

**유포금지**



---

# 디지털원패스 이용기관 연계가이드

---



**행 정 안 전 부**  
**디지털안전정책과**

개발가이드는 인터넷망에 유통되거나 외부로  
반출되지 않도록 보안 관리하여 주시기 바랍니다.

# 【 목 차 】

<b>1. 디지털원패스 개요</b>	<b>1</b>
1.1. 가이드 소개	1
1.2. 용어 정의	1
1.3. 서비스 정의	2
1.4. 서비스 이용현황	2
<b>2. 구성</b>	<b>3</b>
2.1. 디지털원패스 로그인	3
2.2. 디지털원패스 회원가입 절차	3
2.3. 디지털원패스 정부 웹사이트 이용	3
2.4. 연계구성도 흐름	4
2.5. 디지털원패스 로그인 순서도	5
<b>3. 디지털원패스 연계관련 행정절차</b>	<b>6</b>
<b>4. 디지털원패스 연계 준비</b>	<b>7</b>
4.1. 이용기관 사전 준비	7
4.2. 개발 및 운영환경 점검	8
<b>5. 연계 모듈 구성 및 환경파일 설정</b>	<b>9</b>
5.1. 배포 모듈	9
5.2. 배포 모듈 명세	9
5.3. 환경설정 파일	12
<b>6. 로그인 개발</b>	<b>13</b>
6.1. 디지털원패스 로그인 구현	13
6.2. 로그인 요청결과 처리 구현	16
<b>7. 회원정보 정합성</b>	<b>19</b>
7.1. 디지털원패스 이용기관 사용자 정합성 유지	19
<b>8. 기타 기능구현</b>	<b>21</b>
8.1. 회원가입 버튼 별도 구현	21
8.2. 로그인 완료 후 returnUrl 이동	21
<b>9. 테스트 절차</b>	<b>22</b>
<b>10. 코드 정의서</b>	<b>23</b>
10.1. 코드구분	23
10.2. 로그인	23
10.3. API - 디지털원패스 사용자 정보	24
10.4. API - 디지털원패스 이용기관 사용자 정합성 코드 정보	24
<b>11. FAQ</b>	<b>25</b>
11.1. 일반	25
11.2. 이용서비스 was로그 오류 문의 예제	27
11.3. 이메일 문의	28
11.4. 참고자료(붙임)	28

# 1 디지털원패스 개요

## 1.1 가이드 소개

본 문서는 디지털원패스 이용(연계)기관을 대상으로 “디지털원패스” 서비스를 이용기관 서비스에 적용하기 위한 표준 이용기관 연계 가이드입니다.

※ 디지털원패스는 WEB서비스 대상으로 연계기술지원을 하고, 연계기관의 모바일 어플리케이션(APP) 연동에 대한 별도의 가이드 및 기술지원은 제공하지 않습니다.

본 문서의 지침에 따라 디지털원패스를 디지털정부 이용서비스에 연계 작업을 진행하여 주시기 바랍니다. 디지털원패스 연계 확인 및 검수 과정에서 본 문서의 준용 여부를 확인하며 위배될 경우 서비스 오픈이 불가능 합니다. 본 가이드 및 <p.28> 11.4. 참고자료의 추가 가이드를 반드시 참고하시고 이외사항은 문의하여 주시기 바랍니다.

디지털정부 이용서비스 담당공무원 및 개발자는 본 가이드의 숙지가 필요하며 문의 시에 의사소통 매개로써 사용해 주시기 바랍니다.

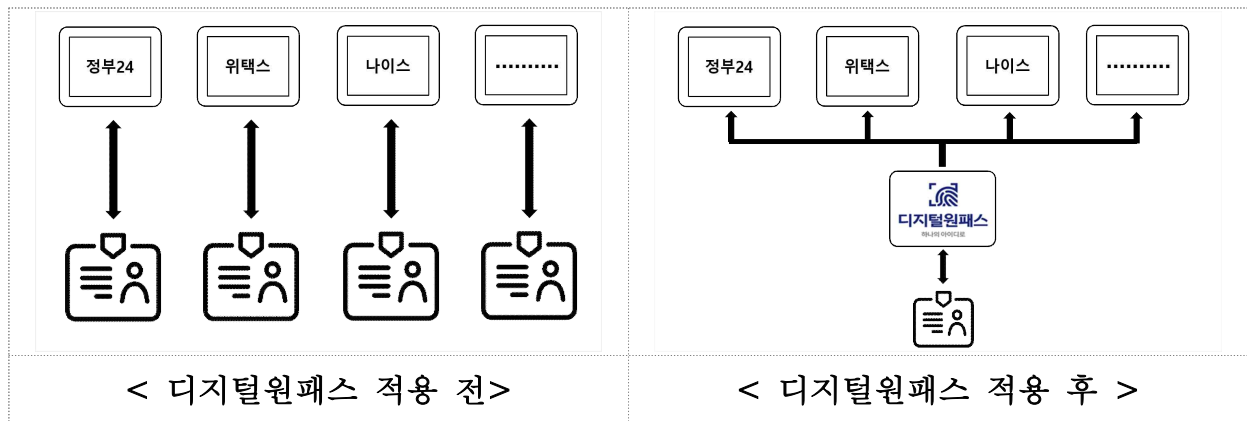
문의사항은 디지털원패스 기술지원 이메일(support@klid.or.kr)을 통하여 해주시기 바랍니다.

## 1.2 용어정의

명 칭	설 명
이용기관	디지털원패스를 도입한 전자정부 기관
이용서비스	디지털원패스를 도입한 전자정부 사이트(서비스)
회원가입	디지털원패스( <a href="https://www.onepass.go.kr">https://www.onepass.go.kr</a> ) 회원가입
회원연동	디지털원패스 회원의 이용기관 서비스 회원가입
회원탈퇴	디지털원패스( <a href="https://www.onepass.go.kr">https://www.onepass.go.kr</a> ) 회원탈퇴
회원연동해지	디지털원패스 회원의 이용기관 서비스 회원탈퇴

## 1.3 서비스 정의

- 국민이 공공사이트 이용 시, 사이트별 ID·패스워드 및 공동인증서 관리 등에 불편이 존재
- 하나의 ID로 선호하는 인증수단(SMS, 지문, 안면, 패턴, PIN 등)을 이용하여 여러 공공웹사이트를 안전하고 편리하게 이용할 수 있는 로그인서비스



## 1.4 서비스 이용현황

- 정부24, 위택스, 나이스 등 53개 기관 172개 사이트, 회원 가입자 약 153만명
- '21년 기준, 웹 대비 앱을 이용한 인증(약 161만건, 전체 약 57%)이 더 많고, 인증수단 중 웹 SMS(약 90만건), 앱 지문(약 69만건)인증 이용이 가장 많음

(’21.5.19 기준)

