게 발송요청한 전자문서의 처리상태(수신 및 열람 상태)를 요청("❸ 전자문서 처리상태 요청 API" 호출)하여 획득
- 4) 전자문서를 수신한 수신자는 전자문서 열람을 위해 본인확인을 한 후, 그 결과에 따라 중계자에게 전자문서 열람을 요청한다. 요청을 받은 중계자는 수신한 전자문서가 "첨부방식"이면 열람 대상 전자문서를 검색한 후 열람문서를, "URL 방식"이면 열람인증토큰 발급 후 토큰을 수신자에게 전달한다.

※ 수신자가 송신자에게 원문열람 요청 시 수신자가 정당한 이용자임을 증명할 수 있도록 (수신) 중계자는 수신자에게 열람인증토큰을 발급(열람인증토큰의 구조는 "공인전자문서중계자 간 중계전송 표준연계 기술규격"의 "A - 부록. 열람인증토큰" 참조)하여야 한다.
- 5) 송신자가 "URL방식"으로 전자문서를 송신한 경우에는 다음 절차가 추가적으로 진행된다.
 - 5-1) 열람인증토큰을 받은 수신자는 전자문서 열람 URL에 인증토큰을 추가한 후, 전자문서 열람을 요청("❹ 전자문서 열람요청 API" 호출)함
 - 5-2) 열람 URL이 호출된 송신자는 열람 요청한 요청자가 중계자로부터 본인확인을 받은 정당한 수신자인지 검증을 위해 전달받은 열람인증토큰을 송신중계자에게 전달하여 검증("❺ 열람인증토큰 검증 API" 호출) 요청함
 - 5-3) 송신중계자는 열람인증토큰을 검증 후 검증결과를 송신자에게 응답하고, 응답을 받은 송신자는 검증이 유효한 경우에는 전자문서를 찾아서 수신자가 열람하도록 전달함
 - 5-4) 송신자는 수신자의 전자문서 열람요청에 대해 성공적으로 응답하고 나면, 열람시각을 포함한 열람정보를 송신중계자에게 전달("❻ 전자문서 열람정보 전달 API" 호출)함
- 6) 전자문서가 정상적으로 열람처리가 되면, 송신중계자는 송신자에게 수신자가 열람한 시각정보를 포함한 열람결과를 전달("❼ 전자문서 처리상태 전달 API" 호출)한다.
- 7) 발송 요청한 전자문서의 처리(수신, 열람 등)상태를 일정시간 수신하지 못한 경우 송신자는 송신중계자에게 해당 전자문서의 처리상태 검색을 요청("❼ 전자문서 처리상태 요청 API" 호출)한다.

□ 처리 절차별 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
❶ 전자문서 발송요청	<ul style="list-style-type: none"> ▪ 송신자가 중계자에게 전자문서의 발송을 요청하는 API ▪ 2건 이상의 전자문서 발송요청을 처리하도록 전자문서 묶음단위의 발송요청 구조 	송신자	송신중계자	

API 명	설명	요청자	응답자	비고
	를 가짐			
② 전자문서 처리(수신)상태 전달	<ul style="list-style-type: none"> 중계자가 전자문서를 송신한 후 수신자의 수신시각 정보 등을 포함한 발송 결과를 송신자에게 전달하는 API 	송신중계자	송신자	송신자가 ①의 응답으로 수신자의 수신상태에 대한 정보를 받지 못한 경우 사용
③ 전자문서 열람요청	<ul style="list-style-type: none"> 수신자가 URL방식으로 전달된 전자문서의 원문 열람을 요청하는 API 	수신자	송신자	
④ 열람인증토큰 검증요청	<ul style="list-style-type: none"> 수신자가 URL방식으로 전달된 전자문서의 원문을 열람하고자 송신자에게 전달한 열람인증토큰에 대해 송신자가 송신중계자에게 유효성 검증을 요청하는 API 	송신자	송신중계자	전자문서 발송 시 원문 열람을 위한 URL과 진본성 정보만을 전달한 경우에 원문 열람을 위해 사용
⑤ 전자문서 열람정보 전달	<ul style="list-style-type: none"> 송신자가 수신자의 전자문서 열람시각정보를 포함한 열람결과를 송신중계자에게 전달하는 API 	송신자	송신중계자	송신자가 “URL방식”으로 전자문서를 발송함으로써 열람서비스를 직접 제공하는 경우
⑥ 전자문서 처리(열람)상태 전달	<ul style="list-style-type: none"> 송신중계자가 수신자의 전자문서 열람시각정보를 포함한 열람결과를 송신자에게 전달하는 API 	송신중계자	송신자	전자문서 발송결과를 전달하기 위한 “② 전자문서 처리상태” API와 동일한 API 사용
⑦ 전자문서 처리상태 요청	<ul style="list-style-type: none"> 송신자가 송신중계자에게 본인이 발송 요청한 전자문서의 송신, 수신, 열람에 대한 상태와 시각정보를 요청하는 API 	송신자	송신중계자	송신자가 ②, ⑤ API를 제공하지 못하거나, 시스템 오류 등으로 인해 송신중계자로부터 결과를 전달받지 못하는 경우 사용

3.2.5. 유통증명서 발급 단계

이 단계는 송신자가 송신중계자를 통해 발송 요청에 성공한 전자문서에 대한 유통증명서 발급을 요청하고 이에 대한 발급결과를 전달받는 단계이다.

송신자는 중계자가 제공하는 API를 통해 송신한 전자문서에 대해 유통증명서 발급을 요청하고, 그에 대한 결과로서 유통증명서를 발급 받는다.

중계자는 유통증명서 발급을 위해 전담기관의 유통허브시스템에 연계하여 유통증명서 발급을 요청하게 되는데, 발급요청에 대한 동기식 응답으로 유통증명서를 전달할 수도 있으며, 발급요청만 접수하고 비동기식으로 유통증명서를 발급한 후 전달하는 것도 가능하다.

□ 처리 절차에 따른 흐름도



[그림 5] 유통증명서 발급단계 절차

□ 처리 절차별 표준 연계 인터페이스

API 명	설명	요청자	응답자	비고
① 유통증명서 발급 요청	송신자가 송신중계자에게 유통증명서 발급 요청하는 API	송신자	송신 중계자	발급 결과(유통증명서)를 동기식 응답으로 돌려주거나 요청접수 후 비동기식으로 전달
② 유통증명서 발급 결과 전달	송신중계자가 유통증명서 발급 처리 후 발급을 요청한 송신자에게 발급결과를 전달하는 API	송신 중계자	송신자	중계자가 유통증명서를 비동기식으로 발급하는 경우 송신자는 “②” “③” API 중 선택하여 발급된 유통증명서를 수신함
③ 유통증명서 발급 결과 요청	송신자가 중계자에게 유통증명서 발급 요청에 대한 발급결과를 요청하는 API	송신자	송신 중계자	

3.2.6. 원본증명서 발급 단계

이 단계는 공인전자문서센터에 보관된 전자문서 원본에 대한 원본증명서 발급을 요청하고, 이에 대한 발급결과를 전달받는 단계이다.

송신자는 송신중계자가 제공하는 API를 통해 전자문서 원본에 대한 원본증명서 발급을 요청하고, 그에 대한 결과로서 원본증명서를 발급받는다.

중계자는 원본증명서 발급을 위해 공인전자문서센터와 연계하여 원본증명서 발급을 요청하며, 발급요청에 대한 동기식 응답으로 원본증명서를 전달할 수 있고, 발급요청만 접수한 후 비동기식으로 원본증명서를 발급하여 전달하는 것도 가능하다.

처리 절차 및 흐름은 3.2.5 유통증명서 발급 단계와 동일하므로 본 절에서는 생략한다.

3.2.7. 전자증명서 발급 단계

이 단계는 송신한 전자문서에 대한 전자증명서 발급을 요청하고, 이에 대한 발급결과를 전달받는 단계이다.

송신자는 송신중계자가 제공하는 API를 통해 전자증명서 발급을 요청하고, 그에 대한 결과로서 전자증명서를 발급받는다.

중계자는 전자증명서 발급을 위해 전담기관의 유통허브시스템 및 공인전자문서센터와 연계하여 유통증명서와 원본증명서 발급을 요청하며, 발급요청에 대한 동기식 응답으로 전자증명서를 전달할 수 있고, 발급요청만 접수한 후 비동기식으로 전자증명서를 발급하여 전달하는 것도 가능하다.

처리 절차 및 흐름은 3.2.5 유통증명서 발급 단계와 동일하므로 본 절에서는 생략한다.

4. 연계 인터페이스(API)

4.1. API 개요

중계자와 송신자 시스템 간 연계 방안의 각 단계에서 도출한 표준 연계 인터페이스(API)는 다음과 같다.

