
PAYCO 연동 가이드

저작권

Copyright © 2014 NHN Entertainment Corp. All Rights Reserved.

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

이 문서는 정보 제공의 목적으로만 제공됩니다. NHN Entertainment(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, NHN Entertainment(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.

NHN Entertainment(주)는 이 문서의 내용을 예고 없이 변경할 수 있습니다.

문서 정보

문서 개요

이 문서는 가맹점 사이트에서 PAYCO로 구매 & 결제 할 수 있도록 PAYCO 연동 가이드에 따라 상품 정보 및 구매 연동 시스템을 개발하는 방법에 대해 설명한다.

- PAYCO 구매하기 버튼 연동 : 버튼 노출 방법 설명
- PAYCO 에서 제공하는 연동 API 명세 : API 목록 및 상세 명세
- 가맹점에서 구현해야 하는 API 명세 가이드

독자

이 문서의 독자는 PAYCO 가맹점 사이트 개발자이다.

문의처

이 문서의 내용에 오류가 있거나 내용과 관련한 의문 사항이 있으면 아래의 연락처에 문의한다.

- 기술지원 : PAYCO 기술지원 (dl_payco_ts@nhnent.com)
- 문서 버전 및 이력

| 버전 | 일자 | 이력사항 |
|-----|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.0 | 2014-12-18 | 초안 작성 |
| | 2014-12-26 | 주문 및 결제 취소 가능 여부 조회 API, 주문 및 결제 취소 처리 API 변경 환불 결과 확인 API 삭제 |
| | 2015-01-08 | - PAYCO 연동 : 간편결제 타입(A2) 추가, 버튼 파라미터에 sellerkey 추가 - 주문예약 API 설명 보충, 추가 배송비에 대한 설명 추가 - 연동 시 필요한 키 값 정의 추가 |
| | 2015-01-28 | - 주문예약 API orderSheetUiType 컬럼 추가 (주문서 UI 선택) - 주문 취소 API URL 수정 - 주문 취소 API orderQuantity 컬럼 삭제 - 추가 배송비 조회 API 명세 변경 (주소지 추가) - 연동 키 체계 설명 수정 (20p) - 마일리지 적립 취소 API 추가 (3.2.5) - 구매 확정 API 삭제 (주문상태변경으로 통합) |

| | | |
|-------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | - 연동 키 유효성 체크 API 추가 |
| | 2015-02-02 | <ul style="list-style-type: none"> - 주문예약, 주문 취소, 주문 완료 통보 API 의 추가 배송비 부분 수정 - 주문예약 API 의 productUnitPrice, productPaymentUnitPrice 컬럼삭제 - 연동 키 유효성 체크 API 명세 변경 |
| | 2015-02-10 | <ul style="list-style-type: none"> - 주문예약 API deliveryGroupNo 컬럼 삭제 - 모든 API memberChannelCode, memberKey 컬럼 삭제 - 추가배송비 조회 API detailRemoteAreaDeliveryFees 컬럼 삭제 |
| | 2015-02-16 | <ul style="list-style-type: none"> - 주문예약 extraData 추가 (payExpiryYmdt, virtualAccountExpiryYmd, appUrl, cancelMobileUrl 컬럼추가) |
| 1.1 | 2015-03-17 | <ul style="list-style-type: none"> - 취소가능여부조회, 취소처리 API 수수료(cancelTotalFeeAmt) 항목 제거 - 취소처리 API 현금영수증 receiptIssueYn 컬럼제거 - 개요설명추가 - 주문예약 결제모드(payMode) 컬럼추가 - 결제 승인 API (결제모드 PAY2 타입) 추가 - 결제 인증 완료 (결제모드 PAY2 타입) API 추가 |
| 1.1.1 | 2015-03-24 | <ul style="list-style-type: none"> - 주문완료통보 API cardSettleInfo 전달항목 - 결제승인 API cardSettleInfo 응답항목 수정 : 카드결제내역에 할부 개월, 카드사 승인번호 추가 installmentMm, cardAdmissionNo 컬럼 추가 |
| 1.1.2 | 2015-03-27 | <ul style="list-style-type: none"> - 결제승인 API input 항목 sellerOrderReferenceKey 컬럼추가 - 결제인증완료 sellerOrderReferenceKey 컬럼추가 |
| 1.1.3 | 2015-03-31 | <ul style="list-style-type: none"> - 개요 내용 추가 & 부가 설명 추가 - 시퀀스다이어그램 수정 (추가배송비 flow 추가) - 문구수정 (체크아웃 => 간편구매) - 주문예약 API taxationType 컬럼추가 - 주문예약 API extraData > appUrl 컬럼삭제 - 주문예약 serviceUrlParam, returnUrlParam 샘플추가 - 결제인증완료 returnUrlParam 삭제 및 설명추가 code 리턴코드 컬럼 추가 - 카드결제내역 할부개월 installmentMm -> cardInstallmentMonthNumber 컬럼명변경 - 결제승인 API 샘플소스 추가 - |
| 1.1.4 | 2015-04-03 | <ul style="list-style-type: none"> - 주문예약 API taxationType 컬럼삭제 |

| | | |
|---------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | - 주문예약 API) totalTaxfreeAmt , totalTaxableAmt , totalVatAmt 컬럼추가 |
| 1.1.5 | 2015-04-10 | <ul style="list-style-type: none"> - 카드사코드, 통신사코드, 은행코드 정의 추가 - 주문예약 API) 상품별정보 taxationType 컬럼추가 - 결제취소 API) remainCancelPossibleAmt, totalCancelTaxfreeAmt, totalCancelTaxableAmt, totalCancelVatAmt ,totalCancelPossibleAmt 컬럼추가 |
| 1.1.6 | 2015-04-10 | <ul style="list-style-type: none"> - 주문완료통보) cardInterestFreeYn 카드무이자여부, corporateCardYn 법인카드여부, partCancelPossibleYn 부분취소가능여부 컬럼추가 - 결제승인 API) cardInterestFreeYn 카드무이자여부, corporateCardYn 법인카드여부, partCancelPossibleYn 부분취소가능여부 컬럼추가 - 결제예약 API) INPUT > totalOrderAmt 컬럼제거 - 결제취소 API) orderNo 컬럼추가 - 주문상품상태변경 API) orderNo 컬럼추가 - 유의사항 추가 - 에러코드) 시스템공지 ERRORCODE 추가 (9000) |
| 1.1.7 | 2015-05-04 | <ul style="list-style-type: none"> - 영수증 추가 - 3.5.5 결제 수단 > 쿠폰 추가 - paymentDetails 설명 보충 |
| 1.1.7.2 | 2015-05-08 | - 영수증 도메인정보 추가 |
| 1.1.7.3 | 2015-05-11 | - 추가배송비 과세처리방법 안내사항 추가 |
| 1.1.7.4 | 2015-05-13 | <ul style="list-style-type: none"> - 영수증 샘플 URL 추가 - nonBankbookPaymentInformUrl 설명부분 nonBankbookDepositInformUrl 로 변경 |
| 1.1.8 | 2015-05-15 | <ul style="list-style-type: none"> - 사전준비사항 주문인증키(orderCertifyKey) 관련 문구 추가 - 주문완료통보시 주문인증키 orderCertifyKey 값 추가 - 주문취소 API URL 변경 /outseller/order/cancel/request => /outseller/order/cancel 주문인증키 orderCertifyKey 필수값 추가) |
| 1.1.8.2 | 2015-05-21 | - 결제승인 API 주문인증키 orderCertifyKey 값 추가 |
| 1.1.8.3 | 2015-05-27 | <ul style="list-style-type: none"> - 주문유형 설명추가 - 결제수단 및 코드조회 API 추가 - 주문예약 팝업차단관련 TIP 문구 추가 - 바로이체(35) 결제수단코드 추가 |

| | | |
|----------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.1.8.4 | 2015-06-01 | - 가맹점용 영수증 추가 |
| 1.1.8.5 | 2015-06-08 | - inApp 부가설명 추가 |
| 1.1.8.6 | 2015-06-09 | - 주문예약 팝업오픈 IE11 관련 유의사항 추가 - PAY2 방식 승인 API 중복 방어코드 작성 안내 추가 - Iframe 부가설명 추가 |
| 1.1.8.7 | 2015-06-15 | - inApp 적용가이드 : 쿠키관련 내용 및 모바일 백신프로그램관련 안내 추가 - 주문예약 extraData 의 viewOptions 항목 추가 |
| 1.1.8.8 | 2015-06-25 | - 주문예약 customUrlSchemeUseYn 항목 추가 |
| 1.1.9.1 | 2015-07-16 | - 결제승인 API timeout 30 초 설정 안내문구 추가 - 주문 예약 > sellerOrderReferenceKey 중복허용 - 주문 예약 > sellerOrderReferenceKeyType 외부가맹점 주문번호 타입 추가 - sellerOrderReferenceKey 키 연동부분 orderNo 연동으로 변경 : 주문 상품 상태 변경 (sellerOrderReferenceKey 삭제 , orderNo 필수로 변경) : 주문 및 결제 취소 가능여부 조회 (sellerOrderReferenceKey 삭제 , orderNo 필수로 변경) : 주문 및 결제 취소 처리 (sellerOrderReferenceKey 삭제 , orderNo 필수로 변경) : 마일리지 적립 취소 (sellerOrderReferenceKey 삭제 , orderNo 필수로 변경) |
| 1.1.9.2 | 2015-07-29 | - iframe 적용가이드 scroll 관련 script 가이드 추가 |
| 1.1.9.3 | 2015-08-05 | - 주문 및 결제 취소 처리 API requestMemo 추가 |
| 1.1.9.4 | 2015-09-07 | - paymentCertifyToken Length 수정 |
| 1.1.9.5 | 2015-10-06 | - 주문 예약 > payMode default 관련 문구 추가 |
| 1.1.9.6 | 2015-10-08 | - 기술지원 메일링 주소변경 (dl_payco_ts@nhnent.com) |
| 1.1.9.7 | 2015-10-12 | - 결제 인증 완료 > discountAmt 추가 |
| 1.1.9.8 | 2015-10-13 | - 결제 인증 완료 > pointAmt 추가 - 에러코드 update |
| 1.1.9.9 | 2015-11-03 | - inApp 적용가이드 > 일반신용카드관련 안내추가 - 카드사코드 > 카드발급사기준 설명추가 - 주문 및 결제 취소 처리 > 바로이체 취소관련 안내추가 |
| 1.1.9.10 | 2015-11-17 | - 결제승인 API > couponSettleInfo (쿠폰정보) 추가 - 주문 완료 통보 > couponSettleInfo (쿠폰정보) 추가 |
| 1.1.9.11 | 2015-11-24 | - inApp 적용가이드 > tel, about:blank 처리 관련 내용 추가 |

| | | |
|----------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <ul style="list-style-type: none"> - 무통장 입금 완료 통보 > orderCertifyKey 추가 - Iframe 적용가이드 > PAYCO 앱결제 위한 추가사항 |
| 1.1.9.12 | 2016-01-06 | <ul style="list-style-type: none"> - inApp 적용가이드 > 내용 수정 - 주문예약 > customUrlSchemeUseYn 삭제 - Iframe 적용가이드 > PAYCO 앱결제 위한 추가사항 제거 - 결제인증완료 > GET -> POST 로 변경 |
| 1.1.9.13 | 2016-01-18 | <ul style="list-style-type: none"> - 부록 > 판매자 부가정보 추가(sellerOptions) |
| 1.1.9.14 | 2016-01-19 | <ul style="list-style-type: none"> - 에러코드 update(오류메시지 세분화) - orderNo Length 수정(100->20) - 주문예약 API extraData > appUrl 컬럼추가 |
| 1.1.9.15 | 2016-01-27 | <ul style="list-style-type: none"> - 주문취소 APICancelPaymentDetails > cancelPaymentTradeNo 추가 - inApp 적용가이드 > third-party 쿠키이슈 대응 - pgAdmissionNo 자리수 증가 |
| 1.1.9.16 | 2016-03-02 | <ul style="list-style-type: none"> - 에러코드 update(결제시 포인트 잔액부족 알림 추가) |
| 1.1.9.17 | 2016-03-08 | <ul style="list-style-type: none"> - 부록 -> 판매자 부가정보(가맹점 이벤트용) 추가(sellerOptions) - 에러코드 update(자동결제 정보 조회시 메시지 추가) |
| 1.1.9.18 | 2016-03-17 | <ul style="list-style-type: none"> - 부록 -> 판매자 부가정보(가맹점 혜택 조건 검증) 추가 |
| 1.1.9.19 | 2016-03-23 | <ul style="list-style-type: none"> - 부록 -> 판매자 부가정보(가맹점 이벤트용) 파라미터 추가 (VALID_FIRST_PAYMENT) - |
| 1.1.9.20 | 2016-03-31 | <ul style="list-style-type: none"> - 주문예약 API > sellerOptions > remoteAreaDeliveryFeeYn 추가 - 결제 인증 완료 > totalRemoteAreaDeliveryFeeAmt 추가 - 주문상품상태 > 추가 |
| 1.1.9.21 | 2016-04-20 | <ul style="list-style-type: none"> - 구매하기 버튼 모바일 js 정식 지원 추가(추후예공예정 문구삭제) - 부록 -> 판매자 부가정보(추가배송비) 추가 |
| 1.1.9.22 | 2016-06-01 | <ul style="list-style-type: none"> - 주문예약 -> 가맹점할인금액 추가(totalSellerDiscountAmt) |
| 1.1.9.23 | 2016-06-27 | <ul style="list-style-type: none"> - 주문 결제 취소 가능여부 조회 -> 취소불가사유코드 추가 |
| 1.1.9.24 | 2016-08-02 | <ul style="list-style-type: none"> - 결제수단 코드 추가(96:충전금 환불) - 동일사용자 체크 여부(sameUserCheckYn) 추가 |
| 1.1.9.25 | 2016-09-23 | <ul style="list-style-type: none"> - 주문예약 API orderSheetUiType 타입 추가 |
| 1.1.9.26 | 2016-10-27 | <ul style="list-style-type: none"> - ExtraData > viewOptions 주문서 언어코드 추가 |
| 1.1.9.27 | 2016-10-31 | <ul style="list-style-type: none"> - SettleInfo 관련 Type 정보 오류 수정 |
| 1.1.9.28 | 2016-11-23 | <ul style="list-style-type: none"> - 결제인증완료 > POST -> GET 으로 변경(option POST) |
| 1.1.9.29 | 2016-12-12 | <ul style="list-style-type: none"> - cellphoneSettleInfo-> companyName 값이 삭제됨 |

