NHN KCP MOBILE CERT
WEBLINK_ENC API 연동가이드

NHN KCP

문서 정보

문서 개요

이 문서는 가맹점 사이트에서 NHN KCP MOBILE CERT WEBLINK_ENC API 연동 가이드에 따라 통신사, 휴대폰번호, 명의자명, 생년월일, 성별, 내외국인 구분으로 본인을 확인하는 주민등록번호 실명인증의 대체 인증서비스에 대해 설명합니다.

- 가맹점에서 구현해야 하는 API 명세 가이드
- 서비스 설명 및 NHN KCP 권고 사항 설명
- 각 샘플 페이지 해당 역할 및 파라미터 상세 안내

독자

이 문서의 독자는 NHN KCP 본인확인 서비스를 연동하는 가맹점 사이트 개발자입니다.

문의처

이 문서의 내용에 오류가 있거나 내용과 관련한 문의 사항이 있으면 아래의 메일 주소로 문의 바랍니다.

- NHN KCP 기술지원팀 : **support@kcp.co.kr**
- 문서 버전 및 이력

버전	일자	이력사항	작성자
2.1.0	2020-12-11	가맹점 인증키 발급 추가	
2.1.1	2020-12-21	JSP JDK 허용 버전 변경(1.5 -> 1.6)	
2.1.2	2023-05-18	연동가이드 최신화	김민규

표기 규칙

- **소스 코드 표기**

이 문서에서 소스 코드는 검정색 바탕에 흰색, 주석은 초록색, 중요사항은 주황색으로 표기합니다.

- **파라미터 TYPE**

Column	Type	Description
1	Number	숫자형 Integer : (0~9)
2	String	문자형 String : Alpha numeric (A-Z; 0-9; UTF-8 characters)
3	Etc	기타 Etc.

- **특수문자**

콤마	엠퍼센트	세미콜론	뉴 라인	역 슬래쉬	파이프 라인	작은 따옴표	큰 따옴표	부등호
,	**&**	**;**	**₩n**	**₩**	**|**	‘	“	**<**

※ 스크립트 언어 불가 ex) <script>, eval((.*))

※ ₩u0000 : unicode null

요청 페이지에서 데이터를 입력할 때, 위와 같은 특수 문자를 입력할 경우 오류가 발생할 수 있습니다.
위 특수 문자 목록을 반드시 참고하여, 데이터를 입력할 때에 반드시 체크해 주시기 바랍니다.

목차

1. 개요

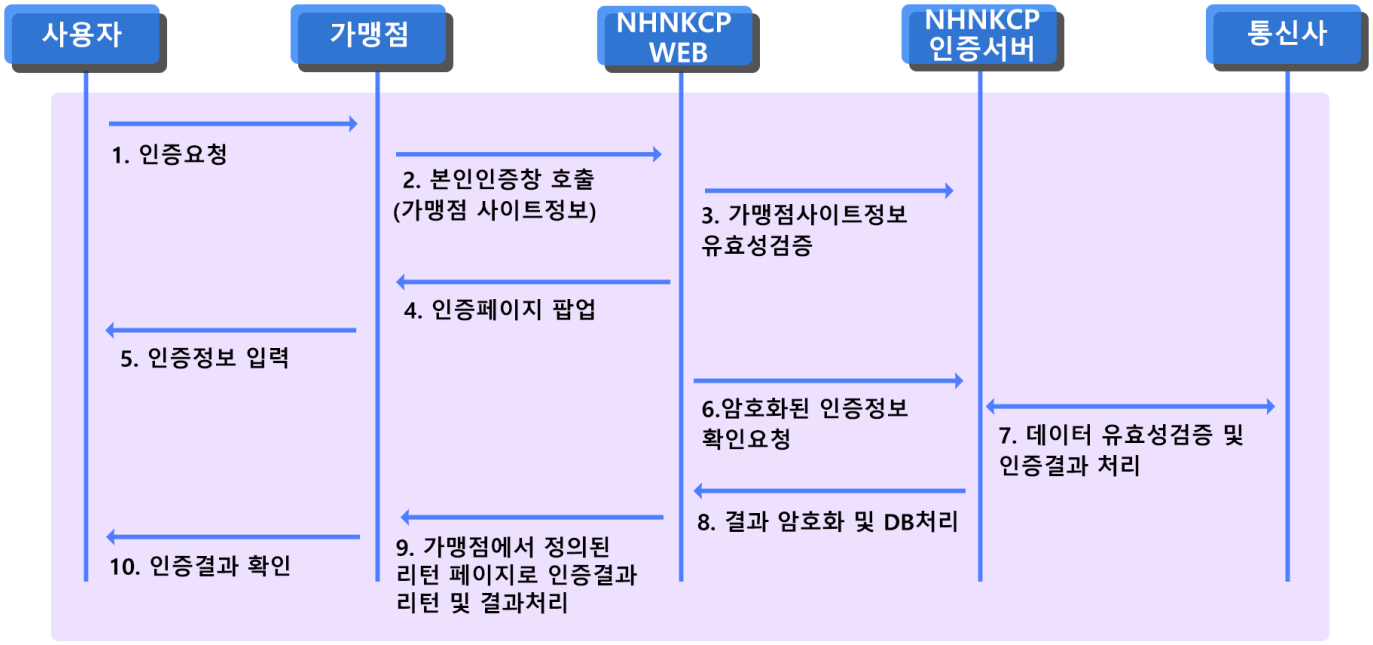
이 장에서는 NHN KCP 본인확인 서비스를 안내하고, 본인확인 처리 절차를 소개합니다.

1.1 NHN KCP 본인확인 서비스 개요 및 안내

1.1.1 제품 개요

- **NHN KCP MOBILE CERT WEBLINK_ENC** 서비스는 통신사, 휴대폰번호, 명의자명, 생년월일, 성별, 내외국인 구분으로 본인을 확인하는 서비스입니다. 기존의 실명과 주민등록번호로 본인을 확인하던 주민등록번호 실명인증의 대체 인증 서비스입니다.

1.1.2 본인확인 서비스 흐름도



① 인증을 필요로 하는 고객은 가맹점에 본인확인을 위한 NHN KCP 인증 전용창을 호출하여 통신사 , 휴대폰번호, 명의자명, 생년월일, 성별, 내외국인 구분을 선택함

② NHN KCP 인증창에서는 NHN KCP 인증전용 서버로 인증전문을 송신함

③ NHN KCP 인증창은 NHN KCP인증서버로 가맹점 사이트의 유효성을 판단함

④ 정상적인 가맹점 상태를 확인하면 NHN KCP 인증 전용창이 호출됨

⑤ 인증요청 고객은 NHN KCP 본인 인증창을 통해 필수 입력 값(명의자명, 생년월일, 성별, 휴대폰번호, 통신사, 내외국인)을 입력

⑥ NHN KCP 인증창에서는 입력 받은 고객 본인확인 요청 값을 위 변조 여부 검증 후 KCP 인증서버로 전송

⑦ NHN KCP 인증서버는 통신사와 인증정보의 유효성을 검증하고 인증결과를 처리하여 OTP 승인번호 입력단계 또는 PASS 인증단계로 이동 OTP 승인번호를 입력하거나 PASS앱 인증하여 통신사와 점유인증이 완료되면 결과값과 결과변수들(암호화된 enc_cert_data2)을 리턴

⑧ NHN KCP 인증서버는 NHN KCP 인증 전용 창으로 해당 본인확인 결과 값을 리턴

⑨ NHN KCP 인증 전용창은 앞단 order 페이지에 정의했던 결과페이지(Ret_URL) 측으로 인증결과를 리턴

⑩ 가맹점은 고객에게 인증결과 정상여부를 리턴

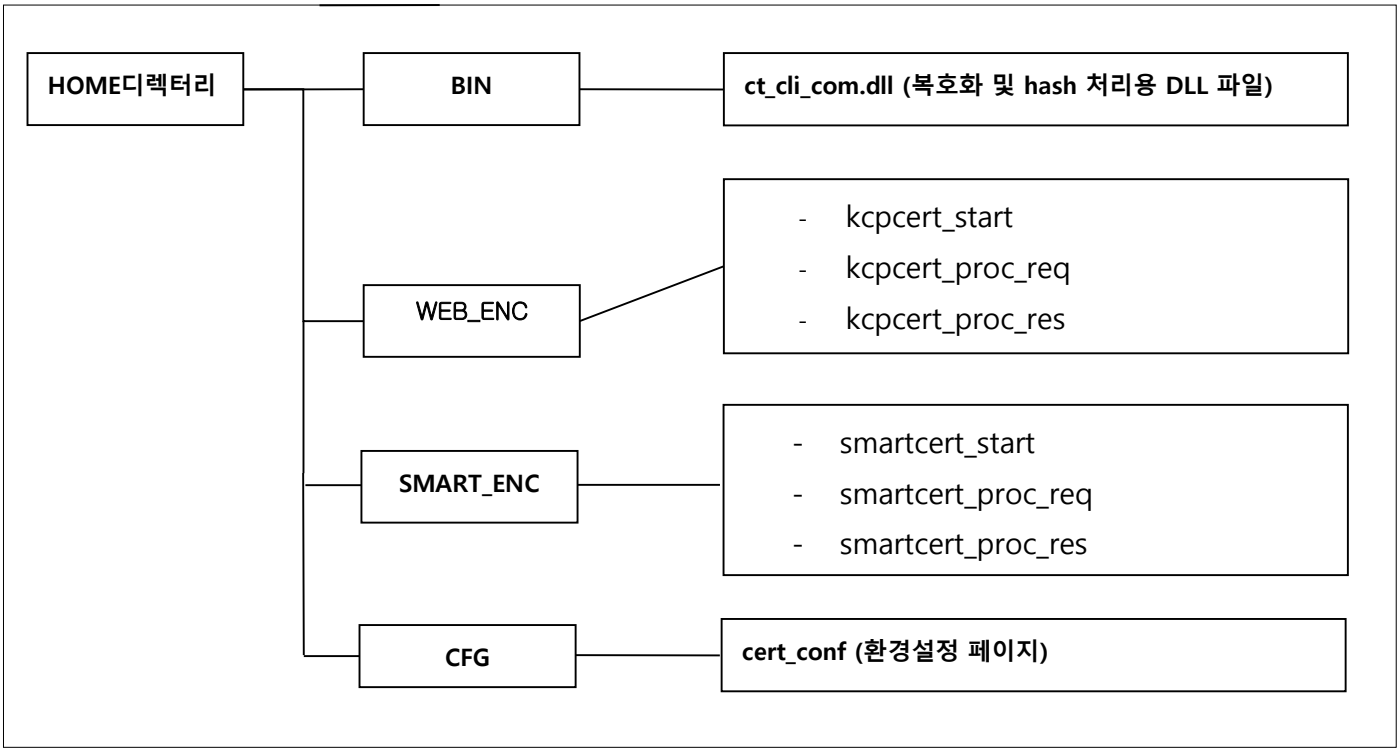
2. API 연동 전 준비사항

이 장에서는 API 연동 및 연동 전 준비사항에 대해 상세하게 설명합니다.

2.1 API 연동 사전 준비사항

2.1.1 ASP 확인사항

ASP 환경에서 구성 된 인증 시스템은 다음과 같습니다.