단계 구분	API		요청자	서비스 제공자 (응답자)	HTTP Method	요청 URI
	번호	API 이름				
(1) 이용자 인증단계	❶	이용자 인증토큰 발급(및 갱신)	송신자	송신 중계자	POST	https://[중계자서비스 도메인]/auth/client
	❷	이용자 인증토큰 (발급 및) 갱신	송신자	송신 중계자	POST	이용자 인증단계의 “❶ 이용자 인증토큰 발급(및 갱신)” API와 동일
(2) 중계자 인증단계	❶	중계자 인증토큰 발급(및 갱신)	송신 중계자	송신자	POST	https://[송신자서비스 도메인]/auth/client
	❷	중계자 인증토큰 (발급 및) 갱신	송신 중계자	송신자	POST	중계자 인증단계의 “❶ 중계자 인증토큰 발급(및 갱신)” API와 동일
(3) 전자문서 발송·열람 단계	❶	전자문서 발송요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인]/api/reqSendedoc
	❷	전자문서 처리(수신) 상태 전달	송신 중계자	송신자	POST	https://[송신자서비스 도메인]/api/resedocStatus
	❸	전자문서 열람요청	수신자	송신자	GET	https://[전자문서 열람 URL]
	❹	열람인증토큰 검증 요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인]/api/verifyReadToken
	❺	전자문서 열람정보 전달	송신자	송신 중계자	POST	https://[중계자서비스 도메인]/api/sendReadInfo
	❻	전자문서 처리(열람) 상태 전달	송신 중계자	송신자	POST	전자문서 발송·열람 단계의 “❷ 전자문서 처리(수신)상태 전달” API와 동일
	❼	전자문서 처리상태 요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqStatusedoc
(4) 유통 증명서 발급단계	❶	유통증명서 발급요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqDistributionCert
	❷	유통증명서 발급결과 전달	송신 중계자	송신자	POST	https://[송신자서비스 도메인] /api/resDistributionCert
	❸	유통증명서 발급결과 요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqStatusDistributionCert
(5) 원본 증명서 발급단계	❶	원본증명서 발급요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqOriginalCert
	❷	원본증명서 발급결과 전달	송신 중계자	송신자	POST	https://[송신자서비스 도메인] /api/resOriginalCert
	❸	원본증명서 발급결과 요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqStatusOriginalCert

단계 구분	API		요청자	서비스 제공자 (응답자)	HTTP Method	요청 URI
	번호	API 이름				
(6) 전자 증명서 발급단계	①	전자증명서 발급요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqElectronicCert
	②	전자증명서 발급결과 전달	송신 중계자	송신자	POST	https://[송신자서비스 도메인] /api/resElectronicCert
	③	전자증명서 발급결과 요청	송신자	송신 중계자	POST	https://[중계자서비스 도메인] /api/reqStatusElectronicCert

4.2. API 공통사항

중계자 간 연계를 위해 필요한 모든 API는 다음 기준에 맞춰 정의한다.

- HTTPS 기반의 REST 아키텍처를 기반으로 함
- 중계자 간 네트워크 보안은 TLS 1.2 이상으로 함
- API 요청자는 서비스 제공자로부터 “1. 사용자 인증토큰 발급”요청을 통해 access token을 획득한 후 access token을 기반으로 API를 호출함

4.2.1. HTTP 헤더

본 규격의 모든 메시지는 요청 및 응답에 따라 아래 명시된 헤더 값을 HTTP 헤더에 필수로 기술하여야 한다. 다만 Access Token을 발급받기 위한 “(1) 이용자 인증단계”의 “① 이용자 인증”과 “(2) 중계자 인증단계”의 “① 중계자 인증” API는 헤더의 값 중 “Authorization” 값은 기술하지 않는다.

“(4) 전자문서 발송·열람 단계”의 “③ 전자문서 열람요청” API 호출 시에는 수신자가 수신중계자로부터 발급받은 열람인증토큰을 헤더의 “Authorization” 값에 넣어 전달한다.

그 외에 각 중계자 및 송신자가 제공하는 나머지 API는 중계자 또는 이용자 인증을 통해 발급받은 인증토큰(Access Token)을 헤더의 “Authorization” 값에 전달하여 사용한다.

□ 요청메시지 HTTP 헤더

- Authorization: API를 제공하는 시스템(중계자 또는 송신자)으로부터 인증절차를 거쳐 발급받은 인증토큰(Access Token)이나 수신중계자로부터 발급 받은 열람인증토큰을 전달
- platform-id: 요청자가 중계자인 경우에는 중계자시스템의 인증번호를, 송신자인 경우에는 송신자의 식별번호를 전달
- req-UUID: 요청메시지의 고유식별값(UUID)을 전달
- req-date: 요청메시지 발송시각을 전달

ex) 2020-10-01 22:34:30

□ 응답메시지 HTTP 헤더

- platform-id: API 호출에 응답하는 시스템이 중계자인 경우는 중계자시스템의 인증번호를, 송신자인 경우에는 송신자의 식별번호를 전달
- res-UUID: 응답메시지의 고유식별값을 전달
- res-date: 응답메시지의 발송시각을 전달
ex) 2020-10-01 22:34:30

□ HTTP 헤더 예시

```
POST /api/eaddr HTTP/1.1

Accept: application/json
Connection: keep-alive
Content-Length: 83
Content-Type: application/json
Host: relaySvc.aaa.com
Authorization: Bearer eyJ0eXAIoiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJ2cdC5jb.....
platform-id: aaa-01-aaaapp
req-UUID: f73746a3-b65a-433f-a62b-8e5ee86aadf2
req-date: 2020-10-01 22:34:30.123
```

4.2.2. 요청 및 응답 메시지

□ 메시지 구조

요청이나 응답메시지가 별도의 첨부문서 없이 업무메시지만으로 구성된 경우에는 메시지 헤더 정보의 content-type이 “application/json(또는 text/plain)”인 Single-MIME 구조를, 업무메시지 외에 첨부문서가 있는 경우에는 “multipart/mixed”인 Multi-MIME 구조로 전달한다.

메시지가 Multi-MIME 구조인 경우, 해당 메시지는 첫 번째 MIME에 업무메시지가, 두 번째 MIME 부터는 첨부문서가 순차적으로 들어가는 구조로 구성된다.

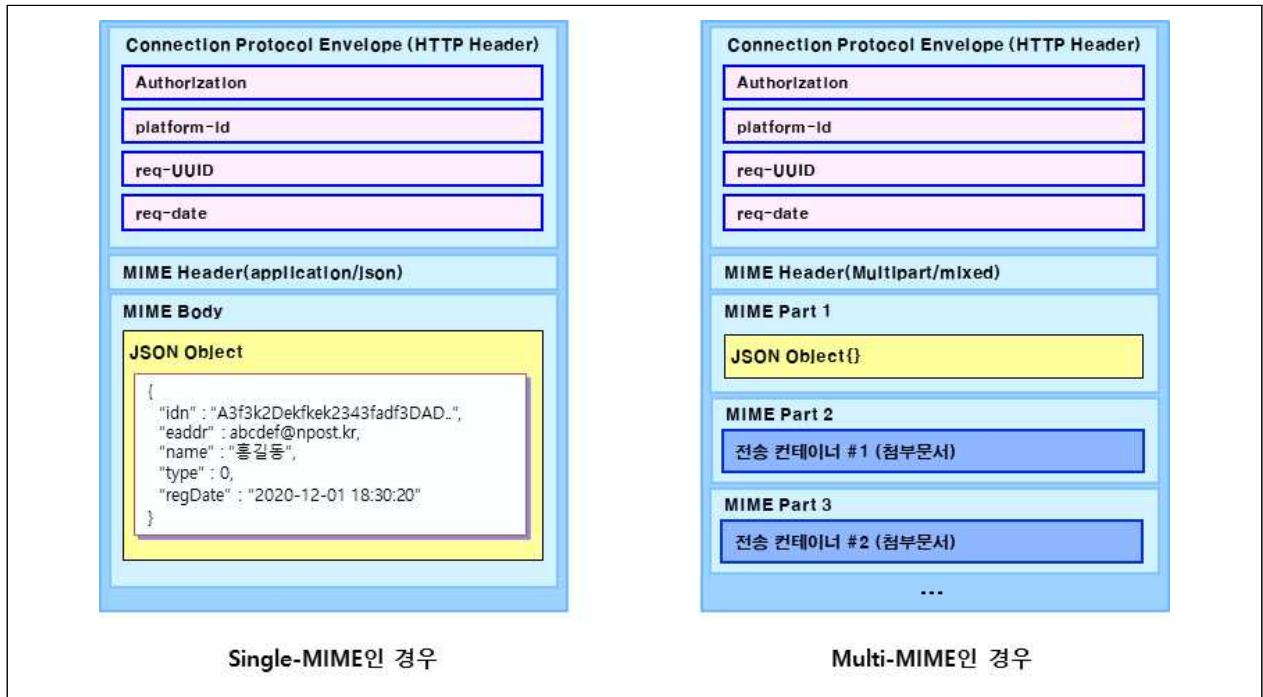
각 API의 요청 및 응답메시지 구조는 API의 상세설명에서 정의한다.

