

# 오너클랜 API 파트너 가이드

인증받은 오너클랜 API 파트너가 다른 오너클랜 사용자들을 위해 API를 사용할 때 각 사용자마다 API 권한을 취득하는 과정을 생략하고 오너클랜 API 사용 권한 획득을 대행하기 위한 API입니다.

사용자 등록과 해제, 등록된 사용자 리스트 조회 기능을 제공합니다.

파트너 인증은 access key / secret key pair로 진행하며 별도로 발급을 요청해야 합니다.

## 유의사항

- 오너클랜 사용자의 API 사용 권한 획득과 파트너 계정(access key / secret key) 발급은 다른 프로세스입니다.
- 기존에 오너클랜에 가입된 사용자들을 위한 서비스를 만들 때 API를 사용하고자 하는 경우, 파트너는 별도의 오너클랜 계정 생성 없이 파트너 계정만 발급받으면 됩니다. 그 이후에 해당 계정(access key / secret key)를 통해 이 가이드에 서술된 것처럼 오너클랜 API에 오너클랜 사용자 계정을 등록하거나 해제할 수 있습니다.

## 사용자 등록

파트너 정보로 오너클랜 API를 사용할 오너클랜 계정을 등록합니다. 단, 기존에 파트너 없이 API를 사용하고 있는 오너클랜 계정은 등록할 수 없습니다.

### 사용자 등록 엔드포인트

- Sandbox 환경: <https://api-sandbox.ownerclan.com/v1/partner/registerUser>
- Production 환경: <https://api.ownerclan.com/v1/partner/registerUser>

### 사용자 등록 방식

사용자 등록은 전용 엔드포인트에 등록할 사용자의 데이터와 파트너 access key / secret key pair를 HTTP POST 요청으로 전송하는 방식으로 이루어집니다. 예시 코드는 다음과 같습니다.

```
// 등록할 사용자의 정보를 설정합니다.
// username: 사용자의 오너클랜 ID
// passwordHash: 사용자의 오너클랜 PW에 대한 md5 hash hex 값입니다.
// userType: 오너클랜 판매사 계정이면 "seller", 공급사 계정이면 "vendor"
var userInfo = {
    username: "USERNAME",
    passwordHash: "PASSWORD_HASH",
    userType: "USERTYPE"
};

// XMLHttpRequest object를 생성합니다.
var client = new XMLHttpRequest();

// XMLHttpRequest의 HTTP 요청 타입과 url을 설정합니다.
// Production 환경인 경우 https://api.ownerclan.com/v1/partner/registerUser 엔드포인트를 사용하면 됩니다.
```

```

client.open("POST", "https://api-sandbox.ownerclan.com/v1/partner/registerUser",
true);

// HTTP POST content-type을 설정합니다. 반드시 application/json이어야 합니다.
client.setRequestHeader("Content-Type", "application/json");

// 파트너 access key / secret key pair 데이터를 설정합니다.
client.setRequestHeader("accesskey", "PARTNER_ACCESS_KEY");
client.setRequestHeader("secretkey", "PARTNER_SECRET_KEY");

// XMLHttpRequest 요청이 완료되었을 때의 callback을 설정합니다.
client.onreadystatechange = function(aEvt) {
    if (client.readyState === 4) {
        if (client.status === 200) {
            // API 서버 응답이 정상인 경우 결과를 콘솔에 씁니다.
            console.log(client.responseText);
        } else {
            // API 서버 응답이 정상이 아닌 경우 에러와 HTTP status code를 콘솔에 씁니다.
            console.error(client.status, client.responseText);
        }
    }
}

// XMLHttpRequest 요청을 전송합니다.
client.send(JSON.stringify(userInfo));

```

#### 요구사항:

- 사용자 데이터는 username, passwordHash, userType 필드가 있는 JSON object이어야 합니다.
- 사용자 데이터는 HTTP POST body로 설정하면 되고, Content-Type header를 application/json으로 설정해야 합니다.
- 사용자 데이터의 userType은 해당 사용자가 오너클랜 판매사인지 공급사인지의 여부이며, 판매사인 경우 seller, 공급사인 경우 vendor를 사용하면 됩니다.
- 사용자 데이터의 passwordHash는 오너클랜 계정의 비밀번호에 대한 md5 hash 값이며, hex로 인코딩 된 결과입니다.

◦ nodejs 예제는 다음과 같습니다.

```

var crypto = require("crypto");
var password = "TARGET_PASSWORD"; // 실제 사용자 비밀번호 값.
var passwordHash =
    crypto.createHash("md5").update(password).digest("hex");

```

- 파트너 access key / secret key는 HTTP header에 각각 accesskey, secretkey라는 이름으로 설정하면 됩니다.
- HTTP 요청은 POST만 가능합니다.

#### 반환 데이터:

- 요청에 성공할 경우 HTTP status code 200과 "OK" 메시지를 반환합니다.
- 요청에 실패할 경우 HTTP status code 4xx와 에러 메시지를 반환합니다.

---

## 사용자 등록 해제

파트너를 통해 등록한 오너클랜 계정의 API 사용 권한을 해제합니다. 파트너를 통해 등록한 계정만 권한을 해제할 수 있습니다.