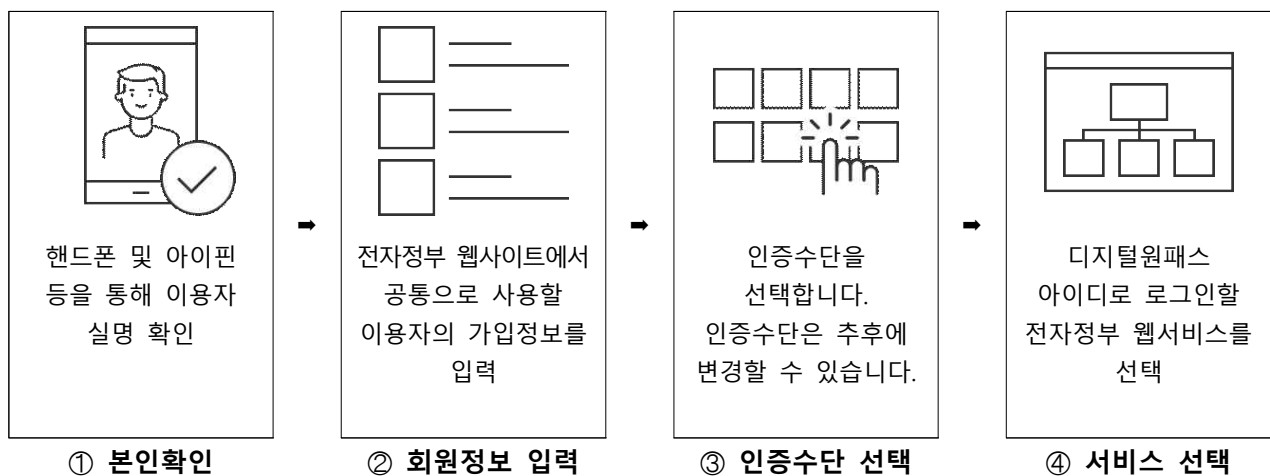
회원가입자 수	기관 수	연계사이트	월별 인증건수(’21년)		
			3월	4월	5월(~19일)
1,535,369	53	172	805,603	672,995	429,726