1) 요청메시지 요건

- HTTP 메소드가 GET인 API의 요청메시지는 일반적인 URI 호출방식을 사용하며 요청 파라미

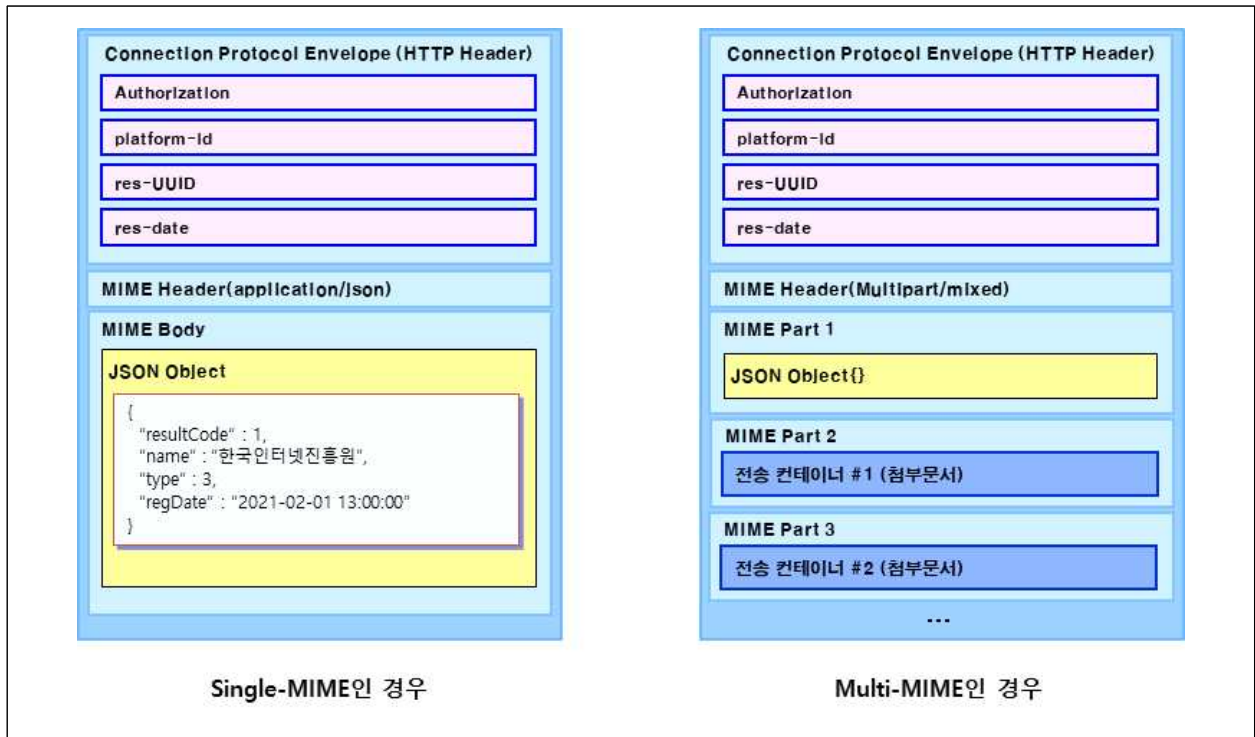
터에 대한 인코딩(UTF-8)을 필수로 수행해야 한다.

- HTTP 메소드가 POST, PUT, PATCH, DELETE 인 API의 요청메시지는 항상 업무메시지에 해당하는 JSON Object를 같이 전달하며, 요청메시지에 첨부문서가 있는 경우에는 Multi-MIME 구조로 구성하여 첨부문서도 같이 전달한다.
- 요청메시지 구성방안



2) 응답메시지 요건

- 본 규격에서 HTTP 상태코드 200은 요청메시지 전달의 성공만을 의미하며 요청 업무에 대한 처리결과 성공하였음을 의미하지는 않는다. HTTP 통신상 오류 발생 시 이에 대해서는 오류원인에 따라 Bad Request(400), Unauthorized(401), Internal Server Error(500) 등의 HTTP 상태 코드로 오류를 응답한다.
- 응답메시지의 업무메시지 영역에 있는 resultCode(성공:1, 실패:0)항목을 사용하여 요청에 대한 업무처리 결과의 성공 및 실패 여부를 전달한다.
- 응답메시지 구성방안



□ MIME 헤더 내 Content-Disposition

업무메시지와 첨부문서를 명확하게 식별하기 위해서 본 규격에서는 Content-Disposition의 name 필드를 활용하며, MIME으로 첨부해서 보낼 수 있는 content는 “업무메시지”, “알림메시지”, “첨부파일”이 있다. 이는 응답메시지도 동일하다.

- 업무메시지 : 각 요청메시지의 본문(JSON Object로 표현)에 해당하며 “msg”로 기술
- 알림메시지 : 전자문서 발송 시 선택적으로 삽입가능한 text/html, text/plain 타입의 content로서 “notice”로 기술함(전자문서 수신 알림메시지)
- 첨부파일 : 전자문서, 유통증명서 등 본 규약에서 주고받는 모든 파일은 “file”로 기술

```
#업무메시지
Content-Disposition: attachment; name="msg"
Content-Type: application/json;

#알림메시지
Content-Disposition: attachment; name="notice"
Content-Type: text/html; charset="UTF-8"

# 첨부문서
Content-Disposition: attachment; name="file" filename="지방세납부고지서.jpg"
Content-Type: application/octet-stream
```

4.2.3. 데이터 표기방안

□ 시각정보 표기방안

본 규격에서 사용하는 표준시는 KST(대한민국 표준시)로 UTC(협정 세계시)에서 9시간을 더한 시간을 표준으로 사용하며, 입력 받을 시간은 24시간을 기준으로 입력받는다.

(예시. 한국시각 기준 20년 12월 1일 오후 9시 35분 43초 일 경우 → 2020-12-01 21:35:43)

□ 데이터 타입별 표기방안

타입	정의	표기(예시)	해석
int	정수형 데이터	3	최대 3자리의 정수(실제 범위는 각 API별 설명참고)
String	문자열 데이터	10, 88	10 또는 88길이의 고정길이 문자열
		3~64	최소길이 3, 최대길이 64인 변동길이 문자열
Array	배열형 데이터	1~500	1~500개 원소를 가지는 배열

본 규격에서 데이터타입에 따른 표기 및 해석은 다음과 같다.

* 문자열 길이는 byte수가 아닌 글자수를 의미한다(ex. “가나다”=3, “abc”=3, “123”=3)

4.3. 중계자와 송신자 시스템 간 연계 API 상세설명

4.3.1. 이용자 인증토큰 발급

본 API는 이용자(송신자)가 중계자시스템이 제공하는 전자문서 발송 API를 호출하기 위한 인증 토큰을 획득하기 위해 제공된다. 인증 방식은 전자문서 발송 형태에 따라 다음의 두 가지 방식으로 제공되며, 중계자는 지원 가능한 전자문서 발송 서비스 범위에 따라 인증 방식을 선택하여 제공할 수 있다.

① 발송시스템과 송신자가 동일하지 않은 경우 (사용자 인증 기반 토큰 발급)

발송시스템이 이용자를 대신하여 전자문서 발송 대행 서비스를 제공하는 형태이다.

□ 인증방식

- authorization_code를 이용한 토큰 발급

- refresh_token을 이용한 액세스 토큰 갱신

※사용자 로그인 및 authorization_code 발급 과정은 중계자시스템이 제공하는 인증 절차 및 정책을 따르며, 본 문서에서는 authorization_code 수신 이후의 토큰 발급 절차만 정의한다.

② 발송시스템과 송신자가 동일한 경우 (서버 간 토큰 발급)

발송시스템을 보유한 기관이 중계자시스템과 직접 연계하여 전자문서를 발송하는 형태이다.