① NHN KCP에서 제공하는 소스 파일을 서버에 복사합니다.

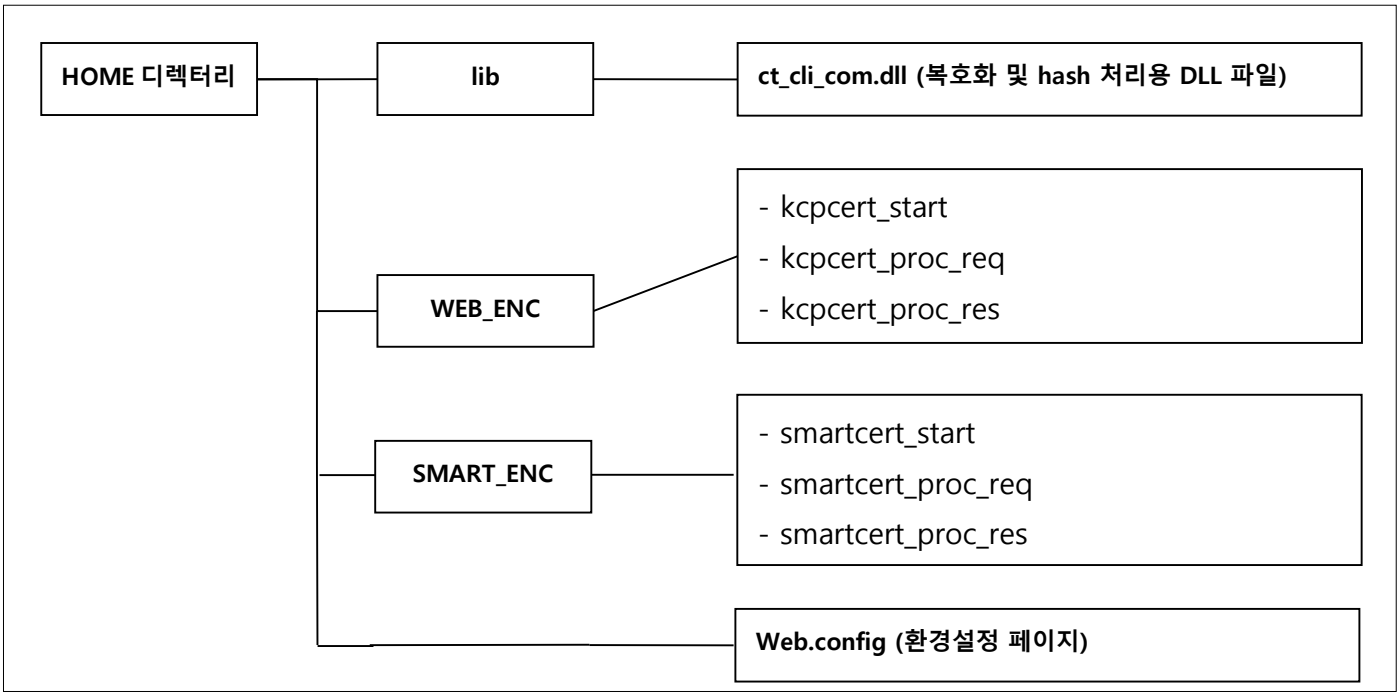
② 소스 중, /bin 디렉토리의 **ct_cli_com.dll** 파일을 서버 레지스트리에 등록합니다.

※ 설치방법 : 같은 디렉토리에 있는 **setup_bat** 실행 또는, cmd모드에서 ct_cli_com.dll이 있는 폴더에 들어가서 **regsvr32 ct_cli_com.dll**를 입력 후 레지스트리 등록 성공 메시지를 확인 하시면 완료됩니다.

※ Server.CreateObject('ct_cli_com.KCP') 구문에서 오류 발생 시 : ct_cli_com.dll이 레지스트리에 정상적으로 등록 되었다면 해당 파일(ct_cli_com.dll)의 사용 권한을 확인해주시기 바랍니다.

2.1.2 ASP.NET 확인사항

ASP.NET 환경에서 구성 된 인증 시스템은 다음과 같습니다.



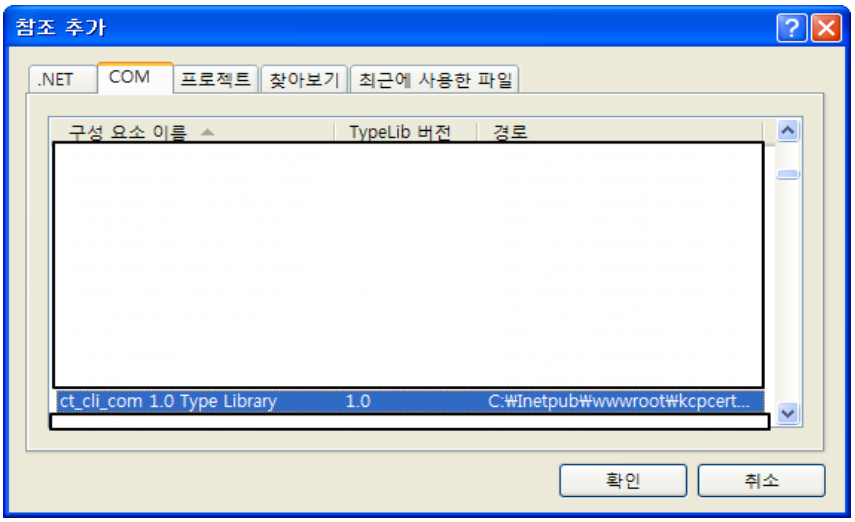
① NHN KCP에서 제공하는 소스 파일을 서버에 복사합니다.

② 소스 중, /bin 디렉토리의 **ct_cli_com.dll** 파일을 서버 레지스트리에 등록합니다.

※ 설치방법 : 같은 디렉토리에 있는 **setup_bat** 실행 또는, cmd모드에서 ct_cli_com.dll이 있는 폴더에 들어가서 **regsvr32 ct_cli_com.dll**를 입력 후 레지스트리 등록 성공 메시지를 확인 하시면 완료됩니다.

※ Server.CreateObject('ct_cli_com.KCP') 구문에서 오류 발생 시 : ct_cli_com.dll이 레지스트리에 정상적으로 등록 되었다면 해당 파일(ct_cli_com.dll)의 사용 권한을 확인해주시기 바랍니다.

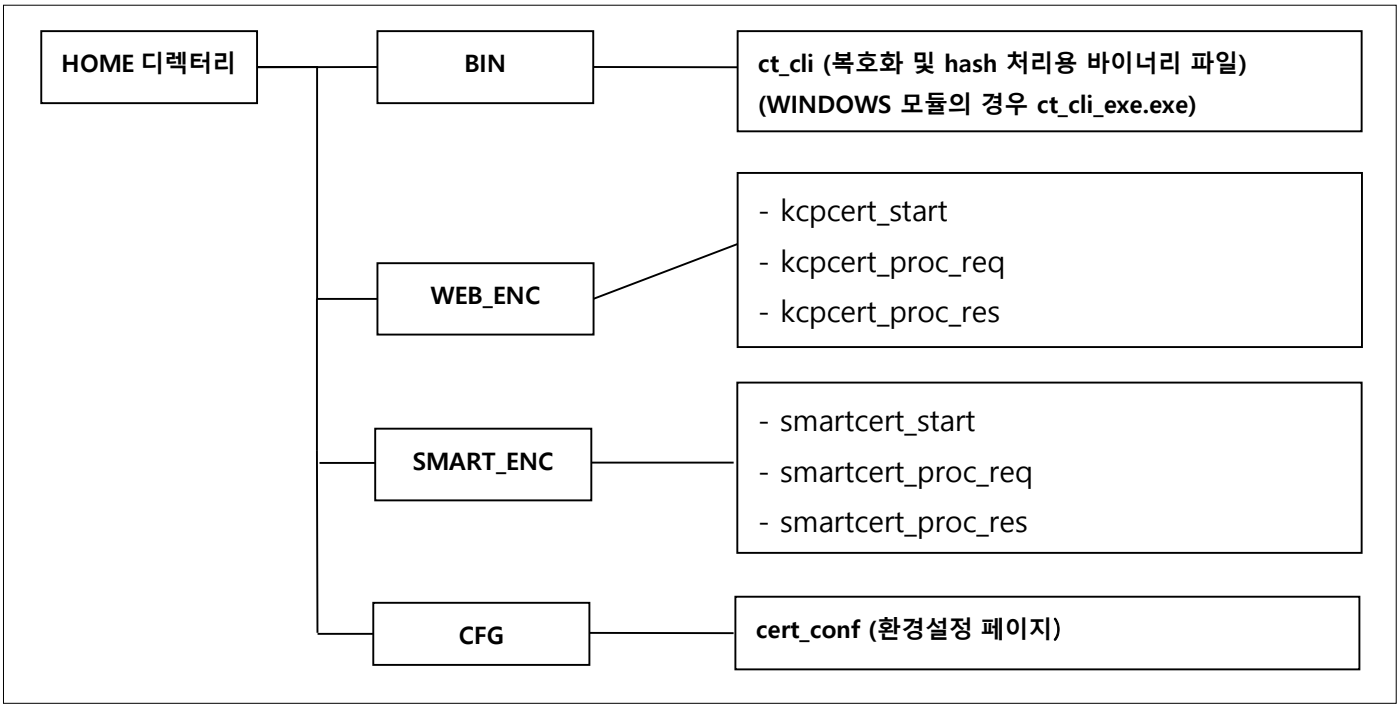
③ 해쉬처리 dll 참조 추가



[ct_cli_com 1.0 Type Libray 선택후 확인]

2.1.3 PHP 확인사항

PHP 환경에서 구성 된 인증 시스템은 다음과 같습니다.



① NHN KCP에서 제공하는 소스 파일을 서버에 복사합니다.

② 소스 중, /bin 디렉토리의 **ct_cli** 파일을 서버에 업로드 합니다.

③ 실행권한을 755 이상으로 줍니다.

[바이너리 파일 정상 업로드 확인여부]
서버 업로드 이후, ct_cli 파일을 실행시, 아래와 같이 출력되면 정상 실행

[test@devpg:/home/kcpcert_enc/bin]**./ct_cli**

```
==========================================================================
ct_cli - KCP ENC MODULE ver 1.08
--------------------------------------------------------------------------
 ▷ Hash 데이터 생성
  → If_CT_CLI__make_hash_data
  → 대상 데이터

 ▷ 암호화 데이터 생성
  → If_CT_CLI__encrypt_enc_cert
  → 업체코드
  → 인증번호
  → 원본 데이터

 ▷ 암호화 데이터 복호화
  → If_CT_CLI__decrypt_enc_cert
  → 업체코드
  → 인증번호
```

→ 암호화 데이터

→ [1: EUC-KR -> UTF-8]

▷ Hash 데이터 검증

→ If_CT_CLI__check_valid_hash

→ Hash 데이터

→ 비교 데이터

==========

이와 같은 메시지 출력이 되지 않는 경우

① 파일 업로드 시 바이너리 타입으로 업로드 되었는지 확인합니다.

② 실행권한을 755 이상으로 주었는지 확인합니다.

