apps in toss

Menu On this page

파트너사 로그인 연동하기

파트너사 서비스에 토스 로그인을 연동해보세요.

시작하기

로그인 연동을 위해서는 사전 셋팅이 필요해요.
아래 정보들을 cert.support@toss.im 으로 전달해주세요.

> 🚨 잠시만요
>
> 파트너사 로그인 어드민 페이지는 없습니다.

목록	설명
웹/앱 여부	웹, 앱, 둘 다
회원 관리 키	CI, e-mail, 휴대폰번호 등
사용자의 권한 목록	이름, 이메일 등 필요한 권한을 아래에서 선택해주세요
로고 이미지	원형, 600px x 600px 권장드려요
약관 목록	약관 제목과 약관 URL, 필수 여부를 보내주세요
redirect_uri	완료 후 보내질 화면 url 을 입력해주세요
연동 예정인 앱 버전	파트너사 앱에도 도입할 경우 해당됩니다
로그인 연결 끊기 API 사용 유무	default 는 사용안함으로 설정되요
네트워크 정보	VPN 사용 유무나 개발환경에서 사용 시 별도 등록이 필요해요

개발을 위한 키 확인하기

연동을 위해서는 아래 정보들을 받아야해요.
위 정보들을 메일로 보내주시면 토스 로그인 담당자가 직접 회신해드립니다.

- clientId
- clientSecret
- 복호화 키

> 🚨 잠시만요
>
> 앱인토스용 로그인의 clientId, 복호화 키와 상이합니다.
> 앱인토스용 로그인이 아니기 때문에 mTLS 인증서 적용이 불필요해요.
> 앱인토스용 로그인과 동일한 userKey 사용을 위해서는 앱인토스 콘솔에서 연동해주세요.

API 공통규격 확인하기

도메인 정보

https://oauth2.cert.toss.im

API 공통 응답

성공

json

```
// 성공일 경우 resultType이 SUCCESS로 설정되며 해당 API의 응답이 success 하위에 적재
{
   "resultType":"SUCCESS",
   "success":{
      "sample":"data"
   }
}
```