## 2 구성

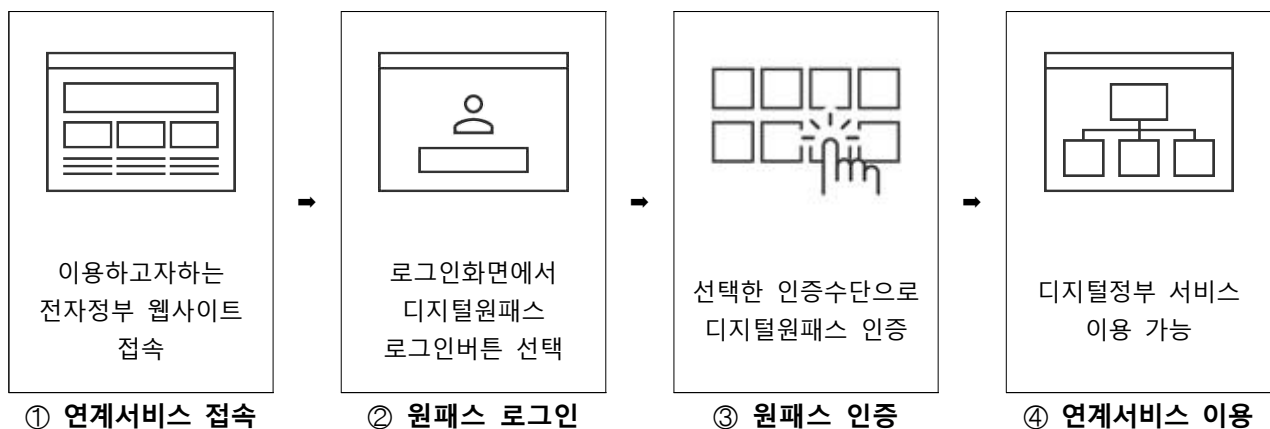
### 2.1 디지털원패스 로그인



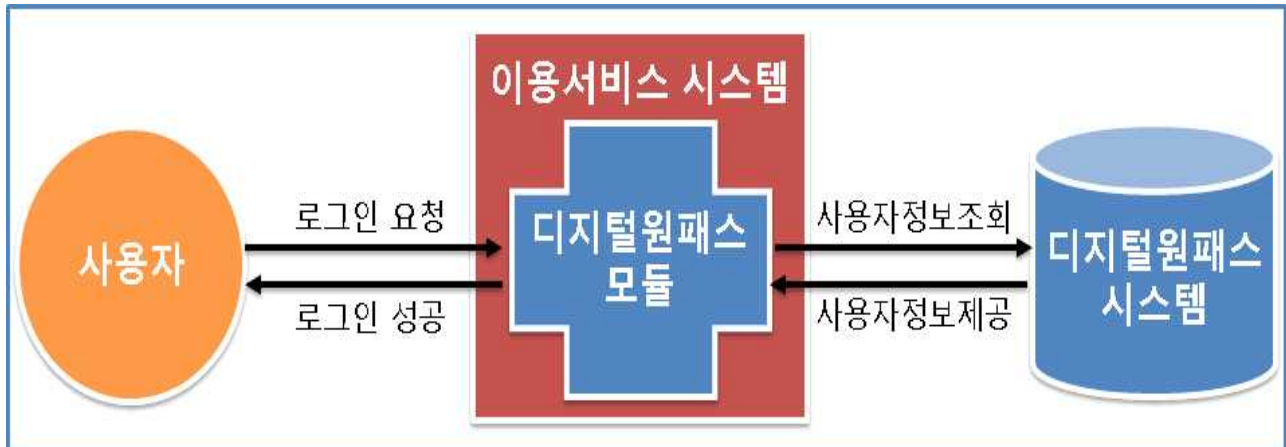
### 2.2 디지털원패스 회원가입 절차



### 2.3 정부 웹사이트 이용

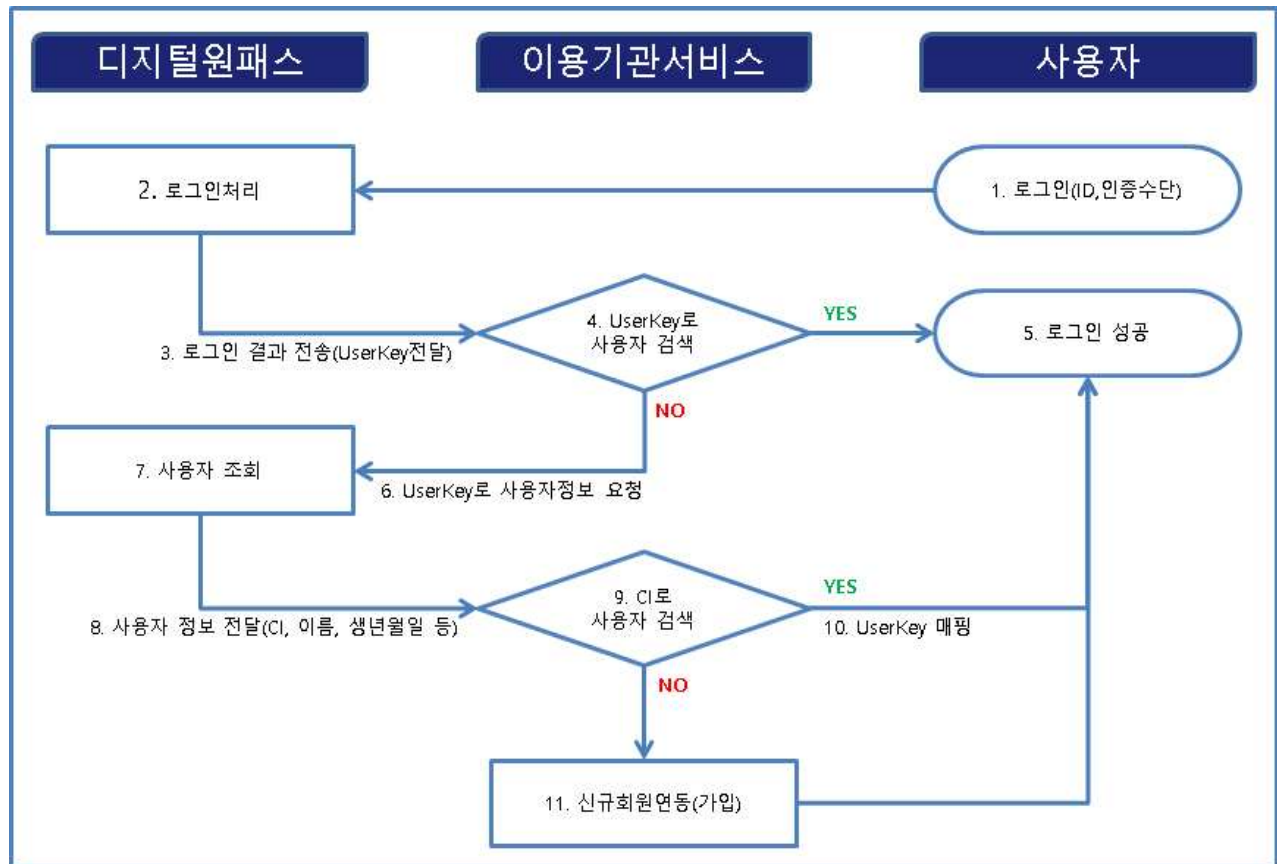


## 2.4 연계 구성도 흐름



구분	설명
사용자	전자정부 서비스를 이용하려는 사용자
이용서비스 시스템	디지털원패스를 도입한 전자정부서비스 시스템
디지털원패스 시스템	디지털원패스 사용자 로그인 인증 시스템

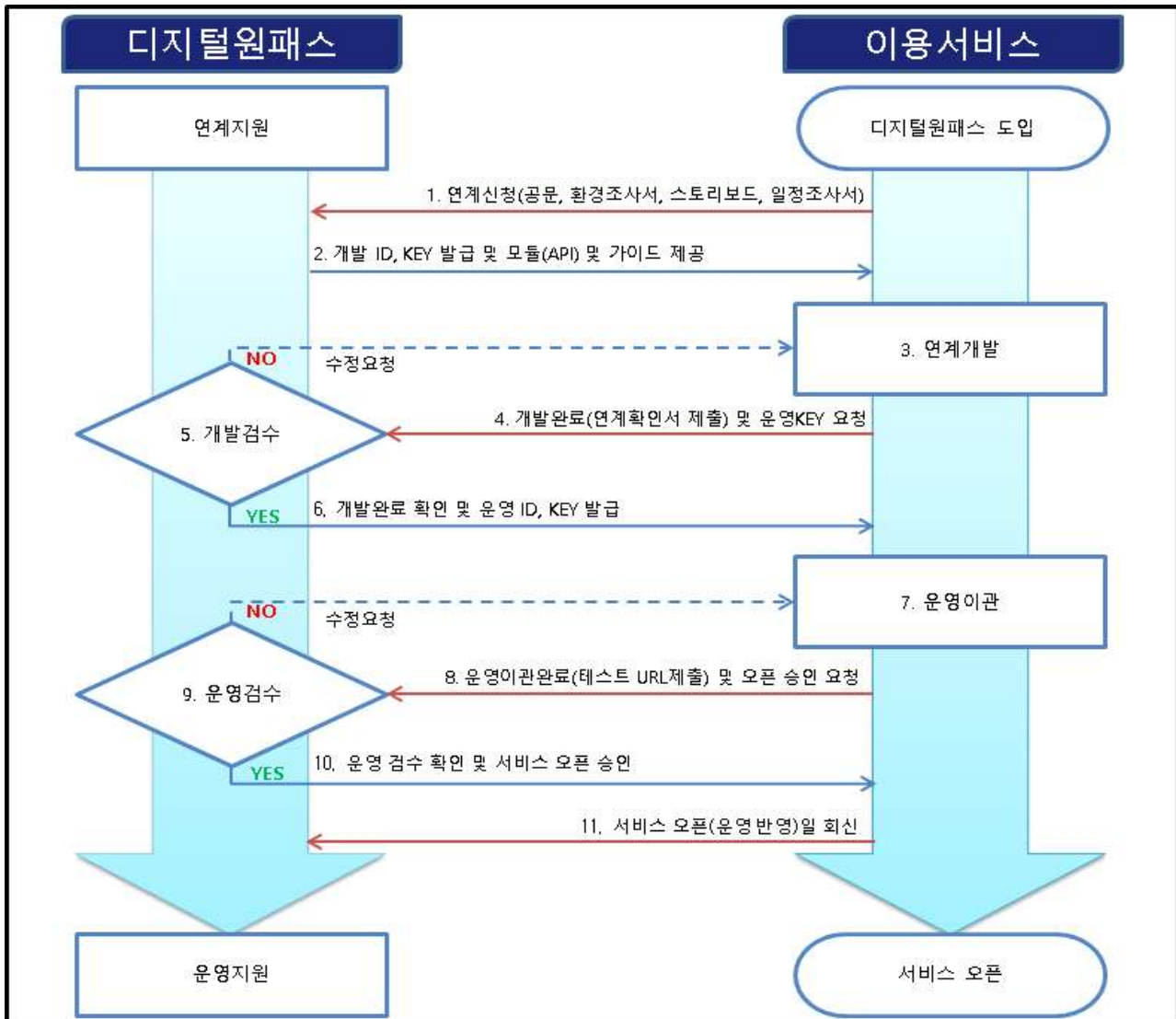
## 2.5 디지털원패스 로그인 순서도



번호	설명
1	사용자가 이용기관서비스 사이트에서 “디지털원패스로그인” 버튼을 클릭하여 로그인 시도.
2	디지털원패스 아이디 및 인증수단을 올바르게 인증하면 로그인 처리.
3	사용자가 인증을 성공하면 사용자의 UserKey를 이용기관 서비스에 전달. onepass.properties파일의 onepass.site.path.acs 항목에 해당하는 URL로 리턴.
4	전달받은 UserKey로 이용기관의 사용자 테이블을 검색.
5	UserKey와 일치하는 사용자가 있을 경우 그 사용자를 로그인 처리.
6	UserKey와 일치하는 사용자가 없을 경우 UserKey와 ApiToken을 사용해서 사용자정보 요청 ※ 사용자 상세정보는 디지털원패스 회원정보 조회 AP(Socket 통신)로 접근 ※ 반드시 4번의 USER KEY 조회를 우선적으로 수행 후 상세정보 요청.
7	사용자 상세정보 조회
8	사용자 상세정보(CI, 이름, 생년월일, 전화번호 등)를 이용기관 서비스에 전달.
9	사용자 상세정보(CI등)를 확인하여 이용기관의 사용자 테이블을 검색.
10	9번 항목에 CI가 존재할 경우, 해당사용자의 테이블에 UserKey를 매핑.
11	9번 항목에 CI가 없을 경우, 사용자 상세정보 및 UserKey를 이용기관에 저장(회원가입 처리)