인증방식

- client_credentials를 이용한 액세스 토큰 발급
- refreshToken은 발급하지 않으며, 액세스 토큰 만료 시 clientId 및 clientSecret을 이용하여 재발급 받는다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/auth/client	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
grantType	String	-	Y	토큰 요청 유형 (authorization_code, refresh_token, client_credentials 중 택일)
clientId	String	16~32	Y	발송기관 클라이언트 ID
clientSecret	String	32~512	Y	발송기관 클라이언트 인증 비밀 키
code	String	20~128	N	사용자 인증 후 발급 된 authorization_code (grantType이 authorization_code인 경우 필수)
redirectUri	String	10~256	N	authorization_code 발급 시 사용된 Redirect URI (grantType이 authorization_code인 경우 필수)
refreshToken	String	32~2048	N	액세스 토큰 재발급용 토큰 (grantType이 refresh_token인 경우 필수)

□ 요청메시지 예시

```
# authorization_code
{
  "grant_type": "authorization_code",
  "code": "SplxIOBeZQQYbYS6WxS...",
  "redirect_uri": "https://sender.example.kr/oauth/callback",
  "client_id": "client123",
  "client_secret": "4P6dpbeW7pKkfM0R9b5m42Bld3a"
}

# refresh_token
{
  "grant_type": "refresh_token",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "client_id": "client123",
  "client_secret": "4P6dpbeW7pKkfM0R9b5m42Bld3a"
}

# client_credentials
{
  "grant_type": "client_credentials",
  "client_id": "client123",
  "client_secret": "4P6dpbeW7pKkfM0R9b5m42Bld3a"
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	accessToken	String	32~2048	Y	API 호출용 액세스 토큰
	tokenType	String	6	Y	“Bearer” 고정
	expiresIn	int	6	Y	토큰 유효시간(초)
	refreshToken	String	32~2048	N	accessToken 재발급용 토큰 (grantType이 authorization_code 또는 refresh_token인 경우 발급 가능)
	scope	String	-	N	API 접근 권한 범위
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

* refreshToken은 운영 정책에 따라 재발급(rotation) 또는 재사용(reuse) 방식으로 처리할 수 있다.

* scope 값은 서비스 제공자가 정의하며, 복수 개의 권한은 공백으로 구분한다.

예) "document:*", "document:full", "document:read document:write",

□ 응답메시지 예시

```
# authorization_code
{
  "access_token": "eyJpc3MiOiJwbGF0Zm9ybS0wMS1hcHAiLCJ...",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 600,
  "scope": "document:write document:read"
}

# refresh_token
{
  "access_token": "eyJpc3MiOiJwbGF0Zm9ybS0wMS1hcHAiLCJ...",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 600,
  "scope": "document:write document:read"
}

# client_credentials
{
  "access_token": "eyJhbGciOiJIUzI1NiIs...",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

4.3.2. 중계자 인증토큰 발급

본 API는 중계자가 이용자(송신자)시스템이 제공하는 API를 호출하기 위한 인증 토큰 획득을 위해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[송신자서비스 도메인]/auth/client	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터		타입	길이	필수 여부	설명
grantType		String	18	Y	"client_credentials" 고정
발급	clientId	String	16~32	Y	중계자의 클라이언트 ID (발급 시 필수)
	clientSecret	String	32~512	N	중계자의 클라이언트 인증 비밀 키

□ 요청메시지 예시

```
{
  "grant_type": "client_credentials",
  "client_id": "client123",
  "client_secret": "4P6dpbeW7pKkfM0R9b5m42Bld3a"
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	accessToken	String	32~2048	Y	액세스 토큰
	tokenType	String	6	Y	"Bearer" 고정
	expiresIn	int	6	Y	토큰 유효시간(초)
	scope	String	-	N	허가된 권한 범위
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

□ 응답메시지 예시

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs...\"",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

4.3.3. 전자문서 발송요청

본 API는 송신자가 수신자에게 전자문서를 전달하기 위해 중계자에게 발송을 요청할 때 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqSendedoc	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명	
edocList	array	1..N	Y	전송하고자 하는 전자문서정보 목록	
edocId	String	36	Y	송신자가 관리하는 전자문서 고유 ID	
subject	String	100	Y	전자문서제목	
edocCode	String	10	N	전자문서유형 정보(발송 전자문서 구분을 위해 송신자가 관리하는 전자문서코드 정보)	
sendEaddr	String	3~64	Y	송신자의 공인전자주소	
sendSubEaddr	String	3~64	N	송신자 공인전자주소의 하위 계정 ^(*)	
rcvInfo	-	-	Y	수신자의 식별번호, 공인전자주소 및 수신자의 개인정보를 기반으로 전송을 요청할 수 있음	
rcvPlatformId	String	25	N	수신자의 중계플랫폼 ID ^(**)	
rcvInfoType	String	1	Y	수신자에 대한 정보유형으로 각 유형에 따라 식별번호, 주소, 개인정보, 암호화된 개인정보 항목을 전달함 “1”: 식별번호 “2”: 주소 “3”: 개인정보 “4”: 암호화된 개인정보 ※ 아래 각 항목의 필수여부는 각 항목이 선택되었을 경우에 필수 값인지 여부를 말함	
식별번호	rcvIdn	String	10, 88	Y	수신자 고유번호(개인: CI, 법인: 사업자번호)
주소	rcvEaddr	String	3~64	Y	수신자의 공인전자주소
	rcvSubEaddr	String	3~64	N	수신자 공인전자주소의 하위 계정 ^(*)
개인정보	rcvBirthDate	String	8	Y	수신자의 생년월일 (YYYYMMDD)

파라미터		타입	길이	필수 여부	설명
암호화된 개인 정보	recvName	String	100	Y	수신자의 이름
	recvPhoneNo	String	11	Y	수신자의 전화번호 ('-' 없이 11자리)
	recvInfoHash	String	128	Y	수신자의 이름, 생년월일, 전화번호를 순서대로 연결한 후 SHA-512 알고리즘으로 해시한 값
sendReqDate		String	19	Y	송신요청일시
transferType		String	4	Y	전자문서 전송방식(url ^(****) 또는 file)
url		String	2048	N	열람 url
contentsList		array	1..N	N	전송대상 알림메시지 및 첨부파일의 해시정보
contentId		String	100	N	전송 전자문서 메시지 패키지 내에서 콘텐츠(알림메시지 또는 첨부파일)를 포함한 MIME의 content-id 값 - 전송메시지 내 multi MIME 중 해당하는 콘텐츠가 첨부된 MIME의 content-id 값을 기술 - 파일이 직접 첨부되지 않은 URL방식의 경우 이 항목을 기술하지 않으나, 이 경우에도 알림메시지를 첨부한 경우에는 이 항목에 안내문이 들어간 MIME의 content-id를 기술
contentHash		String	44, 64	Y	contentID에 해당하는 안내문 및 전자문서의 해쉬값 contentID가 없는 경우에는 URL로 전달하는 전자문서의 해쉬값을 기술
authExpTime ^(****)		String	19	N	문서열람 인증 만료일시
readExpTime ^(*****)		String	19	N	문서열람 만료일시
edocExtension		Any	-	N	문서 단위에서 중계자에게 추가로 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용
reqExtension		Any	-	N	요청메시지 단위에서 중계자에게 추가로 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용

* 공인전자주소의 하위 계정: 송신자나 수신자가 기업·기관의 경우에는 주소 하위에 복수의 계정을 보유할 수 있음

** recvPlatformId: 송신자가 발송요청할 때 수신자의 중계플랫폼 정보를 알고 있는 경우에는 이 정보를 포함하여 전달함으로써 수신 플랫폼을 지정할 수 있으며, 송신자가 수신플랫폼을 지정해서 요청하는 경우, 중계자는 반드시 해당 수신 플랫폼으로 전자문서를 전송하여야 함

*** url: transferType이 url일 경우 필수항목임

**** authExpTime: 발송기관이 설정한 값으로 열람인증이 가능한 일자로 이 시각까지 한번이라도 열람이 되지 않은 경우에는 문서 열람이 불가함

***** readExpTime: 발송기관이 설정한 값으로 이 시각 이후로는 문서의 재열람이 불가함

□ 요청메시지 예시

```

{ "edocList" : [
  { "edocId" : "333746a3-b65a-433f-a62b-435ee86aadf2",
    "subject" : "보험계약서",
    "edocCode": "a00001",
    "sendEaddr" : "aaa_123",
    "recvInfo" : {
      "recvPlatformId" : "torpedo-02-bizframe",
      "recvEaddr" : "abc_12345" },
    "sendReqDate" : "2021-01-07 09:00:00",
    "transferType" : "url",
    "url" : "https://npost.kr/msg/2d6e1a624aafcb60d95728f...",
    "contentsList" : [
      { "contentId" : "notice_001",
        "contentHash" : "074f2c0ad51e2d6e1a624aafcb60d95728fd02854022d0dc8d90334564303b5" },
      { "contentHash" : "0dsr3c0ad5sd3d6e1a624aafcb60d95728fd02854022d0dc8d903ew16435tg5" }
    ],
    "edocExtension" :
      { "legacyKey" : "123456" } } ,
  { "edocId" : "23437464f-b65a-433f-a62b-4r5te86aass1",
    "subject" : "보험계약서",
    "edocCode": "a00001",
    "sendEaddr" : "aaa_123",
    "recvInfo" : {
      "recvPlatformId" : "torpedo-02-bizframe",
      "recvEaddr" : "def_12345" },
    "sendDate" : "2021-01-07 09:00:00",
    "transferType" : "url",
    "url" : "https://npost.kr/msg/2d6e1a624aafcb60d95728f...",
    "contentsList" : [
      { "contentId" : "notice_002",
        "contentHash" : "074f2c0ad51e2d6e1a624aafcb60d95728fd02854022d0dc8d90334564303b5" },
      { "contentHash" : "0dsr3c0ad5sd3d6e1a624aafcb60d95728fd02854022d0dc8d903ew16435tg5" }
    ],
    "edocExtension" :
      { "legacyKey" : "123457" } } ,
  { "edocId" : "3wsdf6a3-3dfa-434f-a62b-435ee8634dfc",
    "subject" : "보험계약서",
    "edocCode": "a00001",
    "sendEaddr" : "aaa_123",
    "recvInfo" : {

```