1,2를 진행하여도 메시지 출력이 되지 않는 경우

uname –a 명령어 실행 후 나온 결과값을 support@kcp.co.kr 로 문의 주시기 바랍니다.

- **ct_cli_lib.php 페이지**

샘플소스 lib 폴더에 제공해 드리고 있는 ct_cli_lib.php 파일의 아래 빨간색 강조된 부분은 보안상 수정하지 마시길 바랍니다.

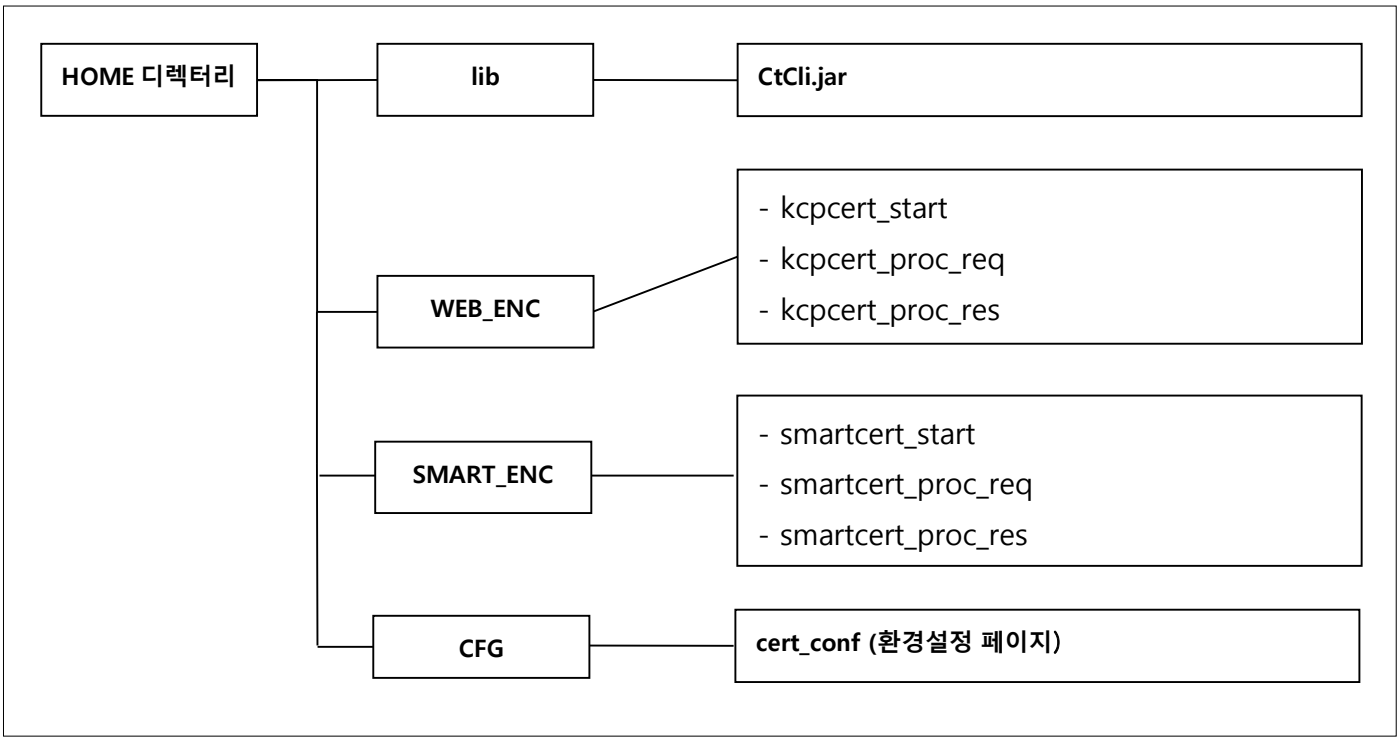
```
function  mf_exec()
{
        $arg = func_get_args();
        if ( is_array( $arg[0] ) )  $arg = $arg[0];
        $exec_cmd = array_shift( $arg );

        while ( list(,$i) = each($arg) )
        {
                $exec_cmd .= " " . escapeshellarg( $i );
        }
        $rt = exec( $exec_cmd );
        return  $rt;
    }
```

∗ PHP 4 이상 버전 필요

- 호스팅을 이용하시는 경우 호스팅사에 'exec' 함수 실행이 가능해야 합니다.

2.1.4 JSP 확인사항

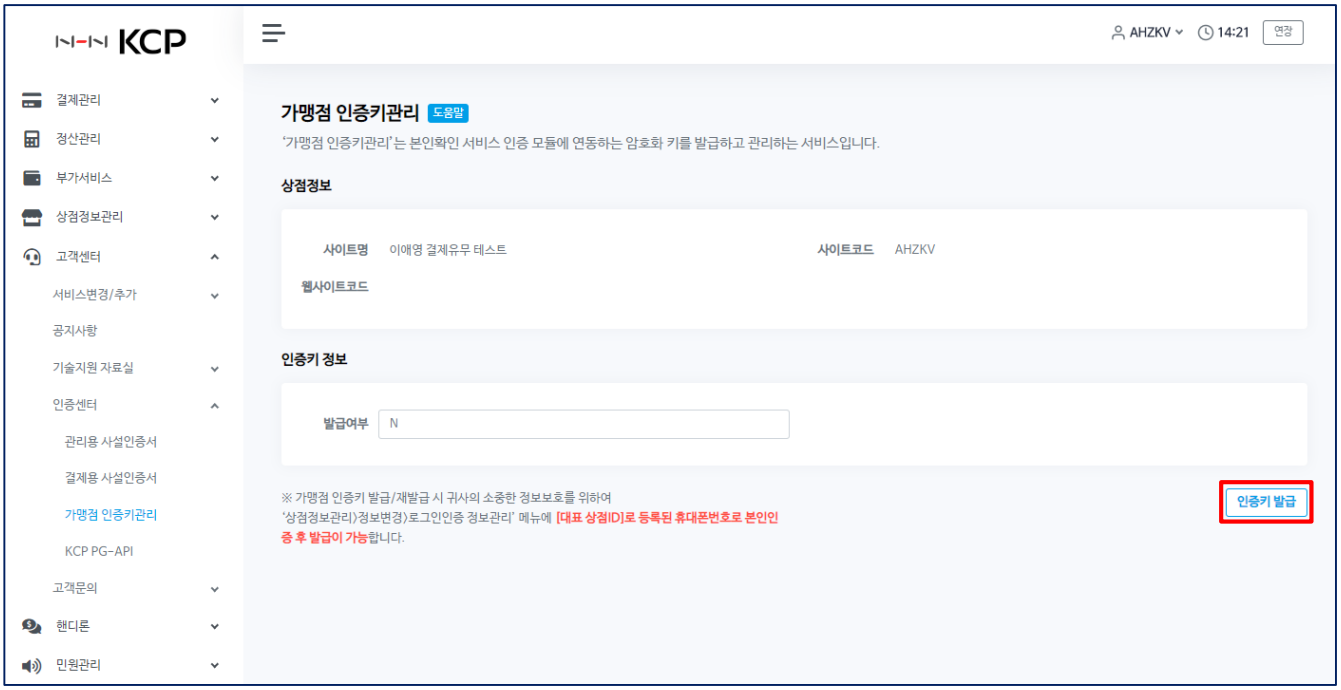
JSP 환경에서 구성 된 인증 시스템은 다음과 같습니다.



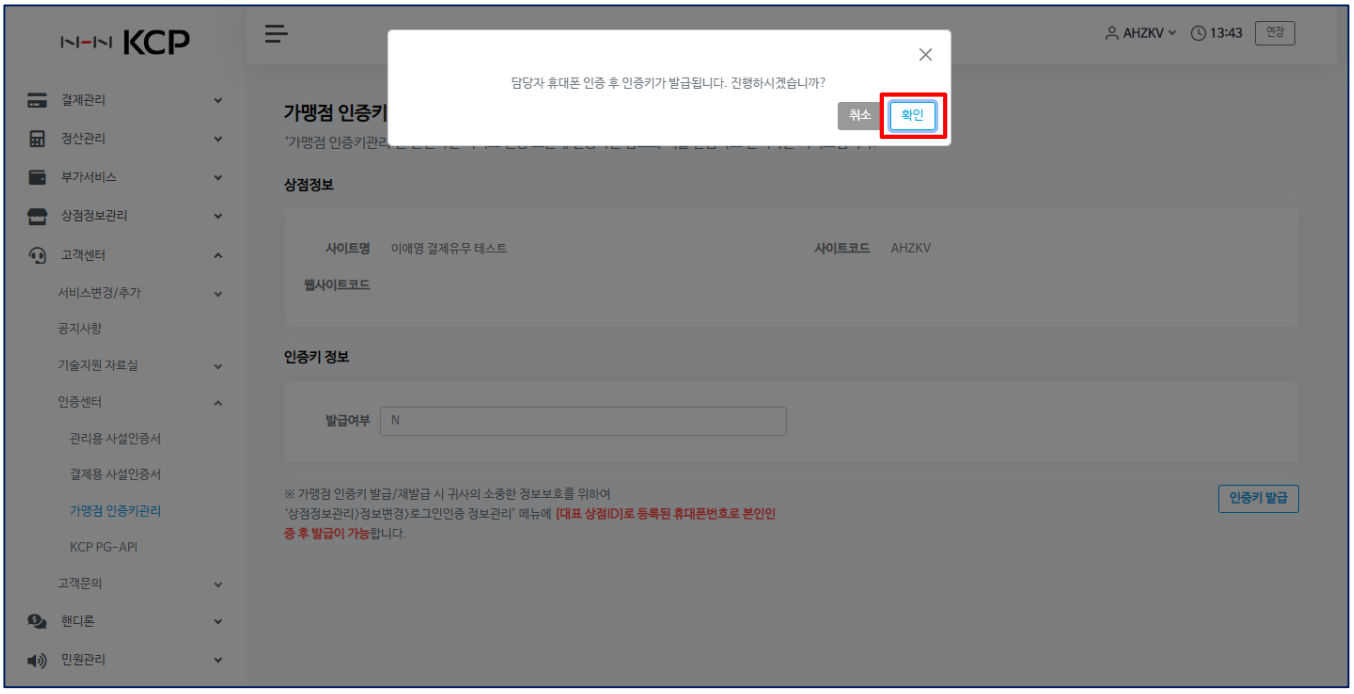
- JDK 1.6 버전 이상에서 구동이 가능합니다.
- 소스 중, ₩lib 디렉터리의 CtCLIE.jar 파일을 서버에 등록해야 합니다.
- jar 파일은 서버(Windows/Linux) 와 비트(32/64bit) 구분 없이 공통으로 사용할 수 있습니다.