### 3 디지털원패스 연계관련 행정 절차



번호	설명
1	연계신청(공문, 환경조사서, 스토리보드, 일정조사서)
2	개발 ID, KEY 발급 및 모듈(API) 및 가이드 제공
3	연계개발
4	개발완료(연계확인서 제출) 및 운영KEY 요청
5	개발검수
6	개발완료 확인 및 운영 ID, KEY 발급
7	운영이관
8	운영이관완료(테스트 URL제출) 및 오픈 승인 요청
9	운영검수
10	운영 검수 확인 및 서비스 오픈 승인
11	서비스 오픈(운영 반영)일 회신

## 4 디지털원패스 연계 준비

### 4.1 이용기관 사전 준비

#### 1. 디지털원패스 회원가입(개발서버 및 운영서버)

- 디지털원패스 개발서버에서 사용가능한 인증수단은 공동인증서 및 패스워드  
※ 모바일인증(APP), SMS(문자)인증 사용불가
- 디지털원패스 운영서버에서 사용가능한 인증수단은 모바일인증(APP), SMS(문자)인증, 공동인증서입니다(패스워드 사용불가)

#### 2. 디지털원패스 서비스에 필요한 IP, PORT 오픈(단방향) 및 호스트 등록

- 개발서버 정보(인터넷망)

구분	URL	IP	PORT	비고
PORTAL	<a href="https://www.onepass-dev.kr">https://www.onepass-dev.kr</a>	59.15.44.130	80 / 443	홈페이지
IDP	<a href="https://saml.onepass-dev.kr">https://saml.onepass-dev.kr</a>		80 / 443	인증페이지
AP	59.15.44.130		53010	사용자 정보조회 모듈

- 운영서버 정보(인터넷망)

구분	URL	IP	PORT	비고
PORTAL	<a href="https://www.onepass.go.kr">https://www.onepass.go.kr</a>	125.60.56.80	80 / 443	홈페이지
IDP	<a href="https://saml.onepass.go.kr">https://saml.onepass.go.kr</a>		80 / 443	인증페이지
AP	ap.onepass.go.kr	125.60.56.91	53010	사용자 정보조회 모듈

#### 3. 디지털원패스 서비스 연계방안 수립

- 로그인 화면구성, 회원연동절차(기존 및 신규 이용자), 정보수정 및 연동해지 등의 연계방안을 수립
- 참고자료(붙임)의 디지털원패스 상세 가이드 반드시 참고
- 디지털원패스 연계 검수 및 점검 시 표준 가이드 준용 여부를 확인하여 위배될 경우 서비스 오픈이 불가능 합니다. 제공 되는 가이드를 반드시 참고하시고 이외사항은 반드시 문의하여 주시기 바랍니다.

## 4.2 개발 및 운영환경 점검

### 1. 디지털원패스 연계를 위한 시스템 개발 및 운영환경

항목	내용	비고
OS	Java WAS 지원 환경 모두	
개발언어	Java (JDK 1.6 이상)	
WEB/WAS	모든 WEB/WAS 지원 (WAS는 JDK 1.6 이상 지원 환경)	
지원 웹브라우저	HTML5를 지원하는 모든 브라우저 호환 (IE 11이상)	Edge, Chrome 등

※ 상기 명시된 규격 외의로 구현된 시스템은 연계지원 불가. 단, 이용기관 자체구현은 가능

### 2. 참고사항

#### - 톰캣 8.5 이상 Cookie 정책

Tomcat8.5 이상 에서 도메인이 .(dot)으로 시작하는 쿠키를 생성할 수 없도록 변경됨.  
(RFC 6265 Cookie Processor 정책)

#### - 해결책

기존 LegacyCookieProcessor를 사용하도록 context.xml에 코드 추가  
<CookieProcessor className="org.apache.tomcat.util.http.LegacyCookieProcessor" />

## 5 연계모듈 구성 및 환경파일 설정

### 5.1 배포 모듈

항목	내용	비고
onepass-client-2.0.0.jar	연계 모듈	Library 저장 폴더
onepass.properties	연계 모듈 설정 파일	classpath 최상단 위치
onepass-client-1.0.0.js	연계API javascript	javascript 배포위치 (옵션)
pubkey_share.der	서명검증 파일(공개키)	임의의 디렉토리 설치 후 onepass.properties 파일에 절대경로 PATH설정
onepass.jks	SSL 인증서 키 체인 스토어	임의의 디렉토리 설치 후 onepass.properties 파일에 클래스패스 기준으로 상대경로 PATH설정
jquery-1.12.4.min.js	외부라이브러리 javascript	javascript 배포위치 (onepass-client-1.0.0.js 사용하는 경우, jquery 미사용시)
loginout.jsp	로그인/로그아웃 모듈	/webapp/onepass/ ※로그인 버튼 직접 구현시 사용안함
loginout_h.jsp	로그인/로그아웃 모듈 (옵션, 높은레벨의 인증수단 요구시 사용)	/webapp/onepass/ ※로그인 버튼 직접 구현시 사용안함

### 5.2 배포 모듈 명세

#### ○ onepass-client-1.0.0.js

클래스 명	메소드	설명
OnepassClientUtil	loadLoginBtn	인자로 전달된 id를 가진 div에 로그인 버튼을 생성

#### ○ onepass-client-2.0.0.jar

클래스 명	메소드	설명
OnepassResponseHandler	check	Callback 호출된 request를 전달하면 전문을 해석하여 OnepassResponse 클래스를 리턴한다.

클래스명	메소드	설명
OnepassResponse	getStatus	응답전문의 상태를 확인한다. 코드정의의 로그인 중 status를 참고한다
	getType	'LOGIN / LOGOUT'
	getResultCode	실패 시 원인이 저장된다. 서버측 오류. 코드정의의 로그인 중 resultCode를 참고한다
	getInvalidCode	실패 시 원인이 저장된다. 전문 오류 코드정의의 로그인 중 resultCode를 참고한다
	getDetailErrorMessage	실패 시 상세 원인이 저장된다.
	getUserKey	사용자 UserKey가 전달 된다.
	getCrtCode	사용자의 인증수단 코드정의의 로그인 중 crtCode를 참고한다
	getIntfToken	사용자가 로그인한 세션의 토큰값. 사용자가 로그아웃 하게 되면 유효하지 않음.
	getResponse	응답전문 전체를 가져옴.
	getFindUser	사용자 개인정보 변경 여부
ApiSendHandler	findUser	개인정보를 조회하기 위한 클래스이다. response에서 받은 intfToken 과 userKey 가 필요하다.
ApiSendHandler	interLockRelease	사용자 디지털원패스 연동해지 또는 탈퇴시에 사용한다. response에서 받은 intfToken 과 userKey 가 필요하다.