```

    "recvPlatformId" : "torpedo-01-bizframe",
    "recvIdn" : "dfefek3r3kalfed.." },
    "sendDate" : "2021-01-07 09:00:00",
    "transferType" : "url",
    "url" : "https://npost.kr/msg/2d6e1a624aafcb60d95728f...",
    "contentsList" : [
        { "contentId" : "notice_003",
          "contentHash" : "074f2c0ad51e2d6e1a624aafcb60d95728fd02854022d0dc8d90334564303b5" },
        { "contentHash" : "0dsr3c0ad5sd3d6e1a624aafcb60d95728fd02854022d0dc8d903ew16435tg5" }
    ],
    "edocExtension" :
        { "legacyKey" : "123456" } }
} }

```

- 송신자가 전송하는 전자문서가 모두 transferType이 "url"이고 전자문서안내문을 보내지 않는 경우에 요청메시지는 single MIME 구조를 가지며, contentsList내에 contentId 항목이 존재하지 않음
- 송신자가 전송하는 전자문서가 transferType이 "url"이고 전자문서안내문을 보내는 경우에 요청메시지는 multi MIME 구조며, 첨부파일이 들어가는 "MIME Part 2"부터 전자문서 안내문 (content-disposition의 name항목이 "notice"임)을 포함하여 전송함. 이 때 contentsList내에 contentId 항목에는 전자문서 안내문을 포함한 MIME의 content-id 값을 기술
- 송신자가 전송하는 전자문서가 transferType이 "file"인 경우에는 전자문서안내문의 전송여부에 관계없이 요청메시지는 multi MIME 구조며, 첨부파일이 들어가는 "MIME Part 2"부터 전자문서안내문 또는 첨부파일(content-disposition의 name항목이 "notice"나 "file"임)을 포함하여 전송함. 이 때 contentsList내에 contentId 항목에는 전자문서안내문 또는 첨부파일을 포함한 MIME의 content-id 값을 기술

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
접수 완료 또는 수	reqAccTime	String	19	Y	요청메시지 접수시간
	resultList	array	1..N	N	전자문서별 송신요청 처리결과 목록 (resultCode가 1인 경우)
	edocId	String	36	Y	송신자가 요청 시 전달한 고유 전자문서 ID

파라미터		타입	길이	필수여부	설명	
신 완 료	recvResultStatus	int	1	Y	전자문서번호별 처리결과 [정상처리 상태] 1 : 접수완료, 2 : 송신완료, 3 : 수신완료 [실패/예외 상태] 10 : 접수실패, 20 : 송신실패, 30 : 수신실패	
	edocNum		String	33	N	송신중계자가 부여한 전자문서번호(전자문서번호 부여 전 송신실패 시 번호 없이 응답)
	성 공	sendDate	String	19	N	전자문서 송신일시(송신중계자가 수신중계자에게 전자문서의 전달요청까지 완료한 경우, i.e. recvResultStatus 값이 2, 3, 30인 경우)
		recvDate	String	19	N	전자문서 송신하여 수신중계자에게 수신시각까지 받은 경우, 수신시각까지 포함하여 응답(recvResultStatus 값이 3인 경우)
	실 패	recvErrCode	String	10	Y	전자문서번호별 전자문서 수신오류 시 오류코드(recvResultStatus 값이 10, 20, 30인 경우)
		recvErrMsg	String	256	N	전자문서번호별 전자문서 수신오류 시 오류메시지
	edocExtension		Any	-	N	문서 단위에서 송신자에게 추가 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용
실 패	errCode	String	10	Y	요청메시지 처리 과정의 오류 시 오류코드(resultCode가 0인 경우)	
	errMsg	String	256	N	요청메시지 처리 과정의 오류 시 오류메시지	
resExtension		Any	-	N	응답메시지 단위에서 중계자에게 추가로 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용	

※ 송신자와 수신자가 동일한 중계자 서비스를 사용하는 경우에는 송신과 수신에 동시에 발생될 수 있으나, 서로 다른 중계자 서비스를 이용하는 경우에는 송·수신 행위 및 처리결과에 대한 정보는 서로 다를 수 있으므로 이를 구분하여 기록.

□ 응답메시지 예시

```
{
  "resultCode" : 1,
  "reqAccTime" : "2021-01-07 09:00:00",
  "resultList": [
    { "edocId" : "333746a3-b65a-433f-a62b-435ee86aadf2",
      "recvResultStatus" : "1"
    },
    { "edocId" : "23437464f-b65a-433f-a62b-4r5te86aass1",
      "recvResultStatus" : "1"
    },
    { "edocId" : "3wsdf6a3-3dfa-434f-a62b-435ee8634dfc",
      "recvResultStatus" : "1"
    }
  ]
}
```

4.3.4. 전자문서 처리상태 전달

본 API는 중계자가 전자문서 발송결과를 전달하기 위해 사용하는 API로 송신자(발송기관)에 의해 제공된다.

상태명	코드값	설명
접수완료	1	발송시스템이 송신중계자에게 전자문서를 전송하였고, 송신중계자 시스템이 해당 문서를 정상적으로 수신(접수)했음을 응답한 상태임
송신완료	2	송신중계자가 수신중계자에게 전자문서 송신은 완료하였으나 아직 수신중계자로부터 수신자에게 전자문서를 정상적으로 전달하였음을 확인받지 못한 상태로서, 송신중계자가 비동기로 수신중계자에게 전자문서를 전송하고, 수신중계자로부터 수신접수까지만 완료된 상태임
수신완료	3	수신중계자가 수신한 전자문서를 정상적으로 수신자에게 전달하였으나 수신자가 아직 열람하지는 않은 상태임
열람완료	4	수신자가 수신한 전자문서를 정상적으로 열람한 상태임
접수실패 (미수신)	10	발송시스템이 송신중계자에게 전자문서를 전송하였으나, 전자문서의 형식 오류, 필수값 누락, 인증 또는 검증 실패 등의 사유로 송신중계자 시스템에 정상적으로 접수되지 못한 상태임.
송신실패	20	수신자가 공인전자주소에 가입하지 않았거나, 수신중계자 시스템의 오류, 네트워크 상의 오류 등의 이유로 송신중계자가 전자문서 송신에 실패한 상태임
수신실패	30	수신중계자가 수신한 전자문서를 수신자에게 전달하지 못한 상태로서, 수신자의 미가입, 탈퇴 등의 이유로 수신자의 공인전자주소가 없는 상태임

열람실패	40	수신자가 수신한 전자문서를 열람하는 과정에서 열람기한 만료, 전자문서 열람시스템의 오류 등의 이유로 전자문서 열람에 실패한 상태임(단, 최초 열람이 성공하여 열람상태가 '4'가 된 상태에서는 이후 열람이 실패하더라도 상태가 '40'로 변경되지는 않음)
------	----	--

※ 송신자와 송신중계자는 다음과 같은 기준으로 전자문서 처리상태 코드 값을 공유한다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[송신자 서비스 도메인]/api/resedocStatus	POST	JSON

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수여부	설명
edocList	array	-	Y	전자문서 처리상태 목록
edocId	String	36	Y	송신자가 발송 요청 시 전달한 고유 전자문서 ID
status	int	1	Y	전자문서번호별 처리결과 [정상 처리 상태] 2 : 송신완료, 3 : 수신완료, 4 : 열람성공 [실패/예외 상태] 20 : 송신실패, 30 : 수신실패, 40 : 열람실패
edocNum	String	33	N	송신중계자가 부여한 전자문서번호(전자문서번호 부여 전 송신실패 시 번호 없이 응답)
sendDate	String	19	N	송신일시 처리결과가 2, 3, 4, 30, 40인 경우 필수
recvDate	String	19	N	수신일시 처리결과가 3, 4, 40인 경우 필수
readDate	String	19	N	열람일시 처리결과가 4인 경우 필수
failCode	String	10	N	전자문서 처리상태가 실패(status가 20, 30, 40인 경우)의 오류코드
failMsg	String	512	N	failCode에 대한 메시지
edocExtension	Any	-	N	문서 단위에서 송신자에게 추가 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용
reqExtension	Any	-	N	요청메시지 단위에서 송신자에게 추가로 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용

□ 요청메시지 예시