| | | |
|----------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.1.9.30 | 2016-12-12 | - 에리코드 추가-> 1305 "진행중인 취소요청건이 있습니다." |
| 1.1.9.31 | 2017-03-09 | - 일부 문서 오류 수정 - 간편구매 -> 바로구매로 변경 - 주문완료통보 응답항목 현금영수증, 면과세 관련 내용 추가 - 무통장입금 완료통보 응답항목 현금영수증, 면과세 관련내용 추가 - 주문 및 결제취소 처리 응답항목 현금영수증, 면과세 관련내용 추가 |
| 1.1.9.32 | 2017-03-13 | - 주문예약 봉사료 항목 추가 > 봉사료 항목 추가 - 주문 및 결제취소 처리 > 봉사료 항목 추가 - 결제승인 API > 봉사료 항목 추가 - 주문 완료 통보 > 봉사료 항목 추가 - 무통장 입금 완료 통보 > 봉사료 항목 추가 |
| 1.1.9.33 | 2017-05-23 | - appUrl 오타수정 |
| 1.1.9.34 | 2017-06-23 | - INPUT, OUTPUT 예제에 대한 최신화 갱신 |
| 1.1.9.35 | 2017-11-02 | - 오류코드 상세사유 추가 - 취소결과 cancelYmdt 추가 |
| 1.1.9.36 | 2017-11-07 | - 주문 및 결제취소 처리 > 응답 항목 추가> cancelReceiptAmt - 주문완료 통보 > 응답항목 추가 > receiptAmt - 무통장 입금 완료 통보 > 응답항목 추가 > receiptAmt - 결제승인 API > 응답항목 추가 > receiptAmt - 현금영수증 관련 항목 Deprecated(2018 년 제거예정) cancelReceiptPaycoPointAmt cancelReceiptPaycoPointTaxfreeAmt cancelReceiptPaycoPointTaxableAmt cancelReceiptPaycoPointVatAmt cancelReceiptPaycoPointServiceAmt |

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

| | |
|---------------------------------|-----------|
| PAYCO 연동 가이드 | 1 |
| 1. 개요 | 13 |
| 1.1 들어가며 | 14 |
| 1.1.1 PAYCO 주문연동 방식 | 14 |
| 1.1.2 PAYCO 결제 모드 방식 | 14 |
| 1.1.3 사전 준비사항 | 15 |
| 1.2 PAYCO 결제연동 방식 | 17 |
| 1.2.1 결제모드 PAY1 타입 | 17 |
| 1.2.2 결제모드 PAY2 타입 | 19 |
| 1.3 PAYCO 결제연동방식 샘플 | 21 |
| 1.3.1 결제모드 PAY1 타입 | 21 |
| 1.3.2 결제모드 PAY2 타입 | 22 |
| 2. PAYCO 구매하기 버튼 적용 방법 | 25 |
| 2.1 PAYCO 구매하기 버튼 삽입 | 26 |
| 2.1.1 PAYCO 구매하기 버튼 종류 | 26 |
| 2.1.2 버튼 이미지 삽입을 위한 자바스크립트 | 28 |
| 2.2 PAYCO 구매하기 버튼 클릭 시 처리 방법 설명 | 31 |
| 3. PAYCO 연동하기 | 35 |
| 3.1 PAYCO 연동 API | 36 |
| 3.1.1 연동 API 목록 | 36 |
| 3.1.2 PAYCO 연동 API 주소 | 37 |
| 3.1.3 PAYCO 연동 API 공통 규약 | 37 |
| 3.1.4 PAYCO 연동에 필요한 KEY 정보 | 38 |
| 3.2 PAYCO 연동 API 상세 명세 | 40 |
| 3.2.1 주문 예약 | 40 |

| | | |
|------------|---------------------------------------|------------|
| 3.2.2 | 주문 상품 상태 변경 | 46 |
| 3.2.3 | 주문 및 결제 취소 가능여부 조회 | 47 |
| 3.2.4 | 주문 및 결제 취소 처리 | 50 |
| 3.2.5 | 마일리지 적립 취소 | 54 |
| 3.2.6 | 가맹점 별 연동 키 유효성 체크 | 55 |
| 3.2.7 | 결제승인 API (결제모드 PAY2타입) | 57 |
| 3.2.8 | 결제수단 및 코드조회 API | 66 |
| 3.3 | 외부 가맹점 API 상세 | 77 |
| 3.3.3 | 추가 배송비 조회 | 77 |
| 3.3.4 | 주문 완료 통보 (결제모드 PAY1타입) | 78 |
| 3.3.5 | 결제 인증 완료 (결제모드 PAY2타입) | 86 |
| 3.3.6 | 무통장 입금 완료 통보 | 87 |
| 3.4 | 유의사항 | 93 |
| 3.5 | 코드 정의 | 94 |
| 3.5.1 | 주문 유형(=결제유형) 파트너센터 주문통계페이지 주문유형별 통계제공 | 94 |
| 3.5.2 | 주문상품상태 | 94 |
| 3.5.3 | 주문채널 (= 결제채널) | 94 |
| 3.5.4 | 결제유형 | 95 |
| 3.5.5 | 결제수단 | 95 |
| 3.5.6 | 상품타입 | 95 |
| 3.5.7 | 과세타입 | 95 |
| 3.5.8 | 카드사코드 | 95 |
| 3.5.9 | 통신사코드 | 96 |
| 3.5.10 | 은행코드 | 97 |
| 3.5.11 | 에러코드 | 98 |
| 3.6 | 영수증 | 111 |
| 3.6.1 | PAYCO 영수증 주소 | 111 |
| 3.6.2 | PAYCO 영수증 노출정의 | 111 |
| 3.7 | 부가설명 | 112 |
| 3.7.1 | inApp 적용가이드 | 112 |
| 3.7.2 | Iframe 적용가이드 | 114 |
| 3.8 | 부록 | 115 |
| 3.8.1 | 판매자 부가정보(게임사 결제용) | 115 |
| 3.8.2 | 판매자 부가정보(가맹점 이벤트용) | 116 |
| 3.8.3 | 판매자 부가정보(가맹점 혜택 조건 검증) | 117 |
| 3.8.4 | 판매자 부가정보(추가배송비) | 118 |
| 3.8.5 | 판매자 부가정보(결제 사용자 체크) | 119 |

표 및 그림 목록

표 목록

| | |
|------------------------------------------------|----|
| • 표 2-1 PAYCO 구매하기 버튼 종류 | 26 |
| • 표 2-2 자바스크립트 파라미터 설명 | 29 |
| • 표 3-1 PAYCO 제공 API 목록 | 36 |
| • 표 3-2 가맹점에서 구현해야 할 API 목록 | 36 |
| • 표 3-3 Default format | 37 |
| • 표 3-4 주문 예약 API > INPUT | 40 |
| • 표 3-5 주문 예약 API > OUTPUT | 45 |
| • 표 3-6 주문 상품 상태 변경 API > INPUT | 46 |
| • 표 3-7 주문 상품 상태 변경 API > OUTPUT | 46 |
| • 표 3-8 주문 결제 취소 가능여부 조회 API > INPUT | 48 |
| • 표 3-9 주문 결제 취소 가능여부 조회 API > OUTPUT | 48 |
| • 표 3-10 주문 결제 취소 처리 API > INPUT | 50 |
| • 표 3-11 주문 결제 취소 처리 API > OUTPUT | 51 |
| • 표 3-12 마일리지 적립 취소 > INPUT | 54 |
| • 표 3-13 마일리지 적립 취소 > OUTPUT | 55 |
| • 표 3-14 가맹점 별 코드 값 사용가능 여부 > INPUT | 55 |
| • 표 3-15 가맹점 별 코드 값 사용가능 여부 > OUTPUT | 56 |
| • 표 3-16 결제승인 API > INPUT | 57 |
| • 표 3-17 결제승인 API > OUTPUT | 58 |
| • 표 3-18 결제수단 및 코드조회 API > INPUT | 66 |
| • 표 3-19 결제수단 및 코드조회 API > OUTPUT | 66 |
| • 표 3-20 추가 배송비 조회 > INPUT | 77 |
| • 표 3-21 추가 배송비 조회 > OUTPUT | 78 |
| • 표 3-22 ServiceUrl 로 전달되는 정보 | 79 |
| • 표 3-23 returnUrl 로 전달되는 정보 | 86 |
| • 표 3-24 NonBankbookDepositInformUrl 로 전달되는 정보 | 87 |

그림 목록

| | |
|--------------------------------|----|
| • 그림 1-1 PAYCO 처리 순서도 (PAY1타입) | 17 |
| • 그림 1-2 PAYCO 처리 순서도 (PAY2타입) | 19 |

1. 개요

이 장에서는 PAYCO 구매하기 연동의 기본 처리 절차를 간략하게 소개한다.

1. 개요

1.1 들어가며

1.1.1 PAYCO 주문연동 방식

| 주문연동 항목 | 간편결제형 | 바로구매형 |
|------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| 개요 | 간편결제수단(신용카드/휴대폰/바로이체) 결제 인증/승인을 제공함 결제수단 목록에 "Payco간편결제" 노출 | 간편결제 기능에 추가로, PAYCO 회원의 배송지 주소록을 제공함 (배송지 주소록이 Payco에서 통합관리됨) 주문버튼 영역에 "PAYCO구매하기 버튼"노출 |
| Flow요약 | Payco주문창을 띄우고 결제를 진행함. 결제완료시 결제성공여부를 가맹점에 전달함. | PAYCO 구매하기 버튼 클릭시, Payco주문창을 띄우고 결제를 진행함. Payco주문창에서 배송지를 입력받아, 결제정보와 <u>배송지정보</u> 를 가맹점에 전달함 |
| 결제모드방식 | PAY1, PAY2 방식 지원 | PAY1, PAY2 방식 지원 |

1.1.2 PAYCO 결제 모드 방식

| 결제모드 항목 | PAY1 | PAY2 |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 개요 | PAYCO결제창에서 인증/승인을 모두 처리하는 결제방식. 결제성공시 가맹점 ServiceUrl을 호출하여, 결제정보를 전달함.(결제전송누락을 막기위해 서버간 통신을 이용하여 결제내역 전송) -가맹점 URL(ServiceUrl)호출이 실패한경우를 대비하여 PAYCO에서주기적으로 재호출하는 기능 제공함. | 인증과 승인을 분리하여 제공하는 방식. PAYCO결제창을 통하여, 결제수단 인증을 하며 이후 인증 토큰값을 가맹점에 전달함. 가맹점에서는 해당 토큰을 이용하여 PAYCO승인 API를 호출하여 결제요청함. |
| 매뉴얼 링크 | 결제모드 PAY1 타입 | 결제모드 PAY2 타입 |

| | | |
|--|-------------------------|---------------------|
| | 인증/승인 분리가 필요하지 않은 경우 이용 | 인증/승인 분리가 필요한 경우 이용 |
|--|-------------------------|---------------------|

1.1.3 사전 준비사항

- 주문연동방식(바로구매형/간편결제형) 과 결제 연동 방식(PAY1, PAY2) 에 대한 결정
- 가맹점 방화벽 오픈 요청
 - PAYCO 결제서비스를 사용하기 위해서는 가맹점 측의 서버 IP에 대해 방화벽 오픈요청을 해야 결제연동이 가능합니다.
(외부가맹점 URL이 80,443 port 외 사용시 사전공유 부탁드립니다.)
- 가맹점코드발급 요청
 - 가맹점코드(sellerKey) , 상점 ID(cpId), 상품 ID(productID) 발급이 필요합니다.
[\[PAYCO 연동에 필요한 KEY 정보 참조\]](#)
 - 가맹점코드 발급시 가맹점사이트(PC, MOBILE), 약관URL 정보가 필요합니다.
- 결제연동시 리턴되는 PAYCO주문번호(orderNo)와 주문인증키(orderCertifyKey)에 대해 가맹점 DB 저장에 필요합니다.
- 관련 사이트
 - <http://www.payco.com> - 페이코 서비스사이트
 - <https://partner.payco.com> - 페이코 가맹점사이트
 - <http://devcenter.payco.com> - 페이코 결제 데모사이트 (payco/payco1234)
 - <https://demo-partner.payco.com> - 페이코 가맹점 데모사이트
- 테스트정보

| | 간편결제형 | 바로구매형 |
|------------|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| sellerKey | S0FSJE | S0FSJE |
| cpId | PARTNERTEST | PARTNERTEST |
| productID | PROD_EASY | PROD_CHK (일반상품) DELIVERY_PROD (배송비상품구분용도) |
| serviceUrl | https://alpha-api-bill.payco.com/test/surl/ok | https://alpha-api-bill.payco.com/test/surl/ok |

