
LINE Pay 기술 연동 가이드

저작권

Copyright © LINE Corp. All Rights Reserved.

이 문서는 LINE(주)의 지적 재산이므로 어떠한 경우에도 LINE(주)의 공식적인 허가 없이 이 문서의 일부 또는 전체를 복제, 전송, 배포하거나 변경하여 사용할 수 없습니다.

이 문서는 정보 제공의 목적으로만 제공됩니다. LINE(주)는 이 문서에 수록된 정보의 완전성과 정확성을 검증하기 위해 노력하였으나, 발생할 수 있는 내용상의 오류나 누락에 대해서는 책임지지 않습니다. 따라서 이 문서의 사용이나 사용 결과에 따른 책임은 전적으로 사용자에게 있으며, LINE(주)는 이에 대해 명시적 혹은 묵시적으로 어떠한 보증도 하지 않습니다.

관련 URL 정보를 포함하여 이 문서에서 언급한 특정 소프트웨어 상품이나 제품은 해당 소유자가 속한 현지 및 국내외 관련법을 따르며, 해당 법률을 준수하지 않음으로 인해 발생하는 모든 결과에 대한 책임은 전적으로 사용자 자신에게 있습니다.

LINE(주)는 이 문서의 내용을 예고 없이 변경할 수 있습니다.

문서 정보

문서 개요

이 문서는 LINE Pay 가맹점의 결제를 LINE Pay 와 연동하는 방법을 설명한다.

독자

이 문서의 독자는 LINE Pay 가맹점 서버/사이트 개발자로서, HTTP 통신과 Restful API 를 이해해야 한다.

연락처

LINE Pay 와 관련한 문의 사항이 있으면 다음 연락처로 문의하십시오.

LINE Pay 기술지원(pay_tech@linecorp.com)

문서 버전 및 이력

버전	일자	이력사항
1.0.0	2014.11.10	초안 작성
1.0.1	2014.12.01	자동결제 추가
1.0.2	2015.01.15	인증/매입 프로세스 추가
1.0.3	2015.01.20	에러코드 추가
1.0.4	2015.03.01	OTK 기능 추가, ConfirmURL 브라우저 설정기능 추가 샌드박스 사용시 화면 예제 추가
1.0.5	2015.06.01	결제프로세스 설명추가 oneTimeKey 설명추가 API 변경기록 추가 부록의 라인페이 서버 IP 정보, QR/바코드 테스트방법 추가
1.0.6	2015.07.30	재시도 가능 returnCode 신규 정의

표기 규칙

소스 코드 표기

이 문서에서 소스 코드는 회색 바탕에 검정색 글씨로 표기한다.

```
COPYDATASTRUCT st;  
st.dwData = PURPLE_OUTBOUND_ENDING;  
st.cbData = sizeof(pp);  
st.lpData = &pp;  
::SendMes(GetTargetHwnd(), WM_COPYDATA, (WPARAM)this->m_hWnd, (LPARAM)&st);
```

목차

LINE Pay 기술 연동 가이드	1
개요	11
LINE Pay 란	11
LINE Pay 가맹점 등록 과정	12
LINE Pay 기능	13
일반 결제	13
자동 결제	13
공통 기능	13
LINE Pay 결제 상태	14
LINE Pay 연동 흐름	15
일반 결제	15
자동 결제	21
LINE Pay 연동 API 사용 방법	22
API 변경 이력	23
2015-06-24	23
2015-04-30	23
결제 Reserve	24
결제 reserve API	24
결제 reserve API 명세	24
결제 reserve API 예제	29
결제 Confirm	30
결제 confirm API	30
결제 confirm API 명세	30
결제 confirm API 예제	32
매입	34
매입 API	34
매입 API 예제	36
인증 무효 처리	37

인증 무효 처리 API	37
인증 무효 처리 API 명세	37
인증 무효 처리 API 예제	38
인증 내역	39
인증 내역 조회 API	39
인증 내역 조회 API 명세	39
인증 내역 조회 API 예제	40
결제 내역	42
결제 내역 조회 API	42
결제 내역 조회 API 명세	42
결제 내역 조회 API 예제	43
결제 환불	45
결제 환불 API	45
결제 환불 API 명세	45
결제 환불 API 예제	46
regKey(자동 결제를 위한 키) 사용 가능 여부 확인	47
regKey 사용 가능 여부 API	47
regKey 사용 가능 여부 확인 API 명세	47
regKey 사용 가능 여부 확인 API 예제	48
자동 결제	49
자동 결제 API	49
자동 결제 API 명세	49
자동 결제 API 예제	51
regKey 만료	52
regKey 만료 API	52
regKey 만료 API 명세	52
regKey 만료 API 예제	53

부록 **54**

LINE Pay 오류 코드	54
PaymentUrl 앱 사용 가이드	59
Android 애플리케이션 예제	59
iPhone 애플리케이션 예제	60
confirmUrl 상세 설명 및 예외 상황	61
confirmUrl 상세 설명	61
confirmUrl 예외 상황	61
ConfirmUrl 을 Server-to-Server 로 전달한 경우	62
ConfirmUrl 명세	62
ConfirmUrl 사용 예제	62
Sandbox 사용방식	63
Sandbox PC 결제	63
Sandbox 모바일 결제	64

표 및 그림 목록

표 목록

표 1 결제 reserve Endpoint 명세	24
표 2 결제 reserve API 요청 파라미터	24
표 3 결제 reserve API 응답	28
표 4 결제 confirm API Endpoint 명세	30
표 5 결제 confirm API 요청 파라미터	30
표 6 결제 confirm API 응답	31
표 7 매입 API Endpoint 명세	34
표 8 매입 API 요청 파라미터	34
표 9 매입 API 응답	35
표 10 인증 무효 처리 API Endpoint 명세	37
표 11 인증 무효 처리 API 응답	38
표 12 인증 내역 조회 Endpoint 명세	39
표 13 인증 내역 조회 API 조회 파라미터	39
표 14 인증 내역 조회 API 응답	39
표 15 결제 내역 조회 Endpoint 명세	42
표 16 결제 내역 조회 API Query 파라미터	42
표 17 결제 내역 조회 API 응답	42
표 18 결제 환불 Endpoint 명세	45
표 19 결제 환불 API 파라미터	45
표 20 결제 환불 API 응답	45
표 21 regKey 사용 가능 여부 확인 Endpoint 명세	47
표 22 regKey 사용 가능 여부 확인 API Query 파라미터	47
표 23 regKey 사용 가능 여부 확인 API 응답	48
표 24 자동 결제 Endpoint 명세	49
표 25 자동 결제 API 파라미터	49
표 26 자동 결제 API 응답	50
표 27 regKey 만료 Endpoint 명세	52
표 28 regKey 만료 API 응답	52
표 29 LINE Pay 오류 코드	54
표 30 API 별 오류코드	57
표 31 confirmUrl Endpoint 명세	62

그림 목록

그림 1 LINE Pay 결제 과정 _____	11
그림 2 가맹점 등록 과정 _____	12
그림 3 PC 결제 reserve - 결제 화면 _____	15
그림 4 모바일 결제 reserve - 결제 화면 _____	16
그림 5 PC 결제 화면-결제 완료 _____	17
그림 6 모바일 결제 화면-결제 완료 _____	18
그림 7 confirmUrl을 Server-to-Server로 전달 _____	19
그림 8 OneTimeKey발급과 결제 reserve _____	20
그림 9 자동 결제 _____	21
그림 10 Sandbox PC 결제 진행화면 예제 _____	63
그림 11 Sandbox 모바일 결제 진행화면 예제 _____	64

개요

이 장에서는 LINE Pay 연동의 기본 개념을 설명한다.

LINE Pay 란

LINE Pay 는 LINE 회원들이 LINE Pay 가맹점 사이트에서 사용할 수 있는 결제 시스템이다.

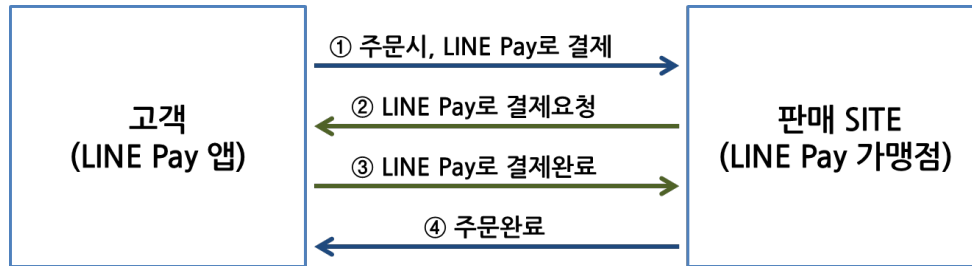


그림 1 LINE Pay 결제 과정

가맹점이 LINE Pay 가맹점으로 가입하면 LINE 을 사용하고 있는 전 세계의 LINE 회원들을 가맹점의 고객으로 유치할 수 있다. 또한 LINE 을 통해 가맹점의 마케팅 채널을 확장하여 매출 향상을 기대할 수 있다.

LINE Pay 를 사용하는 가입자들이 LINE Pay 로 결제하려면, 결제하려는 사이트가 반드시 LINE Pay 가맹점으로 등록되어 있어야 한다.

LINE Pay 가맹점 등록 과정

LINE Pay 가맹점으로 등록하면 연동 키와 테스트 키가 발급된다. 가맹점 등록 과정은 다음과 같다.

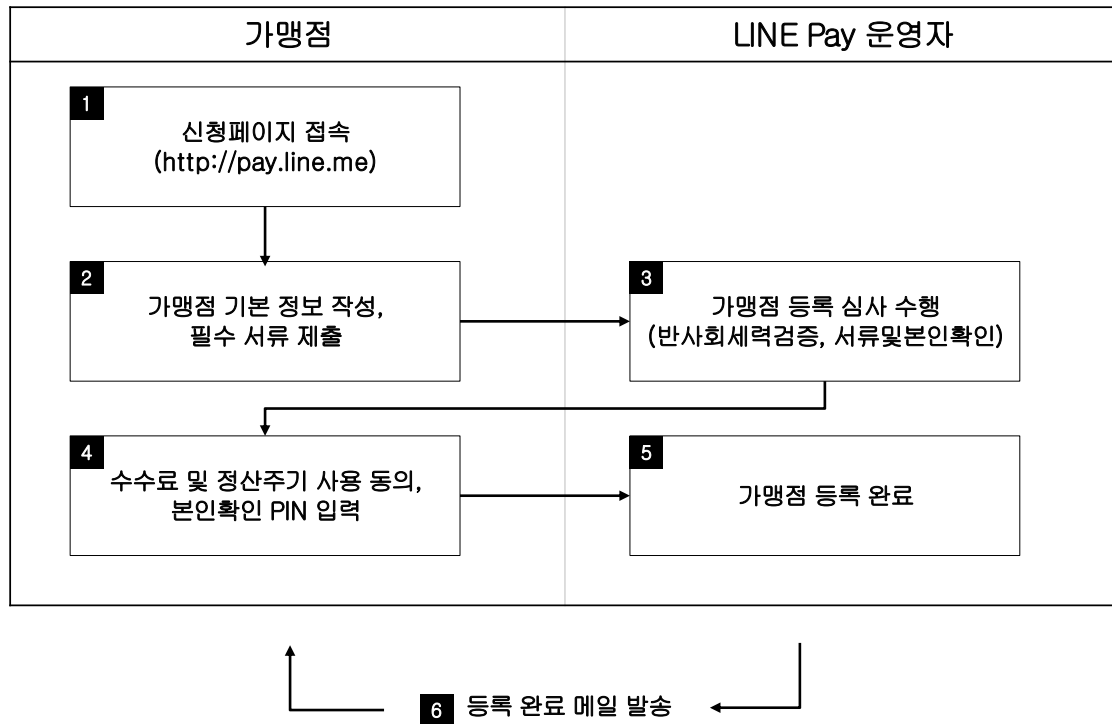


그림 2 가맹점 등록 과정

1. 신청 페이지(http://pay.line.me) 접속
2. 기본 정보 작성 및 필수 서류 제출
3. 가맹점 등록 심사
4. 수수료 및 정산 주기 사용 동의, 본인 확인 PIN 입력
5. 가맹점 등록 완료
6. 등록 완료 메일 발송

LINE Pay 기능

LINE Pay 결제 기능은 아래와 같이 분류된다.

일반 결제

일반적인 결제 기능으로, 결제 요청 시마다 LINE Pay 회원이 LINE Pay에 접근해서 결제 수단을 선택하고, 결제 비밀번호를 확인하여 거래하는 형태이다. 결제 완료의 기준은 매입이며, 가맹점이 인증과 매입을 분리한 경우, "매입 API"를 호출해야 한다. LINE Pay 회원이 인증 시점에 선택한 결제 수단에 따라 잔액 또는 카드 한도의 차감이 발생한다.