실패

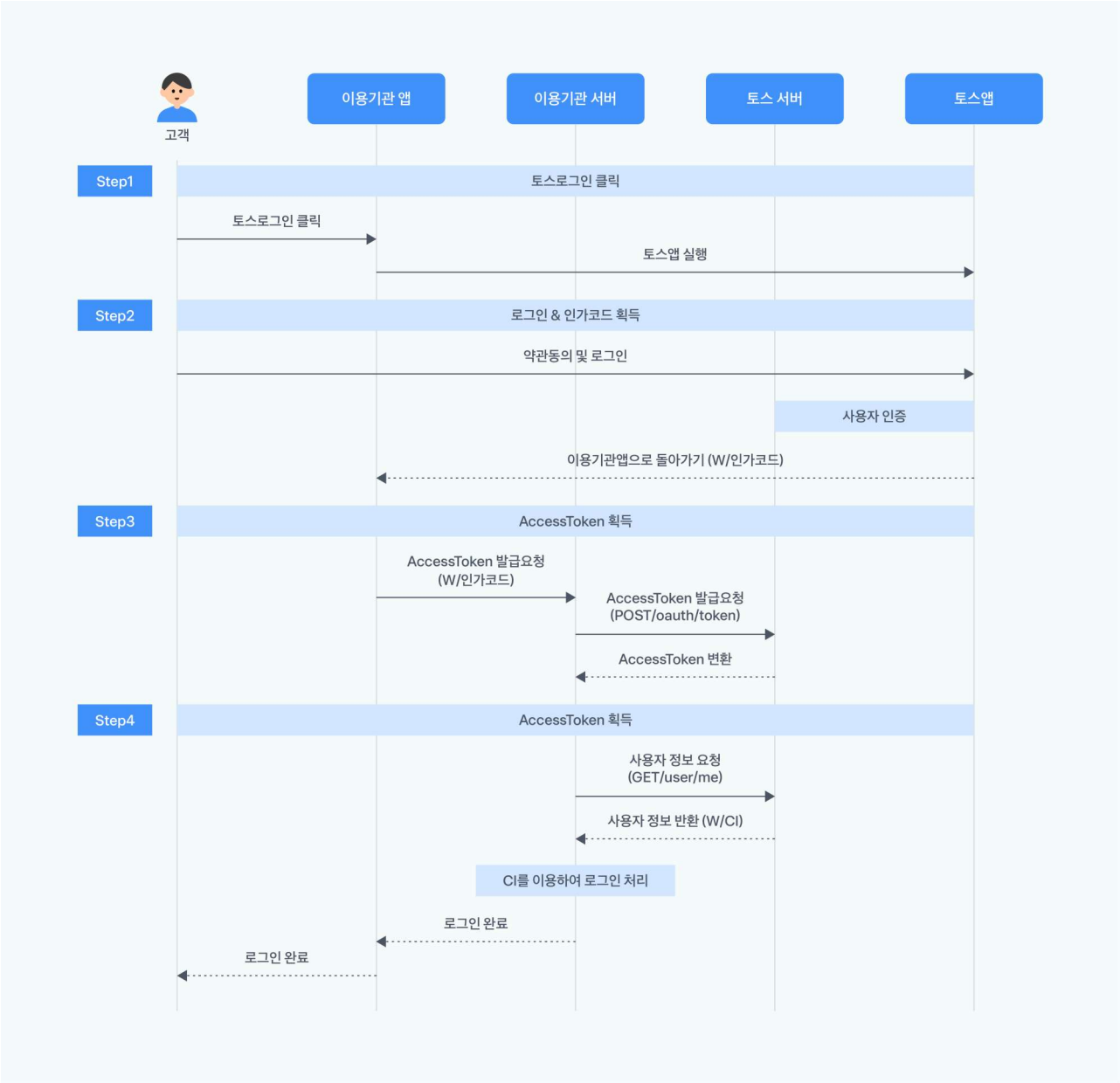
json

```
// 실패일 경우 resultType 이 FAIL로 설정되며 해당 실패 사유가 error 하위에 적재됩니다
{
   "resultType":"FAIL",
   "error":{
      "errorCode":"INVALID_PARAMETER",
      "reason":"요청에 실패했습니다."
```

```
        }
    }
```

개발하기


고객
이용기관 앱
이용기관 서버
토스 서버
토스 로그인 앱
토스앱
Step1
토스로그인창 접속
토스로그인 클릭
OAuth 인증 요청 (GET/oauth/authorize)
토스로그인창으로 302 Redirect
Redirect
토스로그인창 접속
Step2
로그인 & 인가코드 획득
약관동의 및 로그인
토스앱 실행
사용자 인증
인증완료
이용기관앱으로 돌아가기 (W/인가코드)
Step3
AccessToken 획득
Redirect URL 접속
(W/인가코드)
AccessToken 발급요청
(POST/oauth/token)
AccessToken 변환
Step4
사용자 정보 획득 및 로그인 처리
사용자 정보 요청
(GET/user/me)
사용자 정보 반환 (W/CI)
CI를 이용하여 로그인 처리
로그인 완료
로그인 완료



---

1. 인가 코드 받기

사용자의 인증을 요청합니다.
사용자의 인증에 성공하면 인가 코드를 redirect_uri 에 포함시켜 redirect 시켜요.

- Content-type : application/json
- Method : `GET`
- URL : `/authorize`

> 🚨 잠시만요
>
> 인가코드의 유효시간은 10분입니다.