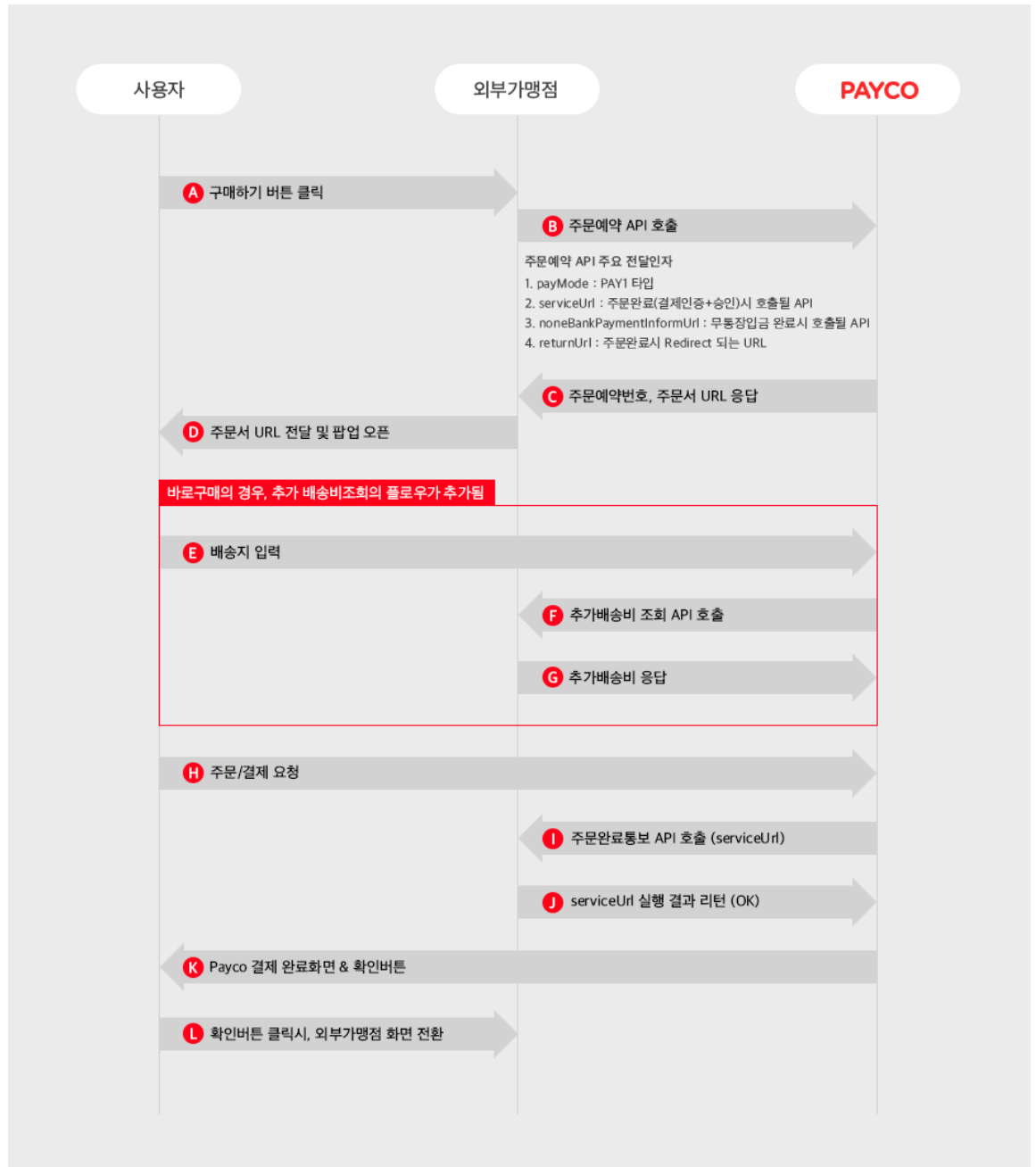
1. 개요

| | | |
|-------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Mock-up) | 항상 OK가 리턴됨 (PAY1결제모드만 사용) | 항상 OK가 리턴됨 (PAY1결제모드만 사용) |
| 배송비 조회 API (Mock-up) | 사용안함 | https://alpha-api-bill.payco.com/test/getRemoteAreaDeliveryFee ↳ 항상 3,000원 고정으로 리턴 |

1.2 PAYCO 결제연동 방식

1.2.1 결제모드 PAY1 타입

[결제인증과 승인 통합모드]



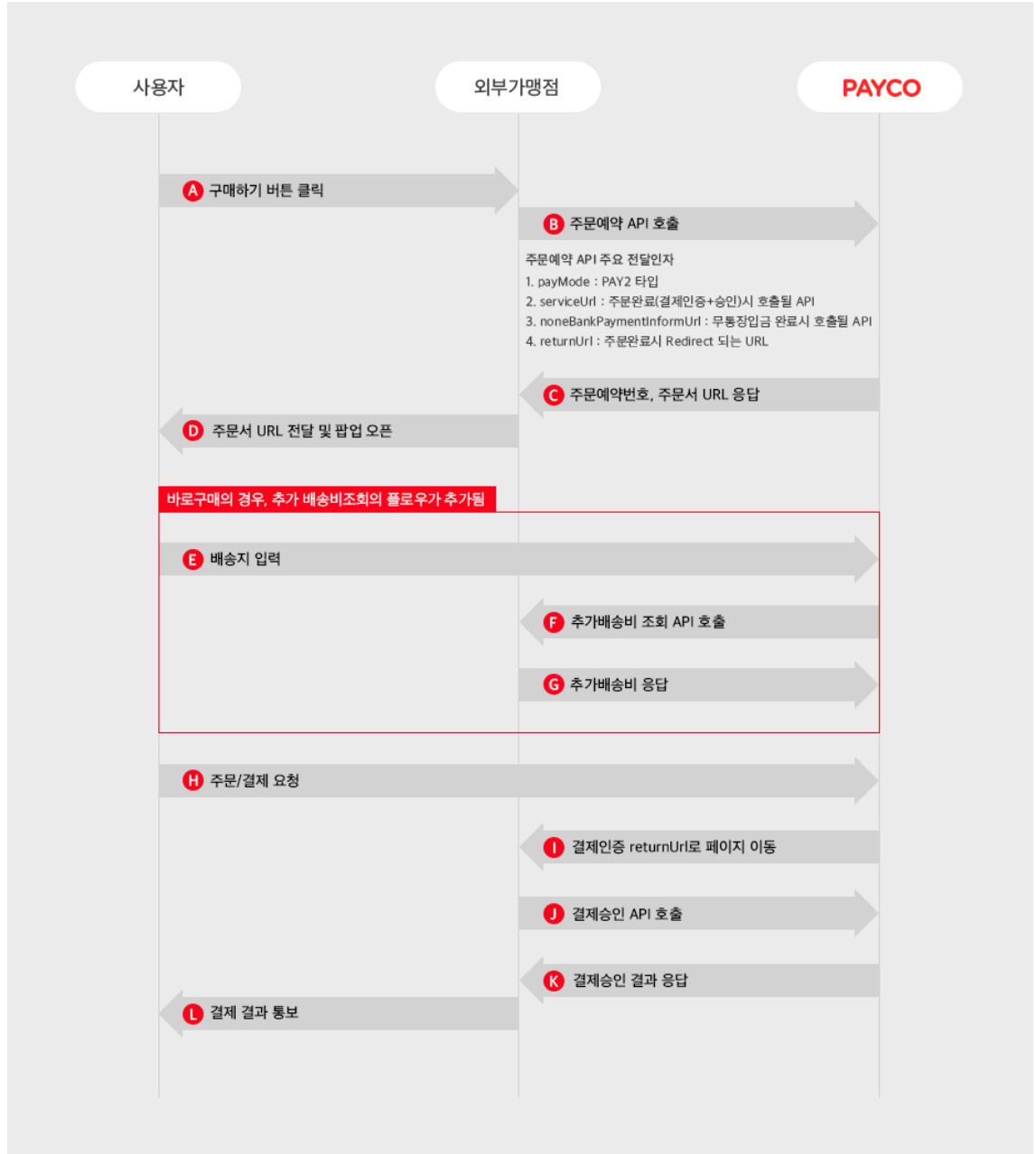
- 그림 1-1 PAYCO 처리 순서도 (PAY1타입)

1. 개요

- ① 외부 가맹점에서는 Payco에서 제공하는 구매하기 버튼 Javascript plugin을 이용하여 구매하기 버튼을 연동해야 한다. A) [\[2. PAYCO 구매하기 버튼 연동\]](#)
- ② 사용자가 구매하기 버튼을 클릭하면 외부가맹점에서 해당 주문 건에 대해 Payco 주문예약 API를 호출한다. B) [\[3.2.1 주문예약 API\]](#)
 - ✓ [주문예약 API 주요 전달인자 부가설명]
 - ✓ serviceUrl : 주문완료 또는 결제완료 시 Payco에서 이 URL(API)를 호출하므로, 외부가맹점에서는 주문완료 또는 결제완료 후 처리되어야 하는 내용을 Payco에서 제공하는 연동 API 명세에 맞춰 구현해야 한다. (하단 [외부가맹점 구현 API Spec](#) 참조)
 - ✓ nonBankbookDepositInformUrl : 무통장 주문완료 또는 무통장 결제완료 시 Payco에서 이 URL(API)를 호출하므로, 외부가맹점에서는 무통장 주문완료 또는 무통장 결제완료 후 처리되어야 하는 내용을 Payco에서 제공하는 연동 API 명세에 맞춰 구현해야 한다.
 - ✓ returnUrl : 주문 또는 결제완료 후 redirect될 외부가맹점의 페이지 URL.
- ③ Payco는 주문예약 API 요청을 받아 주문예약을 처리하고, 외부가맹점에게 주문 예약 번호와 주문서 URL을 전달한다. C)
- ④ 외부가맹점은 전달받은 주문서 URL을 브라우저로 전달하고 외부가맹점은 전달받은 주문서 URL(간편결제의 경우 결제팝업) 로 주문서/결제 팝업을 띄운다. (D,E)
- ⑤ (바로구매형의 경우에만 해당) PAYCO결제창에서 배송지입력후, 가맹점으로 추가배송비 조회 [\[3.3.3 추가배송비조회 API\]](#)
- ⑥ 팝업에서 사용자가 주문 또는 결제 완료한다.(F)
- ⑦ 주문예약 시 가맹점으로부터 전달받은 serviceURL을 호출하고, 가맹점으로부터 결과값을 전달받는다. [\[3.3.4 주문 완료 통보 API연동\]](#)
- ⑧ PAYCO완료화면에서 확인버튼 클릭시 returnUrl로 화면 전환을 한다.(J)

1.2.2 결제모드 PAY2 타입

[결제인증과 승인 분리모드]



- 그림 1-2 PAYCO 처리 순서도 (PAY2타입)

1. 개요

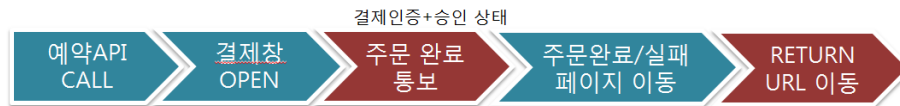
- ① 외부 가맹점에서는 Payco에서 제공하는 구매하기 버튼 Javascript plugin을 이용하여 구매하기 버튼을 연동해야 한다. [\[2. PAYCO 구매하기 버튼 연동\]](#)
- ② 사용자가 구매하기 버튼을 클릭하면 외부가맹점에서 해당 주문 건에 대해 Payco 주문예약 API를 호출한다. - A), B) [\[3.2.1 주문예약 API\]](#)
 - ✓ [주문예약 API 주요 전달인자 부가설명]
 - ✓ returnUrl : 결제인증 후 주문완료 또는 결제완료 시 Payco에서 이 URL(API)를 호출하므로, 외부가맹점에서는 주문완료 또는 결제완료 후 처리되어야 하는 내용을 Payco에서 제공하는 연동 API 명세에 맞춰 구현해야 한다. (하단 [외부가맹점 구현 API Spec](#) 참조)
 - ✓ nonBankbookDepositInformUrl : 무통장 주문완료 또는 무통장 결제완료 시 Payco에서 이 URL(API)를 호출하므로, 외부가맹점에서는 무통장 주문완료 또는 무통장 결제완료 후 처리되어야 하는 내용을 Payco에서 제공하는 연동 API 명세에 맞춰 구현해야 한다.
 - ✓ returnUrl : 주문 또는 결제완료 후 redirect될 외부가맹점의 페이지 URL.
- ③ Payco는 주문예약 API 요청을 받아 주문예약을 처리하고, 외부가맹점에게 주문 예약 번호와 주문서 URL을 전달한다. - C)
- ④ 외부가맹점은 전달받은 주문서 URL(간편결제의 경우 결제팝업) 로 주문서/결제 팝업을 띄운다. - D~E)
- ⑤ (바로구매형의 경우에만 해당) PAYCO결제창에서 배송지입력후, 가맹점으로 추가배송비 조회 [\[3.3.3 추가배송비조회 API\]](#)
- ⑥ 팝업에서 사용자가 결제 완료하면 주문예약 시 가맹점으로부터 전달받은 returnUrl로 결제인증 정보를 전달한다 (G) [\[3.3.5 결제인증 완료\(PAY2\)\]](#)
- ⑦ 외부가맹점은 결제인증정보를 참고하여 결제완료처리가 필요한경우 결제승인API를 호출하여 피드백을 전달받아 주문완료처리를 한다. (H~J) [\[3.2.7 결제승인API연동\(PAY2\)\]](#)

1.3 PAYCO 결제연동방식 샘플

1.3.1 결제모드 PAY1 타입



[결제인증과 승인 통합모드]



[표 3-4참조]

[표 3-18참조]

➤ 주문예약 API > 샘플 CALL(curl이 설치된 장비에서 copy후 실행)

● request

```

curl -H "Content-Type: application/json" -d '\
{ \
  "sellerKey": "DEMO", \
  "sellerOrderReferenceKey": "TESTORDERKEY00000001", \
  "totalDeliveryFeeAmt": "0", \
  "totalPaymentAmt": "3000", \
  "serviceUrl": "https://alpha-api-bill.payco.com/test/surl/ok", \
  "returnUrl": "http://demo-bill.payco.com:11111/result.nhn", \
  "orderMethod": "EASYPAY_F", \
  "orderProducts": [ \
    { \
      "cpId": "DEMO", \
      "productId": "PROD_CHK", \
      "productAmt": "1500", \
      "productPaymentAmt": "1500", \
      "sortOrdering": 1, \
      "productName": "BAG", \
      "orderQuantity": 1, \
      "sellerOrderProductReferenceKey": "TESTPRODKEY00000001" \
    } \
  ] \
} \
' https://alpha-api-bill.payco.com/outseller/order/reserve
  
```

● response

```

{
  "result":
  {
    "reserveOrderNo": "201503172000525469",
    "orderSheetUrl": "https://alpha-bill.payco.com/easyLogin/201503172000525469"
  },
  "code": 0,
  "message": "success"
}
  
```

1.3.2 결제모드 PAY2 타입

[결제인증과 승인 분리모드]



[\[표 3-4참조\]](#)

[\[표 3-20참조\]](#)

[\[표 3-21참조\]](#)

➤ 주문예약 API > 샘플 CALL(curl이 설치된 장비에서 copy후 실행.)