1. 가맹점 서버에서 "결제 reserve API" 호출
2. LINE Pay 결제 화면에서 결제 수단 선택 및 결제 비밀번호 확인
3. 가맹점 서버에서 "결제 confirm API" 호출
4. 거래 완료

자동 결제

자동 결제는 가맹점이 결제가 필요한 시점에 LINE Pay 회원의 개입 없이, 가맹점 서버와 LINE Pay 서버 간에 결제하는 기능이다. LINE Pay 회원이 최초 결제를 하고, LINE Pay에서 가맹점에게 나중에 자동 결제를 하기 위한 키 값인 regKey(자동 결제를 위한 키)를 발급한다. 일반 결제와 유사하게 결제 reserve 및 confirm 과정을 거친다. 결제 confirm 시 regKey를 발급받아, 이후에는 LINE app을 통하지 않고 가맹점과 LINE Pay 간의 자동 결제 API 연동만으로 거래를 완료하는 형태이다.

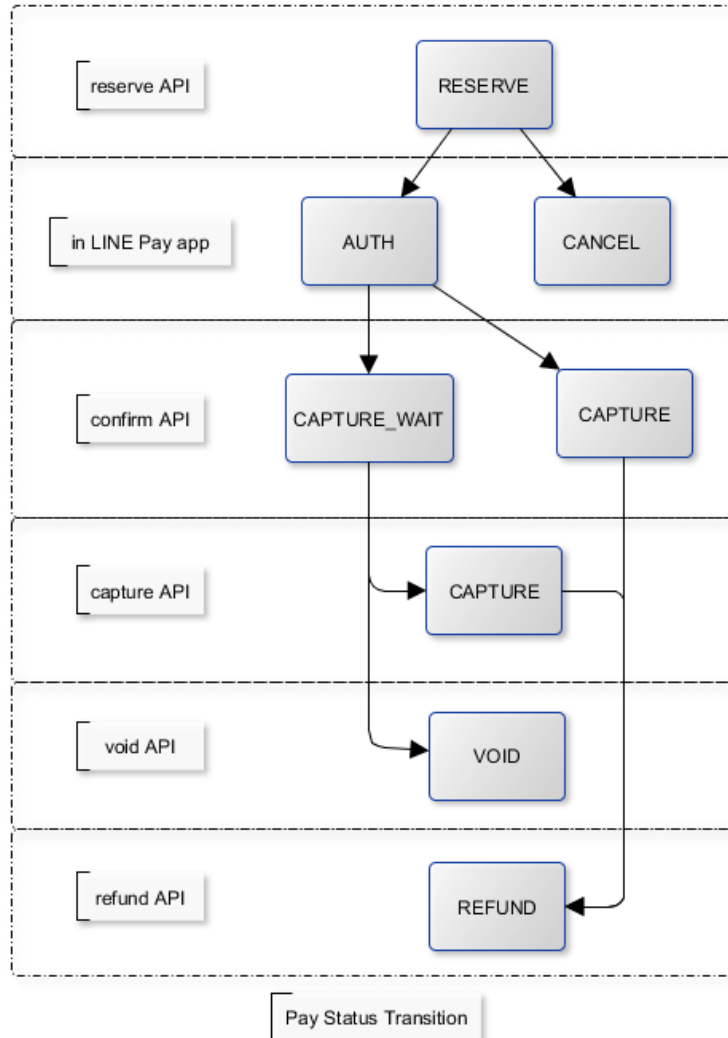
1. 최초 결제 및 regKey(자동 결제를 위한 키) 발급
가맹점 서버에서 결제 reserve API 호출(요청 정보 중 "payType" : "PREAPPROVED")
LINE Pay 결제 화면에서 결제 수단 및 결제 비밀번호 확인 후 returnUrl(결제 reserve API의 파라미터)로 이동
가맹점 서버에서 결제 confirm API 호출 후, 응답 정보 중 regKey 저장
거래 완료
2. 자동 결제
regKey로 자동 결제 API 호출
거래 완료

공통 기능

1. 환불
2. 내역 조회

LINE Pay 결제 상태

LINE pay 의 결제 요청부터 완료까지의 상태를 설명한다.



1. RESERVE : 결제 Reserve API 호출 시, 최초 상태
2. AUTH : LINE Pay app 에서 결제수단 선택과 결제 비밀번호 인증 성공
3. CANCEL : LINE Pay app 에서 결제 취소
4. CAPTURE_WAIT : 매입대기 상태(reserve API 의 “capture” 파라미터가 “false”로 전달된 경우)
5. CAPTURE : 매입
6. VOID : 매입 대기 상태에서 매입 취소
7. REFUND : 매입 후 환불

FAILED : LINE Pay 에서 API 처리 중 실패된 상태로 confirm, capture, void API 에서 발생

LINE Pay 연동 흐름

일반 결제

결제 요청 - 결제 화면

결제 reserve 과정은 다음과 같다. LINE Pay 회원의 가맹점 접속 환경에 따라 연동 흐름이 구분된다.

1. PC 인 경우

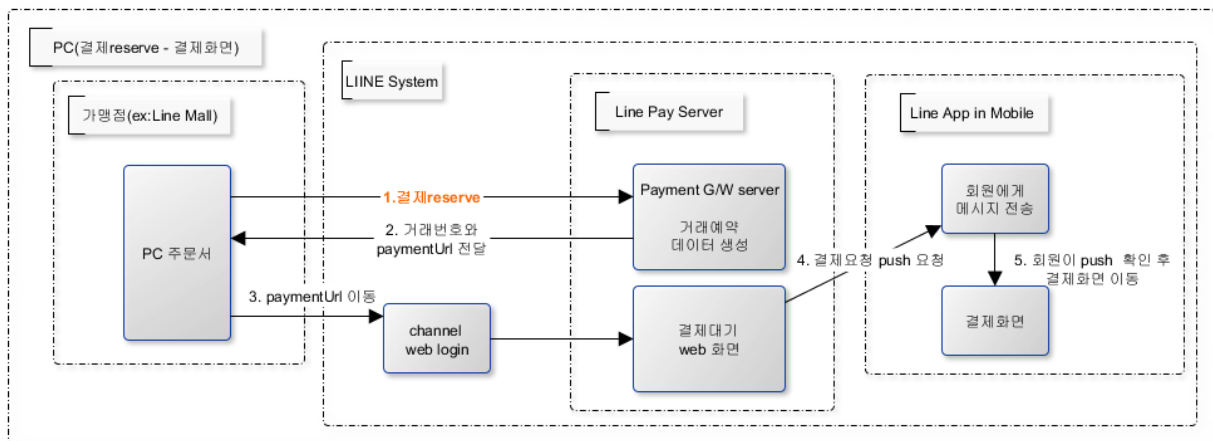


그림 3 PC 결제 reserve - 결제 화면

1. LINE Pay 회원이 가맹점 주문서에서 LINE Pay 결제 수단을 선택한다.
2. 가맹점에서 LINE Pay "결제 reserve API"를 사용해 결제 예약 정보를 생성하고 거래 번호와 "paymentUrl"을 받는다. 거래 번호는 결제 confirm 시 필수 정보로, 가맹점에서 저장하고 있어야 한다.
3. 가맹점은 b 과정에서 전달받은 payment url(LINE pay 제공)을 호출한다.
4. LINE 회원인지 확인하기 위해 channel web login¹을 거친다.
5. 로그인 성공 후, LINE Pay 에서 LINE 으로 결제 요청 PUSH 를 요청한다.
6. LINE 회원은 LINE app 에서 결제 요청 알림을 확인하고, 결제 화면으로 이동한다.

¹ LINE 내 서비스를 이용하기에 앞서 사용자가 LINE 회원인지 확인하기 위해 거치는 과정이다.

가맹점이 LINE 회원의 정보에 접근할 수 있고 결제 reserve 시 mid 를 전달할 수 있다면 channel web login 과정을 거치지 않는다.

2. 모바일인 경우

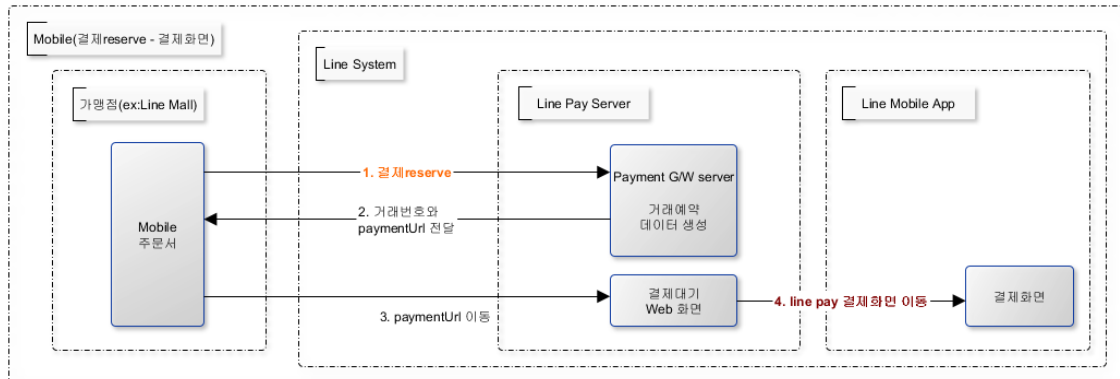


그림 4 모바일 결제 reserve - 결제 화면

1. LINE Pay 회원이 가맹점 주문서에서 LINE Pay 결제 수단을 선택한다.
2. 가맹점에서 LINE Pay "결제 reserve API"를 사용해 결제 예약 정보를 생성하고 LINE Pay 에서 발급한 거래 번호와 "paymentUrl"을 받는다. 거래 번호는 결제 confirm 시 필수 정보로, 가맹점에서 저장하고 있어야 한다.
3. 가맹점은 b 과정에서 전달받은 payment url(LINE pay 제공)을 호출한다.
4. paymentUrl을 사용하여 LINE 앱 내의 결제 화면으로 이동한다. payment URL은 가맹점 연동 환경에 따라, web 과 app 으로 분리되어 있다.
 - web: LINE Pay 에서 제공하는 web 화면. LINE Pay 결제 화면으로 이동하기 전에 LINE 설치 여부를 확인한다.
 - app: LINE Pay 결제 화면으로 직접 이동하는 app scheme url 이다. device 의 LINE 설치 여부, LINE Pay 사용 가능 버전(Android 인 경우만 해당)을 검사한 후 LINE Pay 결제 화면으로 이동하는 예제 등은 이 문서의 부록에 있는 [PaymentUrl 앱 사용 가이드](#) 를 참조한다.

결제 화면 - 결제 완료

LINE Pay 회원은 LINE Pay 결제 화면에서 결제 수단(신용카드 또는 잔고)을 선택하여 결제 reserve 정보를 결제 가능한 상태로 만든다. 다음 결제 reserve 시 가맹점으로부터 전달받은 "confirmUrl"(LINE Pay에서 transactionId 파라미터를 추가한)로 이동한다. 이동 후 가맹점 서버에서는 "결제 confirm API"를 호출하여 결제를 완료한다.