요청

이름	타입	필수값 여부	설명
grant_type	string	Y	인가방법
client_id	string	Y	사전에 발급 받은 client_id
redirect_uri	string	Y	인가 코드를 전달받을 서비스 서버의 URI
response_type	string	Y	code 로 고정
scope	string	N	획득하고자 하는 권한(e.g. user_name)
state	string	N	CSRF 공격 방지 토큰. redirect_uri 에 포함되어 redirect 됩니다
intentUnsupported	string	N	안드로이드 웹뷰에서 intent 파싱이 되지 않는 버전 구분
policy	string	N	OAuth 로그인 연동 동선 구분 = `LOGIN` 으로 값 고정 필요
app	string	N	브라우저에 세션이 남아 있어도, 토스앱을 거치도록 강제하는 옵션. requires 로 주면 강제됩니다.
flow	string	N	브라우저에 세션을 무시하고 동작하도록 하는 옵션. ignore_session 으로 주면 세션이 무시됩니다.

성공 응답

성공시 요청 파라미터의 redirect_uri 로 HTTP 302 redirect 하며 응답이 리턴 됩니다.

이름	타입	필수	설명
code	string	Y	AccessToken 획득을 위한 인가 코드
state	string	N	요청 파라미터로 넣어주신 state 값

실패 응답

실패시 요청 파라미터의 redirect_uri 로 HTTP 302 redirect 하며 응답이 리턴 됩니다.

이름	타입	필수	설명
error	string	Y	에러 코드

이름	타입	필수	설명
error_description	string	N	에러 메시지
state	string	N	요청 파라미터로 넣어주신 state 값

```
// 포맷
https://oauth2.cert.toss.im/authorize?grant_type=$grant_type&client_id=$client_i

// 예시
https://oauth2.cert.toss.im/authorize?grant_type=authorization_code&client_id=wn
```

---

2. AccessToken 받기

사용자 정보 조회 API 를 사용하기 위한 접근 토큰을 발급합니다.

- Content-type : application/json
- Method : `POST`
- URL : `/token`

> 🚨 잠시만요
>
> AccessToken 유효시간은 1시간이에요.

요청

이름	타입	필수값 여부	설명
client_id	string	Y	사전에 발급 받은 client_id
grant_type	string	Y	authorization_code 로 고정
code	string	Y	인가코드받기에서 획득한 code
client_secret	string	Y	사전에 발급 받은 client_secret

이름	타입	필수값 여부	설명
redirect_uri	string	Y	인가코드받기에서 넘겨준 uri

요청 예시

```
// 포맷
curl --request POST 'https://oauth2.cert.toss.im/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'code=$code' \
--data-urlencode 'client_id=$client_id' \
--data-urlencode 'client_secret=$client_secret' \
--data-urlencode 'redirect_uri=$redirect_uri'

// 예시
curl --request POST 'https://oauth2.cert.toss.im/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'code=fKjvI8ILZcj3Q8gX-8lFM2weU-_0Z0tLsNK_OiPG6iCWh0wpQW5QPAL_K
--data-urlencode 'client_id=3ezt6e1th86oday9phzz84u4ovng6ss' \
--data-urlencode 'client_secret={Client_secret}' \
--data-urlencode 'redirect_uri={redirect_uri}'
```

성공 응답

이름	타입	필수	설명
token_type	string	Y	bearer 로 고정
access_token	string	Y	AccessToken
refresh_token	string	Y	RefreshToken
expires_in	number	Y	만료시간(초)
scope	string	Y	인가된 scope(구분)

json

```
// 포맷
{
    "access_token": $access_token,
```

```
    "scope": $scope,
    "token_type": "Bearer",
    "expires_in": $expires_in
}

// 예시
{
    "access_token":"eyJraWQiOiJjZXJ0IiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiJtMHVmMmhaU
    "refresh_token":"xNEYPASwWw0n1AxZUHU9KeGj8BitDyYo4wi8rpfkUcJwByVxpAdUzwtIaWG
    "scope":"user_ci user_birthday user_nationality user_name user_phone user_ge
    "token_type":"Bearer",
    "expires_in":3599
}
```

실패 응답

인가 코드가 만료되었거나 동일한 인가 코드로 AccessToken 을 중복으로 요청할 경우

json

```
{
    "error":"invalid_grant"
}
```

json

```
{
    "resultType":"FAIL",
    "error":{
        "errorCode":"INTERNAL_ERROR",
        "reason":"요청을 처리하는 도중에 문제가 발생했습니다."
    }
}
```

---

3. AccessToken 재발급 받기

사용자 정보 조회 API 를 사용하기 위한 접근 토큰을 재발급합니다.