클래스명	메소드	설명
OnepassUserResponse	getStatus	응답전문의 상태를 확인한다. 코드정의의 사용자정보 중 status를 참고한다
	getUserKey	사용자 UserKey가 전달된다. (최대 50자)
	getBirth	사용자의 생년월일 (최대 8자)
	getCi	사용자의 CI (최대 255자)
	getEmail	사용자의 E-mail (최대 70자)
	getName	사용자의 이름 (최대 70자)
	getPhone	사용자의 휴대전화번호 (최대 14자)
	getId	사용자의 ID (최대 30자)
	getSex	사용자의 성별 (남:M, 여:F)
	getNation	내외국인 구분 (내국인:L, 외국인:F)

※ UserKey는 사용자가 디지털원패스 포털에 회원가입 시 생성되며, 회원탈퇴 시 삭제됩니다.  
디지털원패스 사용자가 탈퇴 후 재가입하게 되면 UserKey값이 바뀌게 됩니다.

## 5.3 환경설정 파일

### ○ onepass.properties

```
## Site info ##
# 붉은색 텍스트만 수정#

# Site URL : 사이트의 URL
onepass.site.url=<사이트 URL 입력>

# Site ID : 사이트의 ID(디지털원패스에서 발급)
onepass.site.id=<디지털원패스에서 제공되는 ID 입력>

# AES secretKey(length:16) : AES 암호화 키(디지털원패스에서 발급)
aes.secretKey=<디지털원패스에서 발급하는 16자리 키 입력>

# PublicKey path(full path) :
keys.public.path=<pubkey_share.der 파일의 절대경로 지정 입력>

#SSL 인증 키스토어 파일경로(클래스패스 기준 상대경로 지정)
ssl.keystore=<onepass.jks 파일의 상대경로(classpath부터 시작하는 경로) 입력>

# ACS path : 디지털원패스 로그인 완료 후 응답받은 callback URL
# 테스트 후 실제 사이트의 로그인/아웃 처리 URL 경로로 변경
onepass.site.path.acs=/onepass/test/acs.jsp

# IDP URL : 디지털원패스 IDP URL(개발:https://saml.onepass-dev.kr, 운영:https://saml.onepass.go.kr)
onepass.idp.url=https://saml.onepass-dev.kr

# Potal URL : API 서버 URL(개발:https://www.onepass-dev.kr, 운영:https://www.onepass.go.kr)
onepass.potal.url=https://www.onepass-dev.kr

# AP URL : AP 서버 URL(개발:59.15.44.130, 운영:ap.onepass.go.kr)
onepass.socket.host=59.15.44.130
onepass.socket.port=53010
```

## 6 로그인 개발

### 6.1 디지털원패스 로그인 구현

#### ○ 로그인 전 view.jsp (case #1 - 직접 버튼 생성)

```
<!-- 직접 구현시는 jquery, onepass js 추가할 필요 없음 -->
.
.
# login button 위치에 div 삽입.

<div id="onepassLogin2" style="position:absolute; right:28px; top:272px;">
  <a href="/onepass/loginout.jsp?serviceType=LOGIN">
    
  </a>
</div>
```

jQuery, onepass-client.js 없이 디지털원패스 버튼 구현시 loginout.jsp를 호출하는태그(serviceType = LOGIN 파라미터 추가)를 만든다.

#### ○ 로그인 전송 직접구현 (case #2 - loginout.jsp 사용하지 않는 경우)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="kr.go.onepass.client.handler.saml.OnepassRequestHandler" %>
.
.
# login button 위치에 div 삽입.

<div id="onepassLogin3" style="position:absolute; right:28px; top:372px;">
  <form id="request" method="post" action="<%=OnepassRequestHandler.LOGIN_DEST%>">
    <input type="hidden" name="<%=OnepassRequestHandler.LOGIN_INPUT_NAME%>"
      value="<%=OnepassRequestHandler.login()%>" />
    <button type="submit">
      
    </button>
  </form>
</div>
```

jQuery, onepass-client.js, loginout.jsp없이 디지털원패스 로그인 버튼 구현시 loginout.jsp파일에 있는 request form을 직접 구현 후에 로그인 버튼 클릭시 submit한다.



## ○ 임베디드 구현 (case #3 - 임베디드)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
session="true" %>
<%@ page import="kr.go.onepass.client.handler.saml.OnepassRequestHandler" %>
```

```
<head>
<%
    String action = OnepassRequestHandler.logoutDest(request);
    String inputName = OnepassRequestHandler.LOGOUT_INPUT_NAME;
    String inputValue = OnepassRequestHandler.logout(request);
    String pageType = OnepassRequestHandler.pageType(request);
    String JSESSIONID = request.getSession().getId();
    action += ";JSESSIONID="+JSESSIONID;
%>
<script type="text/javascript" src="/onepass/test/js/jquery.min.js"></script>
<script type="text/javascript">
    jQuery(document).ready(function() {
        jQuery.ajax({
            url : "<%=action%>",
            type : "POST",
            data: $('#logOutRequest').serialize(),
            withCredentials: true,
            success : function(res) {
                console.log(res + " 로그아웃 성공");
            },
            error : function() {
                console.log("로그아웃 실패");
            }
        });
    })
    function onClickSubmit() {
        var form = document.request;
        var user_id = form.user_id.value;

        if ( user_id == null || user_id == "" ) {
            alert("아이디를 입력해 주세요.");
            form.user_id.focus();
            return;
        }
        form.submit();
    }
</script>
</head>
```

```
<div id="onepassLogin3" style="position:absolute; right:28px; top:372px;">
  <form id="logOutRequest">
    <input type="hidden" name="<%=inputName%>" value="<%=inputValue%>" />
    <input type="hidden" name="pageType" value="<%=pageType%>" />
  </form>
  <form id="request" name="request" method="post"
action="<%=OnepassRequestHandler.LOGIN_DEST%>"
  <input type="hidden" name="<%=OnepassRequestHandler.LOGIN_INPUT_NAME%>"
value="<%=OnepassRequestHandler.login()%>" />
  <div>
    <input type="text" id="user_id" name="user_id" style="ime-mode:disabled;"
placeholder="디지털원패스 아이디" title="디지털원패스 아이디" onkeydown="if(event.keyCode
== 13){onClickSubmit(); return false;};">
  </div>
  <div>
    <a href="javascript:onClickSubmit();">
      </a>
    </div>
  </form>
</div>
```

## 6.2 로그인 요청결과 처리 구현

- 환경설정 파일의 ACS Path에 요청결과 처리 구현
- 성공을 제외한 CASE는 각 사이트의 프로세스에 따라 처리

```

OnepassResponse onepassResponse = OnepassResponseHandler.check(request);
// 응답전문의 상태
onepassResponse.getStatus();

// 응답전문의 타입 : 로그인/로그아웃
onepassResponse.getType();

// 응답전문의 결과코드
onepassResponse.getResultCode();

// 응답전문의 오류 코드
onepassResponse.getInvalidCode();

// 응답전문의 상세 오류 메세지
onepassResponse.getDetailErrorMessage();

// 디지털원패스 사용자 키 (최대 50자)
onepassResponse.getUserKey();

// 디지털원패스 사용자 API 토큰
onepassResponse.getIntfToken();

// 응답전문 전체
onepassResponse.getResponse();

// 사용자 개인정보 변경 여부
onepassResponse.getFindUser();