1. PC 인 경우

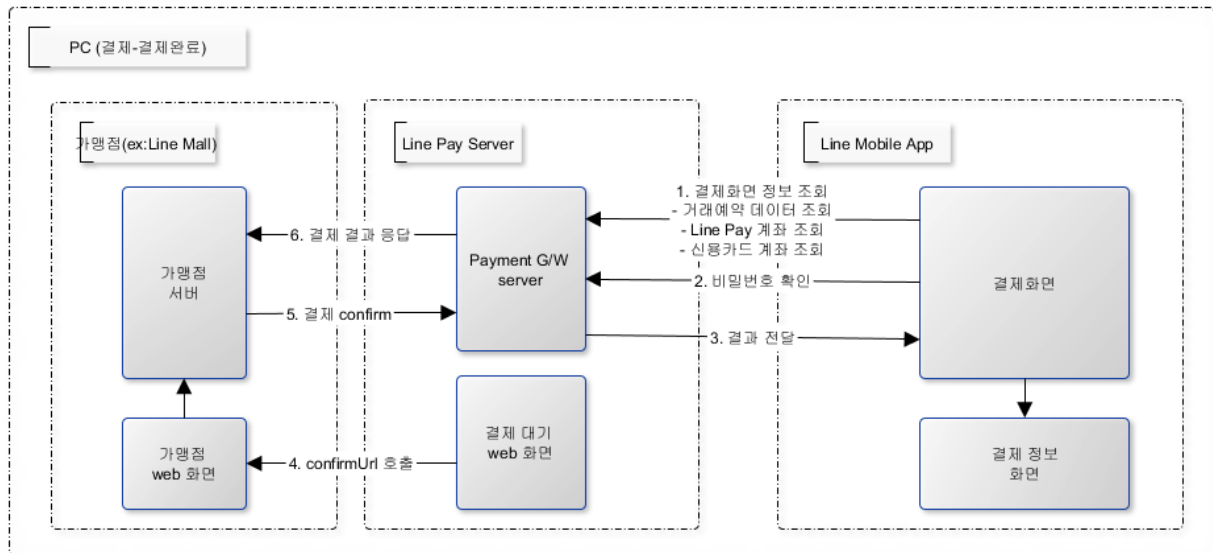


그림 5 PC 결제 화면-결제 완료

1. LINE Pay 회원은 LINE Pay 결제 화면에서 결제 수단을 선택하고 비밀번호를 입력한다.
2. LINE Pay에서는 결제를 위한 결제 수단 정보를 저장하고, 결제 상태 정보를 결제 가능한 상태로 변경한다.
3. LINE Pay 회원은 LINE Pay에서 결제 정보 화면을 확인한다.
4. PC의 결제 대기 화면은 주기적으로 결제 가능한 상태를 확인하고 결제 가능 상태가 되면, 결제 reserve 시 가맹점이 전달한 "confirmUrl"로 이동한다.
5. 가맹점은 LINE Pay로 결제 confirm API를 호출하여 결제를 완료한다.

2. 모바일인 경우

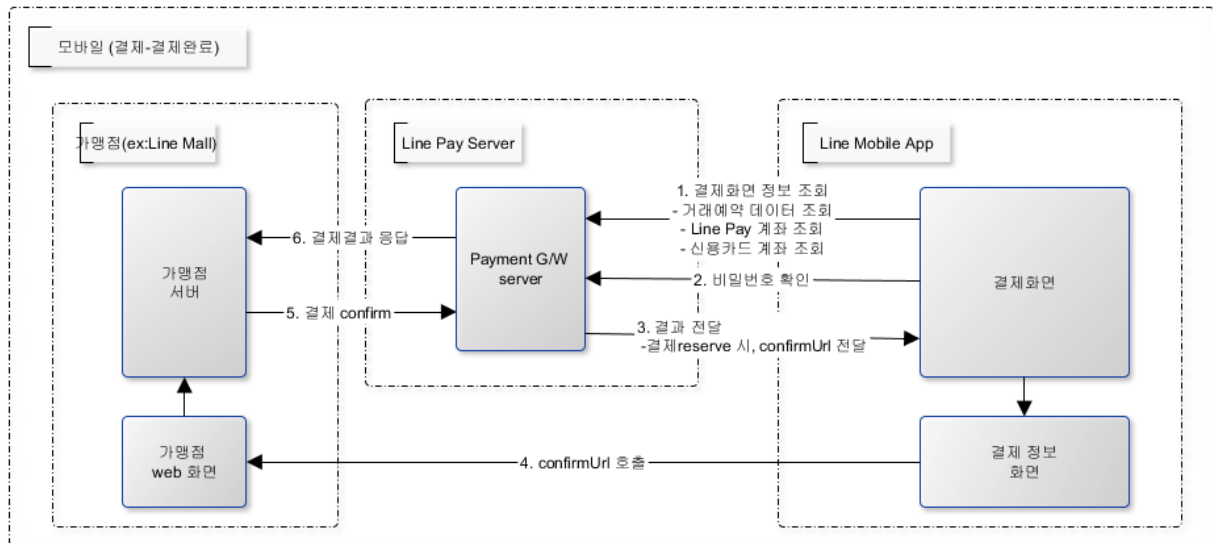


그림 6 모바일 결제 화면-결제 완료

1. LINE Pay 회원은 LINE app 결제 화면에서 결제 수단을 선택하고 비밀번호를 입력한다.
2. LINE Pay에서는 결제를 위한 결제 수단 정보를 저장하고, 결제 상태 정보를 결제 가능한 상태로 변경한다.
3. LINE Pay 회원은 LINE app에서 결제 정보 화면을 확인하고, 화면 하단의 확인 버튼을 눌러 결제 reserve 시 전달받은 "confirmUrl"로 이동한다.
4. 가맹점은 LINE Pay로 결제 confirm API를 호출하여, 결제를 완료한다.

인증과 매입이 분리된 경우

- a. 위와 동일한 과정을 거치지만, 인증과 매입을 분리하려면 결제 reserve API 호출 시, "capture" 파라미터를 "false"로 요청한다.
- b. 결제 confirm API 호출 시 결제는 매입 대기 상태로 저장된다.
- c. 가맹점은 매입 시점에
 - 매입을 원하는 경우: 매입 API를 호출하여 결제를 완료한다.
 - 매입을 원하지 않는 경우: 인증 무효 처리 API를 호출하여 결제를 취소한다.

confirmUrl 을 Server-to-Server 로 전달하는 경우

confirmUrl 전달시 가맹점 서버와 LINE Pay 서버 간의 통신만으로 결제를 진행할 수 있다.

1. 연동 흐름

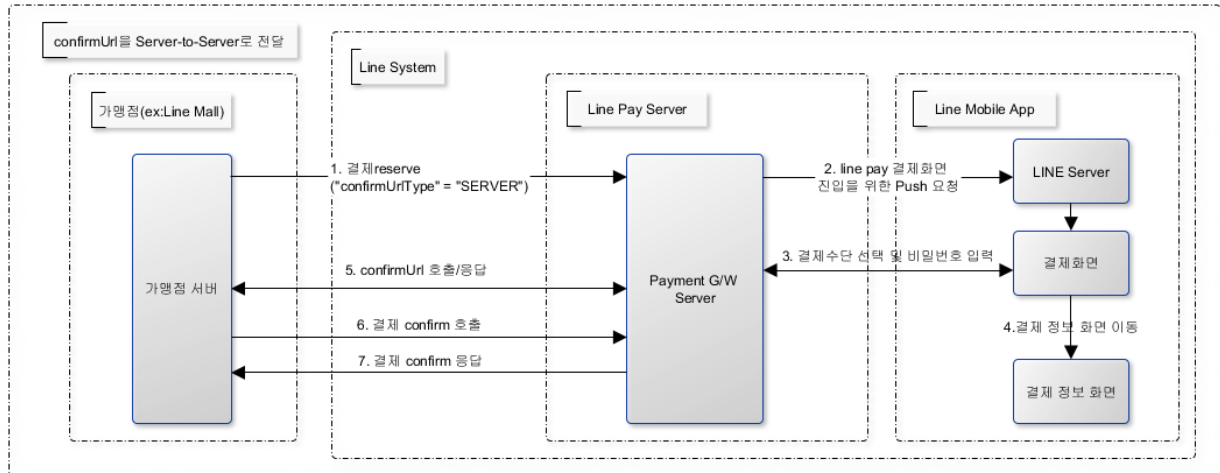


그림 7 confirmUrl 을 Server-to-Server 로 전달

1. 가맹점은 결제 reserve 시, confirmUrlType : "SERVER"로 설정한다.
2. LINE Pay 회원은 LINE Pay 결제 화면 진입 후 결제 수단을 선택하고 비밀번호를 입력한다.
3. LINE Pay 서버는 결제 정보를 저장하고, 결제 reserve 시 가맹점이 전달한 confirmUrl 을 호출한다.
[\[부록\] "ConfirmUrl 을 Server-to-Server 로 전달한 경우"](#)를 참고한다.
4. 가맹점 서버는 결제 confirm API 를 호출하여 결제를 완료한다. confirmUrl 에 대한 응답이 정상적으로 전달 되지 않은 경우에는 confirm API 를 호출할 수 없다.

LINE Pay app 에서 제공하는 one time key 를 이용하여 결제를 진행하는 경우

LINE Pay app 에서는 offline 결제를 지원하기 위해서 oneTimeKey 를 발급한다. oneTimeKey 는 LINE Pay 회원의 정보를 대신하는 정보로 Barcode 또는 QR Code 형태로 확인이 가능하다. 가맹점은 코드를 읽어 oneTimeKey 를 획득하여, 결제를 진행할 수 있다.

1. 전제 조건

oneTimeKey 유효시간 : 5 분(사용자가 oneTimeKey 코드 페이지를 접근한 시간부터)

oneTimeKey 는 결제 reserve 와 동시에 삭제된다.

2. 연동 흐름

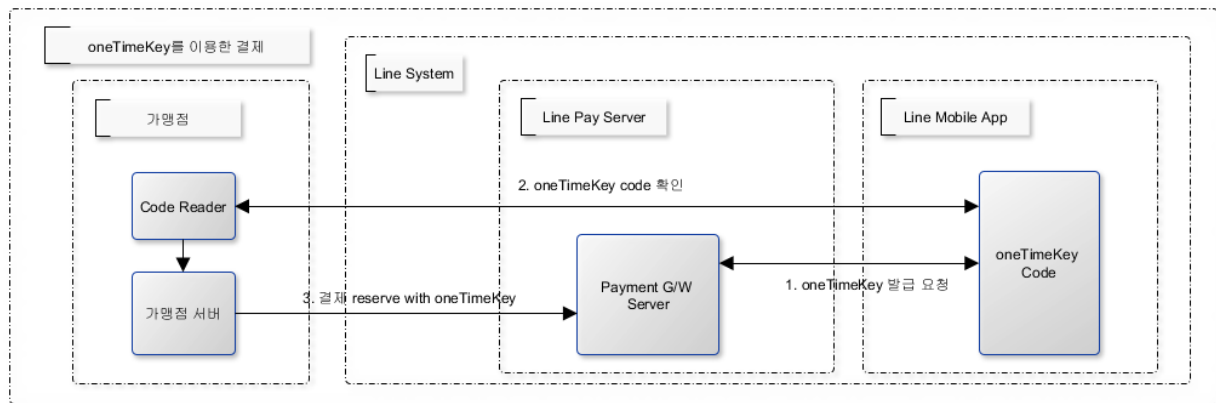


그림 8 OneTimeKey 발급과 결제 reserve

1. 가맹점은 LINE Pay 회원이 보여주는 oneTimeKey 코드(barcode 또는 QR Code)를 읽는다.
2. 가맹점은 획득한 oneTimeKey 정보를 이용하여, 결제 reserve 한다.
3. 결제 reserve 가 되면 LINE Pay 회원은 결제 요청을 LINE 을 통해 확인한다.
4. LINE Pay 회원은 LINE Pay 결제 화면 진입 후 결제 수단을 선택하고 비밀번호를 입력한다.
5. LINE Pay 회원은 결제 정보 화면을 확인한 후, 결제 reserve 시, confirmUrlType 에 따라 흐름이 달라진다
 - CLIENT(기본값) : confirmUrl 이 사용자의 browser 에서 열리며, 이동 후 결제 confirm API 를 호출하여 결제를 완료한다.
 - SERVER : [\[부록\] "ConfirmUrl 을 Server-to-Server 로 전달한 경우"](#)를 참고한다.

가맹점 서버는 결제 confirm API 를 호출하여 결제를 완료한다. confirmUrl 에 대한 응답이 정상적으로 전달되지 않은 경우에는 confirm API 를 호출할 수 없다.

자동 결제

결제 요청 - 결제 화면

전체적인 흐름은 일반 결제와 동일하며, 요청 정보 중 "payType"을 "PREAPPROVED"로 전달하는 부분만 다르다.

결제 화면 - 결제 완료 및 regKey(자동 결제를 위한 키) 발급

일반 결제와 흐름은 동일하지만 confirm 응답 정보 중 regKey가 추가로 전달된다. 나중에 가맹점은 자동 결제를 하기 위해 regKey를 저장해야 한다.

자동 결제

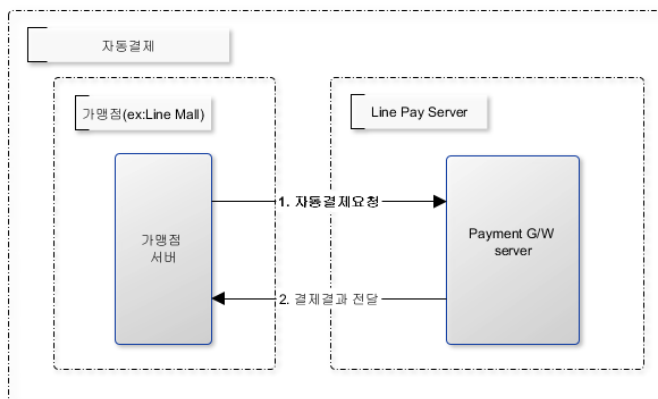


그림 9 자동 결제

결제 confirm 시, 응답 결과인 regKey로 가맹점 서버와 LINE Pay 서버 간에 "자동 결제 API"를 호출하여, 거래를 발생시킨다. 이때 LINE Pay 회원의 개입은 없으나, 결제가 완료되면 LINE Pay 회원에게 결제 완료 알림을 전달한다.

regKey(자동 결제를 위한 키) 만료

regKey를 더 이상 사용하지 않으면 regKey를 만료시킬 수 있다. "regKey 만료"를 참고한다.