```
{ "edocList" : [
  { "edocId" : "333746a3-b65a-433f-a62b-435ee86aadf2",
    "status" : "3",
    "edocNum" : "20220107_KISA000001_0001234567890",
    "sendDate" : "2021-01-07 09:01:00",
    "recvDate" : "2021-01-07 09:01:03" },
  { "edocId" : "23437464f-b65a-433f-a62b-4r5te86aass1",
    "status" : "3",
    "edocNum" : "20220107_KISA000001_0001234567972",
    "sendDate" : "2021-01-07 09:01:01",
    "recvDate" : "2021-01-07 09:01:04" },
  { "edocId" : "3wsdf6a3-3dfa-434f-a62b-435ee8634dfc",
    "status" : "2",
    "edocNum" : "20220107_KISA000001_0003464567812",
    "sendDate" : "2021-01-07 09:01:04" }
] }
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

□ 응답메시지 예시

```
{
  "resultCode" : 1
}
```

4.3.5. 전자문서 열람요청

수신자가 전자문서 열람을 요청하면 (수신)중계자는 열람을 위한 본인인증을 수행한 후 수신자에게 열람인증토큰을 발급해준다.

수신자는 이 열람인증토큰을 HTTP헤더의 "Authorization" 항목에 넣어 송신자가 전달한 전자문서 열람 URL을 호출한다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[전자문서 열람 URL]	GET	HTTP response

2) 메시지 구조

요청메시지 구조

- 비즈니스 요청정보 없음

요청메시지 예시

```
https://[전자문서 열람 URL]
```

응답메시지 구조

- HTTP response 구조

응답메시지 예시

```
HTTP/1.1 200 OK
Server: apache
Content-Type: text/html
Connection: keep-alive
Content-Length: 250

<!-- 열람하고자 하는 전자문서 본문 ---->
```

4.3.6. 열람인증토큰 검증 요청

본 API는 송신자가 수신자로부터 열람 URL을 통해 전자문서 열람을 요청받으면, 요청 시 받은 열람인증토큰을 추출하여 이에 대한 유효성을 송신중계자에게 검증 요청하는 인터페이스이다.

열람인증토큰은 수신자가 “4.3.5. 전자문서 열람요청” 시 HTTP헤더의 “Authorization” 항목에 넣어 송신자에게 전달한 값이다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[증계자서비스 도메인]/api/verifyReadToken	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
readToken	String	-	Y	열람인증토큰

요청메시지 예시

```
{ "readToken" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InBsYXRmb3JtSWQtYWJjZGVmIn0.eyJpc3MiOiJraXNhMDAxMjM0IiwiaXVkljoidHJ1ZSIsImVhdC6MTUxNjIzOTAyMiwiZXhwIjoxNTE2MjM5MDI1LjE2CjJlZG9jTnVtIjoiaWJyMTA3MDFfS0ITQTAwMDAwMV8wMDAxMjM0NTY3ODkwLn0.7StyFcJ0frnhAazbi6u6rUTOGt3a5VHS6JkduQ9A-euZ5mqXi6OoQv5wP4xUKkq54B4o3VyzhOBK_jp5eGKmq" }
```

응답메시지 구조

파라미터	타입	길이	필수 여부	설명
resultCode	int	1	Y	열람인증토큰의 유효성 검증결과 (1: 유효함, 0: 유효하지 않음)
실패	errCode	String	10	오류코드
	errMsg	String	256	오류메시지

응답메시지 예시

```
{
  "resultCode" : 1
}
```

4.3.7. 전자문서 열람정보 전달

본 API는 송신자가 전자문서 열람결과를 송신중계자에게 전달하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/sendReadInfo	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
edocList	array	-	Y	전자문서 처리상태 목록
edocId	String	36	Y	송신자가 발송 요청 시 전달한 고유 전자문서 ID
status	int	1	Y	전자문서번호별 처리결과 (4: 열람성공, 40: 열람실패)
readDate	String	19	N	열람일시 처리결과가 "4"인 경우 필수
failCode	String	10	N	전자문서 처리상태가 실패(status가 40인 경우)의 오류코드
failMsg	String	512	N	failCode에 대한 메시지
edocExtension	Any	-	N	문서 단위에서 중계자에게 추가 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용
reqExtension	Any	-	N	요청메시지 단위에서 중계자에게 추가로 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용

요청메시지 예시

```
{ "edocList" : [
  { "edocId" : "333746a3-b65a-433f-a62b-435ee86aadf2",
    "status" : "4",
    "readDate" : "2021-01-07 09:01:03" },
  ] }
```

응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

응답메시지 예시

```
{
  "resultCode" : 1
}
```

4.3.8. 전자문서 처리상태 정보 요청

본 API는 송신중계자로부터 송신요청 후 처리결과를 받지 못하였거나, 송신 후 열람상태를 받지 못한 송신자가 전자문서 처리(수신, 열람)상태를 확인하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqStatusedoc	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
edocId	String (array)	36	Y	송신자가 발송 요청 시 전달한 고유 전자문서 ID

요청메시지 예시

```
{ "edocId" : [
  "333746a3-b65a-433f-a62b-435ee86aadf2",
  "23437464f-b65a-433f-a62b-4r5te86aass1",
  "3wsdf6a3-3dfa-434f-a62b-435ee8634dfc"
] }
```

□ 응답메시지 구조

파라미터	타입	길이	필수여부	설명	
resultCode	int	1	Y	처리결과 (1:성공, 0:실패) 요청메시지가 문법적으로 맞으면 성공(1)으로 처리	
성공	edocList	array	-	Y	처리상태 검색을 요청한 전자문서목록
	edocId	String	36	Y	송신자가 발송 요청 시 전달한 고유 전자문서 ID
	status	int	1	Y	전자문서번호별 처리결과 [정상 처리 상태] 1 : 접수완료, 2 : 송신완료, 3 : 수신완료, 4 : 열람성공 [실패/예외 상태] 10 : 접수실패, 20 : 송신실패, 30 : 수신실패, 40 : 열람실패)
	edocNum	String	33	N	송신중계자가 부여한 전자문서번호(전자문서번호 부여 전 송신실패 시 번호 없이 응답)
	sendDate	String	19	N	송신일시 처리결과가 2, 3, 4, 30, 40인 경우 필수
	recvDate	String	19	N	수신일시 처리결과가 3, 4, 40인 경우 필수
	readDate	String	19	N	열람일시 처리결과가 4인 경우 필수
	failCode	String	10	N	전자문서 처리상태가 실패(status가 10, 20, 30, 40인 경우)의 오류코드 ※status "10"은 미수신(edocId 알 수 없음)을 포함한다.
	failMsg	String	512	N	failCode에 대한 메시지
edocExtension	Any	-	N	문서 단위에서 송신자에게 추가 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용	
실패	errCode	String	10	Y	요청오류에 대한 코드
	errMsg	String	256	N	요청오류에 대한 메시지
resExtension	Any	-	N	응답메시지 단위에서 송신자에게 추가로 전달해야 하는 정보가 있는 경우 하위에 정보를 추가하여 활용	

□ 응답메시지 예시

```
{ "resultCode" : 1,
  "edocList" : [
    { "edocId" : "333746a3-b65a-433f-a62b-435ee86aadf2",
      "status" : 3,
      "edocNum" : "20220107_KISA000001_0001234567890",
      "sendDate" : "2021-01-07 09:01:00",
      "recvDate" : "2021-01-07 09:01:03" },
    { "edocId" : "23437464f-b65a-433f-a62b-4r5te86aass1",
      "status" : 3,
      "edocNum" : "20220107_KISA000001_0001234567972",
      "sendDate" : "2021-01-07 09:01:01",
      "recvDate" : "2021-01-07 09:01:04" },
    { "edocId" : "3wsdf6a3-3dfa-434f-a62b-435ee8634dfc",
      "status" : 2,
      "edocNum" : "20220107_KISA000001_0003464567812",
      "sendDate" : "2021-01-07 09:01:04" }
  ] }
```

4.3.9. 유통증명서 발급요청

본 API는 송신자가 유통증명서 발급을 요청하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqDistributionCert	POST	JSON, [File]

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	증명서 발급요청에 대한 요청 식별값 (요청자가 유일하게 부여함)
식별자	edocNum	String	33	Y 송신중계자가 부여한 전자문서번호(전자문서번호 부여 전 송신실패 시 번호 없이 응답) 유통증명서 발급요청 시, edocNum 값을 기본으로 하나, 송신자가 edocNum의 관리가 어려울 경우 edocId 값으로 요청함

edocId	String	36	Y	송신자가 발송 요청 시 전달한 고유 전자문서 ID
eaddr	String	100	Y	유통증명서 요청한 이용자의 공인전자주소
reason	String	200	N	유통증명서 발급요청 사유