```

디지털원패스 로그인 완료 후 응답받은 callback URL의 servlet 또는 jsp에서 구현한다.  
OnepassResponseHandler.check 메소드의 파라미터로 request를 전달 하면 OnepassResponse 클래스를 반환 하며 원하는 값을 가져온다.

```
if (STATUS.SUCCESS == onepassResponse.getStatus()) {  
    switch (onepassResponse.getResultCode()) {  
        case STOP_USER:  
            // TODO : 정지된 사용자, 로그인 불가능 처리  
            break;  
        case SUCCESS:  
            switch (onepassResponse.getType()) {  
                case LOGIN:  
                    // TODO : 로그인 성공 process  
                    break;  
                case UNKNOWN:  
                default:  
                    break;  
            }  
            break;  
        case IDP_ERROR:  
        case NO_SUPPORTED_SITE:  
        case SITE_JOIN_REJECT:  
        default:  
            break;  
    }  
} else { // 예외처리 }
```

로그인한 사용자의 onepassResponse.getStatus()와 onepassResponse.getResultCode()가 모두 성공인지 확인한다.

```

/* onepassResponse.getUserKey()를 이용하여 사이트의 사용자 조회
* 사용자가 없는경우 디지털원패스 API를 통하여 사용자 정보 조회.
* 조회된 사용자 정보를 사이트의 사용자 정보에 추가하여 이후에
* API를 사용하지 DB조회를 통하여 사용자 확인
* 디지털원패스 API 사용을 위한 쿠키 추가
*/
// 사용자 정보 조회 API
String userKey = onepassResponse.getUserKey();
String intfToken = onepassResponse.getIntfToken();

ApiSendHandler apiSendHandler = new ApiSendHandler();
OnepassUserResponse findOnepassUser = apiSendHandler.findUser(userKey, intfToken);

if (findOnepassUser != null && findOnepassUser.getStatus() == USER_STATUS.USE) {
    logger.info("getUserKey:{", findOnepassUser.getUserKey());
    logger.info("getBirth:{", findOnepassUser.getBirth());
    logger.info("getCi:{", findOnepassUser.getCi());
    logger.info("getEmail:{", findOnepassUser.getEmail());
    logger.info("getName:{", findOnepassUser.getName());
    logger.info("getPhone:{", findOnepassUser.getPhone());
    logger.info("getId:{", findOnepassUser.getId());
    logger.info("getSex:{", findOnepassUser.getSex());
    logger.info("getNation:{", findOnepassUser.getNation());
} else {
    /* TODO : 정보조회 오류, 로그인 불가 처리 */
}

```

디지털원패스 로그인 성공 후 사용자의 상세정보를 받을때는 ApiSendHandler.findUser 메소드를 사용한다. findUser 메소드의 파라미터로 callback URL의 servlet 또는 jsp에서 응답받은 UserKey와 IntfToken이 필요하다.

findUser 메소드의 결과로 OnepassUserResponse클래스를 반환하며 OnepassUserResponse 클래스의 get메소드를 확인하면 사용자의 상세정보를 확인할 수 있다.

※ IntfToken은 사용자의 디지털원패스 세션이 유효한시간(30분) 동안만 유효한 값이다.

## 7 회원정보 정합성

### 7.1 디지털원패스 이용기관 사용자 정합성 유지

#### ○ 사용자가 이용기관 내에서 연동해지 선택할 경우

```
// 디지털원패스로 연동해지 요청한다.
// 선택한 기관만 연동해지 처리되며 사용자의 디지털원패스 계정은 유지된다

/*
    사용자 디지털원패스 연동 해지 또는 탈퇴시 사용한다.
    디지털원패스로 해당 정보 전송한다.
*/
ApiSendHandler apiSendHandler = new ApiSendHandler();

String userKey = request.getParameter("userKey");
String intfToken = request.getParameter("intfToken");
OnepassUserResponse onepassUser = apiSendHandler.InterLockRelease(userKey, intfToken);
if (onepassUser == null) {
    onepassUser = new OnepassUserResponse();
}

// 연동해지 요청 결과
<td><%=onepassUser.getProcess_result() %></td>
```

사용자 계정이 디지털원패스와 이용기관의 자체 계정이 연동되어 있는경우에 사용자가 디지털원패스 계정 연동을 해지 하길 원하거나 이용기관 자체 계정을 탈퇴하길 원하는 경우에 디지털원패스 연동해지 로직이 수행된다. userkey와 intfToken값을 디지털원패스로 전송하면 디지털원패스 홈페이지 로그인 후 확인되는 나의 서비스 카테고리 이용서비스 정보 목록에 등록되었던 이용서비스가 삭제된다.

CASE 1 : 사용자가 이용기관에 디지털원패스 계정으로 로그인한 경우 사용자가 로그인 성공 후 전달받은 userKey와 intfToken을 ApiSendHandler.InterLockRelease 메소드의 파라미터로 전달한다.

리턴 타입은 OnepassUserResponse클래스 이며 OnepassUserResponse.getProcess\_result() 메소드로 결과를 확인 할 수 있다.

CASE 2 : 사용자가 이용기관의 자체 계정으로 로그인한 경우

사용자가 이용기관의 자체 계정으로 로그인한 경우에 디지털원패스 연동해지를 하기 위해서는 userKey와 intfToken가 필요하다. intfToken를 얻기 위해서는 디지털원패스 로그인을 해야된다.

※ 디지털원패스 로그인 완료 후 **acs url(onepass.properties파일의 onepass.site.path.acs 항목, 디지털원패스 로그인 완료 후 리턴되는 URL)에서** 사용자가 단순 디지털원패스 로그인을 했는지, 연동해지를 위해서 로그인을 했는지 구분할수 있어야 된다. 이때 디지털원패스의 returnUrl을 사용하면 쉽게 구분할 수 있으며 returnUrl의 사용법은 "8. 기타 기능구현 - 나. 로그인 완료 후 returnUrl 이동"을 참고하면 된다.