LINE Pay 연동 API 사용 방법

이 장에서는 LINE Pay 를 연동하는 방법을 상세히 설명한다. 연동에 앞서 가맹점 인증 정보는 심사가 완료된 이후 가맹점 센터2(<http://pay.line.me>)를 통해 확인할 수 있다.

LINE Pay 연동 시 필요한 인증 정보는 아래와 같다.

- channel id
- channel secret key

모두 헤더를 통해 전달되며, 각 API 명세에서 확인할 수 있다.

연동 중 문의 사항이 있거나 내부 오류가 발생한 경우 기술 연동 지원(pay_tech@linecorp.com)으로 문의한다.

연동 환경

- Beta : LINE office network 에서만 테스트 가능한 환경
- Sandbox : 연동 테스트를 위한 환경. Pay app 상에서의 결제를 web simulation page 에서 확인.
- 부록 ([Sandbox 사용방식](#)) 참고
- Real : 실 서비스 환경

² 가맹점 센터는 LINE Pay 가맹점 가입을 완료한 가맹점이 거래 내역과 입금 내역을 확인하거나, 기본적인 가맹점 정보를 수정하고 1:1 문의 등을 요청할 수 있는 채널이다. 가맹점의 기본 정보 내에서 현재 가맹점의 서비스 제공 상태를 확인할 수 있다.

API 변경 이력

주의

- API 기능이 추가/변경되면 API version은 올라간다.
 - 하위 버전 호환성을 보장 하지만 가장 최신 버전 사용할 것을 권장한다.
-

2015-06-24

1. 재시도 가능 returnCode 신규 정의
 - a. API : [결제 환불](#), [인증 무효 처리](#), [매입](#)
 - b. 변경 내용 : LINE Pay 내에서 외부 시스템 연동간 오류로 인한 returnCode 추가(API 요청 재시도 가능)

2015-04-30

1. API Version v2 로 upgrade
2. Promotion 관련 value 추가
 - a. API : [결제 confirm](#), [매입](#), [인증 내역](#), [결제 내역](#)
 - b. 변경 내용 : API 응답 정보 중 "payInfo[].method"의 value 추가
 - DISCOUNT : LINE Pay 내에서 할인이 되는 경우 전달되는 value
3. Token 기반 결제 연동 지원
 - a. API : [결제 reserve](#)
 - b. 변경 내용 : LINE Pay scanner(LINE 5.1 부터 지원)를 통해 결제를 할 수 있도록 하기 위해 token 추가
 - paymentAccessToken : API 응답 정보로 추가되었으며, LINE Pay scanner 의 "input key code"에 입력했을 때, 바로 LINE Pay 결제 화면 진입시키기 위한 정보
4. 자동결제 인증/매입 분리 지원
 - a. API : [자동결제](#)
 - b. 변경 내용 : 자동 결제 시에도 일반 결제와 동일하게 인증/매입을 분리하여 결제를 할 수 있게 파라미터 추가
 - capture : 매입 여부
 - true(기본값): 결제 confirm API 호출 시, 인증과 매입을 한 번에 처리한다
 - false: 결제 confirm API 호출 시, 인증까지 처리하고 "매입 API"를 호출해야 거래가 완료된다
5. regKey 인증 지원
 - a. API : [regKey 사용 가능 여부 API](#)
 - b. 변경 내용 : regKey 가 유효한지 확인하는 과정에서 신용카드 연동사까지 최소금액을 인증하여, 자동결제를 할 수 있는지 확인하는 파라미터 추가
 - c. creditCardAuth : 신용카드 최소금액 인증 여부
 - true : 신용카드 최소 금액을 인증한다.
 - false(기본값) : LINE Pay 내부의 데이터 유효성 검증.

결제 Reserve

결제를 위한 정보를 LINE Pay 에 예약한다.

결제 reserve API

LINE Pay 결제를 진행하기 전에 정상 가맹점인지 판단하고, 결제를 위한 정보를 예약한다. 가맹점은 결제 예약이 성공하면 결제 완료 또는 환불 시까지 사용되는 키 값인 거래 번호를 발급받는다.

결제 reserve API 명세

표 1 결제 reserve Endpoint 명세

항목	설명
Method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/request
Sandbox	https://sandbox-api-pay.line.me/v2/payments/request
Real	https://api-pay.line.me/v2/payments/request

표 2 결제 reserve API 요청 파라미터

항목	데이터 타입	필수 여부	설명
productName	String (4000byte)	Y	상품명(charset: UTF-8)
productImageUrl	String (500byte)	N	상품 이미지 URL 결제 화면에 노출될 이미지 URL 크기: 84 x 84(결제 화면에만 노출되는 이미지. 가이드대로 노출시킬 것을 권장)
amount	number ³	Y	결제 금액
currency	String (3byte)	Y	결제 통화(ISO 4217) 지원 가능한 통화는 아래와 같다. USD

³ 소수점이 가능한 숫자형 데이터 타입

항목	데이터 타입	필수 여부	설명
			- JPY - TWD - THB
mid	String (50byte)	N	LINE member ID 결제하려는 LINE 회원 mid
oneTimeKey	String (12byte)	N	One Time Key LINE Pay app 에서 제공하는 QR/BarCode 정보를 읽은 결과로 LINE Pay 회원의 mid 를 대신한다. 유효시간은 5 분이며, reserve 와 동시에 삭제된다. LINE Pay app 의 QR/BarCode 는 LINE 5.1 버전부터 지원된다.
confirmUrl	String (500byte)	Y	LINE Pay 에서 결제 수단 선택 및 비밀번호 확인 후 이동하는 가맹점 URL - 결제를 완료하기 위해서는 confirmUrl 로 이동 후, confirm API 가 호출되어야 한다. - url 이동시, transactionId 정보가 파라미터로 추가 된다. "confirmUrlType"을 "SERVER"로 전달하는 경우, 꼭 "https" protocol 을 사용해야 하며, real 환경에서는 "신뢰할 수 있는 인증서"를 사용해야 한다. 단, sandbox 환경에서는 연동 단계를 위한 환경임을 고려하여 인증서 오류를 무시한다.

참고

[confirmUrl상세 설명 및 예외 상황](#)

confirmUrlType	String	N	confirmUrl 유형 LINE Pay 에서 결제 수단 선택 및 비밀번호 확인 후 이동될 URL 의 유형 - CLIENT(기본값): 사용자 기반의 URL - SERVER: 서버 기반의 URL. 사용자는 LINE Pay 에서 결제 정보 화면을 확인하는 것으로 시나리오가 종료되며, LINE Pay 에서 가맹점 서버로 결제가 가능한 상태가 되었음을 알린다.
checkConfirmUrlBrowser	Boolean	N	confirmUrl 로 이동시, 사용되는 browser 의 체크 여부 - true: 결제를 요청한 browser 와 confirmUrl 으로 이동되는 browser 가 다른 경우, LINE Pay 에서 원래 browser 로 돌아가라는 안내 페이지를 제공한다. - false(기본값): 이동되는 browser 의 확인 없이 confirmUrl 로 바로 이동된다.

항목	데이터 타입	필수 여부	설명
참고			
confirmUrl상세 설명 및 예외 상황			
cancelUrl	String (500byte)	N	결제 취소 page URL - LINE 앱 결제 화면에서 LINE Pay 회원이 결제를 취소하면 이동하는 URL(가맹점이 모바일에서 접근할 때 취소 시 가맹점 앱 또는 웹으로 이동하기 위한 URL) - 가맹점이 전달한 URL 그대로 이동 - LINE Pay 에서 추가로 전달하는 파라미터 없음.
packageName	String (4000byte)	N	Android 에서 앱 간 전환 시, phishing 방지용 정보
orderId	String (100byte)	Y	가맹점에서 해당 결제 예약 건에 대한 주문 번호 가맹점에서 관리되는 유일한 번호이다.
deliveryPlacePhone	String (100byte)	N	수취인 연락처(for Risk Management)
payType	String (12byte)	N	결제 유형 - NORMAL(기본값): 일반 결제 - PREAPPROVED: 자동 결제
langCd	String	N	결제 대기 화면(paymentUrl)의 언어 코드. 총 6 가지의 언어를 지원한다. - ja: 일본어 - ko: 한국어 - en: 영어 - zh-Hans: 중국어(간체) - zh-Hant: 중국어(번체) - th: 태국어 - 필수로 전달하는 값은 아니다. 이 값이 전달되지 않으면 accept-language header 값을 참조하여 다국어를 지원하도록 한다. - 지원하지 않는 langCd 가 전달되면 영어(en)를 기본으로 사용한다. - BCP-47 형태: http://en.wikipedia.org/wiki/IETF_language_tag
capture	Boolean	N	매입 여부 - true(기본값): 결제 confirm API 호출 시, 인증과 매입을 한 번에 처리한다. - false: 결제 confirm API 호출 시, 인증까지

Error! Style not defined.

항목	데이터 타입	필수 여부	설명
			처리하고 "매입 API"를 호출해야 거래가 완료된다

표 3 결제 reserve API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다. 1106: 헤더 정보 오류 1124: 금액 정보 오류입니다(scale). 1133: 유효하지 않은 oneTimeKey 입니다. 1145: 결제 진행 중 입니다. 1172: 이미 동일 주문번호로 거래된 내역이 존재합니다. 1178: 지원하지 않는 통화입니다. 1194: 자동 결제를 사용할 수 없는 가맹점입니다 9000: 내부 오류
returnMessage	String	결과 메시지 또는 실패 사유. 예를 들면 다음과 같은 경우다. - 결제 불가 가맹점 - 가맹점 인증 정보 오류
info.transactionId	Number	거래 번호(19 자리)
info.paymentUrl.web	String	결제 예약 이후 이동할 웹 URL - 결제 예약이 웹 환경에서 이루어진 경우 사용 - LINE Pay 결제 대기 화면으로 이동하기 위한 URL - 별도의 파라미터 없이 전달된 URL 그대로 이동 - Desktop 에서 팝업 open 시, 사이즈 - Width: 700px, Height : 546px
info.paymentUrl.app	String.	결제 화면 이동 앱 URL - 앱에서 결제 예약이 이루어진 경우 사용 - 가맹점 앱에서 LINE Pay 앱으로 이동하기 위한 URL
부록 PaymentUrl 앱 사용 가이드		
info.paymentAccessToken	String	LINE Pay app 에서 Scanner 를 이용하는 대신 코드를 입력하는 경우 사용하는 코드 값. (12 자리) (LINE Pay app 의 Scanner 는 LINE 5.1 버전부터 지원)

결제 reserve API 예제

결제 reserve API 요청 예제

```
{
  "productName" : "test product",
  "productImageUrl" : "http://testst.com",
  "amount" : 10,
  "currency" : "USD",
  "mid" : "os89dufgoiw8yer9021384rdfeq",
  "orderId" : "20140101123456789",
  "confirmUrl" : "naversearchapp://inappbrowser?url=
http%3A%2F%2FtestMall.com%2FcheckResult.nhn%3ForderId%3D20140101123456789",
  "cancelUrl" : "naversearchapp://inappbrowser?url=
http%3A%2F%2FtestMall.com%2ForderSheet.nhn%3ForderId%3D20140101123456789",
  "capture" : "true",
  "confirmUrlType" : "CLIENT"
}
```