□ 요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad",
  "edocid" : "333746a3-b65a-433f-a62b-435ee86aadf2",
  "eaddr" : "kisa1234"
}
```

□ 응답메시지 구조

파라미터	타입	길이	필수 여부	설명
resultCode	int	1	Y	처리결과 (1:성공, 0:실패)
성공	status	String	1	유통증명서 요청결과 (W : 요청대기, S : 발급완료)
	acceptTime	String	19	유통증명서 발급 접수시간
	certNum	String	19	유통증명서 일련번호
실패	errCode	String	10	오류코드
	errMsg	String	256	오류메시지

※ 중계자가 증명서 발급을 동기식으로 처리하는 경우 multi-mime 구조로 응답하며 응답메시지 본문에는 certNum을 기입하고 유통증명서.pdf 파일을 첨부파일로 전달함

※ 중계자가 증명서 발급을 비동기식으로 처리하는 경우 응답메시지는 single-mime이며 acceptTime만 전달함

□ 응답메시지 예시

```
{
  "resultCode" : 1,
  "status" : "S",
  "certNum" : "abcd-0332-1234-5678"
}
```

4.3.10. 유통증명서 발급결과 전달

본 API는 중계자가 전담기관의 유통허브에 요청 후 유통증명서 발급결과를 받으면 이를 요청한 송신자에게 전달하기 위해 사용하는 API로 송신자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[발송기관서비스 도메인]/api/resDistributionCert	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	유통증명서 발급요청 시의 요청 식별값
status	String	1	Y	유통증명서 요청결과 (F : 발급실패, S : 발급완료)
certNum	String	19	N	유통증명서 일련번호
failMsg	String	256	N	유통증명서 발급실패 사유

※ 유통증명서 발급 성공 시 요청메시지는 multi-mime이며 certNum과 유통증명서.pdf 파일이 전달됨

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad",
  "status" : "S",
  "certNum" : "abcd-0332-1234-5678"
}
```

응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

응답메시지 예시

```
{
  "resultCode" : 1
}
```

4.3.11. 유통증명서 발급결과 요청

본 API는 송신자가 유통증명서 발급요청한 것에 대한 결과를 비동기식으로 수신하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqStatusDistributionCert	POST	JSON, [File]

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	유통증명서 발급요청 시의 요청 식별값

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad"
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	status	String	1	Y	유통증명서 요청결과 (F : 발급실패, W : 발급대기, S : 발급완료)
	certNum	String	19	N	유통증명서 일련번호
	failMsg	String	256	N	유통증명서 발급실패 사유
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

※ 유통증명서 발급이 완료된 경우 응답메시지는 multi-mime이며 certNum과 유통증명서.pdf 파일이 전달됨

□ 응답메시지 예시

```
{
  "resultCode" : 1,
  "status" : "S",
  "certNum" : "abcd-0332-1234-5678"
}
```

4.3.12. 원본증명서 발급요청

본 API는 송신자가 원본증명서 발급을 요청하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqOriginalCert	POST	JSON, [File]

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	증명서 발급요청에 대한 요청 식별값 (요청자가 유일하게 부여함)
식별자	edocNum	String	33	송신중계자가 부여한 전자문서번호(전자문서번호 부여 전 송신실패 시 번호 없이 응답)원본증명서 발급요청 시, edocNum 값을 기본으로 하나, 송신자가 edocNum의 관리가 어려울 경우 edocId 값으로 요청함
	edocId	String	36	송신자가 발송 요청 시 전달한 고유 전자문서 ID
eaddr	String	100	Y	원본증명서 요청한 이용자의 공인전자주소
reason	String	200	N	원본증명서 발급요청 사유

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad",
  "edocid" : "333746a3-b65a-433f-a62b-435ee86aadf2",
  "eaddr" : "kisa1234"
}
```

응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	status	String	1	Y	원본증명서 요청결과 (W : 요청대기, S : 발급완료)
	acceptTime	String	19	Y	원본증명서 발급 접수시간
	serialNum	String	1~20	N	원본증명서 일련번호
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

- ※ 증계자가 증명서 발급을 동기식으로 처리하는 경우 multi-mime 구조로 응답하며 응답메시지 본문에는 certNum을 기입하고 원본증명서.pdf 파일을 첨부파일로 전달함
- ※ 증계자가 증명서 발급을 비동기식으로 처리하는 경우 응답메시지는 single-mime이며 acceptTime만 전달함

응답메시지 예시

```
{
  "resultCode" : 1,
  "status" : "S",
  "serialNum" : "54854813187641"
}
```

4.3.13. 원본증명서 발급결과 전달

본 API는 증계자가 전담기관의 유통허브에 요청 후 원본증명서 발급결과를 받으면 이를 요청한 송신자에게 전달하기 위해 사용하는 API로 송신자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[발송기관서비스 도메인]/api/resOriginalCert	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	원본증명서 발급요청 시의 요청 식별값
status	String	1	Y	원본증명서 요청결과 (F : 발급실패, S : 발급완료)
serialNum	String	1~20	N	원본증명서 일련번호
failMsg	String	256	N	원본증명서 발급실패 사유

※ 원본증명서 발급 성공 시 요청메시지는 multi-mime이며 certNum과 원본증명서.pdf 파일이 전달됨

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad",
  "status" : "S",
  "serialNum" : "54854813187641"
}
```

응답메시지 구조

파라미터	타입	길이	필수 여부	설명
resultCode	int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	오류코드
	errMsg	String	256	오류메시지

응답메시지 예시

```
{
  "resultCode" : 1
}
```

4.3.14. 원본증명서 발급결과 요청

본 API는 송신자가 원본증명서 발급요청한 것에 대한 결과를 비동기식으로 수신하기 위해 사용하는 API 로 증계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqStatusOriginalCert	POST	JSON, [File]

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	원본증명서 발급요청 시의 요청 식별값

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad"
}
```

응답메시지 구조

파라미터	타입	길이	필수 여부	설명	
resultCode	int	1	Y	처리결과 (1:성공, 0:실패)	
성공	status	String	1	Y	원본증명서 요청결과 (F : 발급실패, W : 발급대기, S : 발급완료)
	serialNum	String	1~20	N	원본증명서 일련번호
	failMsg	String	256	N	원본증명서 발급실패 사유
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

※ 원본증명서 발급이 완료된 경우 응답메시지는 multi-mime이며 certNum과 원본증명서.pdf 파일이 전달됨

응답메시지 예시

```
{
  "resultCode" : 1,
  "status" : "S",
  "serialNum" : "54854813187641"
}
```

4.3.15. 전자증명서 발급요청

본 API는 송신자가 전자증명서 발급을 요청하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqElectronicCert	POST	JSON, [File]

2) 메시지 구조

□ 요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	증명서 발급요청에 대한 요청 식별값 (요청자가 유일하게 부여함)
식별자	edocNum	String	33	송신중계자가 부여한 전자문서번호(전자문서번호 부여 전 송신실패 시 번호 없이 응답) 증명서 발급요청 시, edocNum 값을 기본으로 하나, 송신자가 edocNum의 관리가 어려울 경우 edocId 값으로 요청함
	edocId	String	36	송신자가 발송 요청 시 전달한 고유 전자문서 ID
eaddr	String	100	Y	증명서 요청한 이용자의 공인전자주소
reason	String	200	N	증명서 발급요청 사유

□ 요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad",
  "edocid" : "333746a3-b65a-433f-a62b-435ee86aadf2",
  "eaddr" : "kisa1234"
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	status	String	1	Y	증명서 요청결과 (W : 요청대기, S : 발급완료)
	acceptTime	String	19	Y	증명서 발급 접수시간
	certNum	String	19	N	유통증명서 일련번호
	serialNum	String	1~20	N	원본증명서 일련번호
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

- ※ 중계자가 증명서 발급을 동기식으로 처리하는 경우 multi-mime 구조로 응답하며 응답메시지 본문에는 각 증명서 일련번호를 기입하고 유통증명서와 원본증명서 파일을 첨부파일로 전달함
- ※ 유통증명서와 원본증명서가 모두 발급 완료된 경우에만 status=S로 응답한다. 두 증명서 중 하나라도 발급에 실패한 경우 전체 요청은 실패로 처리한다.
- ※ 중계자가 증명서 발급을 비동기식으로 처리하는 경우 응답메시지는 single-mime이며 acceptTime만 전달함

□ 응답메시지 예시