- Content-type : application/json
- Method : `POST`

- URL : `/token`

> 🚨 잠시만요
>
> refreshToken 유효시간은 1일이에요.

요청

이름	타입	필수	설명
client_id	string	Y	사전에 발급 받은 client_id
grant_type	string	Y	refresh_token 로 고정
refresh_token	string	Y	2 에서 획득한 RefreshToken
client_secret	string	Y	사전에 발급 받은 client_secret

요청 예시 입력

```
// 포맷
curl --request POST 'https://oauth2.cert.toss.im/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'refresh_token=$refresh_token' \
--data-urlencode 'client_id=$client_id' \
--data-urlencode 'client_secret=$client_secret' \
--data-urlencode 'redirect_uri=$redirect_uri'

// 예시
curl --request POST 'https://oauth2.cert.toss.im/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'refresh_token=xNEYPASwWw0n1AxZUHU9KeGj8BitDyYo4wi8rpfkUcJwByVx
--data-urlencode 'client_id=3ezt6e1th86oday9p9cuy84u4bbng6ss' \
--data-urlencode 'client_secret={Client_secret}' \
--data-urlencode 'redirect_uri={redirect_uri}'
```

성공 응답

이름	타입	필수	설명
token_type	string	Y	bearer 로 고정

이름	타입	필수	설명
access_token	string	Y	AccessToken
refresh_token	string	Y	RefreshToken
expires_in	number	Y	만료시간(초)
scope	string	Y	인가된 scope(구분)

실패 응답

이름	타입	필수	설명
error	string	Y	에러 코드
error_description	string	Y	에러 메시지

---

4. 사용자 정보 받기

사용자 정보를 조회합니다.
DI는 null 로 내려가며 횟수 제한없이 호출이 가능해요.
개인정보 보호를 위해 암호화된 형태로 제공합니다.

- Content-type : application/json
- Method : `GET`
- URL : `/api-partner/v1/apps-in-toss/user/oauth2/login-me`

> 🚨 잠시만요
>
> 앱인토스용 로그인 시의 userKey와는 다른 userKey가 내려가요.
> 동일한 사업자의 경우 동일한 userKey가 내려가도록 7월에 수정될 예정이니 참고해주세요.

요청 헤더

이름	타입	필수값 여부	설명
Authorization	string	Y	AccessToken으로 인증 요청 `Authorization: Bearer ${AccessToken}`

```
// 포맷
curl --request GET 'https://oauth2.cert.toss.im/oauth2/api/login/user/me/without
--header 'Authorization: Bearer $access_token'


// 예시
curl --request GET 'https://oauth2.cert.toss.im/oauth2/api/login/user/me/without
--header 'Authorization: Bearer eyJraWQiOiJjZXJ0IiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiO
```

성공 응답

이름	타입	필수	암호화 여부	설명
userKey	number	Y		유저식별자
scope	string	Y		인가된 scope(구분)
agreedTerms	list	Y		동의한 약관 목록
policy	string	Y		OAuth 로그인 연동 동선 구분 = `LOGIN` 으로 값 고정 필요
certTxId	string	N		회원가입인 경우, 본인확인 식별자. 앱인토스 파트너사의 경우 해당사항 없음
ci	string		Y	CI
name	string		Y	이름
phone	string		Y	휴대전화번호
gender	string		Y	성별(MALE/FEMALE)
nationality	string		Y	내/외국인여부(LOCAL/FOREIGNER)
birthday	string		Y	생년월일(yyyyMMdd)
email	string		Y	이메일 (점유인증 하지 않은 이메일 정보)

json

```
// 예시
{
    "resultType":"SUCCESS",
    "success":{
        "userKey":443731104,
```

```
        "scope":"user_ci,user_birthday,user_nationality,user_name,user_phone,use
        "agreedTerms":[],
        "policy":"AUTO_SELECT",
        "certTxId":"ad052b57-dc8f-4cdb-a6e2-7b494e28b5ec",
        "name":"ENCRYPTED_VALUE",
        "phone":"ENCRYPTED_VALUE",
        "birthday":"ENCRYPTED_VALUE",
        "ci":"ENCRYPTED_VALUE",
        "di":null,
        "gender":"ENCRYPTED_VALUE",
        "nationality":"ENCRYPTED_VALUE",
        "email":null
    }
}
```