결제 reserve API 응답 예제

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "transactionId" : 123123123123,
    "paymentUrl" : {
      "web" : "http://web-pay.line.me/web/wait?transactionReserveId=blahblah",
      "app" : "line://pay/payment/blahblah"
    },
    "paymentAccessToken" : "187568751124"
  }
}
```

결제 Confirm

결제 confirm API

가맹점이 결제를 최종적으로 완료시키기 위한 API이다. 가맹점에서 결제 confirm API를 호출해야 실제 결제가 완료된다. 단, 결제 reserve 시 "capture" 파라미터가 "false"로 전달되면 confirm API 호출 후 결제는 매입대기 상태가 되며, "매입 API"를 호출해야 결제가 최종 완료된다.

결제 confirm API 명세

표 4 결제 confirm API Endpoint 명세

항목	설명
Method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connect : 5 초 read : 40 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/{ transactionId ⁴ }/confirm
Sandbox	https://sandbox-api-pay.line.me/v2/payments/{ transactionId }/confirm
Real	https://api-pay.line.me/v2/payments/{ transactionId }/confirm

표 5 결제 confirm API 요청 파라미터

항목	데이터 타입	필수 여부	설명
amount	number	Y	결제 금액
currency	String (3byte)	Y	결제 통화(ISO 4217) 지원 가능한 통화는 아래와 같다. • USD • JPY • TWD • THB

⁴ 결제 reserve API 연동 시 응답 정보로 전달받은 transactionId.

표 6 결제 confirm API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1101: 구매 회원이 거래 불가 상태입니다. 1102: 구매 회원이 거래 불가 상태입니. 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1110: 사용할 수 없는 신용카드입니다. 1124: 금액 정보 오류 1141: 계좌 상태 오류 1142: 잔액부족 1150: 거래 내역이 존재하지 않습니다. 1152: 이미 거래되었습니다. 1153: 결제 요청 금액과 결제하려는 금액이 다릅니다. 1159: 결제요청 정보가 없습니다. 1169: LINE Pay 에서 결제 수단 선택과 암호 인증을 해야 합니다. 1170: 회원 계좌의 잔액이 변동되었습니다. 1172: 이미 동일 주문번호로 거래된 내역이 존재합니다. 1180: 결제 유효시간이 지났습니다. 1198: API 호출이 중복 요청되었습니다. 1199: 내부 요청 오류 1280: 신용카드 결제 중 일시적 오류 1281: 신용카드 결제 오류 1282: 신용카드 승인 오류 1283: 부정 사용이 의심되어 결제가 거절되었습니다. 1284: 신용카드 결제가 일시적으로 중단되었습니다. 1285: 신용카드 결제 정보 누락 1286: 신용카드 결제 정보가 잘못되었습니다. 1287: 신용카드 유효 기간이 초과되었습니다. 1288: 신용카드 결제 계좌의 잔고가 부족합니다 1289: 신용카드 한도 초과 1290: 신용카드 건당 결제 한도 초과 1291: 도난 신고된 카드입니다. 1292: 사용이 정지된 카드입니다. 1293: CVN 입력 오류 1294: 블랙리스트에 등록된 카드입니다. 1295: 신용카드 번호가 잘못되었습니다. 1296: 처리할 수 없는 금액입니다. 1298: 카드 사용이 거절되었습니다. 9000: 내부 오류
returnMessage	String	결과 메시지 또는 실패 사유. 예를 들면 다음과 같은 경우다. - 결제 불가 가맹점 - 가맹점 인증 정보 오류
info.orderId	String	결제 예약 시 가맹점에서 전달한 주문 번호
info.transactionId	number	결제 예약 결과로 전달받은 거래 번호(19 자리)

항목	데이터 타입	설명
info.authorizationExpirationDate	String	opt-인증 만료 일시(ISO 8601) 인증(capture=false)만 한 결제 건인 경우 전달
info.payInfo[].method	String	결제 시 사용된 결제 수단 (신용카드: CREDIT_CARD, 잔고: BALANCE, 할인: DISCOUNT)
info.payInfo[].amount	number	결제 금액
info.payInfo[].creditCardNickname	String	opt-자동 결제 시의 신용카드 별명 - LINE Pay 에서 관리하는 신용카드 이름. 등록 시에 설정한 이름이다. - LINE Pay 회원이 별명을 설정하지 않은 경우 빈 문자열이 전달된다. - LINE Pay 에서 회원의 요청으로 변경 가능하며, 변경 내용은 가맹점에 공유되지 않는다.
info.payInfo[].creditCardBrand	String	opt-자동 결제 시의 신용카드 브랜드 - VISA - MASTER - AMEX - DINERS - JCB
info.regKey	String	opt-자동 결제를 위한 키(15 자리)

결제 confirm API 예제

결제 confirm API 요청 예제

```
POST https://sandbox-api-pay.line.me/v2/payments/sdhqiwouehrafdasrqoi123as/confirm
HTTP/1.1
Content-Type: application/json
{"amount":1000,"currency":"JPY" }
```

결제 confirm API 응답 예제

```
// 결제유형("payType")이 "NORMAL" 인 경우
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "orderId" : "order_210124213",
    "transactionId" : 20140101123123123,
    "payInfo" : [
      {"method" : "BALANCE", "amount" : 10},
      {"method" : "DISCOUNT", "amount" : 10}
    ]
  }
}

// 결제유형("payType")이 "PREAPPROVED"인 경우
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
```

Error! Style not defined.

```
"orderId" : "order_210124213",
"transactionId" : 20140101123123123,
"payInfo" : [
{"method" : "CREDIT_CARD", "amount" : 10, "creditCardNickname" : "test",
"creditCardBrand" : "VISA"}
],
"regKey" : "RKf930b19XTf1TX"
}
```

매입

결제 reserve API 호출 시 "capture" 파라미터를 "false"로 전달하면, 매입 과정을 거쳐야 결제가 완료된다.

매입 API

"결제 confirm API"를 호출해서 인증만 한 경우, 가맹점이 매입대기 상태의 결제를 최종적으로 완료시키기 위한 API 이다.

표 7 매입 API Endpoint 명세

항목	설명
Method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connect : 5 초 read : 60 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/authorizations/{transactionId ⁵ }/capture
Sandbox	https://sandbox-api-pay.line.me/v2/payments/authorizations/{transactionId}/capture
Real	https://api-pay.line.me/v2/payments/authorizations/{transactionId}/capture

표 8 매입 API 요청 파라미터

항목	데이터 타입	필수 여부	설명
amount	number	Y	결제 금액
currency	String (3byte)	Y	결제 통화(ISO 4217) 지원 가능한 통화는 아래와 같다. • USD • JPY • TWD • THB

⁵ 결제 reserve API 연동 시 응답 정보로 전달받은 transactionId.

표 9 매입 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1150: 거래 내역이 존재하지 않습니다. 1155: 잘못된 거래번호입니다. 1170: 회원 계좌의 잔액이 변동되었습니다. 1172: 이미 동일 주문번호로 거래된 내역이 존재합니다. 1179: 처리할 수 없는 상태입니다. 1183: 금액 오류 1184: 금액 오류 1198: API 호출이 중복 요청되었거나, 내부 자동 재인증이 되는 도중 매입 API 호출이 되었습니다. (수분 후 재시도) 1199: 내부 요청 오류 1280: 신용카드 결제 중 일시적 오류 1281: 신용카드 결제 오류 1282: 신용카드 승인 오류 1283: 부정 사용이 의심되어 결제가 거절되었습니다. 1284: 신용카드 결제가 일시적으로 중단되었습니다. 1285: 신용카드 결제 정보 누락 1286: 신용카드 결제 정보가 잘못되었습니다. 1287: 신용카드 유효 기간이 초과되었습니다. 1288: 신용카드 결제 계좌의 잔고가 부족합니다 1289: 신용카드 한도 초과 1290: 신용카드 건당 결제 한도 초과 1291: 도난 신고된 카드입니다. 1292: 사용이 정지된 카드입니다. 1293: CVN 입력 오류 1294: 블랙리스트에 등록된 카드입니다. 1295: 신용카드 번호가 잘못되었습니다. 1296: 처리할 수 없는 금액입니다. 1298: 카드 사용이 거절되었습니다. 1900: 일시적 오류입니다. 잠시 후 다시 시도하세요. 1901: 일시적 오류입니다. 잠시 후 다시 시도하세요. 1999: 이전 요청 정보와 다릅니다.(요청 재시도시) 9000: 내부 오류 # 오류코드 1900, 1901, 1999 발생시 재시도 가능하다. # 오류 코드 1199, 1280~1298 가 발생한 경우, 거래는 자동 취소된다.
returnMessage	String	결과 메시지 또는 실패 사유. 예를 들면 다음과 같은 경우다. - 결제 불가 가맹점 - 가맹점 인증 정보 오류
info.orderId	String	결제 예약 시 가맹점에서 전달한 주문 번호
info.transactionId	number	결제 예약 결과로 전달받은 거래 번호(19 자리)
info.payInfo[].method	String	결제 시 사용된 결제 수단

항목	데이터 타입	설명
		(신용카드: CREDIT_CARD, 잔고: BALANCE, 할인: DISCOUNT)
info.payInfo[].amount	number	결제 금액

매입 API 예제

매입 API 요청 예제

```
POST https://sandbox-api-pay.line.me
/v2/payments/authorizations/sdhqiwouehrafdasrgoi123as/capture HTTP/1.1
Content-Type: application/json
{"amount":1000,"currency":"JPY"}
```

매입 API 응답 예제

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "orderId" : "order_210124213",
    "transactionId" : 20140101123123123,
    "payInfo" : [
      {"method" : "BALANCE", "amount" : 10},
      {"method" : "DISCOUNT", "amount" : 10}
    ]
  }
}
```

인증 무효 처리

결제 상태가 인증 상태인 결제 데이터를 무효 처리한다.

인증 무효 처리 API

결제 confirm API 호출 시 인증 상태까지 진행된 경우 인증을 무효로 처리하는 API 이다. 인증 상태의 결제만 무효로 처리할 수 있고, 매입이 완료된 결제는 "결제 환불 API"를 사용해 환불해야 한다.

인증 무효 처리 API 명세

표 10 인증 무효 처리 API Endpoint 명세

항목	설명
Method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/authorizations/{transactionId ⁶ }/void
Sandbox	https://sandbox-api-pay.line.me/v2/payments/authorizations/{transactionId}/void
Real	https://api-pay.line.me/v2/payments/authorizations/{transactionId}/void

⁶ 결제 reserve API 연동 시 응답 정보로 전달받은 transactionId.

표 11 인증 무효 처리 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1101: 구매 회원이 거래 불가 상태입니다. 1102: 구매 회원이 거래 불가 상태입니다. 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1150: 거래 내역이 존재하지 않습니다. 1155: 잘못된 거래번호입니다. 1165: 이미 무효화된 거래입니다. 1170: 회원 계좌의 잔액이 변동되었습니다. 1198: API 호출이 중복 요청되었습니다. 1199: 내부 요청 오류 1900: 일시적 오류입니다. 잠시 후 다시 시도하세요. 1903: 일시적 오류입니다. 잠시 후 다시 시도하세요. 1999: 이전 요청 정보와 다릅니다.(요청 재시도시) 9000: 내부 오류 # 오류코드 1900,1902,1999 발생시 재시도 가능하다.
returnMessage	String	결과 메시지 또는 실패 사유. 예를 들면 다음과 같은 경우다. - 결제 불가 가맹점 - 가맹점 인증 정보 오류

인증 무효 처리 API 예제

인증 무효 처리 API 요청 예제

```
POST https://sandbox-api-pay.line.me
/v2/payments/authorizations/sdhqiwouehrafdsrqoi123as/void HTTP/1.1
Content-Type: application/json
```

인증 무효 처리 API 응답 예제

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
}
```

인증 내역

인증 내역 조회 API

LINE Pay 로 인증된 내역을 조회하는 API 이다. 인증되었거나 인증 무효 처리된 데이터만 조회할 수 있으며, 매입 이후의 데이터는 "결제 내역 조회 API"에서 조회할 수 있다.

인증 내역 조회 API 명세

표 12 인증 내역 조회 Endpoint 명세

항목	설명
Method	GET
Required	X-LINE-ChannelId: {연동 정보 - channelId}
Request Header	X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/authorizations
sandbox	https://sandbox-api-pay.line.me/v2/payments/authorizations
Real	https://api-pay.line.me/v2/payments/authorizations

표 13 인증 내역 조회 API 조회 파라미터

항목	데이터 타입	필수 여부	설명
transactionId	number	N	LINE Pay 에서 발급한 거래 번호
orderId	String	N	가맹점 주문번호

* 조회 요청 시, 하나 이상의 파라미터가 전달되어야 하며, 최대 100 개의 내역을 조회할 수 있다.

표 14 인증 내역 조회 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1150: 거래 내역이 존재하지 않습니다. 1177: 최대 조회 가능한 거래 수 초과(100 개) 9000: 내부오류
returnMessage	String	결과 메시지 또는 실패 사유

항목	데이터 타입	설명
info[].transactionId	number	거래 번호(19 자리)
info[].transactionDate	String	거래 일시(ISO 8601)
info[].transactionType	String	거래 구분 - PAYMENT: 결제 - PAYMENT_REFUND: 환불 - PARTIAL_REFUND: 부분 환불
info[].payStatus	String	결제 상태 - AUTHORIZATION: 인증 - VOIDED_AUTHORIZATION: 인증 무효("인증 무효 처리 API"를 호출한 상태) - EXPIRED_AUTHORIZATION: 인증 만료(LINE Pay 에서 가맹점에게 허용한 인증 기한이 지난 경우)
info[].payInfo[].method	String	결제 시 사용된 결제 수단 (신용카드: CREDIT_CARD, 잔고: BALANCE, 할인: DISCOUNT)
info[].payInfo[].amount	number	거래 금액(거래 번호 발생 시 거래된 거래 금액) 원결제 거래를 조회할 때 최종 거래 금액은 $\text{sum}(\text{info}[\text{.payInfo}[\text{.amount}]) - \text{sum}(\text{refundList}[\text{.refundAmount}])$ 이다.
info[].productName	String	상품명
info[].currency	String	통화(ISO 4217)
info[].orderId	String	가맹점 주문 번호
info[].authorizationExpireDate	String	인증 만료 일시(ISO 8601)

인증 내역 조회 API 예제

인증 내역 조회 API 요청 예제

```
GET https://sandbox-api-pay.line.me
/v2/payments/authorizations?transactionId=2014123123123&transactionId=2014111111222
222 HTTP/1.1
```

인증 내역 조회 API 응답 예제

```
{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": [{
    "transactionId": "1020140728100001997",
    "transactionDate": "2014-07-28T09:48:43Z",
    "transactionType": "PAYMENT",
    "payInfo": [
      { "method": "BALANCE", "amount": 10 },
      { "method": "DISCOUNT", "amount": 10 }
    ],
    "productName": "tes production",
    "currency": "USD",
    "orderId": "20140101123123123",
    "payStatus": "AUTHORIZATION",
  }]
```

Error! Style not defined.