### 사용자 등록 해제 엔드포인트

- Sandbox 환경: <https://api-sandbox.ownerclan.com/v1/partner/unregisterUser>
- Production 환경: <https://api.ownerclan.com/v1/partner/unregisterUser>

### 사용자 등록 해제 방식

사용자 등록 해제는 전용 엔드포인트에 등록을 해제할 사용자의 데이터와 파트너 accessKey / secretKey pair를 HTTP POST 요청으로 전송하는 방식으로 이루어집니다. 예시 코드는 다음과 같습니다.

```
// 등록할 사용자의 정보를 설정합니다.
// username: 사용자의 오너클랜 ID
// userType: 오너클랜 판매사 계정이면 "seller", 공급사 계정이면 "vendor"
var userInfo = {
    username: "USERNAME",
    userType: "USERTYPE"
};

// XMLHttpRequest object를 생성합니다.
var client = new XMLHttpRequest();

// XMLHttpRequest의 HTTP 요청 타입과 url을 설정합니다.
// Production 환경인 경우 https://api.ownerclan.com/v1/partner/unregisterUser 엔드
// 포인트를 사용하면 됩니다.
client.open("POST", "https://api-sandbox.ownerclan.com/v1/partner/unregisterUser",
true);

// HTTP POST content-type을 설정합니다. 반드시 application/json이어야 합니다.
client.setRequestHeader("Content-Type", "application/json");

// 파트너 access key / secret key pair 데이터를 설정합니다.
client.setRequestHeader("accesskey", "PARTNER_ACCESS_KEY");
client.setRequestHeader("secretkey", "PARTNER_SECRET_KEY");

// XMLHttpRequest 요청이 완료되었을 때의 callback을 설정합니다.
client.onreadystatechange = function(aEvt) {
    if (client.readyState === 4) {
        if (client.status === 200) {
            // API 서버 응답이 정상인 경우 결과를 콘솔에 씁니다.
            console.log(client.responseText);
        } else {
            // API 서버 응답이 정상이 아닌 경우 에러와 HTTP status code를 콘솔에 씁니
            다.
        }
    }
}
```

```

        console.error(client.status, client.responseText);
    }
}

// XMLHttpRequest 요청을 전송합니다.
client.send(JSON.stringify(userInfo));

```

#### 요구사항:

- 사용자 데이터는 username, userType 필드가 있는 JSON object이어야 합니다.
- 사용자 데이터는 HTTP POST body로 설정하면 되고, Content-Type header를 application/json으로 설정해야 합니다.
- 사용자 데이터의 userType은 해당 사용자가 오너클랜 판매사인지 공급사인지의 여부이며, 판매사인 경우 seller, 공급사인 경우 vendor를 사용하면 됩니다.
- 파트너 access key / secret key는 HTTP header에 각각 accesskey, secretkey라는 이름으로 설정하면 됩니다.
- HTTP 요청은 POST만 가능합니다.

#### 반환 데이터:

- 요청에 성공할 경우 HTTP status code 200과 "OK" 메시지를 반환합니다.
- 요청에 실패할 경우 HTTP status code 4xx와 에러 메시지를 반환합니다.

## 사용자 리스트 조회

파트너를 통해 등록한 오너클랜 계정의 리스트를 조회합니다.

#### 사용자 리스트 조회 엔드포인트

- Sandbox 환경: <https://api-sandbox.ownerclan.com/v1/partner/listUsers>
- Production 환경: <https://api.ownerclan.com/v1/partner/listUsers>

#### 사용자 리스트 조회 방식

사용자 리스트 조회는 전용 엔드포인트에 파트너 accessKey / secretKey pair를 HTTP POST 요청으로 전송하는 방식으로 이루어집니다. 예시 코드는 다음과 같습니다.

```

// XMLHttpRequest object를 생성합니다.
var client = new XMLHttpRequest();

// XMLHttpRequest의 HTTP 요청 타입과 url을 설정합니다.
// Production 환경인 경우 https://api.ownerclan.com/v1/partner/listUsers 엔드포인트
// 를 사용하면 됩니다.
client.open("GET", "https://api-sandbox.ownerclan.com/v1/partner/listUsers",
true);

// 파트너 access key / secret key pair 데이터를 설정합니다.
client.setRequestHeader("accesskey", "PARTNER_ACCESS_KEY");

```

```

client.setRequestHeader("secretkey", "PARTNER_SECRET_KEY");

// XMLHttpRequest 요청이 완료되었을 때의 callback을 설정합니다.
client.onreadystatechange = function(aEvt) {
    if (client.readyState === 4) {
        if (client.status === 200) {
            // API 서버 응답이 정상인 경우 결과를 콘솔에 씁니다.
            console.log(client.responseText);
        } else {
            // API 서버 응답이 정상이 아닌 경우 에러와 HTTP status code를 콘솔에 씁니다.
            console.error(client.status, client.responseText);
        }
    }
}

// XMLHttpRequest 요청을 전송합니다.
client.send(null);

```

#### 요구사항:

- 파트너 access key / secret key는 HTTP header에 각각 **accesskey**, **secretkey**라는 이름으로 설정하면 됩니다.
- HTTP 요청은 GET만 가능합니다.

#### 반환 데이터:

- 요청에 성공할 경우 HTTP status code 200과 등록된 사용자의 계정명(ID) 및 타입의 배열을 반환합니다.
- 요청에 실패한 경우 HTTP status code 4xx와 에러 메시지를 반환합니다.