## 8 기타기능 구현

### 8.1 회원가입 버튼 별도 구현

- 회원가입 버튼 별도 구현시 parameter 추가
  - 이용기관 메인페이지로 이동
    - `https://www.onepass.go.kr/regist/agree?redirectSiteSeq=<기관siteSeq>`
    - ※ site seq 추가시 회원가입 완료 후 이용기관 메인페이지로 이동
  - 별도의 원하는 페이지로 이동
    - `https://www.onepass.go.kr/regist/agree?redirectSiteUrl=<URL>`
    - ※ `http://www.xxx.com/targetServletUrl` 형식으로 parameter 추가
- parameter 우선순위
  - parameter 없을 경우 디지털원패스 메인 홈페이지 이동
  - redirectSiteSeq, redirectSiteUrl 모두 있을 경우 URL로 이동

### 8.2 로그인 완료 후 returnUrl 이동

- loginout.jsp의 request form에 이동할 url hidden 값 추가
  - 예) `<input type="hidden" name="returnUrl" value="/onepass/test/test?abc=123&bcd=456"/>`
- ACS(onepass.properties의 onepass.site.path.acs항목) 확인
  - ※ `request.getParameter("returnUrl");` 확인
  - 예) 

```
if (returnUrl != null && returnUrl.length() > 0) {
    // 페이지 이동시 필요한 데이터 설정
    ...
    // returnUrl 이동
    response.sendRedirect(returnUrl);
}
```

## 9 테스트 절차

### 9.1 테스트 절차

#### ○ 테스트 사전 작업

- 디지털원패스 개발서버 와 운영서버는 각각 개별로 회원 가입을 하여야 합니다.
- 디지털원패스 개발서버(<https://www.onepass-dev.kr>)에서는 인증수단으로 패스워드, 공동인증서 2개만 사용 가능(모바일 인증, SMS 인증 불가)합니다.
- 디지털원패스 운영서버 (<https://www.onepass.go.kr>)에서는 인증수단으로 모바일(APP)인증, 공동인증서, SMS(문자) 사용 가능(패스워드 사용 불가)합니다.

#### ○ 개발자 모듈 테스트

- 개발자가 배포 받은 디지털원패스 모듈을 로컬 PC 또는 이용 서비스의 개발서버에 적용하여 디지털원패스의 개발서버(<https://www.onepass-dev.kr>)와 연계를 진행하여 디지털원패스 로그인 및 사용자 조화가 정상적으로 실행 되는지 모듈 테스트를 진행 합니다.

#### ○ 개발환경 테스트

- 연계 개발 완료 후 개발 환경 간의 연계테스트(<https://www.onepass-dev.kr>)를 진행하며 이용서비스 기관에서 테스트를 진행합니다. 업무로직 반영에 문제가 없는지 등을 테스트 합니다. 테스트 완료 후 연계확인서(개발)를 작성하여 디지털원패스에 전달하여 검수를 받습니다. 수정사항 있을 시 디지털원패스에서 수정개발요청서를 작성하여 수정 요청을 할 것이며 개발 수정 후 다시 연계 확인서(개발)를 작성하여 디지털원패스에 회신하여 주시면 됩니다.

#### ○ 운영환경 테스트

- 운영이관 완료 후 디지털원패스가 가이드에 맞게 잘 개발 되었는지, 연동이 잘 이루어지는지 실제 이용서비스의 운영시스템에서 실기 테스트를 진행합니다. 이 때 일반 사용자는 접근할 수 없도록 로그인 페이지를 별도 또는 숨김으로 구성하여 해당 URL을 디지털원패스에 전달하여 운영테스트 검수를 받습니다



## 10 코드정의서

### 10.1 코드 구분

구분		설명
로그인	status	응답 데이터 체크 결과 상태
	type	응답 데이터 타입
	resultCode	응답 결과 값 코드
	invalidCode	데이터 유효성 오류 코드
	detailErrorMessage	유효성 검사 실패 상세 원인
	errorCode	로그인 인증수단
API 사용자 정보	status	디지털원패스 사용자 상태
정합성 코드	findUser	사용자 개인정보 변경 필요 여부

### 10.2 로그인

구분	값	설명	
status	SUCCESS	이상없음	성공 처리
	UNKNOWN_ERROR	알 수 없는 오류	오류 처리 (전문 처리시 오류)
	SIGN_ERROR	서명 확인 불가	
	PARAMETER_NOT_FOUND	응답 데이터 없음	
	PARAMETER_ERROR	응답 데이터 오류	
	PARSING_ERROR	응답 XML 파싱 오류	
	VALIDATE_ERROR	데이터의 유효성 오류 (상세정보는 invalidCode참조)	
type	LOGIN	로그인	
resultCode	SUCCESS	성공	성공 처리
	IDP_ERROR	디지털원패스 오류	원패스 처리 중 오류
invalidCode	UNKNOWN	알수없음	
	ELEMENT_EMPTY	필수 항목이 없음	
	ELEMENT_NOT_EQUALS	필수 항목이 일치하지 않음	
	ATTRIBUTE_EMPTY	필수 속성이 없음	
	VALUE_EMPTY	필수 속성 값이 없음	
	VALUE_NOT_EQUALS	필수 항목 값이 일치하지 않음	
	VALUE_INVALID	필수 항목 값이 유효하지 않음	

detailErrorMessage	Conditions.NotBefore	이용기관 WAS시간 동기화	유효성 검사 실패 상세 원인
	Conditions.NotOnOrAfter	이용기관 WAS시간 동기화	
crtCode	CRT1000	패스워드	
	CRT3000	공동인증서(구 공인인증서)	
	CRT5000	모바일 인증 - 지문	
	CRT5100	모바일 인증 - 안면	
	CRT5200	모바일 인증 - 패턴	
	CRT5300	모바일 인증 - 핀	
	CRT5400	모바일 인증 - 공동인증서(지문·안면)	
	CRT5900	모바일 인증	
	CRT7100	SMS(문자 인증)	

### 10.3 API-디지털원패스 사용자 정보

구분	값	설명	처리
status	USER_STATUS.USE	정상	정상처리
	USER_STATUS.STOP	정지	오류처리
	USER_STATUS.OUT	탈퇴	
	USER_STATUS.NOT_FOUND	없음	
	USER_STATUS.UNKNOWN	오류	

### 10.4 API-디지털원패스 이용기관 사용자 정합성 코드 정보

구분	값	설명	처리
findUser	NEED	정보 변경 필요 (사용자 정보 변경 이력 존재)	ApiSendHandler.findUser() 조회 후 사용자 개인정보 업데이트
	NEEDLESS	정보 변경 불필요	

## 11 FAQ

### 11.1 일반

#### ○ 현재 서비스에 https가 적용되어 있지 않습니다. 연계 가능한가요?

- 디지털원패스에서 전달하는 값 중 중요데이터는 암호화하여 전달되고, SAML 표준에 맞추어 서명이 이루어져서 무결성이 검증됩니다.
- 데이터는 안전하게 전송이 되나 더욱 안전한 통신을 위해 [https 적용을 권고](#)합니다.

#### ○ 디지털원패스 회원이 아닌 경우의 프로세스가 궁금합니다.

- 이용기관 입장에서는 디지털 원패스 회원가입 프로세스에 대해서는 신경 쓰실 필요가 없습니다.
- 다만, 디지털원패스를 이용하여 이용기관의 서비스를 이용하고자 하는 사용자를 위해서 로그인영역에 [디지털원패스 회원가입 링크](#)를 걸어두시는걸 권고합니다.
- 디지털원패스 회원이 아닌 경우 회원연동(회원가입) 프로세스를 거친 후 일반 로그인과 같이 처리 되므로 로그인 처리만 하시면 됩니다.

