



# 원모어패스

## 클라우드

### REST API - U2F

#### 목차

- U2F 란?
- 준비 사항
- OMPASS 적용 프로세스
- OMPASS-U2F
- OMPASS 등록 및 U2F 인증
- 인증 토큰 받기
- 인증 토큰 검증
- API 에러 메시지

## U2F 란?

U2F는 패스워드 인증이 완료된 후 OMPASS 2차 인증을 통해 최종 로그인이 가능한 인증 방식입니다.

## 준비 사항

OMPASS 관리자 페이지에서 어플리케이션을 등록과 동시에 자동 할당된 Secret Key 정보를 확인하시기 바랍니다.

**Secret Key가 외부에 노출될 경우 보안 문제가 발생하오니 절대 노출되지 않게 주의해야 합니다.**

## OMPASS 적용 프로세스

1 사용자 아이디 패스워드 검증

3 OMPASS 등록 및 U2F 인증

5 인증 토큰 검증

2 OMPASS 등록 · 인증 API

4 인증 토큰 받기

6 로그인

## OMPASS-U2F

server-side

server-side에서 사용자의 패스워드 인증이 완료된 후 HTTP HEADER에는 Secret Key를 포함하고 Request Body에는 사용자의 아이디를 포함하여 OMPASS API를 호출합니다.

## ■ OMPASS 등록 · 인증 API

POST

URL /v1/ompass/u2f

URL EXAMPLE - <https://interface-api.ompasscloud.com/v1/ompass/u2f>

## ■ Header

Key	Type	Description
Authorization	Bearer	· 어플리케이션에 할당된 Secret Key
		· Authorization 타입으로 'Bearer'를 반드시 명시하고 'Bearer' 와 'Secret Key' 사이에 공백 필수
		· Example : Bearer djfk3dkfdl39dldjmgjd4idls83jflghidhs83jfk

## ■ Request Body (JSON)

Key	Type	Description
user_id	String	사용자 아이디
lang_init	String	응답받을 OMPASS URI의 초기 언어 설정 값 (KR 또는 EN)

## ■ Example of Request Body

```
{
  "user_id": "omsecurity",
  "lang_init": "KR"
}
```

## ■ Response (JSON)

· 인증 성공시

Key	Type	Description
user_id	String	사용자 아이디
is_register	Boolean	해당 사용자의 OMPASS 등록 여부
ompass_uri	String	해당 사용자가 OMPASS 에 등록되어 있지 않으면 등록 페이지 URI 반환
		해당 사용자가 OMPASS 에 등록되어 있으면 인증 페이지 URI 반환

```
{
  "code": 200,
  "message": "ok",
  "data": {
    "user_id": "omsecurity",
    "is_register": false,
    "ompass_uri": "https://interface-api.ompasscloud.com/register/did/14?do..."
  }
}
```

# OMPASS 등록 및 U2F 인증

client-side

응답받은 OMPASS URI를 브라우저(client-side)에서 호출합니다.