2.1.5 가맹점 인증키 발급방법

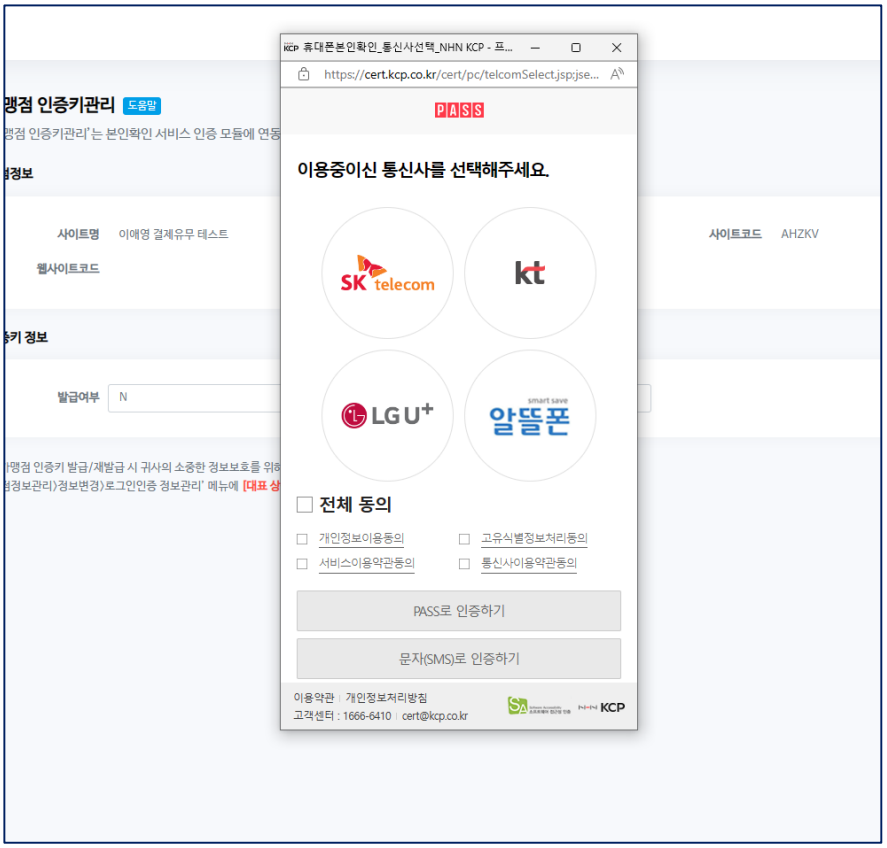
1. partner.kcp.co.kr 로그인 -> 고객센터 -> 인증센터 -> 가맹점 인증키 관리



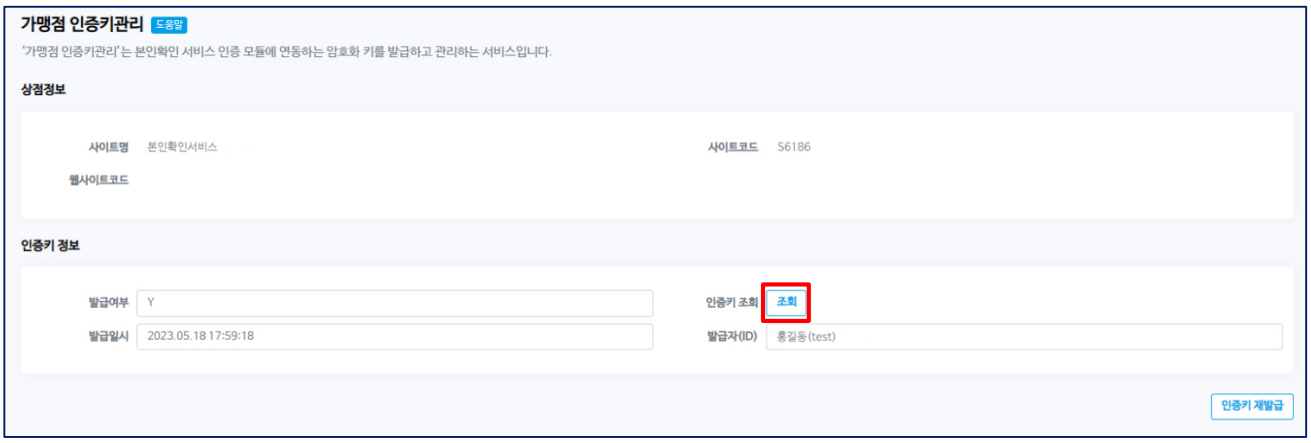
2. 인증키 발급



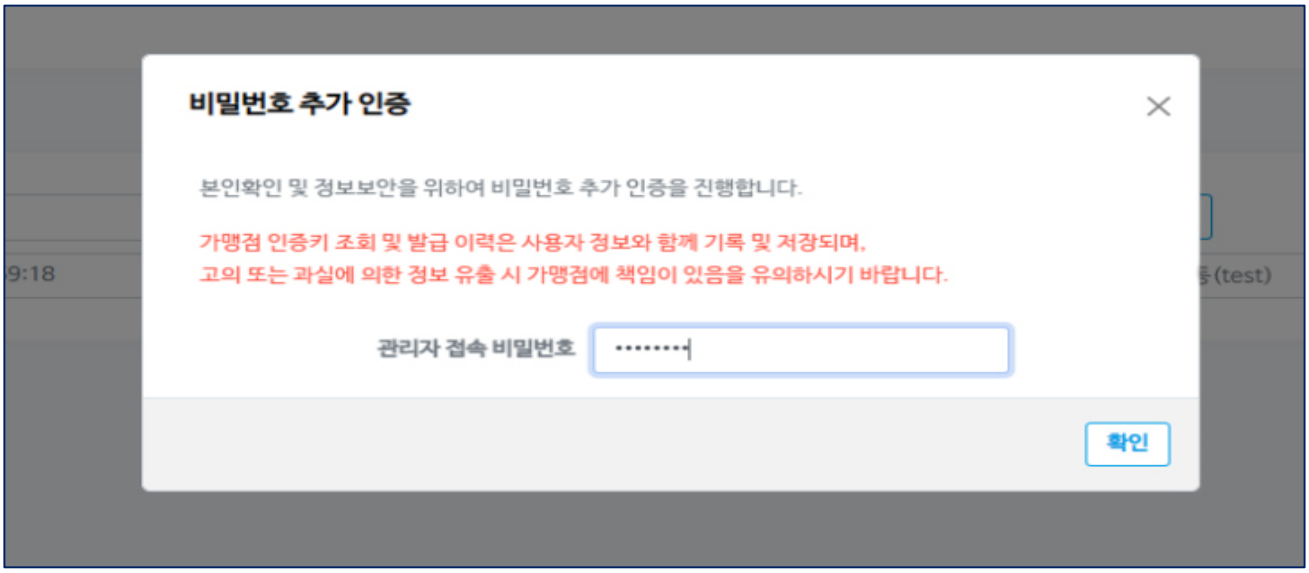
3. 휴대폰 인증



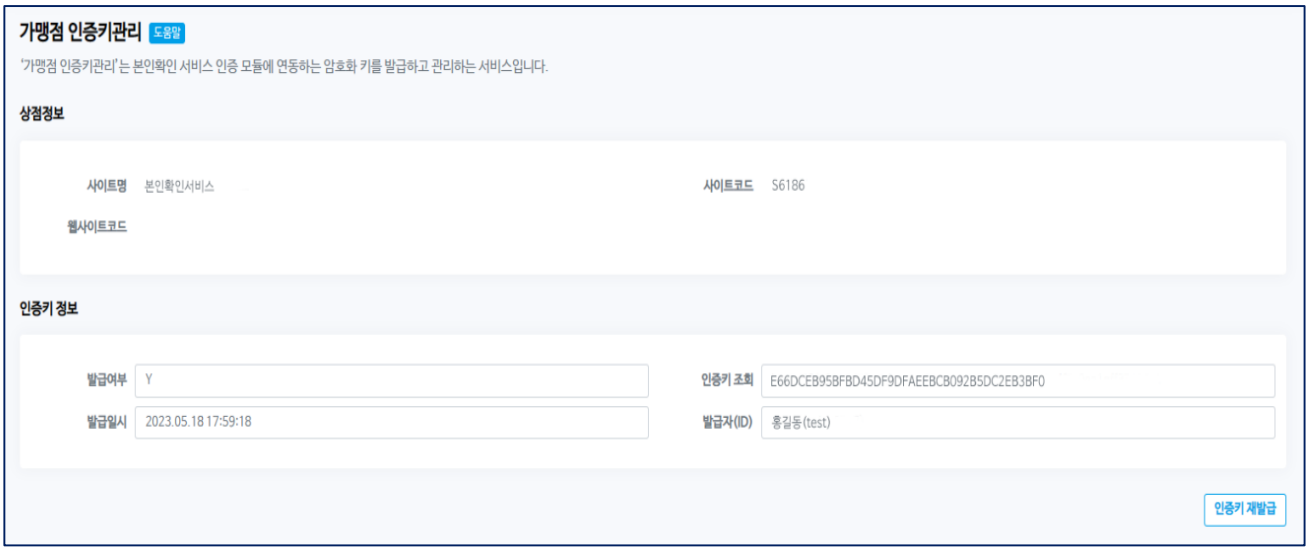
4. 인증키 조회



5. 가맹점 관리자 비밀번호 검증



6. 가맹점 인증키 확인



7. 가맹점 인증키 적용

실 본인인증 진행 시 가맹점 관리자에서 발급 받은 인증키를 API 에 변경하시어 적용바랍니다.
테스트로 진행 시에는 샘플 내에 있는 공용 ENC_KEY 값으로 사용바랍니다.

```
/* = * g_conf_ENC_KEY  설정                                                    = */
/* = 실 본인인증 진행시 가맹점 관리자에서 발급받은 인증키로 변경하시어 적용바랍니다.          = */
/* = 반드시 외부에 노출되지 않도록 관리하시기 바랍니다.                              = */
String g_conf_ENC_KEY       = "E66DCEB95BFBD45DF9DFAEEBCB092B5DC2EB3BF0";
```

2.1.6 주의사항

- 본인확인 요청 시 사용되는 필드 중 up_hash (상점요청데이터 검증 필드)값을 반드시 넘겨주셔야 합니다.
(up_hash 를 넘기지 않을 시 오류가 발생할 수 있습니다.)

- 가맹점 인증키 재발급 시 기존 인증키 즉시 사용 불가
인증키 재발급 시 모바일 사업팀 협의 必 (문의 : mbiz@kcp.co.kr)

3. 휴대폰 본인확인요청 페이지

이 장에서는 PC버전이나 모바일버전의 인증창을 호출하여 인증을 처리하는 페이지를 설명합니다.

3.1 휴대폰 본인확인요청(kcpcert_start)

3.1.1 요청페이지 변수 안내

휴대폰 본인확인 WEBLINK 방식은 요청자로부터 인증 요청 정보를 입력 받고, 가맹점이 설정한 데이터를 체크 후,
NHN KCP에서 제공하는 인증창을 호출하여 인증을 처리하는 페이지입니다. NHN KCP가 제공하는 샘플소스 상에서는
[kcpcert_start] 페이지에 해당하며, 실제 페이지 작성시에는 샘플소스와 매뉴얼을 참고하시어 가맹점에 맞게 수정, 적용하시기 바랍니다.