실패 응답

유효하지 않은 토큰을 사용할 경우, 현재 사용 중인 access_token의 유효시간을 확인하고 재발급을 진행해주세요.

json


```
// 예시
{
    "error":"invalid_grant"
}
```


서버 에러 응답 예시

errorCode	설명
INTERNAL_ERROR	내부 서버 에러
USER_KEY_NOT_FOUND	로그인 서비스에 접속한 유저 키 값을 찾을 수 없음
USER_NOT_FOUND	토스 유저 정보를 찾을 수 없음
BAD_REQUEST_RETRIEVE_CERT_RESULT_EXCEEDED_LIMIT	조회 가능 횟수 초과 동일한 토큰으로 `/api/login/user/me/without-di` API 조회하면 정상적으로 조회되나, di 필드는 null 값으로 내려감

json

```
// 예시
{
    "resultType":"FAIL",
    "error":{
        "errorCode":"INTERNAL_ERROR",
        "reason":"요청을 처리하는 도중에 문제가 발생했습니다."
    }
}
```

---

5. 사용자 정보 복호화하기

이메일로 받은 `복호화 키` 와 `AAD(Additional Authenticated DATA)` 로 진행해주세요.

암호화 알고리즘

- AES 대칭키 암호화
- 키 길이 : 256비트
- 모드 : GCM
- AAD : 복호화 키와 함께 이메일로 전달드립니다.

데이터 교환방식

- 암호화된 데이터의 앞 부분에 IV/NONCE 값을 첨부하여 드립니다.
- 복호화할 때는 암호화된 데이터의 앞 부분에서 IV/NONCE 값을 추출하여 사용하시길 바랍니다.

복호화 샘플 코드

▶ Kotlin 예제

▶ PHP 예제

▶ JAVA 예제

6. 로그인 끊기

발급받은 AccessToken을 더 이상 사용하지 않거나 사용자의 요구에 의해 만료시킬 경우 토큰을 삭제(만료)해주세요.

- Content-type : application/json
- Method : `POST`
- URL : `/api/login/access/remove`

AccessToken 으로 로그인 연결 끊기

```
// 포맷
curl --request POST 'https://oauth2.cert.toss.im/oauth2/api/login/access/remove'
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer $access_token'

// 예시
curl --request POST 'https://oauth2.cert.toss.im/oauth2/api/login/access/remove'
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJraWQiOiJjZXJ0IizzYWxnIjoiUlMyNTYifQ.eyJzdWIiO
```

userKey 를 이용하여 고객사에서 새로운 access_token을 발급 후 로그인 연결 끊기

Client Credentials Grant (클라이언트 자격증명 승인 방식) 방식으로 획득한 access_token을 사용 : 권한이 있는 Client에서 로그인 연결 해제를 요청하고 있는지를 검증하기 위해 client_credentials 방식으로 고객사에서 access_token을 발급해서 특정 사용자의 로그인 연결을 끊는 방식

> 🚨 주의해주세요
>
> 연결 끊기 요청의 키로 userKey를 사용하기 때문에 유저 연결 관리를 위해 사용자의 userKey 개별적으로 보관 필요
>
> scope에 연결 끊기 요청을 사용할 수 있는 Client임이 설정되어 있어야 함(참고하여 사전 요청 필요)