```
"authorizationExpireDate" : "2014-07-28T09:48:43Z"  
  }]  
}
```

결제 내역

결제 내역 조회 API

LINE Pay 로 결제된 결제 건을 조회한다. 매입 완료된 데이터만 조회할 수 있다.

결제 내역 조회 API 명세

표 15 결제 내역 조회 Endpoint 명세

항목	설명
Method	GET
Required	X-LINE-ChannelId: {연동 정보 - channelId}
Request Header	X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments
sandbox	https://sandbox-api-pay.line.me/v2/payments
Real	https://api-pay.line.me/v2/payments

표 16 결제 내역 조회 API Query 파라미터

항목	데이터 타입	필수 여부	설명
transactionId	number	N	LINE Pay 에서 발급한 결제 또는 환불 거래번호
orderId	String	N	가맹점 주문 번호

* 조회 요청 시 하나 이상의 파라미터가 전달되어야 하며, 최대 100 개의 내역을 조회할 수 있다.

표 17 결제 내역 조회 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1150: 거래 내역이 존재하지 않습니다. 1177: 최대 조회 가능한 거래 수 초과(100 개) 9000: 내부오류
returnMessage	String	결과 메시지 또는 실패 사유
info[].transactionId	number	거래 번호(19 자리)

항목	데이터 타입	설명
info[].transactionDate	String	거래 일시(ISO 8601)
info[].transactionType	String	거래 구분 - PAYMENT: 결제 - PAYMENT_REFUND: 환불 - PARTIAL_REFUND: 부분 환불
info[].payInfo[].method	String	결제 시 사용된 결제 수단 (신용카드: CREDIT_CARD, 잔고: BALANCE, 할인: DISCOUNT)
info[].payInfo[].amount	number	거래 금액(거래 번호 발생 시 거래된 거래 금액) 원결제 거래를 조회할 때 최종 거래 금액은 $\text{sum}(\text{info}[\text{.payInfo}[\text{.amount}]) - \text{sum}(\text{refundList}[\text{.refundAmount}])$ 이다.
info[].productName	String	상품명
info[].currency	String	통화(ISO 4217)
info[].orderId	String	가맹점 주문 번호

원결제 거래 조회 및 환불 거래가 있는 경우

info[].refundList[].refundTransactionId	number	환불 거래 번호(19 자리)
info[].refundList[].refundTransactionType	String	거래 구분 - PAYMENT_REFUND: 환불 - PARTIAL_REFUND: 부분 환불
info[].refundList[].refundAmount	number	환불 금액
info[].refundList[].refundTransactionDate	String	환불 거래 일시(ISO 8601)

환불 거래 조회 시

info[].originalTransactionId	number	원결제 거래 번호(19 자리)
------------------------------	--------	------------------

결제 내역 조회 API 예제

결제 내역 조회 API 요청 예제

```
GET https://sandbox-api-pay.line.me/v2/payments?orderId=20140101123123123&orderId=201411111122222 HTTP/1.1
```

결제 내역 조회 API 응답 예제(원 결제 거래 조회 시)

```
{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": [{
    "transactionId": "1020140728100001997",
    "transactionDate": "2014-07-28T09:48:43Z",
    "transactionType": "PAYMENT",
```

```

    "payInfo": [
      { "method": "BALANCE", "amount": 10 },
      { "method": "DISCOUNT", "amount": 10 }
    ],
    "productName": "tes production",
    "currency": "USD",
    "orderId": "20140101123123123",
    "refundList": [
      {
        "refundTransactionId": 20213901923123,
        "transactionType": "PARTIAL_REFUND",
        "refundAmount": -1,
        "refundTransactionDate": "2014-07-28T09:48:52Z"
      },
      {
        "refundTransactionId": 20213901923123,
        "transactionType": "PARTIAL_REFUND",
        "refundAmount": -1,
        "refundTransactionDate": "2014-07-28T09:49:28Z"
      },
      {
        "refundTransactionId": 20213901923123,
        "transactionType": "PARTIAL_REFUND",
        "refundAmount": -1,
        "refundTransactionDate": "2014-07-28T09:53:18Z"
      }
    ]
  }
}

```

결제 내역 조회 API 응답 예제(환불 거래 조회 시)

```

{
  "returnCode": "0000",
  "returnMessage": "success",
  "info": [
    {
      "transactionId": 1020140728100001997,
      "transactionDate": "2014-07-28T09:48:43Z",
      "transactionType": "PARTIAL_REFUND",
      "amount": -5,
      "productName": "",
      "currency": "USD",
      "orderId": "20140101123123123",
      "originalTransactionId": 1020140728100001999
    }
  ]
}

```

결제 환불

결제 환불 API

LINE Pay 로 결제완료(매입완료)된 건의 환불을 요청한다. 환불 시 LINE Pay 회원의 결제 거래 번호는 필수로 전달되어야 하며, 부분 환불도 가능하다.

결제 환불 API 명세

표 18 결제 환불 Endpoint 명세

항목	설명
method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/{transactionId}/refund
sandbox	https://sandbox-api-pay.line.me/v2/payments/{transactionId}/refund
Real	https://api-pay.line.me/v2/payments/{transactionId}/refund

- transactionId 는 결제 거래 번호를 의미한다.

표 19 결제 환불 API 파라미터

항목	데이터 타입	필수여부	설명
refundAmount	number	N	환불 금액 - 전달되지 않은 경우 전체 환불

표 20 결제 환불 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 • 0000: 성공 • 1101: 구매 회원이 거래 불가 상태입니다. • 1102: 구매 회원이 거래 불가 상태입니다. • 1104: 가맹점이 존재하지 않습니다. • 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 • 1106: 헤더 정보 오류 • 1124: 금액 정보 오류

항목	데이터 타입	설명
		1150: 거래 내역이 존재하지 않습니다. 1155: 환불할 수 없는 거래 유형의 거래 번호입니다. 1163: 환불이 불가능합니다.(환불 가능일 초과) 1164: 환불 가능 금액을 초과하였습니다. 1165: 이미 환불된 거래입니다. 1179: 처리할 수 없는 상태입니다. 1198: API 호출이 중복 요청되었습니다. 1199: 내부 요청 오류 1900: 일시적 오류입니다. 잠시 후 다시 시도하세요. 1902: 일시적 오류입니다. 잠시 후 다시 시도하세요. 1999: 이전 요청 정보와 다릅니다.(요청 재시도시) 9000: 내부 오류
		# 오류코드 1900,1902,1999 발생시 재시도 가능하다.
returnMessage	String	결과 메시지 또는 실패 사유
info.refundTransactionId	number	환불 거래 번호(신규 발급-19 자리)
info.refundTransactionDate	String	환불 거래 일시(ISO 8601)

결제 환불 API 예제

결제 환불 API 요청 예제

```
// 부분 환불인 경우
{
  "refundAmount" : 1000
}

// 전체 환불인 경우
{}
```

결제 환불 API 응답 예제

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "refundTransactionId" : 123123123123,
    "refundTransactionDate" : "2014-01-01T06:17:41Z",
  }
}
```

regKey(자동 결제를 위한 키) 사용 가능 여부 확인

regKey 사용 가능 여부 API

자동 결제 API 를 사용하기 전에 regKey 가 사용 가능한 상태인지 확인한다.

regKey 사용 가능 여부 확인 API 명세

표 21 regKey 사용 가능 여부 확인 Endpoint 명세

항목	설명
Method	GET
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/preapprovedPay/{regKey}/check
Sandbox	https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/{regKey}/check
Real	https://api-pay.line.me/v2/payments/preapprovedPay/{regKey}/check

표 22 regKey 사용 가능 여부 확인 API Query 파라미터

항목	데이터 타입	필수 여부	설명
creditCardAuth	Boolean	N	regKey 에 설정된 신용카드의 최소금액 인증 여부 - true : - LINE Pay 내부의 데이터 유효성 검증. - 신용카드 최소금액 인증 - 가맹점 정보에서 최소금액 인증 기능을 활성화해야 사용 가능하며, LINE Pay 관리자의 검수가 필요하다. - false(기본값) : - LINE Pay 내부의 데이터 유효성 검증.

표 23 regKey 사용 가능 여부 확인 API 응답

항목	데이터 타입	설명
returnCode	String(4byte)	결과 코드 0000: 성공 1101: 구매 회원이 거래 불가 상태입니다. 1102: 구매 회원이 거래 불가 상태입니다. 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1141: regKey 에 설정된 신용카드를 인증 할 수 없습니다. 1154: 자동 결제 계좌를 이용할 수 없는 상태입니다. 1190: regKey 가 존재하지 않습니다. 1193: regKey 가 만료되었습니다.
returnMessage	String	결과 메시지 또는 실패 사유

regKey 사용 가능 여부 확인 API 예제

regKey 사용 가능 여부 확인 API 요청 예제

```
GET https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK7862119XTF1TT/check HTTP/1.1
X-Line-ChannelId: 1414228945
X-Line-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
Content-Type: application/json
```

regKey 사용 가능 여부 확인 API 응답 예제

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK"
}
```

자동 결제

자동 결제 API

결제 예약(Reserver API)시 결제유형(type)이 PREAPPROVED 로 결제된 경우, 결제 Confirm API 결과 수신 시 regKey 를 전달받는다. 자동결제 API 는 regKey 를 이용하여 LINE 앱을 통한 사용자 재인증 과정없이 결제(매입)가 가능하다

자동 결제 API 명세

표 24 자동 결제 Endpoint 명세

항목	설명
Method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 40 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/preapprovedPay/{regKey}/payment
Sandbox	https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/{regKey}/payment
Real	https://api-pay.line.me/v2/payments/preapprovedPay/{regKey}/payment

표 25 자동 결제 API 파라미터

항목	데이터 타입	필수 여부	설명
productName	String (4000byte)	Y	상품명
amount	number	Y	결제 금액
currency	String (3byte)	Y	결제 통화(ISO 4217) 지원 가능한 통화는 아래와 같다. • USD • JPY • TWD • THB
orderId	String (100byte)	Y	해당 결제 건에 대한 가맹점의 주문 번호. 가맹점에서 관리하는 고유 키.
capture	Boolean	N	매입 여부 • true(기본값): 인증과 매입을 한번에 처리한다. • false: 인증까지 처리하고 "매입 API"를 호출

항목	데이터 타입	필수 여부	설명
해야 거래가 완료된다			

표 26 자동 결제 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1101: 구매 회원이 거래 불가 상태입니다. 1102: 구매 회원이 거래 불가 상태입니다. 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다 1106: 헤더 정보 오류 1110: 사용할 수 없는 신용카드입니다. 1124: 금액 정보 오류 1141: 결제 계좌 상태 오류 1154: 자동 결제 계좌를 이용할 수 없는 상태입니다. 1172: 이미 동일 주문번호로 거래된 내역이 존재합니다. 1183: 결제 금액이 0 보다 커야 합니다. 1190: regKey 가 존재하지 않습니다. 1193: regKey 가 만료되었습니다. 1198: API 호출이 중복 요청되었습니다. 1199: 내부 요청 오류 1280: 신용카드 결제 중 일시적 오류 1281: 신용카드 결제 오류 1282: 신용카드 승인 오류 1283: 부정 사용이 의심되어 결제가 거절되었습니다. 1284: 신용카드 결제가 일시적으로 중단되었습니다. 1285: 신용카드 결제 정보 누락 1286: 신용카드 결제 정보가 잘못되었습니다. 1287: 신용카드 유효기간이 초과되었습니다. 1288: 신용카드 결제 계좌 잔고가 부족합니다 1289: 신용카드 한도 초과 1290: 신용카드 건당 결제 한도 초과 1291: 도난 신고된 카드입니다. 1292: 사용이 정지된 카드입니다. 1293: CVN 입력 오류 1294: 블랙리스트에 등록된 카드입니다. 1295: 신용카드 번호가 잘못되었습니다. 1296: 잘못된 금액 오류 1298: 카드 사용이 거절되었습니다. 9000: 내부 오류 # 오류 코드 1280~1287, 1290~1294 가 발생한 경우 regKey 를 만료 처리한다.
returnMessage	String	결과 메시지 또는 실패 사유
info.transactionId	number	거래 번호(19 자리)
info.transactionDate	String	거래 일시(ISO 8601)

항목	데이터 타입	설명
info.authorizationExpire Date	String	인증 만료 일시(ISO 8601) 인증(capture=false)만 한 결제 건인 경우 전달

자동 결제 API 예제

자동 결제 API 요청 예제