● request

```
curl -H "Content-Type: application/json" -d '\
{\
  "sellerKey": "DEMO", \
  "sellerOrderReferenceKey": "TESTORDERKEY00000001", \
  "totalDeliveryFeeAmt": "0", \
  "totalPaymentAmt": "3000", \
  "payMode": "PAY2", \
  "returnUrl": "http://demo-bill.payco.com:11111/result.nhn", \
  "orderMethod": "EASYPAY_F", \
  "orderProducts": [ \
    { \
      "cpId": "DEMO", \
      "productId": "PROD_CHK", \
      "productAmt": "1500", \
      "productPaymentAmt": "1500", \
      "sortOrdering": 1, \
      "productName": "BAG", \
      "orderQuantity": 1, \
      "sellerOrderProductReferenceKey": "TESTPRODKEY00000001" \
    } \
  ] \
}\
' https://alpha-api-bill.payco.com/outseller/order/reserve
```

● response

```
{
  "result":
  {
    "reserveOrderNo": "201503172000525469",
    "orderSheetUrl": "https://alpha-bill.payco.com/easyLogin/201503172000525469"
  },
  "code": 0,
  "message": "success"
}
```

➤ 결제 승인 API

● request

```
curl -H "Content-Type: application/json" -d \
{ \
  "sellerKey": "DEMO", \
  "reserveOrderNo": "201503172000525469", \
  "sellerOrderReferenceKey": "88888", \
  "paymentCertifyToken": "9FAFF0967482F8DE25AB92716C073842335235", \
  "totalPaymentAmt": "3000" \
} \
```

● response

```
{
  "result":
  {
    "reserveOrderNo": "201408211000000720",
    "orderNo": "201408211000000721",
    "sellerOrderReferenceKey": "201408212212505952",
    "orderChannel": "PC",
    "totalDeliveryFeeAmt": 2500.0,
    "totalRemoteAreaDeliveryFeeAmt": 0.0,
    "totalPaymentAmt" : 119000.0,
    "paymentCompletionYn" : "N",
    "memberName" : "홍*동",
    "memberEmail" : "ho*@nhnent.com",
    "deliveryPlace": {
      "recipient": "이진희",
      "address1": "경기도 성남시 분당구 삼평구",
      "address2": "629",
      "zipcode": "123123",
      "telephone": "02-0111-2222",
      "deliveryMemo": "경비실에 맡겨주세요"
    }
  },
  "code": 0,
  "message": "success"
}
```

1. 개요

```
"paymentDetails": [{
  "paymentTradeNo" : "20140821100000022",
  "paymentMethodName": "무통장입금",
  "paymentMethodCode": "02",
  "paymentAmt": 109000.0,
  "tradeYmdt": "20140821221250",
  "pgAdmissionNo": "12980185",
  "pgAdmissionYmdt": "20140821221255",
  "easyPaymentYn": "N",
  "nonBankbookSettleInfo": {
    "accountHolderName": "(주)지누스포츠 (팍슈즈)",
    "bankName": "국민",
    "accountNo": "465101-01-217098",
    "expirationYmd": "20140901"
  }
}, {
  "paymentTradeNo" : "20140821100000022",
  "paymentMethodName": "PAYCO포인트",
  "paymentMethodCode": "98",
  "paymentAmt": 10000.0,
  "tradeYmdt": "20140821221250",
  "pgAdmissionNo": "20140821100000023",
  "pgAdmissionYmdt": "20140821221255",
  "easyPaymentYn": "N"
}],
"orderProducts" : [{
  "productId" : "PS_TOASTCOMC1",
  "cpId" : "PS_TOASTCO",
  "orderProductNo" : "20140821100000022",
  "sellerOrderProductReferenceKey" : "1",
  "orderProductStatusCode" : "PAYED",
  "orderProductStatusName" : "결제완료"
},
{
  "productId" : "PS_TOASTCOMC2",
  "cpId" : "PS_TOASTCO",
  "orderProductNo" : "20140821100000023",
  "sellerOrderProductReferenceKey" : "2",
  "orderProductStatusCode" : "PAYED",
  "orderProductStatusName" : "결제완료"
}]
}
```


2. PAYCO 구매하기 버튼 적용 방법



이 장에서는 PAYCO 구매하기 버튼 삽입 방법과 사용자가 “구매하기” 버튼 클릭 시 이벤트 처리 방법에 대해 설명한다.

2.1 PAYCO 구매하기 버튼 삽입



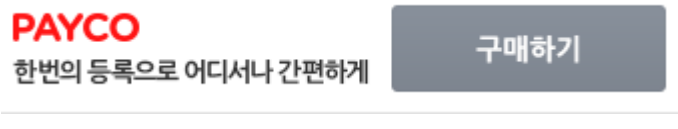

2.1.1 PAYCO 구매하기 버튼 종류

PAYCO 구매하기 버튼은 주문유형(결제유형)에 따라 버튼 타입이 존재한다.

- CHECKOUT : 바로구매형
- EASYPAY : 간편결제형

• 표 2-1 PAYCO 구매하기 버튼 종류

| 주문유형 | 버튼 타입 | 크기 | 버튼 이미지 |
|----------|-------|--------|--------------------------------------------------------------------------------------|
| CHECKOUT | A1 | 277X70 |  |
| | A2 | 277X70 |  |
| | A3 | 277X70 |  |
| | A4 | 277X70 |  |
| | A5 | 277X70 |  |
| | A6 | 277X70 |  |
| CHECKOUT | B1 | 338X84 |  |

| 주문유형 | 버튼 타입 | 크기 | 버튼 이미지 |
|----------|-------|--------|--------------------------------------------------------------------------------------|
| | B2 | 338X84 |  |
| | B3 | 338X84 |  |
| | B4 | 338X84 |  |
| | B5 | 338X84 |  |
| | B6 | 338X84 |  |
| CHECKOUT | C1 | 296X84 |  |

2. PAYCO 구매하기 버튼 적용 방법

| 주문유형 | 버튼 타입 | 크기 | 버튼 이미지 |
|---------|-------|--------|---------------------------------------------------------|
| | C2 | 296X84 | <div>한번의 등록으로 어디서나 간편하게</div> <div>PAYCO 구매하기</div> |
| | C3 | 296X84 | <div>한번의 등록으로 어디서나 간편하게</div> <div>PAYCO 구매하기</div> |
| | C4 | 296X84 | <div>한번의 등록으로 어디서나 간편하게</div> <div>PAYCO 구매하기</div> |
| | C5 | 296X84 | <div>한번의 등록으로 어디서나 간편하게</div> <div>PAYCO 구매하기</div> |
| | C6 | 296X84 | <div>한번의 등록으로 어디서나 간편하게</div> <div>PAYCO 구매하기</div> |
| EASYPAY | A1 | | PAYCO 간편결제 |
| | A2 | | PAYCO 간편결제 |

2.1.2 버튼 이미지 삽입을 위한 자바스크립트

- 버튼 이미지 삽입을 위한 자바 스크립트는 아래와 같다.

- 제공하는 자바스크립트 URL 을 추가한다.

PC 버전 : <https://static-bill.nhnent.com/payco/checkout/js/payco.js>

모바일 : https://static-bill.nhnent.com/payco/checkout/js/payco_mobile.js

- 노출하고자 하는 버튼 타입 및 옵션을 파라미터로 하여 자바스크립트 함수를 호출한다

```
<!DOCTYPE html>
<html lang="ko">
<head>
<title>Payco buy button sample</title>
<script type="text/javascript" src="https://static-bill.nhnent.com/payco/checkout/js/payco.js" charset="UTF-8"></script>
<script type="text/javascript">
</head>
<body>
<div id="payco_type1"></div>
<script type="text/javascript">
//<![CDATA[
    Payco.Button.register({
        SELLER_KEY : '0988BA',           // 가맹점 코드
        ORDER_METHOD : "CHECKOUT",       // 주문유형 : CHECKOUT / EASYPAY
        BUTTON_TYPE : "A1",              // 버튼타입 선택 (A1,A2,..B1,.. C2.....)
        BUTTON_HANDLER : order,          // 구매하기 버튼 이벤트 Handler 함수 등록
        BUTTON_HANDLER_ARG : ['param1', 'param2'], // handler 함수 argument
        DISPLAY_PROMOTION : "Y",         // 이벤트 문구 출력 여부
        DISPLAY_ELEMENT_ID : "payco_type1", // 노출될 element id
        "" : ""
    });
}
//]]>
</script>
</body>
</html>
```

• 표 2-2 자바스크립트 파라미터 설명

| 컬럼명 | 설명 | 선택가능 값 |
|--------------|----------|--------------------------------------|
| SELLER_KEY | 가맹점 코드 | 파트너센터에서 발급받은 가맹점 코드 |
| ORDER_METHOD | 주문 유형 | CHECKOUT : 바로구매형, EASYPAY : 간편결제형 |
| BUTTON_TYPE | 버튼 타입 선택 | A1, A2 ... |

2. PAYCO 구매하기 버튼 적용 방법

| 컬럼명 | 설명 | 선택가능 값 |
|--------------------|--------------------------------------------|-----------------------------------------|
| BUTTON_HANDLER | 구매하기 버튼 클릭 시 호출할 이벤트 Handler 함수 (callback) | 자바스크립트 함수명 ex) order(p1,p2) => order |
| BUTTON_HANDLER_ARG | 구매하기 버튼 클릭 시 호출할 이벤트 Handler 함수 인자(필요 시) | ex) order(p1,p2) => [p1, p2] |
| DISPLAY_PROMOTION | 프로모션 문구 출력 여부 | Y, N (기본값 Y) |
| DISPLAY_ELEMENT_ID | 버튼 노출 위치의 Element ID | 자체적으로 정의한 값 |

2.2 PAYCO 구매하기 버튼 클릭 시 처리 방법 설명

1. 사용자가 구매하기 버튼을 클릭하면 가맹점은 구매상품의 색상, 타입 등 필수 옵션 사항을 선택 했는지 체크한다.
2. 사용자가 옵션을 선택 했거나, 선택해야 할 옵션이 없는 상품의 경우 주문 정보 등록을 위한 처리를 진행한다 ([PAYCO 주문 예약 API](#) 이용)
3. 주문 정보 등록 후 응답 받은 URL 로 주문서 페이지를 띄운다.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html lang="ko">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Payco buy button sample</title>
  <script type="text/javascript" src="https://static-bill.nhnent.com/payco/checkout/js/payco.js" charset="UTF-8"></script>
  <script type="text/javascript">
    function order() {

      // ===== 1. 선택 박스 필수 옵션을 체크 함 =====/

      if (document.getElementById('frm').option.value == "") {
        alert('옵션을 선택해주세요');
        return false;
      }

      if (document.getElementById('frm').buy_cnt.value == 0) {
        alert('구매수량을 선택해주세요');
        return false;
      }

      /*
      1. payco로 주문 정보를 등록 처리를 하는 가맹점 페이지로 이동
      2. 해당 페이지에서 페이코 주문 예약 API를 이용하여 주문 정보를 등록
      3. 리턴받은 주문서 페이지로 이동 처리한다.
      */

      // ( test code ) 응답받은 json data라고 가정.
      var jsonData = {
        "result": {
          "reserveOrderNo": "201408121000000273",
          "orderSheetUrl": "https://bill.toast.com/orderLogin/201412101000009544"
        },
      },
```

2. PAYCO 구매하기 버튼 적용 방법

```
        "message" : "success",
        "code" : 0
    };

    // 성공 시 주문서 URL 팝업 ( 모바일의 경우 redirect )
    if ( jsonData.code == 0 ) {
        var orderurl = jsonData.result.orderSheetUrl;
        window.open(orderurl, 'popupPayco','top=100, left=300, width=727px, height=512px, resizable=no,
scrollbars=yes');
    } else {
        // 실패 처리
    }