	Parameter	Type	Max Length	필수 여부	Description
1	**site_cd**	String	5	Y	상점 사이트코드
2	**ordr_idxx**	String	40	Y	상점관리 요청번호
3	**req_tx**	String	4	Y	요청의 종류를 구분하는 변수 CERT : 고정값
4	**cert_method**	Number	2	Y	인증 수단 01 : 고정값
5	**user_name**	String	100	N	명의자명 값이 있으면 인증 창은 입력 비활성, 없으면 입력 활성 SK - 40 Byte , KT - 80 , LGT Byte T - 30 Byte
6	**up_hash**	String	40	Y	요청 hash data (40byte의 hash 값) site_cd + ordr_idxx + 개인정보(이름,생년월일,성별,내외국인 구분)
7	**cert_otp_use**	String	1	Y	본인확인 인증요청 시 OTP 승인여부 Y : 실명확인 + OTP 점유 확인
8	**cert_able_yn**	String	1	N	비활성화 설정 Y : input 데이터 활성화 처리 "" : input 데이터 비활성화 처리
9	**web_siteid**	String	12	N	사이트 식별코드 CI – DI를 위한 중복확인 식별 아이디 Web_siteid 값이 없으면 NHN KCP에서 지정한 값으로 설정됨
10	**param_opt_1~3**	String	500	N	업체 추가 변수
11	**action**	String	256	Y	NHN KCP 인증창 URL Test : https://testcert.kcp.co.kr/kcp_cert/cert_view.jsp Real : https://cert.kcp.co.kr/kcp_cert/cert_view.jsp
12	**Ret_URL**	String	256	Y	본인확인 인증결과 리턴페이지 설정 * 반드시 업체 리턴페이지 주소를 FULL로 설정하십시요. **샘플소스의 kcpcert_proc_req 페이지를 참고하여** **작성하시기 바랍니다.**
13	**web_siteid_hashYN**	String	1	N	웹사이트아이디 위변조 검증 "Y" 이면 up_hash 생성시 web_siteid 값 추가 "N"이거나 값이 없으면 NHN KCP에서 지정한 값으로 설정됨
14	**cert_enc_use_ext**	String	1	Y	암호화 사용여부 Y : 고정값

3.1.2 암호화 모듈 사용유무

변수명	항목값	상세내용
cert_enc_use_ext	Y	개인정보에 해당하는 값을 enc_cert_data2 변수에 암호화 데이터로 리턴. (AES 암호화 방식 이용 - NHN KCP 가 제공하는 복호화 모듈을 통해 복호화) 암호화 대상 데이터 : **phone_no** **comm_id** **user_name** **birth_day** **sex_code** **local_code** **web_siteid** **ci** **di**

3.1.3 인증요청 정보 암호화 방법

가. JSP

배포 시 첨부한 파일 CtCli.jar 파일을 아래 소스와 같이 import 합니다.

up_hash : 인증창으로 site_cd 값과 ordr_idxx를 해쉬처리하여 전달함.
(앞단에서 개인정보를 넘기는 경우 개인정보도 함께 해쉬) (업체 적용)

```
<%@ page import="kr.co.kcp.CT_CLI"%>
up_hash = cc.makeHashData( g_conf_ENC_KEY, site_cd + ordr_idxx + 개인정보 ); // 해쉬처리 함수
```

dn_hash : 인증창에서 업체로부터 받은 site_cd값과 ordr_idxx 값에 성공 시 cert_no 값을 추가하여 해쉬처리

나. PHP

배포시 첨부한 파일 ct_cli 파일을 서버에 바이너리 타입으로 업로드합니다.

실행권한은 755 이상으로 줍니다.

up_hash : 인증창으로 site_cd 값과 ordr_idxx 를 해쉬 처리하여 전달함.

(앞단에서 개인정보를 넘기는 경우 개인정보도 함께 해쉬) (업체 적용)

```
$ct_cert = new C_CT_CLI;
$up_hash = $ct_cert->make_hash_data( $home_dir , $g_conf_ENC_KEY,
                                     $site_cd.$ordr_idxx.$user_name.$year.$month.$day.$sex_code.$local_code );
```

*** web_siteid_hashYN 값이 Y 인 경우 up_hahsh 생성예제**

```
$up_hash = $ct_cert->make_hash_data( $home_dir , $g_conf_ENC_KEY,
                                     $site_cd.$ordr_idxx.$user_name.$year.$month.$day.$sex_code.$local_code .$web_siteid);
```

dn_hash : 인증창에서 업체로부터 받은 site_cd 값과 ordr_idxx 값에 성공 시 cert_no 값을 추가하여 해쉬 처리

다. ASP

배포 시 첨부한 파일 ct_cli_com.dll 파일을 레지스트리에 등록합니다.
up_hash : 인증창으로 site_cd 값과 ordr_idxx를 해쉬 처리하여 전달함.
(앞단에서 개인정보를 넘기는 경우 개인정보도 함께 해쉬) (업체 적용)

```
set ct_test = Server.CreateObject( "Ct_cli_com.CTKCP" ) //Ct_cli_com.dll 구동
up_hash = ct_test.lf_CT_CLI__make_hash_data ( g_conf_ENC_KEY, site_cd & ordr_idxx & web_siteid & user_name & birth_day & sex_code & local_code ) // 해쉬처리 함수
```

dn_hash : 인증창에서 업체로부터 받은 site_cd 값과 ordr_idxx 값에 성공 시 cert_no값을 추가하여 해쉬처리

라. ASP.NET

배포 시 첨부한 파일 ct_cli_com.dll 파일을 레지스트리에 등록합니다.
up_hash : 인증창으로 site_cd 값과 ordr_idxx를 해쉬처리하여 전달함.
(앞단에서 개인정보를 넘기는 경우 개인정보도 함께 해쉬) (업체 적용)

```
CT_CLI_COMLib.CTKCP ct_cert = new CT_CLI_COMLib.CTKCP();
up_hash = ct_cert.lf_CT_CLI__make_hash_data( ENC_KEY, site_cd + ordr_idxx + web_siteid + user_name + year + month + day + sex_code + local_code);
```

dn_hash : 인증창에서 업체로부터 받은 site_cd값과 ordr_idxx 값에 성공시 cert_no값을 추가하여 해쉬처리

3.1.4 요청한 up_hash 값과 dn_hash값 검증 샘플

가. JSP

기본적으로 up_hash 값을 내부적으로 관리하다가 리턴 받은 up_hash 값과 비교하여 위변조 여부를 확인 하시기 바랍니다.
추가로, NHN KCP 인증 창이 결과로 내려주는 dn_hash 값과 (site_cd + ordr_idxx + cert_no)의 hash data와 비교하여 결과값의 위변조 여부를 반드시 검증하시기 바랍니다.
(site_cd와 cert_no는 enc_cert_data2를 복호화 하기 위한 암호화 key로 사용됩니다.)

[hash 검증 예제 샘플]

기본적으로 NHN KCP에서 제공하는 아래 hash 검증 함수를 이용하시기 바랍니다.
dn_hash 처리는 반드시 하셔야 하며, up_hash 처리는 아래 예제를 참고하시거나 DB 처리를 통해 검증을 권고합니다.

<dn_hash 검증 예제>

설명 : 해당 함수는 NHN KCP가 리턴하는 dn_hash 값을 기준으로 site_cd 값과 ordr_idxx 값과 cert_no 를 검증.

```
// dn_hash 검증
 if ( cc.checkValidHash( g_conf_ENC_KEY, dn_hash, ( site_cd + ordr_idxx + cert_no ) ) != true )
 {
        System.out.println("dn_hash 변조 위험있음");
// 오류 처리 ( dn_hash 변조 위험있음)
 }
```

설명 : 아래 해당 함수는 가맹점에서 인증요청 시 생성한 up_hash 데이터를 부모창에 두고 검증.

1) uphash 생성 이후, 부모창에 남겨 놓기(kcpcert_proc_req.jsp 페이지 154 line)

```
opener.document.form_auth.veri_up_hash.value = frm.up_hash.value;
```

2) 인증 프로세스 진행 이후, 결과 데이터에 리턴되는 값과 veri_up_hash 데이터와 검증

```
if( frm.up_hash.value != auth_form.veri_up_hash.value )
{
  alert("up_hash 변조 위험있음");
   // 오류 처리 ( dn_hash 변조 위험있음)
}
```

- 라이브러리 버전 관리

본인확인 라이브러리의 추후 취약점 발견 시 유지보수를 위하여 버전관리를 진행합니다. 본인인증 요청 시 해당함수를 호출하여 얻은 값을 넘겨 주시면 됩니다.

```
// NHN KCP 본인확인 라이브러리 버전 정보
  sbParam.append( "<input type=₩"hidden₩" name=₩"kcp_cert_lib_ver₩" value=₩"" + cc.getKCPLibVer() + "₩"/>" );
```

나. PHP

기본적으로 up_hash 값을 내부적으로 관리하다가 리턴 받은 up_hash 값과 비교하여 위변조 여부를 확인 하시기 바랍니다.
추가로, NHN KCP가 인증 창이 결과로 내려주는 dn_hash 값과 (site_cd + ordr_idxx + cert_no)의 hash data와 비교하여 결과값의 위변조 여부를 반드시 검증하시기 바랍니다.
(site_cd와 cert_no 는 enc_cert_data2 를 복호화 하기 위한 암호화 key로 사용됩니다.)

[hash 검증 예제 샘플]

기본적으로 NHN KCP에서 제공하는 아래 hash 검증 함수를 이용하시기 바랍니다.
dn_hash 처리는 반드시 하셔야 하며
up_hash처리는 아래 예제를 참고하시거나 DB 처리를 통해 검증을 권고합니다.

<dn_hash 검증 예제>

설명 : 해당 함수는 NHN KCP가 리턴하는 dn_hash값을 site_cd +ordr_idxx + cert_no를 키값으로 사용하여로 검증하며,
일치시 **"1"** 리턴

```
if ( $ct_cert->check_valid_hash ( $home_dir , $g_conf_ENC_KEY, $dn_hash, ( $site_cd . $ordr_idxx . $cert_no ) ) != "1" )
{
        // 오류 처리 ( dn_hash 변조 위험있음)
}
```

<up_hash 검증 예제>

설명 : 아래 해당 함수는 가맹점에서 인증요청시 생성한 up_hash 데이터를 부모창에 두고 검증.

1) uphash 생성 이후, 부모창에 남겨 놓기(kcpcert_proc_req.jsp 페이지 154 line)

```
opener.document.form_auth.veri_up_hash.value = frm.up_hash.value;
```

2) 인증 프로세스 진행이후, 결과 데이터에 리턴되는 값과 **veri_up_hash** 데이터와 검증

```
if( frm.up_hash.value != auth_form.veri_up_hash.value )
{
  alert("up_hash 변조 위험있음");
  // 오류 처리 ( dn_hash 변조 위험있음)
}
```

- 라이브러리 버전 관리

본인확인 라이브러리의 추후 취약점 발견 시 유지보수를 위하여 버전관리를 진행합니다. 본인인증 요청 시 해당함수를 호출하여 얻은 값을 넘겨 주시면 됩니다.

```
' 암호화 라이브러리 버전 정보
tmp_form = tmp_form & "<input type=""hidden"" name=""kcp_cert_lib_ver"" value=""" & ct_cert.lf_CT_CLI__get_kcp_lib_ver () & """/>"}
```

다. ASP

기본적으로 up_hash 값을 내부적으로 관리하다가 리턴 받은 up_hash 값과 비교하여 위변조 여부를 확인 하시기 바랍니다.
추가로, NHN KCP 인증 창이 결과로 내려주는 dn_hash 값과 (site_cd + ordr_idxx + cert_no)의 hash data와 비교하여 결과값의 위변조 여부를 반드시 검증하시기 바랍니다.
(site_cd와 cert_no는 enc_cert_data2를 복호화 하기 위한 암호화 key로 사용됩니다.)