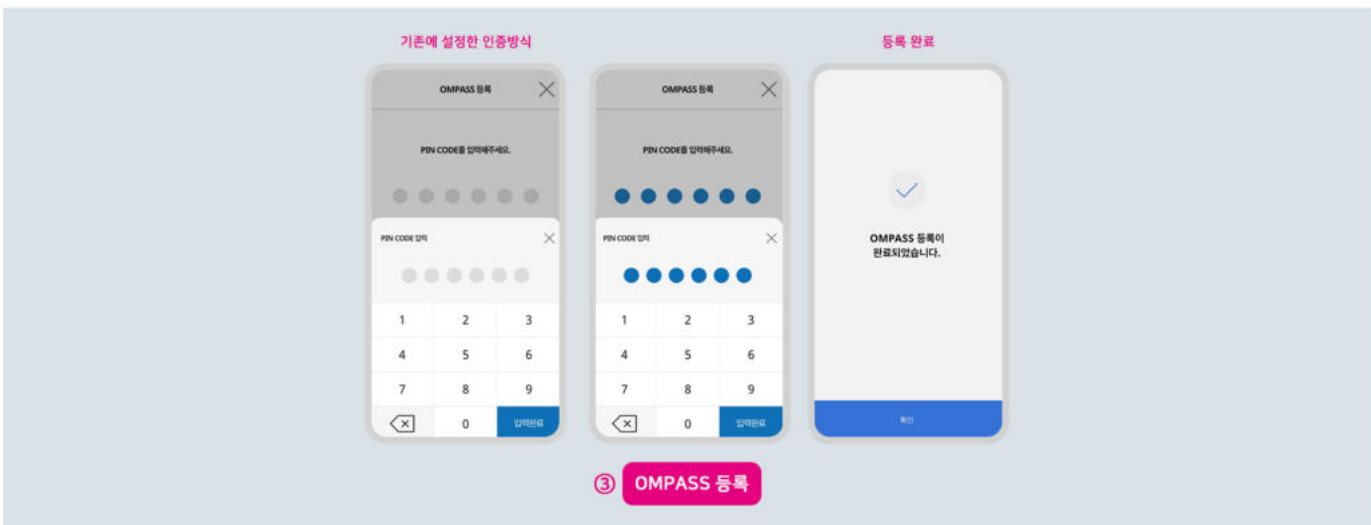
등록 인터페이스 호출 예시 (팝업 창)

- OMPASS에 등록되어 있지 않은 사용자인 경우 아래와 같은 등록 팝업 창이 표시됩니다.



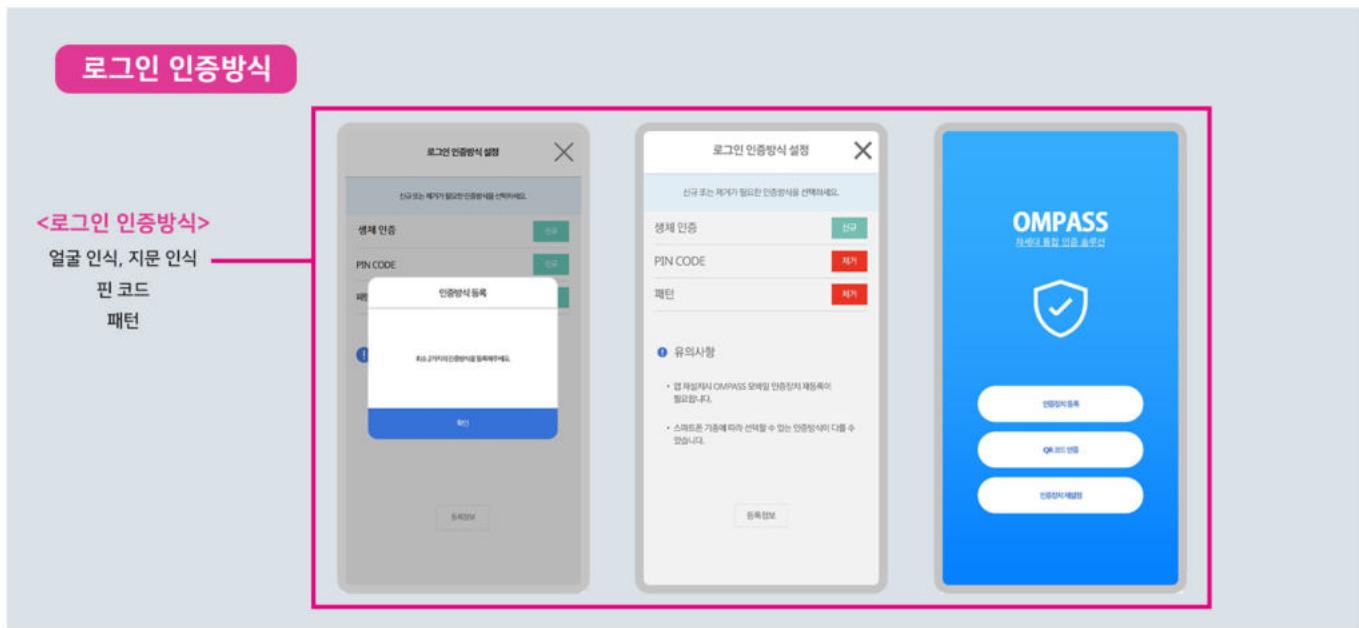
① “2차인증 등록하기” 버튼을 클릭해주세요.