    }
</script>
</head>
<body>
<h1> CHECKOUT TYPE</h1><hr/>
    <form id="frm" action="">
        <div id="product">
            <ul>
                <li>상품명 : 옷</li>
                <li><label>색상 : </label> <select id="option">
                    <option value="">-- 옵션선택 --</option>
                    <option value="red">red</option>
                    <option value="blue">blue</option>
                </select> <br /> </li>
                <li><label>구매수량 : </label> <select id="buy_cnt">
                    <option value="0">-- 구매수량 --</option>
                    <option value="1">1</option>
                    <option value="2">2</option>
                </select> </li>
            </ul>
            <div id="payco_btn_type_A1"> </div>
        </div>
    </form>

    <script type="text/javascript">
        <![CDATA[
            Payco.Button.register({
                SELLER_KEY : 'DEMO' ,
                ORDER_METHOD : "CHECKOUT",           // 주문유형 : CHECKOUT - 바로구매형 / EASYPAY - 간편결제형
                BUTTON_TYPE : "A1",                  // 버튼타입 선택 (A1,A2)
                BUTTON_HANDLER : order,              // 구매하기 버튼 이벤트 Handler 함수 등록
                DISPLAY_PROMOTION : "Y",              // 이벤트 문구 출력 여부
            });
        ]>
    </script>
```



```
        DISPLAY_ELEMENT_ID : "payco_btn_type_A1", // 노출될 element id
        "" : ""
    });
    //]]>
</script>

<br /><br/>

<h1> EASYPAY TYPE</h1><hr/>
<div id="payco_type2">
    <script type="text/javascript">
        <![CDATA[
            Payco.Button.register({
                SELLER_KEY : 'DEMO' ,
                ORDER_METHOD : "EASYPAY", // 주문유형 : CHECKOUT - 바로구매형 / EASYPAY - 간편결제형
                BUTTON_TYPE : "A1", // 버튼타입 선택 (T1,T2)
                DISPLAY_PROMOTION : "Y" // 이벤트 문구 출력 여부
            });

            //]]>
        </script>
    </div>
</body>
</html>
```

3. PAYCO 연동하기

이 장에서는 PAYCO 연동 API 목록 및 API 별 상세 명세에 대해 설명한다.

3.1 PAYCO 연동 API

3.1.1 연동 API 목록

• 표 3-1 PAYCO 제공 API 목록

| API명 | 간편결제형 | | 바로구매형 | | 기타 참고 |
|-------------------------------------|-------|------|-------|------|------------------|
| | PAY1 | PAY2 | PAY1 | PAY2 | |
| 주문예약 | ○ | ○ | ○ | ○ | |
| 결제승인 | X | ○ | X | ○ | PAY2 결제연동 방식만 사용 |
| 주문상태변경 | ▲ | ▲ | ▲ | ▲ | |
| 주문 및 결제 취소 가능 여부 조회 | ▲ | ▲ | ▲ | ▲ | |
| 주문 및 결제 취소 처리 | ▲ | ▲ | ▲ | ▲ | |
| 마일리지 적립 취소 | ▲ | ▲ | ▲ | ▲ | |
| 연동 키 유효성 체크 | ▲ | ▲ | ▲ | ▲ | |

✓ 해당 정보는 PAYCO 에서 제공하는 API 목록입니다..

• 표 3-2 가맹점에서 구현해야 할 API 목록

| API명 | 간편결제형 | | 바로구매형 | | 기타 참고 |
|-----------------------------|-------|------|-------|------|-------------------------------------------------------------------------------------------------------------------------------------------|
| | PAY1 | PAY2 | PAY1 | PAY2 | |
| 주문 완료 통보 | ○ | X | ○ | X | PAY1 결제연동 방식만 사용. 주문완료 이후 가맹점으로 결과를 통지할 URL. 주문 예약 시 serviceUrl에 해당 URL 정보를 전달. |
| 추가 배송비 조회 | X | X | ○ | ○ | 추가배송비를 조회하는 API. PAYCO결제창에서 배송지입력 후, 추가 배송비 여부를 확인하기 위하여 가맹점에서 제공하는 추가 배송비 조회API를 호출함. (URL은 PAYCO어드민에서 관리되므로, 가맹점 코드 발급시 URL을 전달함) |
| 무통장 입금 완료통보 | ▲ | ▲ | ▲ | ▲ | 무통장입금시, 입금결과 통지할 URL 주문예약 시 nonBankbookDepositInformUrl에 해당 정보를 전달 (무통장입금 결제수단을 사용하는 경우 필수) |

- ✓ 해당 정보는 가맹점에서 구현해야 할 API 목록입니다...

3.1.2 PAYCO 연동 API 주소

| 용도 | 도메인 |
|-------|-----------------------------------------------------------------------------------|
| 테스트 용 | https://alpha-api-bill.payco.com/ |
| 서비스 용 | https://api-bill.payco.com |

➤ 도메인 하위 주소는 API 상세 명세 참조.

3.1.3 PAYCO 연동 API 공통 규약

- Content-type : application/json
- Protocol : Payco API 호출 시 https 프로토콜(443포트)을 사용해야 합니다. HTTP는 제공하지 않습니다.

INPUT (Request)

method=POST & Json 타입으로 값을 전달받습니다.

OUTPUT (Response)

API Response는 사전에 예고되지 않은 field가 추가될 수 있습니다.

추가되더라도 오류가 발생되지 않도록 연동 개발 부탁드립니다

✓ code

code=0 이면 성공

code>0 이면 실패 [\[에러코드 참고\]](#)

✓ result

각 API에 따라 다양한 Type으로 사용이 됩니다.

✓ pgErrorCode

기관 오류코드(기관에서 오류가 발생한 경우 해당 오류코드가 전달됩니다.)

- 표 3-3 Default format

3. PAYCO 연동하기

| | Column | Type | Description |
|---|-------------|--------|------------------------------------------------------------|
| 1 | code | Number | 성공여부 |
| 2 | message | String | 성공/실패 message |
| 3 | result | object | API 별 응답 result Json 의 배열 형태로 결과를 반환. 오류 시에는 포함되지 않음 |
| 4 | pgErrorCode | String | 기관 오류코드 기관 오류 시에만 전달됨 |

- **Sample output**

```
{
  "result": {
    ...
  },
  "message": "success",
  "code": 0
}
```

3.1.4 PAYCO 연동에 필요한 KEY 정보

- 아래 항목은 가맹점 별로 발행 해 드리는 항목으로 주문 예약 및 취소 처리 시 필요한 값 들 입니다.
 - sellerKey (가맹점 코드) : 파트너 센터에서 발급 받은 가맹점 코드
 - cpId (상점 ID) : PAYCO에서 발급
 - productID (상품 ID) : PAYCO에서 발급
 - 간편결제형의 경우 단일 코드 발급
 - 바로구매형의 경우 상품, 배송비 각각 발급 (배송비를 별도 관리하기 위함)



3.2 PAYCO 연동 API 상세 명세

PAYCO에서 제공하는 API 들의 상세 명세를 기술한다.

3.2.1 주문 예약

- PAYCO 주문서를 띄우기 전에 주문예약 API를 통해 필요한 정보를 전달한다.
 - 가맹점에서는 주문 예약 응답으로 받은 주문서 URL 을 팝업으로 띄운다.
 - 외부가맹점에서 관리하는 주문상품 연동 키(sellerOrderProductReferenceKey)는 주문 별로 고유한 key이어야 한다.
- 1) **배송비 상품 설명**(바로구매형인 경우에만 전달필요)
 - 배송비를 상품으로 간주하고 상품금액은 총 배송비로 하여 주문 상품 List(orderProducts)에 하나의 상품으로 추가해야 한다
배송비가 없는 경우에도 금액을 0으로 셋팅하여 주문 상품 List(orderProducts)에 추가해야 한다.
추가 배송비가 발생 한 경우 배송비 상품의 productPaymentAmt 에 추가 배송비가 합산 된다.
위 배송비 상품의 cpId, productId는 서비스 별로 고정된 값을 발급한다.
 - 2) **PAYCO 주문예약 후 결제팝업 오픈시 팝업차단을 위한 개발 TIP!!**
 - orderSheetUrl로 PAYCO 팝업오픈시엔
가맹점페이지 오픈 후 피드백받은 orderSheetUrl을 redirect처리하도록 한다.
 - 팝업 오픈은 POST방식의 submit은 지양하도록 한다. (IE11 관리자모드 about:blank 새창 오픈)
- METHOD : POST
 - URL : /outseller/order/reserve

• 표 3-4 주문 예약 API > INPUT

| Column | Type | Max Length | 필수 여부 | Description |
|-------------|--------|------------|-------|-------------------------------------------------------------------|
| 1 sellerKey | String | 100 | Y | 가맹점 코드 - 파트너센터에서 알려주는 값으로, 초기 연동 시 PAYCO에서 쇼핑몰에 값을 전달한다. |

| | | | | | |
|----|-----------------------------|--------|------|---|-------------------------------------------------------------------------------------------------------------------------------|
| 2 | sellerOrderReferenceKey | String | 100 | Y | 외부가맹점의 주문번호 |
| 3 | sellerOrderReferenceKeyType | String | 50 | N | 외부가맹점의 주문번호 타입 UNIQUE_KEY 유니크 키 - 기본값, DUPLICATE_KEY 중복 가능한 키 (외부가맹점의 주문번호가 중복 가능한 경우 사용) |
| 4 | currency | String | 3 | N | 통화(default=KRW) |
| 5 | totalDeliveryFeeAmt | Number | | N | 총 배송비 (화면노출용) |
| 6 | totalSellerDiscountAmt | Number | | N | 가맹점 할인 금액(바로구매시) |
| 7 | totalPaymentAmt | Number | | Y | 총 결제 할 금액 (총 배송비 포함) - 면세금액,과세금액,부가세, 봉사료 의 합 = totalTaxfreeAmt + totalTaxableAmt + totalVatAmt + totalServiceAmt |
| 8 | totalTaxfreeAmt | Number | | N | 면세금액(면세상품의 공급가액 합) |
| 9 | totalTaxableAmt | Number | | N | 과세금액(과세상품의 공급가액 합) |
| 10 | totalVatAmt | Number | | N | 부가세(과세상품의 부가세 합) |
| 11 | totalServiceAmt | Number | | N | 봉사료 |
| 12 | orderTitle | String | 300 | N | 주문 타이틀 |
| 13 | serviceUrl | String | 1000 | N | (PAY1결제연동시 전달) 주문완료 시 호출되는 API |
| 14 | serviceUrlParam | String | 1000 | N | (PAY1결제연동시 전달) 주문완료 시 호출되는 API 시 함께 전달되어야 하는 파라미터 (Json 형 태의 String) |
| 15 | returnUrl | String | 1000 | N | 주문완료 후 Redirect 되는 Url (PAY2결제연동시,인증완료후 redirect 됨) |
| 16 | returnUrlParam | String | 1000 | N | 주문완료 후 Redirect 되는 Url 시 함께 전달되어야 하는 파라미터 (Json 형태의 String) |
| 17 | nonBankbookDepositInformUrl | String | 1000 | N | 무통장입금완료통보 URL |

3. PAYCO 연동하기

| | | | | | |
|----|-------------------------------|--------|------|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18 | orderMethod | String | N/A | Y | <u>주문유형(=결제유형)</u> <ul style="list-style-type: none"> • 바로구매형 : CHECKOUT • 간편결제형+가맹점 회원결제: EASYPAY_F • 간편결제형+가맹점 비회원결제 : EASYPAY |
| 19 | orderChannel | String | N/A | N | <u>주문채널</u> (default. PC) |
| 20 | inAppYn | String | 1 | N | 인앱결제 여부(Y/N), default=N (in-app 결제인 경우 Y) Y인 경우 inApp 적용 가이드 추가 필요 |
| 21 | individualCustomNoInputYn | String | 1 | N | 개인통관고유번호 입력 여부 (Y/N) default=N |
| 22 | orderSheetUiType | String | N | N | 주문서 UI 타입 선택 선택 가능 값 : (RED ,GRAY, APP_BRIDGE_A) |
| 23 | payMode | String | N | N | <u>결제모드</u> (default. PAY1)) <ul style="list-style-type: none"> • PAY1 타입(결제인증, 승인통합) • PAY2 타입 (결제인증, 승인분리) |
| 24 | orderProducts | Array | | Y | 주문상품List |
| 25 | extraData | String | 3000 | N | 부가정보 (Json 형태의 String) |
| 26 | sellerOptions | String | | N | 판매자 부가정보((Json 형태의 String) |

⇒ 표 3-4의 22번 컬럼 **orderProducts** 주문상품 List

| | Column | Type | Max Length | 필수여부 | Description |
|---|--------------------|--------|------------|------|----------------------|
| 1 | cpId | String | 30 | Y | 상점 ID |
| 2 | productId | String | 50 | Y | 상품 ID |
| 3 | productAmt | Number | | Y | 상품금액(상품단가 * 수량) |
| 4 | productPayment Amt | Number | | Y | 상품 결제금액(상품결제단가 * 수량) |

| | | | | | |
|----|--------------------------------|--------|------|---|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 5 | orderQuantity | Number | 10 | Y | 주문수량 (배송비 상품인 경우 1로 세팅) |
| 6 | option | String | 100 | N | 상품 옵션 |
| 7 | sortOrdering | Number | 10 | Y | 상품노출순서 |
| 8 | productName | String | 160 | Y | 상품명 |
| 9 | orderConfirmUrl | String | 4000 | N | 주문완료 후 주문상품을 확인할 수 있는 url |
| 10 | orderConfirmMobileUrl | String | 1000 | N | 주문완료 후 주문상품을 확인할 수 있는 모바일 url |
| 11 | productImageUrl | String | 4000 | N | 이미지 URL (배송비 상품이 아닌 경우는 필수) |
| 12 | sellerOrderProductReferenceKey | String | 100 | Y | 외부가맹점에서 관리하는 주문상품 연동 키 |
| 13 | taxationType | String | 50 | N | 과세타입 <ul style="list-style-type: none"> • 면세상품 : DUTYFREE • 과세상품 : TAXATION (기본) • 결합상품 : COMBINE |

✓ 주문 상품 List(orderProducts) 부가 설명

<간편 결제형>

orderProducts 리스트에 상품(total)으로 하나만 등록한다.

(총결제할금액 totalPaymentAmt 과 상품별 결제금액 합 productPaymentAmt은 일치하도록 전송)

<간편 구매형>

상품별 리스트와 배송비 상품을 함께 등록한다.

예시1) 5개의 상품이 각각 배송비가 존재할 경우 : 상품 리스트(5개) + 배송비상품(총배송비금액) (1개) = 총 6 개 리스트

예시2) 5개의 상품이 배송비 1회 부과일 경우: 상품 리스트(5개) + 배송비 상품(1개) = 총 6개 리스트

- 배송비를 상품으로 간주하여 등록 해야 함. orderProducts에 배송비 상품 정보를 등록 시엔 본 상품의 정보가 아니라 배송비 정보로 등록해야 함 (즉, 5만원 상품 구매 시 배송비가 3천원인 경우, 배송비 상품 정보를 orderProducts에 등록할 경우 상품 단가, 결제 금액은 5만원이 아니라 3천원으로 등록해야 함)

3. PAYCO 연동하기

⇒ 표 3-4의 23번 컬럼 **extraData**

| | Column | Type | Max Length | 필수여부 | Description |
|---|-------------------------|--------|------------|------|-----------------------------------------------------------------------------------|
| 1 | payExpiryYmdt | String | 14 | N | 해당 주문예약건 만료 처리 일시 (해당 일시 이후에는 결제 불가) Ex) "payExpiryYmdt" : "20150210180000" |
| 2 | virtualAccountExpiryYmd | String | 14 | N | 가상계좌만료일시 YYYYMMDD or YYYYMMDDHHmmss Default 3 일, YYYYMMDD 일 경우 그날 24 시 만료 |
| 3 | cancelMobileUrl | String | 300 | N | 모바일 결제페이지에서 취소 버튼 클릭시 이동할 URL (결제창 이전 URL 등). 미입력시 메인 URL 로 이동 |
| 4 | viewOptions | | | N | 화면 UI 옵션 |
| 5 | sameUserCheckYn | String | | N | 동일사용자 체크 여부 Ex) Y or N |
| 6 | userName | String | 100 | N | 사용자 이름 (sameUserCheckYn = Y 인경우 필수) Ex) 홍길동 |
| 7 | userBirth | String | 100 | N | 사용자 생년월일(yyyyMMdd) (sameUserCheckYn = Y 인경우 필수) Ex) 19890919 |

⇒ 표 3-4의 23번 컬럼 **extraData**의 **viewOptions** 화면UI 옵션

| | Column | Type | Max Length | 필수여부 | Description |
|---|--------------------|--------|------------|------|--------------------------------|
| 1 | showMobileTopGnbYn | String | 1 | N | 모바일 상단 GNB 노출여부 |
| 2 | iframeYn | String | 1 | N | Iframe 호출 |
| 3 | languageCode | String | 100 | N | 주문서 언어 코드 (default KR) - EN/KR |

• 표 3-5 주문 예약 API > OUTPUT

| | Column | Type | Max Length | Description |
|---|----------------|--------|------------|-------------------------|
| 1 | reserveOrderNo | String | 20 | 주문예약번호 |
| 2 | orderSheetUrl | String | 1000 | PAYCO 주문서 url (팝업 open) |

표 2-3 Default format 부분 중 result 컬럼 부분만 표시 하였음

• 표 3-4의 24번 컬럼 sellerOptions

| | Column | Type | Max Length | 필수여부 | Description |
|---|-------------------------|--------|------------|------|-------------------------------------------------------------------|
| 1 | remoteAreaDeliveryFeeYn | String | 1 | N | 추가배송비 존재여부 (Y or N) Y : 추가배송비 API 호출됨 N : 추가배송비 API 호출하지 않음 |

• 주문 예약 API > 샘플 코드

> 샘플 코드 INPUT