```
{
  "resultCode" : 1,
  "status" : "S",
  "certNum" : "abcd-0332-1234-5678"
}
```

4.3.16. 전자증명서 발급결과 전달

본 API는 중계자가 유통허브시스템 및 공인전자문서센터에 유통증명서와 원본증명서를 발급을 요청하여 결과를 받으면 이를 요청한 송신자에게 전달하기 위해 사용하는 API로 송신자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[발송기관서비스 도메인]/api/resElectronicCert	POST	JSON

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	전자증명서 발급요청 시의 요청식별자
status	String	1	Y	전자증명서 요청결과 (F : 발급실패, S : 발급완료)
certNum	String	19	N	유통증명서 일련번호
serialNum	String	1~20	N	원본증명서 일련번호
failMsg	String	256	N	전자증명서 발급실패 사유

※ 전자증명서 발급 성공 시 결과 전달 메시지는 multi-mime이며 각 증명서의 일련번호화 증명서 파일이 전달됨

※ 유통증명서와 원본증명서가 모두 발급 완료된 경우에만 status=S로 전달한다. 두 증명서 중 하나라도 발급에 실패한 경우 전체 요청은 실패로 처리한다.

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad",
  "status" : "S",
  "certNum" : "abcd-0332-1234-5678",
  "serialNum" : "54854813187641"
}
```

응답메시지 구조

파라미터	타입	길이	필수 여부	설명
resultCode	int	1	Y	처리결과 (1:성공, 0:실패)
실패	errCode	String	10	오류코드
	errMsg	String	256	오류메시지

응답메시지 예시

```
{
  "resultCode" : 1
}
```

4.3.17. 전자증명서 발급결과 요청

본 API는 송신자가 전자증명서 발급요청한 것에 대한 결과를 비동기식으로 수신하기 위해 사용하는 API로 중계자에 의해 제공된다.

1) API 기본정보

요청 URI	메소드	응답형식
https://[중계자서비스 도메인]/api/reqStatusElectronicCert	POST	JSON, [File]

2) 메시지 구조

요청메시지 구조

파라미터	타입	길이	필수 여부	설명
certReqId	String	36	Y	전자증명서 발급요청 시의 요청 식별값

요청메시지 예시

```
{
  "certReqId" : "15831a3f-5ab6-343f-ab62-4635ee8af2ad"
}
```

□ 응답메시지 구조

파라미터		타입	길이	필수 여부	설명
resultCode		int	1	Y	처리결과 (1:성공, 0:실패)
성공	status	String	1	Y	전자증명서 요청결과 (F : 발급실패, W : 발급대기, S : 발급완료)
	certNum	String	19	N	유통증명서 일련번호
	serialNum	String	1~20	N	원본증명서 일련번호
	failMsg	String	256	N	전자증명서 발급실패 사유
실패	errCode	String	10	Y	오류코드
	errMsg	String	256	N	오류메시지

※ 전자증명서 발급이 완료된 경우 응답메시지는 multi-mime이며 certNum과 원본증명서.pdf 파일이 전달됨

※ 유통증명서와 원본증명서가 모두 발급 완료된 경우에만 status=S로 전달한다. 두 증명서 중 하나라도 실패 시 status=F로 응답하고, 발급대기 상태면 status=W로 응답한다.

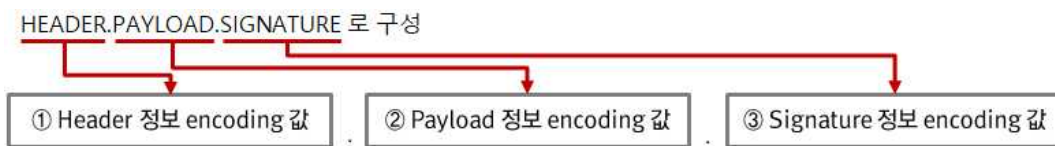
□ 응답메시지 예시

```
{
  "resultCode" : 1,
  "status" : "S",
  "certNum" : "abcd-0332-1234-5678",
  "serialNum" : "54854813187641"
}
```

A. 부록 - 이용자 인증토큰

1) 이용자 인증 관련 토큰 구조의 개요

- 이용자(송신자)가 중계자 시스템에 연계하기 전 인증을 위한 “이용자 인증토큰 발급 및 갱신” API에서 사용하는 토큰은 총 2가지로 “accessToken”, “refreshToken”이 있다.
- 토큰은 모두 JWT(JSON Web Token) 구조체로 기본 구조는 동일하다.
- 기본구조



2) 이용자 인증 관련 토큰의 항목별 구성내역

토큰의 구성항목인 Header, payload, signature의 값은 2개 토큰이 모두 동일한 구조를 가지고 있으며, 각 항목을 구성하는 정보구조는 다음과 같다.

□ Header 정보 encoding 값

- JWT를 검증하는데 필요한 정보로 typ(토큰 타입), alg(서명알고리즘), kid(검증에 사용할 공개키 식별값)로 구성됨
- Header 정보를 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

```
{
  "alg": "ES256"
  "typ": "JWT",
  "kid": "05c43a1f"
}
```

➔ header

```
HEADER = base64UrlEncode(header) =
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJraWQiOiIwNWc0M2E1ZiJ9.
```


□ Payload 정보 encoding 값

- Palyload는 토큰에 전달하고자 하는 값을 포함하는 파트임
- payload 정보 구성
(필수) iss(토큰발급자인 중계자 플랫폼-ID), sub(토큰수신자의 공인전자주소), iat(토큰 발급 시각),

exp(토큰 만료 시각)로 구성됨

(선택) aud, scope, clientId 및 중계자시스템 업무 수행에 필요한 사용자 정의 클레임(Custom Claim)등을 추가 가능

- 각 토큰에 따라 구성된 Payload 정보를 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용
- payload 예시

<pre>{ "iss" : "platform-01-app", "sub" : "eaddr123", "iat" : 1768010222, "exp": 1768010822, "aud": "document-api", "clientId": "client1234", "scope": "document:write", "serviceCode": "1154" }</pre>		<p>payload</p>
<p>PAYLOAD = base64UrlEncode(payload) = eyJpc3MiOiJwbGF0Zm9ybS0wMMS1hcHAiLCJzdWliOiJlYWRkcjEyMyIsImhhdCI6MTc2ODAxMDIyMiwiZXhwIjoxNzY4MDEwODIyLCJhdWQiOiJkb2N1bWVudC1hcGkiLCJjbGllbnRlZCI6ImNsaWVudDEyMzQyZyZyZSI6ImRvY3VtZW50OndyaXRlliwic2VydmliZUNvZGU0iixMTU0In0</p>		

□ Signature 정보 encoding 값

- 인코딩된 헤더와 페이로드를 점(.)으로 연결한 후 헤더에 정의된 알고리즘과 개인키를 이용하여 서명한 값으로 구성됨
- 서명한 값을 Base64 URL-Safe 인코딩된 문자열로 변환하여 사용

<p>SIGNATURE = ECDSASHA256(HEADER + "." + PAYLOAD, privateKey) = kVvMhYzZAYy8Fc8QMaGwTAYy8vmh-sZywBbi7MwDkWCx7QJvk4HqjUhrVFcIQ2ZEnp74He32hhUoB0sbidT3OQ</p>

□ 이용자 인증토큰 예시

- Header, Payload, Signature 값을 "."으로 연결한 값

B. 부록 -오류코드 정의

번호	코드 값	코드 값 정의	설명
1	ERR-03-101	송신자 인증번호 번호 오류	HTTP 헤더에 platform-id값이 없거나 유효하지 않음
2	ERR-03-102	송신자 인증 오류	발송기관의 clientId가 존재하지 않거나 clientSecret이 일치하지 않는 경우
3	ERR-03-103	송신자 인증토큰 오류	HTTP 헤더에 송신자 인증토큰 값이 없거나 유효하지 않은 경우
4	ERR-03-104	송신자 인증토큰 만료 오류	인증토큰 기간이 만료된 경우
5	ERR-03-201	요청메시지 구조 오류	요청메시지의 JSON 스키마가 구조가 맞지 않거나 값이 유효하지 않은 경우
6	ERR-03-301	공인전자주소 권한 없음	송신 공인전자주소에 대한 권한이 없을 경우
7	ERR-03-302	송신자 공인전자주소 미존재	송신 중계플랫폼에 공인전자주소가 없는 경우
8	ERR-03-304	수신자 공인전자주소 미존재	수신 중계플랫폼에 공인전자주소가 없는 경우
9	ERR-03-401	전자문서 발송요청 미존재	edocId에 해당하는 전자문서 발송 정보가 없는 경우
10	ERR-03-901	시스템 내부 오류	시스템 내부 처리 중 오류 발생할 경우
11	ERR-03-999	기타 오류	정의된 오류 이외의 오류

규격 연혁

버전	제 · 개정일	제 · 개정내역
v1.0	2026년 5월 22일	· 제정