② QR코드를 OMPASS 앱으로 스캔해주세요.



③ 인증방식 등록 완료 후 선택한 인증방식으로 등록을 완료해주세요.

### ③번의 인증방식 등록 방법 예시)



### 인증 인터페이스 호출 예시 (팝업 창)

- OMPASS에 등록되어 있는 사용자인 경우 아래와 같은 인증 팝업 창이 표시됩니다.



- ① “2차 인증하기” 버튼을 클릭해주세요.
- ② 모바일로 OMPASS 인증 알림이 옵니다.
- ③ 기존에 선택했던 인증방식으로 인증을 완료해주세요.

## 인증 토큰 받기

client-side

OMPASS 등록 또는 인증을 완료했을 때 OMPASS 페이지 팝업 창에서 해당 어플리케이션에 지정되어 있는 리다이렉트 URI로 HTTP 리다이렉트되며, 인증 토큰이 담긴 쿼리 스트링 (Query String)을 포함합니다. 리다이렉트된 인증 토큰을 파싱(client-side)하여 server-side로 전달합니다.

## 인증 토큰 검증

server-side

client-side에서 전달받은 인증 토큰을 포함하여 OMPASS 인증 토큰 검증 API를 호출하여 토큰의 유효성을 검증 받습니다. API 요청의 응답 HTTP STATUS CODE가 200 이면 해당 아이디를 확인 후 로그인 처리합니다.

### ■ OMPASS 성공 토큰 검증 API

POST

URL /v1/ompass/token-verification

URL EXAMPLE – <https://interface-api.ompasscloud.com/v1/ompass/token-verification>

### ■ Header

Key	Type	Description
Authorization	Bearer	· 어플리케이션에 할당된 Secret Key
		· Authorization 타입으로 'Bearer'를 반드시 명시하고 'Bearer' 와 'Secret Key' 사이에 공백 필수
		· Example : Bearer djfk39dkfdl39dldjmgjd4idls83jflghidhs83jfk

### ■ Request Body (JSON)

Key	Type	Description
user_id	String	사용자 아이디
access_token	String	리다이렉트 URI로 전달 받은 access_token

### ■ Example of Request Body

```
{
  "user_id": "omsecurity",
  "access_token": "dfj2ld92lldj29cldl29llduuufnbsd229312000df12ldio2o019029dj10wj"
}
```

## ■ Response (JSON)

### · 인증 성공시

Key	Type	Description
user_id	String	사용자 아이디

```
{
  "code": 200,
  "message": "ok",
  "data": {
    "user_id": "omsecurity"
  }
}
```

## API 에러 메시지

code	message	description
000	Required Request Body is missing.	Request Body 가 누락되었습니다.
001	Please make a request including the secret key.	Secret Key 를 포함하여 요청해주세요.
002	Please make a request including the user ID.	사용자 ID 를 포함하여 요청해주세요.
003	Please make a request including the access token.	access token 을 포함하여 요청해주세요.
004	Invalid secret key.	유효하지 않은 Secret Key 입니다.
005	The secret key format does not match. example) Bearer dl239d29dl292kmdjf139f2ds	Secret Key 의 형식이 올바르지 않습니다. 예) Bearer dl239d29dl292kmdjf139f2ds")
006	User ID cannot exceed 30 digits.	사용자 ID 의 길이는 30자를 초과할 수 없습니다.
011	The token has expired.	токен이 만료되었습니다.
012	It is a token of an unsupported format.	지원하지 않는 형식의 토큰입니다.
013	The token is not configured correctly.	토큰의 구성이 올바르지 않습니다.
014	Failed to verify the existing signature.	기존 서명을 확인하지 못했습니다.