```
{
  "productName" : "네이버 뮤직 월 정기 이용권",
  "amount" : 10000,
  "currency" : "JPY",
  "orderId" : "testOrd2014121200000001"
}
```

자동 결제 API 응답 예제

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "transactionId" : 123123123123,
    "transactionDate" : "2014-01-01T06:17:41Z"
  }
}
```

regKey 만료

regKey 만료 API

자동 결제로 등록된 regKey 정보를 만료시키는 API 다. 이 API 를 호출한 후에는 해당 regKey 를 사용하여 자동 결제를 할 수 없다.

regKey 만료 API 명세

표 27 regKey 만료 Endpoint 명세

항목	설명
Method	POST
Required	Content-Type: application/json; charset=UTF-8
Request Header	X-LINE-ChannelId: {연동 정보 - channelId} X-LINE-ChannelSecret: {연동 정보 - channelSecretKey}
Request timeout	connection : 5 초 read : 20 초
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/preapprovedPay/{regKey}/expire
Sandbox	https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/{regKey}/expire
Real	https://api-pay.line.me/v2/payments/preapprovedPay/{regKey}/expire

표 28 regKey 만료 API 응답

항목	데이터 타입	설명
returnCode	String (4byte)	결과 코드 0000: 성공 1104: 가맹점이 존재하지 않습니다. 1105: 현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다. 1106: 헤더 정보 오류 1190: regKey 가 존재하지 않습니다. 1193: regKey 가 만료되었습니다. 9000: 내부 오류
returnMessage	String	결과 메시지 또는 실패 사유

regKey 만료 API 예제

regKey 만료 API 요청 예제

```
POST https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK123asd213/expire
HTTP/1.1
Content-Type: application/json
```

regKey 만료 API 응답 Body 예시

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
}
```

부록

LINE Pay 오류 코드

LINE Pay 에서 발생하는 오류 코드에 대해 정의한다. 오류 코드에 대한 returnMessage 는 영어로 제공되며, 별도로 메시지가 없는 경우에는 하이픈(-)이 전달된다.

표 29 LINE Pay 오류 코드

코드 (returnCode)	설명(300byte)
1101	LINE Pay 회원이 아닙니다.
1102	구매 회원이 거래 불가 상태입니다.
1104	가맹점이 존재하지 않습니다.
1105	현 가맹점은 LINE Pay 를 이용할 수 없는 상태입니다.
1106	헤더 정보 오류
1110	사용할 수 없는 신용카드입니다.
1124	금액 정보 오류입니다(scale).
1133	유효하지 않은 oneTimeKey 입니다.
1141	결제 계좌 상태 오류
1142	잔액 부족
1145	결제 진행 중 입니다.
1150	거래 내역이 존재하지 않습니다.
1152	이미 거래가 존재합니다.
1153	결제 reserve 시 금액과 요청한 금액이 다릅니다.
1154	자동 결제에 설정한 결제수단을 이용할 수 없는 상태입니다.
1155	환불할 수 없는 거래 유형의 거래 번호입니다.
1159	결제 요청정보가 없습니다.
1163	환불 가능일이 지나 환불이 불가능합니다.
1164	환불 가능 금액을 초과하였습니다.
1165	이미 환불된 거래입니다.
1169	결제 confirm 을 위한 정보 오류(LINE Pay 에서 결제 수단 선택과 암호 인증을 해야 합니다.)
1170	회원 계좌의 잔액이 변동되었습니다.
1172	이미 동일 주문번호로 거래된 내역이 존재합니다.
1177	최대 조회 가능한 거래 수 초과(100 개)
1178	가맹점이 지원하지 않는 통화입니다.
1179	처리할 수 없는 상태입니다.

코드 (returnCode)	설명(300byte)
1180	결제 유효시간이 지났습니다.
1183	결제금액이 0 보다 커야 합니다.
1184	결제금액이 요청시 금액을 초과합니다.
1190	regKey 가 존재하지 않습니다.
1193	regKey 가 만료되었습니다.
1194	자동 결제를 사용할 수 없는 가맹점
1197	regKey 로 이미 결제 중입니다.
1198	요청 처리 중입니다.
1199	내부 요청 오류
1280	신용카드 결제 중 일시적 오류
1281	신용카드 결제 오류
1282	신용카드 승인 오류
1283	부정 사용이 의심되어 결제가 거절되었습니다.
1284	신용카드 결제가 일시적으로 중단 되었습니다.
1285	신용카드 결제 정보 누락
1286	잘못된 신용카드 결제 정보
1287	신용카드 유효기간이 초과되었습니다.
1288	신용카드 결제 은행 잔고가 부족합니다
1289	신용카드 한도 초과
1290	신용카드 건당 결제 한도 초과
1291	도난 신고된 카드입니다.
1292	사용이 정지된 카드입니다.
1293	CVN 입력 오류
1294	블랙리스트에 등록된 카드입니다.
1295	신용카드 번호가 잘못되었습니다.
1296	처리할 수 없는 금액입니다.
1298	카드 사용이 거절되었습니다.
1900	일시적 오류입니다. 잠시 후 다시 시도하세요.
1901	일시적 오류입니다. 잠시 후 다시 시도하세요.
1902	일시적 오류입니다. 잠시 후 다시 시도하세요.
1903	일시적 오류입니다. 잠시 후 다시 시도하세요.
1999	이전 요청 정보와 다릅니다.(요청 재시도시)
2101	파라미터 오류

코드 (returnCode)	설명(300byte)
2102	JSON 데이터 포맷 오류
2103	잘못된 요청입니다. returnMessage 를 확인해주세요.
2104	잘못된 요청입니다. returnMessage 를 확인해주세요.
9000	내부 오류

내부 오류 인 1199, 9000 발생 시에는 기술연동지원(pay_tech@linecorp.com)으로 문의한다.

표 30 API 별 오류코드

오류 코드	reserve	confirm	매입	인증 무효	환불	인증 내역	결제 내역	자동 결제	regKey 확인	regKey 만료
1101		v		v	v			v	v	
1102		v		v	v			v	v	
1104	v	v	v	v	v	v	v	v	v	v
1105	v	v	v	v	v	v	v	v	v	v
1106	v	v	v	v	v	v	v	v	v	v
1110		v						v		
1124	v	v			v			v		
1133	v								v	
1141		v						v		
1142		v								
1145	v									
1150			v	v	v	v	v			
1152		v								
1153		v								
1154								v	v	
1155			v	v	v					
1159		v								
1163					v					
1164					v					
1165				v						
1169		v								
1170		v	v	v						
1172	v	v	v					v		
1177						v	v			
1178	v									
1179			v	v	v					
1180		v								
1183	v		v					v		
1184			v							
1190								v	v	v
1193								v	v	
1194	v									
1198		v	v	v	v			v		
1199		v	v	v	v			v		
1900			v	v	v					

오류 코드	reserve	confirm	매입	인증 무효	환불	인증 내역	결제 내역	자동 결제	regKey 확인	regKey 만료
1901			v							
1902					v					
1903				v						
1999			v	v	v					
2101	v	v	v	v	v	v	v	v	v	v
2102	v	v	v	v	v	v	v	v	v	v
2103	v	v	v	v	v	v	v	v	v	v
2104	v	v	v	v	v	v	v	v	v	v
9000	v	v	v	v	v	v	v	v	v	v

오류코드 1280~1298 는 카드사에서 발생하는 오류로 별도 언급하지 않는다.

PaymentUrl 앱 사용 가이드

가맹점 앱에서 결제를 요청한 이후에 LINE 앱으로 앱 간 이동을 하기 위한 가이드다.

Android 애플리케이션 예제

아래의 예제 코드로 LINE 앱 설치 여부와 사용 가능한 LINE Pay의 버전을 확인할 수 있다. LINE 앱이 설치되어 있고 사용 가능한 LINE Pay의 버전이 확인되면 LINE Pay 결제 화면으로 이동한다.

```
int linePaySupportedVersion = 230;
String paymentUrl = "..."; // This is "paymentUrl.app" URL String.

Context context = getActivity();
try {
    PackageManager pm = context.getPackageManager();
    PackageInfo packageInfo = pm.getPackageInfo("jp.naver.line.android", 0);
    int versionCode = packageInfo.versionCode;
    if (linePaySupportedVersion <= versionCode) {
        launchUri(paymentUrl);
    } else {
        confirmLineInstall(context);
    }
} catch (NameNotFoundException e) {
    confirmLineInstall(context);
}

private void confirmLineInstall(Context context) {
    new AlertDialog.Builder(context)
        .setTitle("LINE Pay")
        .setMessage(getString(R.String.linepay_confirm))
        .setCancelable(false)
        .setPositiveButton(getString(R.String.linepay_install), new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                launchUri("market://details?id=jp.naver.line.android");
            }
        })
        .setNegativeButton(getString(R.String.linepay_cancel), new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .show();
}

private void launchUri(String uriString) {
    Uri uri = Uri.parse(uriString);
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
}
```

File : res/values/Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    ...
    <String name="linepay_confirm">Supported by Android/iPhone LINE versions 4.8.0
or higher.</String>
    <String name="linepay_install">Get it now</String>
    <String name="linepay_cancel">cancel</String>
    ...
</resources>
```

iPhone 애플리케이션 예제

아래의 예제 코드로 LINE 앱의 설치 여부를 확인할 수 있다. LINE 앱이 설치되어 있으면 LINE Pay 결제 화면으로 이동한다.

```
NSString* lineScheme = @"line://";
BOOL installed = [[UIApplication sharedApplication]
                  canOpenURL:[NSURL URLWithString:lineScheme]];
if (installed) {
    UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:@"LINE Pay"
        message:NSLocalizedString(@"linepay.confirm", nil)
        delegate:self
        cancelButtonTitle:NSLocalizedString(@"linepay.ok", nil) otherButtonTitles:nil];
    alert.tag = 1;
    [alert show];
} else {
    UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:@"LINE Pay"
        message:NSLocalizedString(@"linepay.confirm", nil)
        delegate:self
        cancelButtonTitle:NSLocalizedString(@"linepay.cancel", nil)
        otherButtonTitles:NSLocalizedString(@"linepay.install",
        nil), nil];
    alert.tag = 2;
    [alert show];
}

- (void)alertView:(UIAlertView*)alertView
clickedButtonAtIndex:(NSInteger)buttonIndex {
    if (alertView.tag == 1 && buttonIndex == 0) {
        NSString *paymentUrl = ...; // This is "paymentUrl.app" URL String.
        [self launchUrl:paymentUrl];
    } else if (alertView.tag == 2 && buttonIndex == 1) {
        [self launchUrl:@"itms-
apps://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=443904275&mt=8"];
    }
}

- (void)launchUrl:(NSString*)urlString {
    NSURL *url = [NSURL URLWithString:urlString];
    [[UIApplication sharedApplication] openURL:url];
}
```

File : en.lproj/Localized.Strings

```
"linepay.confirm" = "Supported by Android/iPhone LINE versions 4.8.0 or higher.";
"linepay.ok" = "OK";
"linepay.cancel" = "Cancel";
"linepay.install" = "Get it now";
```

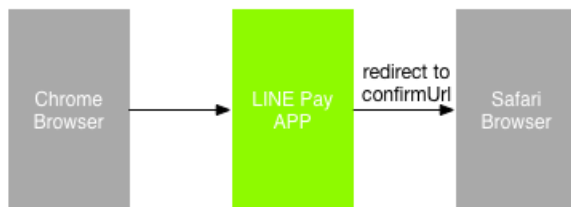
confirmUrl 상세 설명 및 예외 상황

confirmUrl 상세 설명

사용자가 LINE Pay app 에서 결제 인증(결제 수단 선택과 결제 비밀번호 확인) 후 돌아가게 될 callback url
결제가 완료되었음을 의미하는 url 이 아니며, LINE Pay 에서 제공하는 confirm API 를 호출하여 최종 결제를 완료하는
가맹점 페이지의 url

confirmUrl 예외 상황

LINE Pay 에서 사용자 결제 인증 후에 confirmUrl 호출 시에 사용자의 환경에 따라 결제를 요청했던 Browser 와
confirmUrl 을 받는 Browser 가 다를 수 있다.