[hash 검증 예제 샘플]

기본적으로 NHN KCP에서 제공하는 아래 hash 검증 함수를 이용하시기 바랍니다.
dn_hash 처리는 반드시 하셔야 하며, up_hash 처리는 아래 예제를 참고하시거나 DB 처리를 통해 검증을 권고합니다.

<dn_hash 검증 예제>

설명 : 해당 함수는 NHN KCP가 리턴하는 dn_hash 값을 기준으로 site_cd 값과 ordr_idxx 값과 cert_no 를 검증.

```
IF( ct_test.lf_CT_CLI__check_valid_hash( g_conf_ENC_KEY, dn_hash,( site_cd + ordr_idxx + cert_no )) = "FAIL") THEN
  '오류 처리 ( dn_hash 변조 위험 있음)
```

<up_hash 검증 예제>

설명 : kcpcert_proc_req 에서 생성된 up_hash 데이터와, 인증 성공 이후 리턴 되는 up_hash 데이터와 비교하여 검증 바랍니다.

kcpcert_proc_req.jsp 131번 라인 (검증을 위해 부모페이지에 임시로 저장)

parent.document.form_auth.**veri_up_hash**.value = frm.**up_hash**.value; // up_hash 데이터 검증을 위한 필드

이후 승인 정상 리턴 이후

kcpcert_start.jsp 페이지에 리턴 된 up_hash 와 부모페이지의 veri_up_hash 비교 로직 구현

- 라이브러리 버전 관리

본인확인 라이브러리의 추후 취약점 발견시 유지보수를 위하여 버전관리를 진행합니다. 본인인증 요청시 해당함수를 호출하여 얻은 값을 넘겨 주시면 됩니다.

```
' 암호화 라이브러리 버전 정보
tmp_form = tmp_form & "<input type=""hidden"" name=""kcp_cert_lib_ver"" value=""" & ct_cert.lf_CT_CLI__get_kcp_lib_ver () & """/>"
```

라. ASP.NET

기본적으로 up_hash 값을 내부적으로 관리하다가 리턴 받은 up_hash 값과 비교하여 위변조 여부를 확인 하시기 바랍니다.

추가로, NHN KCP가 인증 창이 결과로 내려주는 dn_hash 값과 (site_cd + ordr_idxx + cert_no)의 hash data와 비교하여 결과값의 위변조 여부를 반드시 검증하시기 바랍니다.

(site_cd와 cert_no는 enc_cert_data2를 복호화 하기 위한 암호화 key로 사용됩니다.)

[hash 검증 예제 샘플]

기본적으로 NHN KCP에서 제공하는 아래 hash 검증 함수를 이용하시기 바랍니다.

dn_hash 처리는 반드시 하셔야 하며

up_hash 처리는 아래 예제를 참고하시거나 DB 처리를 통해 검증을 권고합니다.

<dn_hash 검증 예제>

설명 : 해당 함수는 NHN KCP가 리턴하는 dn_hash 값을 기준으로 site_cd 값과 ordr_idxx 값과 cert_no 를 검증.

```
if (ct_cert.lf_CT_CLI__check_valid_hash(ENC_KEY, dn_hash, veri_str).Equals("FAIL"))
{
    //오류 처리 영역 ( dn_hash 변조 위험있음)
}
```

<up_hash 검증 예제>

설명 : kcpcert_proc_req 에서 생성된 up_hash 데이터와, 인증 성공 이 후 리턴 되는 up_hash 데이터와 비교하여 검증 바랍니다.

kcpcert_proc_req.aspx 16번 라인 (검증을 위해 부모페이지에 임시로 저장)

parent.document.form_auth.**veri_up_hash**.value = frm.**up_hash**.value; // up_hash 데이터 검증을 위한 필드

이후 승인 정상 리턴 이후

kcpcert_start.jsp 페이지에 리턴된 up_hash 와 부모페이지의 veri_up_hash 비교 로직 구현

- 라이브러리 버전 관리

본인확인 라이브러리의 추후 취약점 발견시 유지보수를 위하여 버전관리를 진행합니다. 본인인증 요청시 해당함수를 호출하여 얻은 값을 넘겨 주시면 됩니다.

```
// NHN KCP 본인확인 라이브러리 버전 정보
    tmp_form += "<input type='hidden' name='kcp_cert_lib_ver' value='" + ct_cert.lf_CT_CLI__get_kcp_lib_ver() + "'>";
```

4. 결과 처리 페이지

이 장에서는 PC버전이나 모바일버전의 인증창으로 인증 요청에 대한 결과를 처리하는 페이지를 설명합니다.

4.1 결과 처리 페이지 (kcpcert_proc_req) (kcpcert_proc_res)

4.1.1 인증처리 후 인증 창 전달 값

결과처리 페이지는 인증 요청에 대한 결과를 처리하는 페이지입니다. 이 페이지에서는 웹 인증창을 통해 가맹점과 인증 요청 건에 대한 결과를 암호화한 데이터로 리턴 받고 리턴 받은 암호화 데이터를 NHN KCP에서 제공한 복호화 모듈을 통해 복호화하여 결과를 산출하는 페이지입니다.

요청에 대한 처리 페이지 : **kcpcert_proc_req (up_hash 생성)**
응답에 대한 결과처리 페이지 : **kcpcert_proc_res (enc_cert_data2 복호화 처리, dn_hash 검증)**

- 샘플소스내의 페이지명(예. kcpcert_start, kcpcert_proc_req, kcpcert_proc_res 등)은 상점페이지명에 맞게 수정하시기 바랍니다. 특히, kcpcert_proc_req, kcpcert_proc_res 페이지명은 그대로 사용하지 않는 것을 권장 드립니다.

- 결과 처리된 인증정보 중 성명을 제외한 정보는 외부노출을 막기 위해 SSL을 통해 다음 처리 페이지로 전달하거나, **세션이나 Cookie 등에 담겨지지 않는 안전한 공간에 보관**하여야 합니다.

- 세션이나 쿠키를 사용하지 않는 경우 CI, DI 등의 정보를 text, hidden Value 등으로 남겨 처리하지 마시고, 암호화된 응답 데이터를 Submit 하여 회원가입 처리 페이지에서 다시 **복호화 한 후 사용**하시기 바랍니다. (권고 사항)

[인증 성공 혹은 실패 시 응답 파라미터]

	Parameter	Type	Max Length	필수 여부	Description
1	**res_cd**	String	4	Y	결과코드
2	**res_msg**	String	100	Y	결과메세지

[인증 완료 후 리턴 되는 공통 변수]

	Parameter	Type	Max Length	필수 여부	Description
1	**cert_no**	Number	14	Y	본인확인 고유한 인증 거래번호
2	**ordr_idxx**	String	40	Y	상점관리 요청번호
3	**enc_cert_data2**	String		Y	개인정보 암호화값 cert_enc_use_ext 값이 Y인 경우 리턴 되며, NHN KCP에서 제공한 복호화 모듈로 해당 값을 복호화 할 수 있음
4	**up_hash**	String	40	Y	요청 암호화 값 (요청 hash data) NHN KCP가 제공한 hash 알고리즘을 사용하여 생성한 hash data (site_cd + ordr_idxx + 개인정보)를 해쉬처리 요청번호 외 옵션으로 입력하는 개인정보의 위변조 여부를 NHN KCP 인증창에서 검증하기 위해 사용됨. 가맹점 요청번호는 인증결과 수신 후 반드시 자체 위변조 여부 검증 바랍니다.

5	**dn_hash**	String	40	Y	리턴 암호화 값 (리턴 hash data) NHN KCP 인증창이 인증결과 hash 알고리즘을 사용하여 생성한 hash data 실패시에는 요청암호화값(up_hash)와 동일 (sitd_cd + ordr_idxx + cert_no)을 NHN KCP에서 해쉬처리하여 리턴 정상처리시 NHN KCP의 인증거래번호의 위변조 여부를 검증하기 위해 사용됨. 정상처리결과를 수신하였으나 enc_cert_data2가 리턴되지 않았을 경우 반드시 실패 또는 위변조 위험으로 판단하시기 바랍니다.
6	**param_opt_1** **param_opt_2** **param_opt_3**	String	500		업체 추가 변수

[본인확인 서비스 이용 시 리턴 되는 복호화 변수]

	Parameter	Type	Max Length	필수 여부	Description
1	**phone_no**	Number	11	Y	본인확인 휴대폰 번호
2	**comm_id**	String	3	Y	본인확인 통신사 SK 텔레콤 - SKT LG U+ - LGT KT - KTF SKT의 알뜰폰 - SKM LG U+의 알뜰폰 - LGM KT의 알뜰폰 - KTM
3	**user_name**	String	100	Y	명의자명
4	**birth_day**	Number	8	Y	생년월일 (YYYYMMDD)
5	**sex_code**	Number	2	Y	성별 01: 남성 02: 여성
6	**local_code**	Number	2	Y	내외국인 코드 01: 내국인 02: 외국인
7	**ci**	String	88	Y	연계정보 값 특정 웹사이트가 타 웹사이트와의 제휴 사업을 수행할 경우, 동일 고객을 확인하기 위한 값
8	**di**	String	64	Y	중복확인 값 특정 웹 사이트 내에서 중복가입 및 내부회원 관리시, 동일 고객을 확인하기 위한 값
9	**ci_url**	String	264	Y	CI URL인코딩 데이터 특수문자가 포함될 수 있으며, 해당 특수문자가 포함되기 때문에 실제 CI, DI 값 차이가 발생할 수 있습니다. 이런 누락을 방지하기 위하여,

				URL 인코딩 하여 데이터를 내려드리며, 해당 값을 URL 디코딩하여 CI,DI 값으로 처리하시기 바랍니다.
10	**di_url**	String	192	Y	DI URL인코딩 데이터 (위와 내용 동일)
11	**web_siteid**	String	12	Y	웹사이트 아이디 인증요청된 웹사이트 아이디값이 리턴됩니다.

- 응답결과의 경우 통신사측 정책에 따라 변경되거나 추가될 수 있습니다.
- **URL 디코딩 방법 가이드**

- **JSP**

```
<%@ page import="java.net.URLDecoder"%>
System.out.println( "CI_URL" + URLDecoder.decode( cc.getKeyValue("ci_url") ) );
System.out.println( "DI_URL" + URLDecoder.decode( cc.getKeyValue("di_url") ) );
```

- **PHP**

복호화 데이터($enc_cert_data2)의 인코딩이 깨져보이는 경우 $opt 옵션을 1로 변경하여 복호화 해주시기 바랍니다. (PHP 전용옵션)

- **ASP**

```
unescape( ct_cert.lf_CT_CLI__get_key_value("ci_url" ) )
unescape( ct_cert.lf_CT_CLI__get_key_value("di_url" ) )
```

- **ASP.NET**

```
Server.UrlDecode(ct_cert.lf_CT_CLI__get_key_value("ci_url"));
Server.UrlDecode(ct_cert.lf_CT_CLI__get_key_value("di_url"));
```

4.1.2 복호화 모듈 페이지 작성 (kcpcert_proc_res)

[JSP]

[샘플페이지]

배포 시 CtCli.jar파일을 아래 소스와 같이 import 합니다.