```
{
  "totalTaxfreeAmt": 0,
  "sellerKey": "S0FSJE",
  "extraData":
  "{ \"cancelMobileUrl\": \"http://www.payco.com\", \"payExpiryYmdt\": \"20171231180000\", \"virtualAccountExpiryYmd\": \"20171231180000\", \"appUrl\": \"testapp://open\", \"viewOptions\": { \"iframeYn\": \"N\", \"showMobileTopGnbYn\": \"N\", \"languageCode\": \"KR\" } }",
  "payMode": "PAY2",
  "orderProducts": [{
    "productImageUrl": "",
    "orderConfirmMobileUrl": "",
    "productId": "PROD_EASY",
    "productPaymentAmt": 81500,
    "productAmt": 79000,
    "cpId": "PARTNERTEST",
    "sortOrdering": 1,
    "orderQuantity": 1,
    "productName": "[GLOBE] TILT KIDS (NAVY/ORANGE)",
    "orderConfirmUrl": "",
    "taxationType": "TAXATION",
    "sellerOrderProductReferenceKey": "ITEM 100001",
    "option": "280"
  }],
  "totalTaxableAmt": 74091,
  "orderChannel": "PC",
  "sellerOrderReferenceKey": "TEST3662848455",
  "returnUrlParam": "",
  "orderMethod": "EASYPAY",
  "nonBankbookDepositInformUrl":
  "http://nemusoft.iptime.org/payco_without_bankbook_jsp_easypay_pay2.asp",
  "individualCustomNoInputYn": "N",
  "totalPaymentAmt": 81500,
  "totalVatAmt": 7409,
  "currency": "KRW",
  "inAppYn": "N",
  "orderSheetUiType": "GRAY",
  "orderTitle": "payco 결제 테스트 (JSP)",
}
```

3. PAYCO 연동하기

```
"returnUrl": "http://localhost/jsp/easypay/pay2/payco_return.jsp",
"sellerOrderReferenceKeyType": "UNIQUE KEY"
}
```

> 샘플 코드 OUTPUT

```
{
  "result": {
    "reserveOrderNo": "201706232004537495",
    "orderSheetUrl": "https://alpha-
bill.payco.com/easyLogin/201706232004537495"
  },
  "code": 0,
  "message": "success"
}
```

3.2.2 주문 상품 상태 변경

- 가맹점이 변경 가능한 주문상품상태는 '배송 시작 / 취소(수기환불의 경우) / 구매확정
- 에스크로가 적용되는 결제수단(무통장입금, 계좌이체, 바로이체)인 경우 배송 시작, 구매 확정 상태는 정산에 영향을 미칠 수 있으므로 반드시 둘 중 하나는 호출이 필요 함
- URL : /outseller/order/updateOrderProductStatus

• 표 3-6 주문 상품 상태 변경 API > INPUT

| Column | Type | Max Length | 필수 여부 | 설명 |
|----------------------------------|--------|------------|-------|------------------------------------------------------------|
| 1 sellerKey | String | 100 | Y | 가맹점 코드 - 파트너센터에서 알려주는 값으로, 초기 연동시 Payco에서 쇼핑물에 값을 전달한다. |
| 3 orderNo | String | 20 | Y | 주문번호 |
| 4 sellerOrderProductReferenceKey | String | 100 | Y | 외부가맹점에서 관리하는 주문상품연동키 (팝슈즈의 경우 ord_opt_no) |
| 5 orderProductStatus | String | N/A | Y | 변경할 주문상품상태 |

• 표 3-7 주문 상품 상태 변경 API > OUTPUT

| Column | Type | Description |
|---------|--------|---------------|
| code | Number | 성공여부 |
| message | String | 성공/실패 message |

➔ 표 2-3 Default format 중 result 컬럼 부분 없음

- **Sample**

- **INPUT**

```
{
  "sellerKey" : "DEMO",
  "sellerOrderReferenceKey" : " TESTORDERKEY00000001",
  "sellerOrderProductReferenceKey" : "1170961",
  "orderProductStatus" : "DELIVERY START"
}
```

- **OUTPUT**

```
{
  "message": "success",
  "code": 0
}
```

3.2.3 주문 및 결제 취소 가능여부 조회

- 취소 처리 API를 호출하기 이전에 취소 처리를 진행할 수 있는지 확인하기 위해 사용한다.
- cancelPossibleYn이 N인 경우 PG취소가능금액(pgCancelPossibleAmt)만큼만 "주문 및 결제 취소 처리 API"로 취소가 가능하다.
- **cancelPossibleYn 이 N인 경우**
 - 결제 건 중 PG결제 취소가 불가능한 내역이 있음
 - 부분 취소가 필요한 경우 "취소할 총 금액 <= PG 취소가능금액" 으로 요청해야 함
- **PG결제 취소 불가능한 경우**
 - 핸드폰 익월 취소요청
 - 무통장 입금 취소요청
 - 부분취소 불가능한 PG결제의 부분취소 요청
- 취소 할 배송비가 있는 경우 (부분 취소의 경우만 해당)
 - 취소 배송비를 취소할 상품 List에 추가해야 한다.
 - 추가 배송비도 함께 취소해야 하는 경우 배송비 상품의 productAmt에 추가 배송비를 합산한 금액으로 넘긴다.
 - 주문예약 시 전달했던 배송비 상품의 sellerOrderProductReferenceKey, cpId, productId 가 동일해야 한다.
- 수기 환불 시 : 적립예정이거나 적립 완료 된 포인트에 대해 수기 환불 금액만큼 포인트 취소를 요청해야 한다.

3. PAYCO 연동하기

- METHOD : POST
- URL : /outseller/order/cancel/checkAvailability

• 표 3-8 주문 결제 취소 가능여부 조회 API > INPUT

| | Column | Type | Max Length | 필수 여부 | 설명 |
|---|----------------|--------|------------|-------|--------------------------------------------------------------|
| 1 | sellerKey | String | 100 | Y | 가맹점 코드 - 파트너센터에서 알려주는 값으로, 초기 연동 시 Payco 에서 쇼핑몰에 값을 전달한다. |
| 2 | orderNo | String | 20 | Y | 주문 번호 |
| 3 | cancelTotalAmt | Number | | Y | 취소할 총 금액 |
| 4 | orderProducts | Array | N/A | N | 취소할 상품 List(부분취소인 경우 사용, 입력하지 않는 경우 전체 취소) |

⇒ orderProducts

| | Column | Type | Max Length | 필수 여부 | 설명 |
|---|--------------------------------|--------|------------|-------|---------------------|
| 1 | sellerOrderProductReferenceKey | String | 100 | Y | 외부가맹점에서 관리하는 주문상품번호 |
| 2 | cpId | String | 30 | Y | 상점 ID |
| 3 | productId | String | 50 | Y | 상품 ID |
| 4 | productAmt | Number | | Y | 취소 할 주문상품 금액 |

• 표 3-9 주문 결제 취소 가능여부 조회 API > OUTPUT

| | Column | Type | Max Length | Description |
|---|------------------------|--------|------------|-------------|
| 1 | cancelPossibleYn | String | 1 | 취소가능여부 |
| 2 | cancelImpossibleReason | String | 500 | 취소불가이유 |

| | Column | Type | Max Length | Description |
|---|----------------------------|--------|------------|----------------------------------------|
| 3 | partCancelPossibleYn | String | 1 | 부분취소가능여부 |
| 4 | pgCancelPossibleAmt | Number | | PG취소가능금액 (PG결제취소 불가인 결제수단을 제외한 취소가능금액) |
| 5 | cancelImpossibleReasonCode | Number | | 취소불가사유코드 |

→ 표 2-3 Default format 부분 중 result 컬럼 부분만 표시하였음

- Sample 코드

- 샘플 INPUT

```
{
  "sellerKey": "S0FSJE",
  "orderNo": "201706232002275277",
  "cancelTotalAmt": 50000,
  "orderProducts": [{
    "sellerOrderProductReferenceKey": "ITEM 100001",
    "cpId": "PARTNERTEST",
    "productId": "PROD EASY",
    "productAmt": 50000
  }]
}
```

- 샘플 OUTPUT 취소 가능일 때

```
{
  "result": {
    "cancelPossibleYn": "Y",
    "partCancelPossibleYn": "Y",
    "pgCancelPossibleAmt": 50000,
    "cancelImpossibleReason": "",
    "cancelImpossibleReasonCode": null,
    "orderNo": "201706232002275277",
    "cancelExpectPaymentDetails": null,
    "cancelPaymentDetails": null
  },
  "code": 0,
  "message": "success"
}
```

- 샘플 OUTPUT 취소 안될 때

```
{
  "result": {
    "cancelPossibleYn": "N",
    "partCancelPossibleYn": "N",
    "pgCancelPossibleAmt": 500,
    "cancelImpossibleReason": " PAYCO 포인트 잔액 부족으로 주문을 취소할 수 없습니다. PAYCO
고객센터 : 1544-6891 ",
    "cancelImpossibleReasonCode": 1303
  },
  "message": "success",
  "code": 0
}
```