결제 reserve API 의 “checkConfirmUrlBrowser”가 “true”인 경우

LINE Pay 에서 browser 를 확인하여 결제를 시작한 browser 와 이동된 browser 가 같은지 판단한다.
판단 결과에 따라

- 같은 경우 : confirmUrl 로 이동
- 다른 경우 : 결제를 요청했던 browser 로 이동 안내

결제 reserve API 의 “checkConfirmUrlBrowser”가 “false”(기본값)인 경우

가맹점에서 Browser 의 session data(e.g. 로그인 session)에 기반하여 처리하는 내용이 있다면, confirmUrl 호출 시에
사용자의 결제가 정상적으로 진행되지 않을 수 있다.

따라서, 가맹점은 LINE Pay 의 결제 연동시, 아래와 같이 구현해야 한다.

1. confirmUrl 이 호출 될 때, session data 기반의 인증을 수행하지 않고 confirmUrl 에 담긴 정보로 결제를 완료
2. 그 이후 session data 에 대한 처리 예
 - a. session data 확인이 된다면, 기존 가맹점 결제 결과 화면 노출
 - b. session data 확인이 되지 않는다면, 아래 2 가지 방법 중 가맹점이 선택하여 구현
 - session data 재생성(e.g. 로그인) 후 결제 결과 화면 노출
 - 결제를 요청했던 browser 로 돌아가도록 사용자에게 안내

ConfirmUrl을 Server-to-Server로 전달한 경우

LINE Pay 에서 가맹점 서버로 ConfirmUrl 을 호출한 경우를 정의한다.

가맹점이 ACL(Access Control List)를 관리하는 경우

가맹점은 LINE Pay 서버의 IP 를 등록/관리해야 한다. LINE Pay 의 환경 별 IP 정보는 아래와 같다.

- Sandbox : 182.162.196.200
- Real : 211.249.40.1~211.249.40.30

ConfirmUrl 명세

표 31 confirmUrl Endpoint 명세

항목	설명
Protocol	HTTPS
Method	GET
Request timeout	Connection: 5 초 Read: 20 초

표 32 confirmURL 파라미터

파라미터	설명
orderId	결제 reserve 시 가맹점에서 전달한 주문 번호
transactionId	결제 reserve 시 결과로 전달받은 거래 번호

가맹점에서는 응답 결과로 별도의 정보를 전달하지 않고, LINE Pay에서는 HTTP 응답 코드로 성공 여부를 판단한다. HTTP 응답 코드가 성공(200 OK)이 아니면 LINE Pay 회원에게 결제가 정상적으로 처리되지 않았다는 알림을 보낸다.

ConfirmUrl 사용 예제

가맹점이 제공한 confirmUrl 이 'http://testmall.com/pay/result'일 때 예제는 다음과 같다.

요청(LINE Pay -> 가맹점)

```
https://testmall.com/pay/result?orderId=2014xxx1232132&transactionId=201408011234567890
```

응답

```
// 성공인 경우
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Charset: utf-8
Content-Type: text/html
Content-Length: 2
Date: Sat, 11 Oct 2014 02:45:48 GMT

// 실패인 경우
HTTP/1.1 404 NOT_FOUND
Server: Apache-Coyote/1.1
Accept-Charset: utf-8
Content-Type: text/html
Content-Length: 4
Date: Sat, 11 Oct 2014 02:45:48 GMT
```

Sandbox 사용방식

LINE Pay 에서 Sandbox 환경에서 테스트 할 경우, 본 연동가이드에서 제공하는 API Endpoint 명세에 기재된 sandbox 를 활용하여 결제를 시도해야 하며, 결제를 위한 페이지는 가맹점이 준비를 해야 한다.
Sandbox 환경에서 결제를 시도하는 경우, 다음과 같이 처리된다.

※ 특이사항

Sandbox 환경에서는 PC 를 활용하건 모바일을 활용하건 모두 채널 로그인을 요구한다.

(단, 리얼에서 연동하는 경우, 스마트폰을 활용시 라인을 통해 유저의 MID 정보를 받아 사용하기 때문에 채널 로그인 이 제외된다.)

Sandbox PC 결제

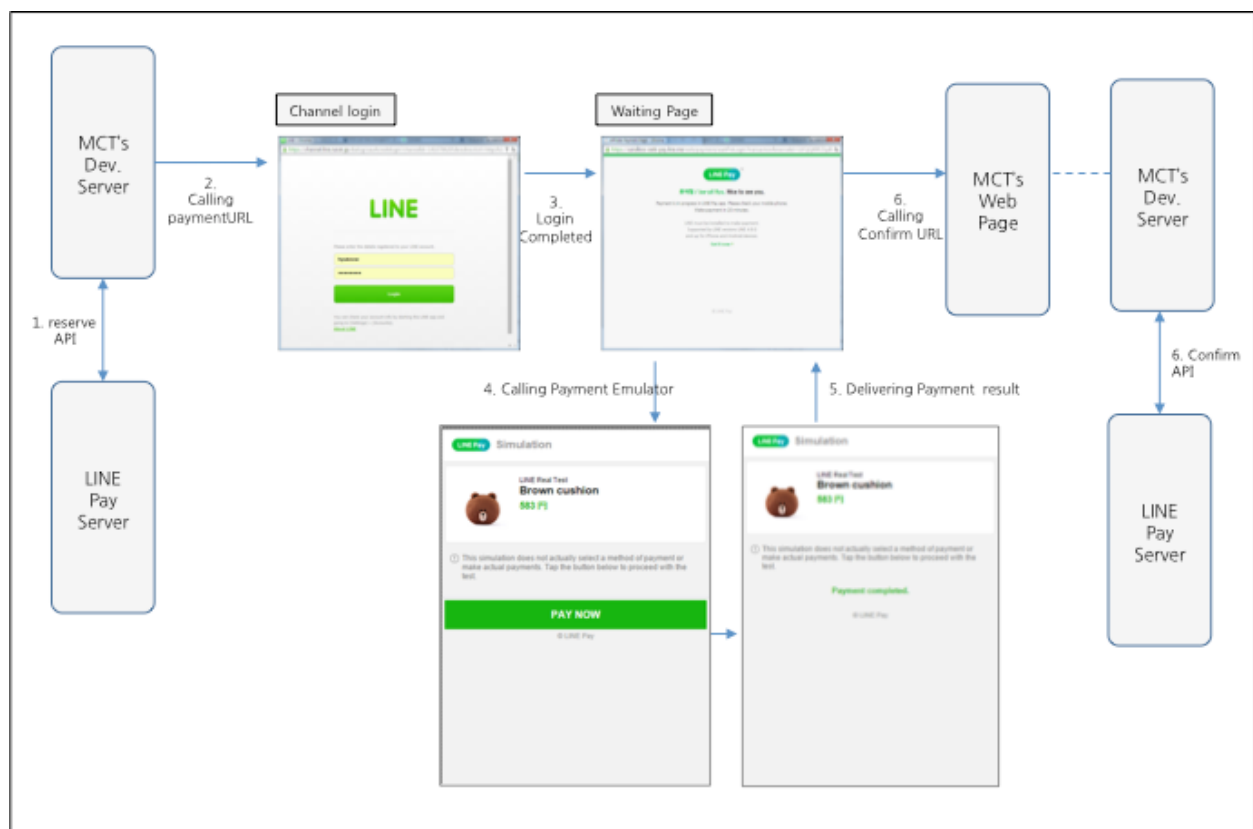


그림 10 Sandbox PC 결제 진행화면 예제

Sandbox 모바일 결제

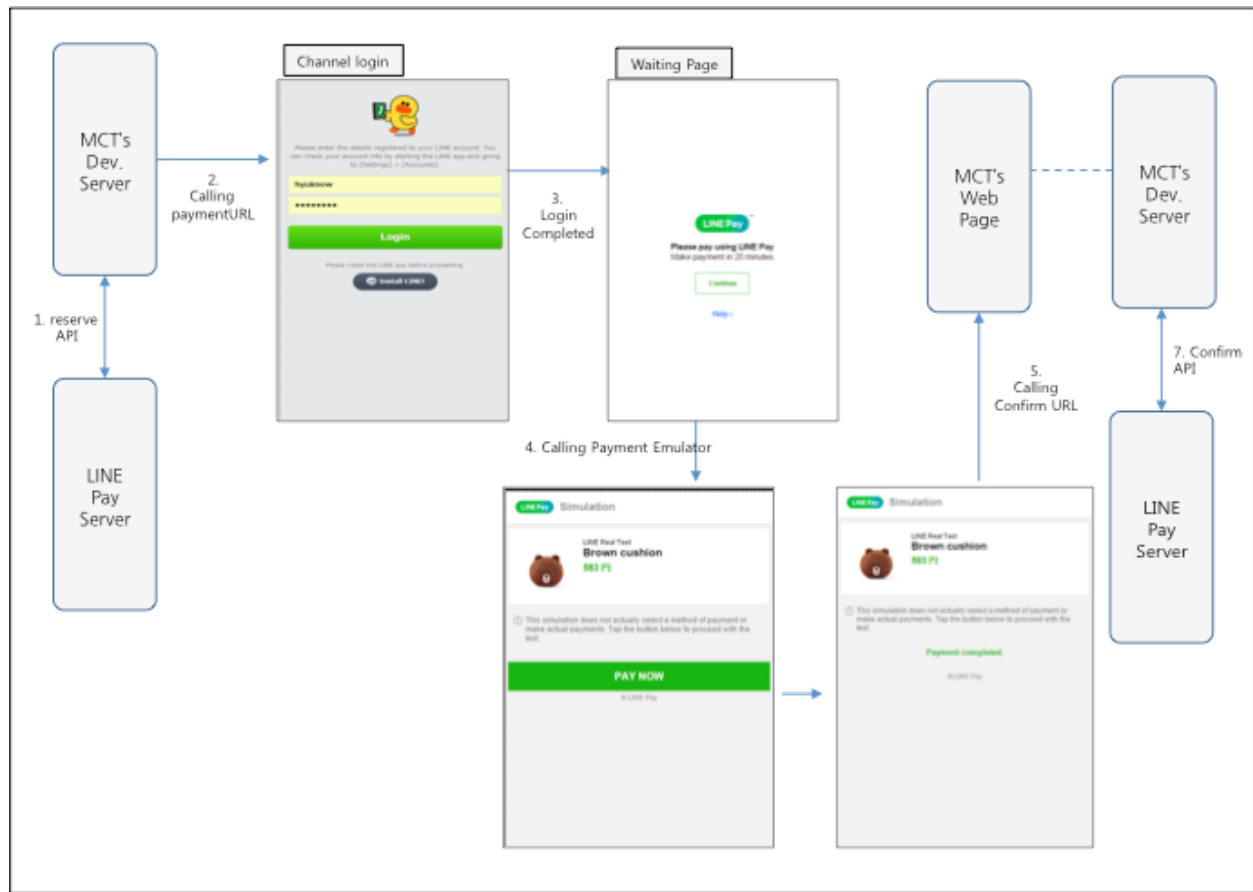


그림 11 Sandbox 모바일 결제 진행화면 예제

Sandbox OneTimeKey 결제

사용자가 LINE Pay app 의 “My Code”에서 보여주는 QR/Bar Code 를 가맹점이 읽어 결제를 진행할 수 있다.

Sandbox 에서도 “My Code”를 web page 로 제공하며, 결제 테스트를 할 수 있다. Sandbox 환경에서 가맹점은 My Code 를 읽어, 결제 Reserve API 를 호출하면 My Code 웹 페이지는 결제 simulation 페이지로 이동한다.

- Sandbox My code url : <https://sandbox-web-pay.line.me/web/sandbox/payment/otk>