아래 샘플소스를 참고하여 암호화된 인증결과데이터(enc_cert_data2)를 복호화하여 결과처리합니다.

[kcpcert_proc_res 페이지 샘플소스 예제 파일]

```
<%@ page language="java" contentType="text/html;charset=euc-kr"%>
<%@ page import="java.util.Enumeration" %>
<%@ page import="kr.co.kcp.CT_CLI"%> 복호화 모듈 클래스 파일
<%
        String site_cd      = "";  // 사이트코드 (NHN KCP 가맹점 아이디)
        String ordr_idxx    = "";  // 주문번호(인증확인용 가맹점 번호)
        String cert_no      = "";  // 인증거래번호
        String enc_cert_data2 = "";  // 개인정보 암호화 데이터
        String req_tx       = "";  // 거래구분 (cert)
        String res_cd       = "";  // 응답코드
        String res_msg      = "";  // 응답메세지
```

```
/*-------------------------------------------------------------------------*/
/*  :: 전체 파라미터 남기기 ::                                             */
 /*-------------------------------------------------------------------------*/
    StringBuffer sbParam = new StringBuffer();
    CT_CLI       cc      = new CT_CLI(); // 복호화를 위한 객체생성
    Enumeration params = request.getParameterNames();
    while(params.hasMoreElements())
    {
        String nmParam = (String) params.nextElement();
        String valParam[] = request.getParameterValues(nmParam);
        for(int i = 0; i < valParam.length;i++)
        {
            if( nmParam.equals( "res_cd" ) )
            {
                res_cd = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "site_cd" ) )
            {
                site_cd = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "ordr_idxx" ) )
            {
                ordr_idxx = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "req_tx" ) )
            {
                req_tx = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "cert_no" ) )
            {
                cert_no = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "enc_cert_data2" ) )
            {
                enc_cert_data2 = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "enc_info" ) )
            {
                enc_info = f_get_parm( valParam[i] );
            }
            if( nmParam.equals( "enc_data" ) )  ➔ 개인정보 암호화 데이터
            {
                enc_data = f_get_parm( valParam[i] );
            }
            sbParam.append( "<input type=₩"hidden₩" name=₩"" + nmParam + "₩" value=₩"" + f_get_parm( valParam[i] ) +
"₩"/>" );
        }
    }
```

```
    if( req_tx.equals( "result" ) )
    {
            if( res_cd.equals( "0000" ) )
            {
                    System.out.println(site_cd);
                    System.out.println(cert_no);
                    System.out.println(enc_cert_data2);
                    cc.decryptEncCert( site_cd, cert_no, enc_cert_data2 );  복호화 실행
                    System.out.println( cc.getKeyValue( "phone_no"    ) );
                    System.out.println( cc.getKeyValue( "user_name"   ) );
                    System.out.println( cc.getKeyValue( "birth_day"   ) );
                    System.out.println( cc.getKeyValue( "sex_code"    ) );
                    System.out.println( cc.getKeyValue( "local_code" ) );
                    System.out.println( cc.getKeyValue( "comm_id"     ) );
                    System.out.println( cc.getKeyValue( "ci"          ) );
                  System.out.println( cc.getKeyValue( "di"          ) );
                  System.out.println( cc.getKeyValue( "web_siteid"  ) );
                  System.out.println( cc.getKeyValue( "res_cd"      ) );
                  System.out.println( cc.getKeyValue( "res_msg"     ) );
            }
    }

cc=null;
```

[PHP]

[샘플페이지]

```
( $home_dir , $g_conf_ENC_KEY, $dn_hash , $veri_str ) != "1" )

{
    echo "dn_hash 변조 위험있음";
    // 오류 처리 ( dn_hash 변조 위험있음)
}

// 가맹점 DB 처리 페이지 영역
echo "사이트 코드"    .    $site_cd;
echo "인증 번호"      .    $cert_no;
echo "암호된 인증정보".    $enc_cert_data2;

$opt = "0" ; // 복호화 인코딩 옵션 ( UTF - 8 사용시 "1" )
$ct_cert->decrypt_enc_cert( $home_dir , $g_conf_ENC_KEY, $site_cd , $cert_no , $enc_cert_data2 , $opt );

echo "이동통신사 코드"    . $ct_cert->mf_get_key_value("comm_id"    )."<br>"; // 이동통신사 코드
echo "전화번호"           . $ct_cert->mf_get_key_value("phone_no"   )."<br>"; // 전화번호
echo "이름"               . $ct_cert->mf_get_key_value("user_name"  )."<br>"; // 이름
echo "생년월일"           . $ct_cert->mf_get_key_value("birth_day"  )."<br>"; // 생년월일
echo "성별코드"           . $ct_cert->mf_get_key_value("sex_code"   )."<br>"; // 성별코드
echo "내/외국인 정보 "    . $ct_cert->mf_get_key_value("local_code" )."<br>"; // 내/외국인 정보
echo "CI"                 . $ct_cert->mf_get_key_value("ci"         )."<br>"; // CI
```

```
echo "DI 중복가입 확인값" . $ct_cert->mf_get_key_value("di"          )."<br>"; // DI 중복가입 확인값
echo "웹사이트 아이디   " . $ct_cert->mf_get_key_value("web_siteid" )."<br>"; // 웹사이트 아이디
echo "암호화된 결과코드"  . $ct_cert->mf_get_key_value("res_cd"     )."<br>"; // 암호화된 결과코드
echo "암호화된 결과메시지". $ct_cert->mf_get_key_value("res_msg"    )."<br>"; // 암호화된 결과메시지
```

[ASP]

[샘플페이지]

배포 시 ct_cli_com 파일을 레지스트리에 등록합니다.

아래 샘플소스를 참고하여 암호화된 인증결과데이터(enc_cert_data2)를 복호화하여 결과처리합니다.

[kcpcert_proc_res.asp 페이지 샘플소스 예제 파일]

```
IF (req_tx = "result") THEN
        IF (res_cd = "0000") THEN
           IF( ct_test.lf_CT_CLI__check_valid_hash( g_conf_ENC_KEY, dn_hash, ( site_cd + ordr_idxx + cert_no )) = "FAIL")
                                                    ' dn_hash 변조 위험있음
        END IF
' 가맹점 DB 처리 영역

        Response.Write "사이트코드"          &    site_cd
        Response.Write "인증 거래번호"       &    cert_no
        Response.Write "암호화 인증데이터"   &    enc_cert_data2

        c_cert.lf_CT_CLI__decrypt_enc_cert ( g_conf_ENC_KEY, site_cd , cert_no , enc_cert_data2 )

        Response.Write "이동통신사 코드"     &    ct_test.lf_CT_CLI__get_key_value("comm_id"     )
        Response.Write "전화번호"            &    ct_test.lf_CT_CLI__get_key_value("phone_no"    )
        Response.Write "이름"                &    ct_test.lf_CT_CLI__get_key_value("user_name"   )
        Response.Write "생년월일"            &    ct_test.lf_CT_CLI__get_key_value("birth_day"   )
        Response.Write "성별코드"            &    ct_test.lf_CT_CLI__get_key_value("sex_code"    )
        Response.Write "내/외국인 정보 "     &    ct_test.lf_CT_CLI__get_key_value("local_code"  )
        Response.Write "CI"                  &    ct_test.lf_CT_CLI__get_key_value("ci"          )
        Response.Write "DI 중복가입 확인값"  &    ct_test.lf_CT_CLI__get_key_value("di"          )
        Response.Write "웹사이트 아이디"     &    ct_test.lf_CT_CLI__get_key_value("web_siteid"  )
        Response.Write "암호화된 결과코드"   &    ct_test.lf_CT_CLI__get_key_value("res_cd"      )
        Response.Write "암호화된 결과메시지" &    ct_test.lf_CT_CLI__get_key_value("res_msg"     )
        END IF
            END IF
%>
```

[ASP.NET]

[샘플페이지]
배포 시 Ct_cli_com 파일을 레지스트리에 등록합니다.
아래 샘플소스를 참고하여 암호화된 인증결과데이터(enc_cert_data2)를 복호화하여 결과처리합니다.

[kcpcert_proc_res.aspx 페이지 샘플소스 예제 파일]

```
 if (res_cd.Equals("0000"))
            {
                CT_CLI_COMLib.CTKCP ct_cert = new CT_CLI_COMLib.CTKCP();

                // dn_hash 검증
                // NHN KCP 가 리턴해 드리는 dn_hash 와 사이트 코드, 주문번호 , 인증번호를 검증하여
                // 해당 데이터의 위변조를 방지합니다
                string veri_str = site_cd + ordr_idxx + cert_no;

                if (ct_cert.lf_CT_CLI__check_valid_hash(ENC_KEY, dn_hash, veri_str).Equals("FAIL"))
                {
                    //오류 처리 영역 ( dn_hash 변조 위험있음)
                }
                ct_cert.lf_CT_CLI__decrypt_enc_cert(ENC_KEY, site_cd, cert_no, enc_cert_data2);
                ct_cert.lf_CT_CLI__get_key_value("comm_id");
                ct_cert.lf_CT_CLI__get_key_value("phone_no");
                ct_cert.lf_CT_CLI__get_key_value("user_name");
                ct_cert.lf_CT_CLI__get_key_value("birth_day");
                ct_cert.lf_CT_CLI__get_key_value("sex_code");
                ct_cert.lf_CT_CLI__get_key_value("local_code");
                ct_cert.lf_CT_CLI__get_key_value("ci");
                ct_cert.lf_CT_CLI__get_key_value("di");
                Server.UrlDecode(ct_cert.lf_CT_CLI__get_key_value("ci_url"));
                Server.UrlDecode(ct_cert.lf_CT_CLI__get_key_value("di_url"));
                ct_cert.lf_CT_CLI__get_key_value("web_siteid");
                ct_cert.lf_CT_CLI__get_key_value("res_cd");
                ct_cert.lf_CT_CLI__get_key_value("res_msg");
            }
```

5. 참고사항

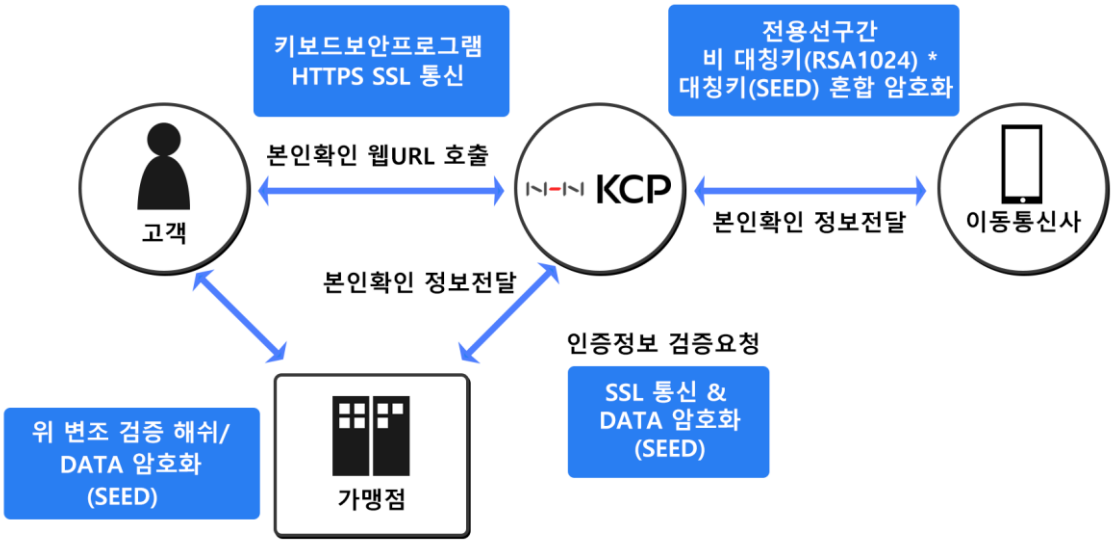
이 장에서는 인증 후 리턴 되는 응답 코드와 간편 연동방법에 대해 설명합니다.