```
}
```

3.2.4 주문 및 결제 취소 처리

- 기존에 생성된 주문 및 결제내역을 취소 처리하는 경우 사용한다.
PG취소가 가능한 금액으로 요청 했을 때만 취소된다.
- 취소 할 배송비가 있는 경우 (부분 취소의 경우만 해당)
 - 취소 배송비를 취소할 상품 List에 추가해야 한다.
 - 추가 배송비도 함께 취소해야 하는 경우 배송비 상품의 productAmt에 추가 배송비를 합산한 금액으로 넘긴다.
 - 주문예약 시 전달했던 배송비 상품의 sellerOrderProductReferenceKey, cpId, productId 가 동일해야 한다.
 - 쿠폰사용시에는 결제 부분취소가 불가능한 케이스가 존재 함.
(부분취소금액 <= 쿠폰사용금액 조건인 경우 부분취소 불가)
 - 바로이체의 경우 11월 5일부터 전체취소 지원
단, 부분취소는 지원이 안되며
주문상태가 배송시작이거나 구매확정인 경우 또는 주문일시가 120일이 지난 건의 경우는 취소 불가.
- METHOD : POST
- URL : /outseller/order/cancel

표 3-10 주문 결제 취소 처리 API > INPUT

| | Column | Type | Max Length | 필수 여부 | 설명 |
|---|----------------|--------|------------|-------|-------------------------------------------------------------------------|
| 1 | sellerKey | String | 100 | Y | 가맹점 코드 - 파트너센터에서 알려주는 값으로, 초기 연동 시 PAYCO에서 쇼핑몰에 값을 전달한다. |
| 2 | orderNo | String | 20 | Y | 주문번호 (sellerOrderReferenceKeyType 을 DUPLICATE_KEY로 넘긴 경우 orderNo 필수) |
| 3 | cancelTotalAmt | Number | | Y | 취소할 총 금액 = totalCancelTaxfreeAmt + |

| | | | | |
|----|------------------------|--------|---|-------------------------------------------------------------------------|
| | | | | totalCancelTaxableAmt + totalCancelVatAmt + totalCancelServiceAmt |
| 4 | totalCancelTaxfreeAmt | Number | N | 총 취소할 면세금액 |
| 5 | totalCancelTaxableAmt | Number | N | 총 취소할 과세금액 |
| 6 | totalCancelVatAmt | Number | N | 총 취소할 부가세 |
| 7 | totalCancelServiceAmt | Number | N | 총 취소할 봉사료 |
| 8 | totalCancelPossibleAmt | Number | N | 총 취소가능금액(현재기준) : 취소가능금액 검증 |
| 9 | orderProducts | Array | N | 취소할 상품 List(부분취소인 경우 사용, 입력하지 않는 경우 전체 취소) |
| 10 | orderCertifyKey | String | Y | 주문완료통보시 내려받은 인증값 |
| 11 | requestMemo | String | N | 취소처리 요청메모 |

⇒ 8번 컬럼 orderProducts

| Column | Type | Max Length | 필수 여부 | 설명 |
|---------------------------------|--------|------------|-------|---------------------|
| 1 sellerOrderProductRefrenceKey | String | 100 | Y | 외부가맹점에서 관리하는 주문상품번호 |
| 2 cpId | String | 30 | Y | 상점 ID |
| 3 productId | String | 50 | Y | 상품 ID |
| 4 productAmt | Number | | Y | 취소 할 주문상품 금액 |
| 5 cancelDetailContent | String | 160 | N | 취소 상세 사유 |

• 표 3-11 주문 결제 취소 처리 API > OUTPUT

| Column | Type | Max Length | Description |
|-------------------------|--------|------------|-------------|
| 1 cancelTradeSeq | Number | 19 | 취소내역번호 |

3. PAYCO 연동하기

| | | | | |
|----|-----------------------------------|--------|--------------------------------------------------------------|---------------|
| 2 | totalCancelPaymentAmt | Number | 총 취소금액 | |
| 3 | remainCancelPossibleAmt | Number | 잔여 취소가능금액 | |
| 4 | cancelReceiptAmt | Number | 현금영수증 발급 대상 취소 금액 | |
| 5 | cancelReceiptPaycoPointAmt | Number | 현금영수증 발급 대상 Payco 포인트 취소 금액 (deprecated, 2018년 제거예정) | |
| 6 | cancelReceiptPaycoPointTaxfreeAmt | Number | 현금영수증 발급 대상 Payco 포인트 취소 면세금액 (deprecated, 2018년 제거예정) | |
| 7 | cancelReceiptPaycoPointTaxableAmt | Number | 현금영수증 발급 대상 Payco 포인트 취소 과세금액 (deprecated, 2018년 제거예정) | |
| 8 | cancelReceiptPaycoPointVatAmt | Number | 현금영수증 발급 대상 Payco 포인트 취소 부가세 (deprecated, 2018년 제거예정) | |
| 9 | cancelReceiptPaycoPointServiceAmt | Number | 현금영수증 발급 대상 Payco 포인트 취소 봉사료 (deprecated, 2018년 제거예정) | |
| 10 | cancelPaymentDetails | Array | N/A | 취소 결제 내역 List |
| 11 | cancelYmdt | String | 취소 처리 시간 | |

⇒ 표 2-3 Default format 부분 중 result 컬럼 부분만 표시하였음

⇒ 표3-11 의 cancelPaymentDetails

| | Column | Type | Max Length | Description | Example |
|---|----------------------|--------|------------|-----------------|--------------------|
| 1 | paymentTradeNo | String | 20 | 결제번호 | 201409191000001142 |
| 2 | cancelPaymentTradeNo | String | 20 | 취소거래내역 결제 번호 | 201409191000001142 |
| 3 | paymentMethodCode | String | 10 | 결제수단코드 | 01 |
| 4 | paymentMethodName | String | | 결제수단명 | 신용카드 |
| 5 | cancelPaymentAmt | Number | | 취소금액 | 119000 |

| | | | | |
|----|------------------|-----------|--------------------------|----------------|
| 6 | cancelTaxfreeAmt | Number | 취소 면세금액 | 9000 |
| 7 | cancelTaxableAmt | Number | 취소 과세금액 | 100000 |
| 8 | cancelVatAmt | Number | 취소 부가세 | 10000 |
| 9 | cancelServiceAmt | Number | 취소 봉사료 | 1000 |
| 10 | tradeYmdt | String 14 | 취소일시 (yyyyMMddHHmmss) | 20140822120201 |

- Sample 코드

⇒ INPUT

```
{
  "requestMemo": "결제취소 테스트입니다.",
  "orderNo": "201706232002275216",
  "orderCertifyKey": "A7CZSz9A489nkOUWSMM3NZ1OAHpdhKnpG2BBMh3D458YCC",
  "cancelTotalAmt": 81500,
  "sellerKey": "S0FSJE",
  "totalCancelTaxableAmt": "74091",
  "totalCancelVatAmt": "7409",
  "totalCancelPossibleAmt": "81500",
  "totalCancelTaxfreeAmt": "0",
  "orderProducts": [{
    "sellerOrderProductReferenceKey": "1171997",
    "cpId": "PS TOASTCO",
    "productId": "PS TOASTCOMC",
    "productAmt": 81500,
    "cancelDetailContent": "TEST"
  }]
}
```

⇒ OUTPUT

```
{
  "result": {
    "orderNo": "201706232002275216",
    "cancelTradeSeq": 2000266924,
    "totalCancelPaymentAmt": 81500.0,
    "remainCancelPossibleAmt": 0.0,
    "cancelPaymentDetails": [{
      "paymentTradeNo": "201706232002995447",
      "paymentMethodCode": "98",
      "paymentMethodName": "PAYCO 포인트",
      "cancelPaymentAmt": 10000.0,
      "tradeYmdt": "20170623135100",
      "cancelPaymentTradeNo": "201706232002995449",
      "cancelTaxableAmt": 9090.0,
      "cancelTaxfreeAmt": 0.0,
      "cancelVatAmt": 910.0,
      "cancelServiceAmt": 0.0
    }]
  }
}
```

3. PAYCO 연동하기

```
{, {
  "paymentTradeNo": "201706232002995446",
  "paymentMethodCode": "75",
  "paymentMethodName": "PAYCO 쿠폰",
  "cancelPaymentAmt": 2000.0,
  "tradeYmdt": "20170623135100",
  "cancelPaymentTradeNo": "201706232002995450",
  "cancelTaxableAmt": 1820.0,
  "cancelTaxfreeAmt": 0.0,
  "cancelVatAmt": 180.0,
  "cancelServiceAmt": 0.0
}, {
  "paymentTradeNo": "201706232002995442",
  "paymentMethodCode": "31",
  "paymentMethodName": "신용카드",
  "cancelPaymentAmt": 69500.0,
  "tradeYmdt": "20170623135100",
  "cancelPaymentTradeNo": "201706232002995452",
  "cancelTaxableAmt": 63181.0,
  "cancelTaxfreeAmt": 0.0,
  "cancelVatAmt": 6319.0,
  "cancelServiceAmt": 0.0
}],
"cancelReceiptPaycoPointAmt": 9700.0,
"cancelReceiptPaycoPointTaxfreeAmt": 0.0,
"cancelReceiptPaycoPointTaxableAmt": 8818.0,
"cancelReceiptPaycoPointVatAmt": 882.0,
"cancelReceiptPaycoPointServiceAmt": 0.0,
"cancelYmdt": "20170623135100"
},
"code": 0,
"message": "success"
}
```

3.2.5 마일리지 적립 취소

- 수기 환불의 경우 취소 할 결제금액을 전달하면 결제금액에 해당하는 마일리지를 계산하여 적립전체/부분취소 처리를 한다.
- URL : /outseller/order/cancel/partMileage

• 표 3-12 마일리지 적립 취소 > INPUT

| 변수명 | Type | Max Lenth | 필수여부 | 설명 |
|-----------------------|--------|--------------|------|-----------------------------------------------------------|
| 1 sellerKey | String | 100 | Y | 판매자Key |
| 2 orderNo | String | 20 | Y | 주문번호 |
| 3 cancelPaymentAmount | Number | | Y | 마일리지 부분 취소할 결제금액(파라미터로 전달한 결제금액에 마일리지 적립율을 곱한 액수를 취소하도록 처 |

리되어 있음)

• 표 3-13 마일리지 적립 취소 > OUTPUT

| 변수명 | Type | Max Lenth | 설명 |
|-----------------------------|--------|--------------|--------------|
| 1 canceledMileageAcmAmount | Number | | 취소된 마일리지 금액 |
| 2 remainingMileageAcmAmount | Number | | 남아있는 마일리지 금액 |

• Sample 코드

⇒ INPUT

```
{
  "cancelPaymentAmount": 81500,
  "orderNo": "201706232002275245",
  "sellerKey": "S0FSJE"
}
```

⇒ OUTPUT

```
{
  "result": {
    "canceledMileageAcmAmount": 815.0,
    "remainingMileageAcmAmount": 0.0
  },
  "code": 0,
  "message": "success"
}
```

3.2.6 가맹점 별 연동 키 유효성 체크

- PAYCO API를 사용하기 위해 사전에 발급해 준 가맹점 별 인증 키에 대한 사용가능 여부를 체크 한다.
- URL : /outseller/order/code/checkUsability

• 표 3-14 가맹점 별 코드 값 사용가능 여부 > INPUT

| 변수명 | Type | Max Lenth | 필수여부 | 설명 |
|-----|------|--------------|------|----|
|-----|------|--------------|------|----|

3. PAYCO 연동하기

| | | | | | |
|---|-----------|--------|-----|---|----------|
| 1 | sellerKey | String | 100 | Y | 판매자Key |
| 2 | codes | Array | 100 | Y | 코드 값 리스트 |

⇒ 2번 컬럼 codes

| Column | Type | Max Length | 필수 여부 | 설명 |
|------------------|--------|------------|-------|----------------------------------------------------------------------|
| 1 codeKind | String | 100 | Y | 코드종류 <값> CP_ID(상점 ID) PRODUCT_ID(상품 ID) |
| 2 codeValue | String | 50 | Y | 코드 값 |
| 3 upperCodeValue | String | 50 | - | codeKind=PRODUCT_ID 인 경우 필수 - upperCodeValue 에 cpId 를 넘겨준다. |

• 표 3-15 가맹점 별 코드 값 사용가능 여부 > OUTPUT

| 변수명 | Type | Max Lenth | 설명 |
|-----------------------|--------|-----------|------------------------------------------|
| 1 codeKind | String | 50 | 코드 종류 CP_ID(상점ID) PRODUCT_ID(상품ID) |
| 2 codeValue | String | 50 | 코드 값 |
| 3 upperCodeValue | String | 50 | 상위 코드값 |
| 4 usabilityYn | String | 1 | 등록가능여부(Y/N) |
| 5 useImpossibleReason | String | 1000 | 등록불가이유 |

• Sample 코드

⇒ INPUT

```
{
```



```

"codes": [{
  "codeKind": "CP ID",
  "codeValue": "PARTNERTEST"
}, {
  "upperCodeValue": "PARTNERTEST",
  "codeKind": "PRODUCT ID",
  "codeValue": "PROD EASY"
}],
"sellerKey": "S0FSJE"
}

```

⇒ OUTPUT