1. access_token 발급 요청 : 요청 시 scope에 `login:access_remove` 꼭 포함되어 있어야 함

- Content-type: application/x-www-form-urlencoded;charset=utf-8

요청

이름	타입	필수	설명
grant_type	string	Y	client_credentials 로 고정
client_id	string	Y	사전에 발급 받은 client_id
client_secret	string	Y	사전에 발급 받은 client_secret
scope	string	Y	login:access_remove

```
// 포맷
curl --request POST 'https://oauth2.cert.toss.im/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=$client_id' \
--data-urlencode 'client_secret=$client_secret' \
--data-urlencode 'scope=$scope'

// 예시
curl --request POST 'https://oauth2.cert.toss.im/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=56gnin234285rm8lq7tm6xl1jtlsevhm' \
--data-urlencode 'client_secret=7SzN7nMGlciRIt2MXndx534Dnskdi3v0po2h4Vunaf1ChP9L
--data-urlencode 'scope=login:access_remove'
```

json

```
{
    "access_token": "eyJraWQiOiJjZXJ0IiwiYWxnIjoiUlMyNS3kdix.esk3XWIiOiI1NmduaW4
    "scope": "login:access_remove",
    "token_type": "Bearer",
    "expires_in": 3599
}
```

로그인 연결 끊기 scope를 사용할 수 있는 client가 아닌 경우

json

```
{"error":"invalid_scope"}
```

2. 로그인 연결 끊기

- Content-type: Content-type: application/json;charset=utf-8

```
// 포맷
curl --request POST 'https://oauth2.cert.toss.im/oauth2/api/login/access/remove'
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer $access_token' \
--data '{"userKey": $user_key}'

// 예시
curl --request POST 'https://oauth2.cert.toss.im/oauth2/api/login/access/remove'
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJraWQiOiJjZXJ0IizzYWxnIjoiUlMyNTYifQ.eyJzdWIiO
--data '{"userKey": 443731103}'
```

json

```
{
    "resultType": "SUCCESS",
    "success": {
        "userKey": 443731103
    }
}
```

요청시 사용한 userKey로 연결된 유저가 없는 경우

json

```
{
    "resultType": "FAIL",
    "error": {
        "errorType": 0,
        "errorCode": "USER_KEY_NOT_FOUND",
        "reason": "UserKeyNotFound",
        "data": {},
        "title": null
    }
}
```

요청시 사용한 access_token의 scope에 로그인 연결 끊기가 포함되지 않은 경우

json

```
{
    "resultType": "FAIL",
    "error": {
        "errorType": 0,
        "errorCode": "BAD_REQUEST_NOT_ALLOWED_SCOPE",
```

```
        "reason": "잘못된 요청입니다.",
        "data": {},
        "title": null
    }
}
```

---

7. 콜백을 통해 로그인 끊기

사용자가 토스앱 내에서 서비스와의 연결을 해제한 경우 가맹점 서버로 알려드려요.
서비스에서 연결이 끊긴 사용자에 대한 처리가 필요한 경우 활용할 수 있어요. 콜백을 받을 URL과 basic Auth 헤더는 콘솔에서 입력할 수 있어요.
서비스에서 직접 로그인 연결 끊기 요청을 호출한 경우 콜백이 호출되지 않아요.

GET 방식

- 요청 requestParam에 `userKey` 와 `referrer` 을 포함합니다.

```
// 포맷
curl --request GET '$callback_url?userKey=$userKey&referrer=$referrer'

// 예시
curl --request GET '$callback_url?userKey=443731103&referrer=UNLINK'
```

POST 방식

- 요청 body에 `userKey` 와 `referrer` 을 포함합니다.

```
// 포맷
curl --request POST '$callback_url' \
--header 'Content-Type: application/json' \
--data '{"userKey": $user_key, "referrer": $referrer}'

// 예시
curl --request POST '$callback_url' \
--header 'Content-Type: application/json' \
--data '{"userKey": 443731103, "referrer": "UNLINK"}'
```

referrer 은 연결 끊기 요청 경로에요.

referrer	설명
`UNLINK`	사용자가 앱에서 연결 끊기
`WITHDRAWAL_TERMS`	로그인 서비스 약관 철회
`WITHDRAWAL_TOSS`	토스 회원 탈퇴

Previous page
토스 로그인 연동하기

Next page
푸시, 알림 연동하기