5.1 응답코드 및 참고사항

5.1.1 응답코드표

응답코드	응답메세지
0000	정상처리
CTD7	간편인증 PUSH발송 실패
CTD8	간편본인확인이 실패하였습니다. U+인증 APP 설치 후 이용 부탁 드립니다.
CTD9	간편인증 APP 재등록 필요
CTC1	OTP 발송 실패
CTC3	OTP 번호 상이
CTC4	OTP 번호 유효시간 초과
CT02	CP 설정 오류
CT03	미등록 가맹점
CT93	인증기관 시스템오류
CT97	이통사 전문포맷오류
CT98	이통사 통신장애
CT99	KCP로 문의(☎ 1666-6410)
CTCA	OTP 발송 실패(유효시간 초과 또는 기타오류)
CTCB	OTP 검증 실패(유효시간 초과 또는 불일치 또는 기타오류)
CTBE	인증실패(인증정보 불일치 또는 기타오류)
CTBB	인증실패(알뜰폰 가입자)
CTC7	USIM OTP PUSH 서비스 불가
CTBD	부가서비스 가입자
CTD1	인증실패(인증 대기 중)
CTD2	인증실패(인증 유효시간 초과)
CTD3	인증실패(사용자에 의한 인증 취소)
CTD4	인증실패(간편인증 미 가입자) \n※간편인증App설치 또는 본인확인(문자) 이용
CTD5	간편인증 실패
CTBF	인증실패(법인 인증 대기 상태)
CTDA	휴대폰본인확인 일반 인증(SMS)이 차단된 고객입니다. 인증문자 차단 해제 또는 간편 인증(App)을 이용해 주시기 바랍니다.
CTD6	인증실패(스마트폰 아님)
4999	KCP로 문의(☎ 1666-6410)

5.1.2 보안체계 및 암호화 안내



**** 개인정보의 노출이 우려되는 모든 통신 구간의 보안 수단 적용 및 구성**

5.1.3 중복가입확인정보 관리 방안

- 중복가입확인을 위한 DI 값은 인증 창 호출 시 전달하는 **web_siteid** 값을 기존에 이용하시던 신용평가기관에서 발급받은 **web_siteid** 로 전달해주시면 동일한 신용평가사에서 제공하는 DI 값으로 이용 가능합니다.
- 해당 web_siteid 값을 넘겨주시지 않으면 NHN KCP가 발급한 **web_siteid** 아이디로 설정되오니 유의바랍니다.

** **web_siteid는 기 발급한 신용평가사에 확인하시어 정확한 web_siteid 값을 전송해주시기 바랍니다.**

사이트코드 계약 시, 기존 신용평가사에 등록된 web_siteid 값을 KCP로 전달해주시고, 해당 값을 정확하게 소스에 적용해주시기 바랍니다.

5.1.4 스마트폰 환경 적용 방법

- 샘플소스의 SMART_ENC 폴더 샘플을 이용하시면 스마트폰 앱에서 구동이 가능합니다.
- WEB_ENC/SMART_ENC 폴더 샘플의 디바이스 및 웹 환경에 따른 사용 가능 부분은 아래와 같습니다.
  아래 표를 참고하시어 가맹점에서 사용자에게 제공하려고 하는 환경에 적합한 샘플로 연동하시기 바랍니다.

	WEB_ENC 샘플	SMART_ENC 샘플
PC 웹	O	O
스마트폰 웹	O	O
스마트폰 앱(웹뷰)	X	O
NHN KCP 인증창 호출 방식	팝업	PC 웹 ➔ 팝업 스마트폰 웹/앱 ➔ iframe

5.1.5 IOS PASS 앱 관련 스키마 등록 방법

- 통신사의 간편본인확인 앱이 업데이트됨에 따라 ios 에서 가맹점 앱서비스를 제공하는 경우,
  IOS9부터 보안을 강화하는 목적으로 앱을 호출할 때 앱스키마 리스트를 등록해주어야 합니다.
- Info.plist 파일에 LSApplicationQueriesSchemes 배열을 정의하여 앱 스키마 리스트를 등록합니다.

(1) Information Property List에 LSApplicationQueriesSchemes를 Array 타입으로 추가
(2) LSApplicationQueriesSchemes 하위 리스트에 String 타입으로 Item 추가
(3) Item 마다 앱 스키마를 입력

통신사	앱스키마
SKT	tauthlink
KT	ktauthexternalcall
LGU+	upluscorporation

PASS 앱 관련 스키마 이후 추가 변수

PASS앱의 스키마 등록이 완료되면 본인인증시에 추가변수를 넘겨주시면 IOS 환경에서 PASS 앱 사용이 가능합니다.

```
<input type="hidden" name="kcp_cert_pass_use"  value="Y"/>
```

5.1.6 안드로이드 intent URL 호출 처리

- 통신사의 간편본인확인 앱이 업데이트 됨에 따라 안드로이드에서 가맹점 앱서비스를 제공하는 경우,
  Intent URL 의 대한 처리를 추가로 해주셔야 합니다.
  같이 첨부 드린 예시 파일을 참고하여 사용하시는 WebViewClient 클래스에 shouldOverrideUrlLoading 함수를 오버라이드 하는 부분을 추가(또는 수정)하여 처리해 주시면 됩니다.

PASS 앱 관련 intent URL 처리 후 추가 변수

PASS 앱 intent URL 처리가 완료되면 본인인증 시에 추가변수를 넘겨주시면 안드로이드 환경에서 PASS 앱사용이 가능합니다.

<input type="hidden" name="kcp_cert_intent_use" value="Y" />

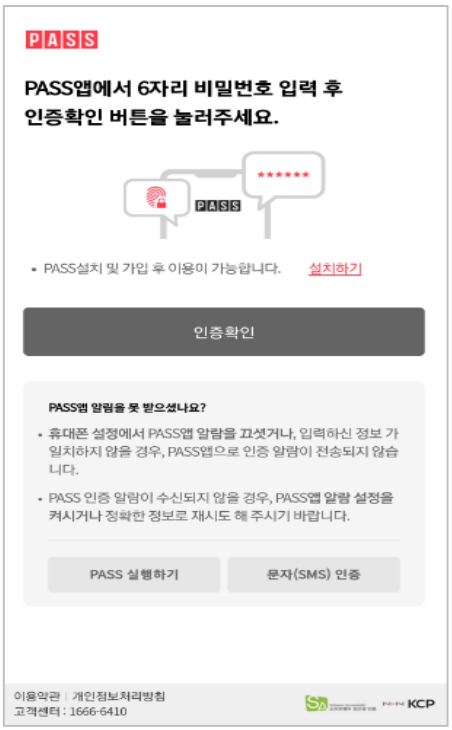
예)

```
private class mWebViewClient extends WebViewClient
{
    @Override
    public boolean shouldOverrideUrlLoading( WebView view, String url )
    {
        if (url != null && !url.equals("about:blank"))
        {
            if ( url.contains("intent:") || url.contains("market:") )
            {
                return url_scheme_intent( url );
            }
        }

        return true;
    }
}

private boolean url_scheme_intent( String url )
{
    // intent 방식
    if ( url.startsWith( "intent" ) )
    {
        Intent intent = null;

        try {
            intent = Intent.parseUri( url, Intent.URI_INTENT_SCHEME );
        } catch ( URISyntaxException ex ) {
            return false;
        }

        // 앱설치 체크를 합니다.
        if ( getPackageManager().resolveActivity( intent, 0 ) == null )
```

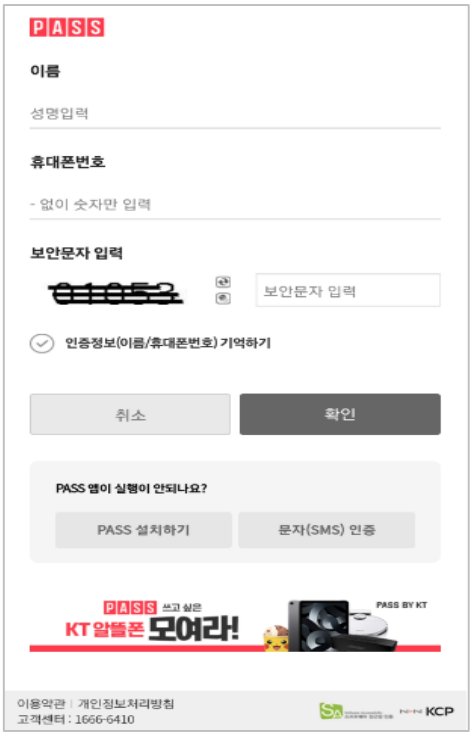
```
        {
            String packagename = intent.getPackage();

            if ( packagename != null )
            {
                startActivity( new Intent( Intent.ACTION_VIEW, Uri.parse( "market://details?id=" + packagename ) ) );

                return true;
            }
        }

        intent = new Intent( Intent.ACTION_VIEW, Uri.parse( intent.getDataString() ) );

        try{
            startActivity( intent );
        }catch( ActivityNotFoundException e ) {
            return false;
        }
    }
    // 기존 스키마 방식
    else
    {
        try
        {
            startActivity( new Intent( Intent.ACTION_VIEW, Uri.parse( url ) ) );
        }
        catch(Exception e)
        {
            // 어플이 설치 안되어 있을 경우 오류 발생. 해당 부분은 업체에 맞게 구현
            Toast.makeText(this, "해당 어플을 설치해 주세요.", Toast.LENGTH_LONG).show();
        }
    }

    return true;
}
```

※ APP 서비스 이용하시지 않고, WEB 서비스만 사용시 참고 가이드 입니다.

[PASS UI 화면]



[PUSH UI 화면]



*** 통신사 선택 이후에 노출되는 화면에 따른 아래의 내용 참고하여 적용 부탁 드립니다.**

[PASS UI] 와 같이 호출이 되는 경우

- Android OS : 수정 없이 사용
- IOS : 수정 없이 사용

[PUSH UI] 와 같이 호출이 되는 경우

- Android OS : **intent URL 처리 없이** 해당 **변수만 전달** 부탁 드립니다.
- IOS : **URL Scheme 처리 없이** 해당 **변수만 전달** 부탁 드립니다.