#### ○ 디지털원패스에서 제공하는 정보 외의 추가 정보가 필요합니다.

- 디지털원패스에서는 최소한의 개인정보만을 수집하고 있습니다.
- 업무상 추가 개인정보가 더 필요한 경우에는 사용자에게 개인정보를 입력받는 프로세스가 [이용기관 측 업무로직에 추가](#)되어 입력받아야 합니다.
- 단, ID 및 PW는 수집을 금지하고 있으니 참고하여 관리 부탁드립니다.

#### ○ 전자정부 프레임워크를 사용하지 않는 경우에도 지원이 되나요?

- 디지털원패스 라이브러리는 전자정부 프레임워크 [의존성 없이](#) 사용이 가능합니다.

#### ○ 개발 / 운영 환경이 인터넷망 연결이 되어 있지 않습니다.

- 디지털원패스는 대국민서비스이며, 인터넷망에서 서비스 되고 있습니다.
- 방화벽 해제와 망 연결 등을 통해 [인터넷망에 통신이 가능해야 서비스](#)됩니다.

#### ○ 기관 홈페이지는 PHP로 개발하였습니다. 언어가 JSP가 아니어도 연계 가능한가요?

- 불가능합니다

○ 우리기관에서는 CI는 보유하고 있지 않고 DI만 보유하고 있습니다.  
이런 경우 디지털원패스 연계가 가능한가요?

- 네 가능합니다.
- 다만, CI가 없으므로 **디지털원패스 회원과 기관 회원이 동일한 사람**인지 확인하기 위해 본인확인 단계를 거쳐야 합니다.

○ 수요조사 때, 연계일정을 '22년도로 제출하였는데  
기관사정상 '21년(올해)로 변경하고 싶은데 가능한가요?

- 네 가능합니다.
- 다만, 기존에 확정된 연계 기관이 있기 때문에 자세한 일정은 협의 부탁드립니다.  
최대한 일정 맞출 수 있도록 지원하겠습니다.

○ 회원과 비회원의 차이가 궁금합니다

- (회원) 기관에서 회원정보를 관리하는 경우 회원 연계라고 이야기하고 있습니다.
- (비회원) 회원정보를 관리하지 않는 경우를 이야기 합니다.  
예를 들어 **게시판 형태의 민원서비스**가 있습니다.

○ 디지털원패스를 “본인확인” 및 “전자서명”용으로 사용하고 싶습니다. 가능한가요?

- 불가능합니다. 디지털원패스는 **“통합로그인 서비스”**입니다.

○ 우리기관은 내부 정책상 ID 및 PW를 꼭 사용해야 합니다.  
이런 경우 어떻게 연계가 진행이 되나요?

- 불가능합니다. 정확한 상황 판단을 위해서 협의 부탁드립니다.

○ 우리기관은 회원의 CI도 없고 DI도 없습니다.  
이런 경우 ID 혹은 다른 정보로 연계가 가능 한가요

- 불가능합니다.
- 기본적으로 CI를 통해서 연계를 진행하고 있습니다.

## 11.2 이용서비스 was로그 오류 문의 예제

### ○ was로그 'socketTimeoutException : connection timed out' 확인 시

```
2020-06-05 10:40:10 INFO h.hp.onepass.web.HpOnepassController - RequestURI : /hp/onepass/userInfo.do
java.net.SocketTimeoutException: connect timed out
    at java.net.PlainSocketImpl.socketConnect(Native Method)
    at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
    at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
    at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
    at java.net.Socket.connect(Socket.java:589)
    at kr.go.onepass.client.utils.SocketRequestUtil.call(SocketRequestUtil.java:47)
    at kr.go.onepass.client.handler.api.ApiSendHandler.findUser(ApiSendHandler.java:159)
    at hlda.hp.onepass.web.HpOnepassController.userInfo(HpOnepassController.java:48)
    at sun.reflect.GeneratedMethodAccessor301.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
```

- 이용서비스 was와 디지털원패스 ap 서버간 소켓통신이 안되는 현상입니다.
    - onepass.properties의 ap서버 설정이 올바르게 작성됐는지 확인 합니다
    - 해당기관의 방화벽 정책에 ap서버 ip, port 허용여부를 확인 합니다
    - 이메일 문의 시 필수 첨부파일(3종)
      - ① was log ② properties파일 ③ 방화벽 오픈된 증적화면(53010포트 텔넷 접속)
      - ④ acs.jsp, userInfo.jsp페이지에 확인되는 코드
- <p.28> 11.3 이메일 문의 참고

### ○ was로그 '500 for URL : https://www.onepass.go.kr/api/user' 확인

```
java.io.IOException: Server returned HTTP response code: 500 for URL: https://www.onepass.go.kr/api/user
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream0(HttpURLConnection.java:1900)
    at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1498)
    at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(HttpsURLConnectionImpl.java:268)
    at kr.go.onepass.client.utils.HttpsRequestUtil.call(HttpsRequestUtil.java:106)
    at kr.go.onepass.client.handler.api.ApiSendHandler.findUserWeb(ApiSendHandler.java:119)
    at kr.go.onepass.client.handler.api.ApiSendHandler.findUser(ApiSendHandler.java:162)
    at egovframework.nlip.process.mn.service.impl.EgovLoginServiceImpl.getUserCi(EgovLoginServiceImpl.java:299)
    at egovframework.nlip.process.mn.service.impl.EgovLoginServiceImpl.onepassLogin(EgovLoginServiceImpl.java:141)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

- 이용서비스 was와 디지털원패스 ap 서버간 소켓통신이 안되는 현상입니다.
    - onepass-client-2.0.0.jar 파일이 다른 위치에 있는지 확인 합니다
    - 해당기관의 방화벽 정책에 ap서버 ip, port 허용여부를 확인 합니다
    - 이메일 문의 시 필수 첨부파일(3종)
      - ① was log ② properties파일 ③ 방화벽 오픈된 증적화면(53010포트 텔넷 접속)
      - ④ acs.jsp, userInfo.jsp페이지에 확인되는 코드
- <p.28> 11.3 이메일 문의 참고

## 11.3 이메일 문의(support@klid.or.kr)

### ○ 이메일 문의 전 체크 사항

- 방화벽 오픈 확인
  - 개발 : 59.15.44.130 port:53010
  - 운영 : ap.onepass.go.kr port:53010
- host등록 여부 확인
- 파일 위치 확인(3가지 파일 모두 동일 경로에 두는 걸 권장)
  - onepass.properties(classpath 최상단)
  - onepass.jks(상대경로)
  - pubkey\_share.der(절대경로)

### ○ 이메일 문의 시 첨부 파일

- onepass.properties 파일
- 에러 화면 캡처
- 디버그 로그(이용사이트 was 로그)
- 방화벽 오픈된 증적화면(53010포트 텔넷 접속)
- acs.jsp, userInfo.jsp페이지에 확인되는 코드 캡처

## 11.4 참고자료(붙임)

1. 디지털원패스 연계기관별가이드 회원유형
2. 디지털원패스 연계기관별가이드 비회원유형
3. 디지털원패스 UiUx표준가이드
4. 디지털원패스 BIS Manual
5. 디지털원패스 연계확인서(시스템명)