```

{
  "result": [{
    "codeKind": "CP ID",
    "codeValue": "PARTNERTEST",
    "usabilityYn": "Y"
  }, {
    "codeKind": "PRODUCT ID",
    "codeValue": "PROD EASY",
    "usabilityYn": "Y"
  }],
  "code": 0,
  "message": "success"
}

```

3.2.7 결제승인 API (결제모드 PAY2 타입)

- 결제인증 후 실제 결제완료 처리가 되는 API 이다.
- 외부가맹점에서 주문 가능한 조건이 되는 경우 결제승인 API를 호출한다. (ex. 재고확인)
- 승인처리 성공 시 PAYCO 회원에게 PAYCO앱에 주문완료 push알림 및 email을 발송한다.
- 결제인증 후 5분내 승인처리가 이루어져야 한다.
- 중복 승인 요청이 일어나지 않도록 방어코드 작성 필요.
- timeout 최대 30초 설정.

• URL : /outseller/payment/approval

• 표 3-16 결제승인 API > INPUT

| | Column | Type | Max Length | 필수 여부 | Description |
|---|----------------|--------|------------|-------|-------------------------------------------------------------------|
| 1 | sellerKey | String | 100 | Y | 가맹점 코드 - 파트너센터에서 알려주는 값으로, 초기 연동 시 PAYCO에서 쇼핑물에 값을 전달한다. |
| 2 | reserveOrderNo | String | 20 | Y | 주문예약번호 |

3. PAYCO 연동하기

| | | | | | |
|---|-------------------------|--------|-----|---|-------------------------------------------------------------------------------------------------------------------|
| 3 | sellerOrderReferenceKey | String | 100 | Y | 외부가맹점에서 관리하는 주문연동 Key |
| 4 | paymentCertifyToken | String | 100 | Y | 결제인증토큰 - 결제인증완료 후 returnUrl로전달 받은 정보 |
| 5 | totalPaymentAmt | Number | | Y | 총 결제 할 금액 (총 배송비 포함) - 면세금액,과세금액,부가세의 합 = totalTaxfreeAmt + totalTaxableAmt + totalVatAmt + totalServiceAmt |
| 6 | totalTaxfreeAmt | String | | N | 면세금액(면세상품의 공급가액 합) |
| 7 | totalTaxableAmt | String | | N | 과세금액(과세상품의 공급가액 합) |
| 8 | totalVatAmt | String | | N | 부가세(과세상품의 부가세 합) |
| 9 | totalServiceAmt | String | | N | 봉사료 |

• 표 3-17 결제승인 API > OUTPUT

- totalOrderAmt(총 주문금액) + totalDeliveryFeeAmt(총 배송비) + totalRemoteAreaDeliveryFeeAmt(총 도서산간비용) = 총 결제금액 (totalPaymentAmt)

| 변수명 | Type | Max Length | 설명 | 예 |
|---------------------------|--------|---------------|--------------------------|----------------------------|
| 1 sellerOrderReferenceKey | String | 100 | 외부가맹점에서 발급하는 주문 연동 Key | 팝슈즈 예)20140821221250595 |
| 2 reserveOrderNo | String | 20 | 주문예약번호 | 201408211000000720 |
| 3 orderNo | String | 20 | 주문번호 | 20140821100000072 |
| 4 memberName | String | 15 | 주문자명(간편결제형만 일부 마스킹처리) | |
| 5 memberEmail | String | 200 | 주문자EMAIL(일부 마스킹처리) | |
| 6 orderChannel | String | N/A | 주문채널 (PC/MOBILE) | PC |
| 7 totalOrderAmt | Number | | 총 주문금액 | 116500 |

| | | | | |
|----|----------------------------------------|-------------------|---------------------------------|---------------------------------------------------------------------------------------|
| 8 | totalDeliveryFeeAmt | Number | 총 배송비 | 2500 |
| 9 | totalRemoteAreaDeliveryFeeAmt | Number | 총 도서산간비 | 0 |
| 10 | totalPaymentAmt | Number | 총 결제금액 | 11900 |
| 11 | paymentCompletionYn | String 1 | 결제완료여부 (Y/N) | Y |
| 12 | deliveryPlace | Object | N/A 배송지 정보 | |
| 13 | orderProducts | Array | N/A 주문상품 List | |
| 14 | paymentDetails | Array | N/A 결제내역 List | - 복합 결제의 경우 3.5.5의 결제수단 별 array list로 구성되어 리턴 되며, 결제 수단은 추후 추가될 수 있다. |
| 15 | orderCertifyKey | String 100 | 주문인증키 | 0 |
| 17 | receiptAmt | Number | 현금영수증 발급 대상 금액 | 0 |
| 18 | receiptPaycoPointAmt | Number | 현금영수증 발급 대상 포인트 결제금액 | Deprecated |
| 19 | receiptPaycoPointTaxfreeAmt | Number | 현금영수증 발급 대상 포인트 면세금액 | Deprecated |
| 20 | receiptPaycoPointTaxableAmt | Number | 현금영수증 발급 대상 포인트 과세금액 | Deprecated |
| 21 | receiptPaycoPointVatAmt | Number | 현금영수증 발급 대상 포인트 부가세 | Deprecated |
| 22 | receiptPaycoPointServiceAmt | Number | 현금영수증 발급 대상 포인트 봉사료 | Deprecated |

⇒ **deliveryPlace (배송지)**

| 변수 | type | Max Length | 설명 | 예제 |
|-------------|--------|------------|-----|-----|
| 1 recipient | String | 15 | 수령인 | 홍길동 |

3. PAYCO 연동하기

| | | | | | |
|---|------------------------|--------|-----|---------|-----------------|
| 2 | englishReceipient | String | 50 | 수령인 영문명 | LEEJINHEE |
| 3 | address1 | String | 300 | 기본주소 | 경기도 성남시 분당구 삼평동 |
| 4 | address2 | String | 300 | 상세주소 | 629 플레이뮤지엄 |
| 5 | zipcode | String | 50 | 우편번호 | 463-400 |
| 6 | deliveryMemo | String | 30 | 배송메모 | 경비실에 맡겨주세요 |
| 7 | telephone | String | 15 | 연락처 | 010-1111-2222 |
| 8 | individualCustomUniqNo | String | 50 | 개인통관번호 | |

⇒ orderProducts (주문상품List)

| 변수 | Type | Max Length | 설명 | 예제 | |
|----|--------------------------------|------------|-----|----------------------------------------------------------------|-------------------|
| 1 | orderProductNo | String | 20 | 주문상품 번호 | 20140821100000022 |
| 2 | cpId | String | 30 | 상점ID | PS_TOASTCO |
| 3 | productId | String | 50 | 상품ID | PS_TOASTCOMC1 |
| 4 | sellerOrderProductReferenceKey | String | 100 | 외부가맹점에서 발급한 주문상품연동Key | 19801895 |
| 5 | orderProductStatusCode | String | 50 | 주문상품 상태코드 | PAYED |
| 6 | orderProductStatusName | String | 10 | 주문상품 상태명 | 결제완료 |
| 7 | productKindCode | String | | 상품타입 | GENERAL_PRODUCT |
| 8 | productPaymentAmt | Float | | 상품결제금액 : 배송비 5000 상품의 경우 (원 상품결제금액 + 추가 배송비) 로 변경 될 수 있음 | |
| 9 | originalProductPaymentAmt | Float | | 원 상품결제금액 (배송비 상품의 경우 배송비 금액) | 2000 |

주문 상품 List(orderProducts) 부가 설명

- 추가 배송비가 발생한 경우, productPaymentAmt 에 추가 배송비와 일반 배송비가 합산 된 금액으로 전달 된다.

- 예) 배송비 2000 , 추가 배송비 3000원일 경우

배송비 상품 orderProducts > originalProductPaymentAmt = 2000원

배송비 상품 orderProducts > productPaymentAmt = 5000원

⇒ **paymentDetails (결제내역 List)**

| 변수 | Type | Max Length | 설명 | 예제 |
|---------------------------------------------|--------|------------|-----------------------------|--------------------|
| 1 paymentTradeNo | String | 20 | 결제번호 | 201409191000001142 |
| 2 paymentMethodCode | String | 10 | 결제수단코드 | 01 |
| 3 paymentMethodName | String | 15 | 결제수단명 | 신용카드 |
| 4 paymentAmt | Number | | 결제금액 | 119000 |
| 5 taxfreeAmt | Number | | 면세금액 | 0 |
| 6 taxableAmt | Number | | 과세금액 | 0 |
| 7 vatAmt | Number | | 부가세 | 0 |
| 8 serviceAmt | Number | | 봉사료 | 0 |
| 9 tradeYmdt | String | 14 | 결제일시 (yyyyMMddHHmmss) | 20140822120201 |
| 10 pgAdmissionNo | String | 100 | 원천사승인번호 | 209380 |
| 11 pgAdmissionYmdt | String | 14 | 원천사승인일시 (yyyyMMddHHmmss) | 20140822120201 |
| 12 easyPaymentYn | String | 1 | 간편결제여부 (Y/N) | N |
| 13 cardSettleInfo | Object | N/A | 신용카드 결제 정보 | ↓ |
| 14 cellphoneSettleInfo | Object | N/A | 핸드폰 결제 정보 | ↓ |
| 15 realtimeAccountTransferSettleInfo | Object | N/A | 실시간계좌이체 결제정보 | ↓ |
| 16 nonBankbookSettleInfo | Object | N/A | 무통장입금 결제정보 | ↓ |

3. PAYCO 연동하기

| | | | | |
|----|-------------------------|--------|-----|------|
| 17 | couponSettleInfo | Object | N/A | 쿠폰정보 |
|----|-------------------------|--------|-----|------|

⇒ cardSettleInfo (카드결제정보)

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|------------------------------|--------|------------|-----------------------------------|-----------|
| 1 | cardCompanyName | String | 15 | 카드사명 | 하나SK |
| 2 | cardCompanyCode | String | 50 | 카드사코드 | CCSK |
| 3 | cardNo | String | 300 | 카드번호 | *****1809 |
| 4 | cardInstallmentMonthNumber | String | 2 | 할부개월(MM) | 00 |
| 5 | cardAdmissionNo | String | 20 | 카드사 승인번호 | 31213333 |
| 6 | cardInterestFreeYn | String | 1 | 무이자여부(Y/N) | Y |
| 7 | corporateCardYn | String | 1 | 법인카드여부 (개인 N ,법인 Y) | Y |
| 8 | partCancelPossibleYn | String | 1 | 부분취소가능유무(Y/N) | Y |
| 9 | cardType (별도요청 가맹점만 추가적용) | String | 100 | 카드타입 (CREDIT 신용카드, CHECK 체크카드) | CHECK |

⇒ cellphoneSettleInfo (핸드폰 결제 정보)

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|-------------|--------|------------|-------|---------------|
| 1 | cellphoneNo | String | 32 | 휴대폰번호 | 010-1111-2222 |

⇒ realtimeAccountTransferSettleInfo (실시간계좌이체,바로이체 결제정보)

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------|--------|------------|----------------------|----|
| 1 | bankName | String | 15 | 은행명 | 신한 |
| 2 | bankCode | String | 50 | 은행코드 | 88 |

⇒ **nonBankbookSettleInfo (무통장입금 결제정보)**

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------------------|--------|------------|-------|---------------|
| 1 | bankName | String | 15 | 은행명 | 신한 |
| 2 | bankCode | String | 50 | 은행코드 | 88 |
| 3 | accountNo | String | 200 | 계좌번호 | 1901801980225 |
| 4 | paymentExpirationYmd | String | 8 | 입금만료일 | 20141208 |

⇒ **couponSettleInfo (쿠폰정보)**

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------------------|--------|------------|--------|-------|
| 1 | discountAmt | String | 15 | 쿠폰할인금액 | 10000 |
| 2 | discountConditionAmt | String | 50 | 쿠폰할인조건 | 50000 |

• 결제승인 API > Sample

➤ 샘플 코드 INPUT

```
{
  "totalPaymentAmt": 81500,
  "sellerKey": "S0FSJE",
  "reserveOrderNo": "201706232004537270",
  "sellerOrderReferenceKey": "TEST2233867668",
  "paymentCertifyToken":
  "Sg3P nctRELyNi7qppJiW0ZEvxqkhXj9OPrWjVkiq70KSUvjuF7xZNh8SYU7UiN"
}
```

➤ 샘플 코드 OUTPUT

```
{
  "result": {
    "sellerOrderReferenceKey": "TEST2233867668",
    "reserveOrderNo": "201706232004537270",
    "orderNo": "201706232002275216",
    "memberName": "강**",
    "memberEmail": "na*****",
    "orderChannel": "PC",
    "totalOrderAmt": 81500,
    "totalDeliveryFeeAmt": 0,
    "totalRemoteAreaDeliveryFeeAmt": 0,
    "totalPaymentAmt": 81500,
    "receiptPaycoPointAmt": 9700,
    "receiptPaycoPointTaxfreeAmt": 0,
  }
}
```

```
"receiptPaycoPointTaxableAmt": 8818,
"receiptPaycoPointVatAmt": 882,
"receiptPaycoPointServiceAmt": 0,
"orderProducts": [{
  "orderProductNo": "201706232002428725",
  "sellerOrderProductReferenceKey": "ITEM 100001",
  "orderProductStatusCode": "OPSPAED",
  "orderProductStatusName": "결제완료",
  "cpId": "PARTNERTEST",
  "productId": "PROD_EASY",
  "productKindCode": "GENERAL_PRODUCT",
  "productPaymentAmt": 81500,
  "originalProductPaymentAmt": 81500
}],
"paymentDetails": [{
  "couponSettleInfo": {
    "discountAmt": 2000.0,
    "discountConditionAmt": 7000.0,
    "encCouponId": null
  },
  "paymentTradeNo": "201706232002995446",
  "paymentMethodCode": "75",
  "paymentMethodName": "PAYCO 쿠폰",
  "paymentAmt": 2000,
  "taxfreeAmt": 0,
  "taxableAmt": 1820,
  "vatAmt": 180,
  "serviceAmt": 0,
  "tradeYmdt": "20170623135056",
  "pgAdmissionNo": "34408",
  "pgAdmissionYmdt": "20170623135055",
  "easyPaymentYn": "N"
}, {
  "paymentTradeNo": "201706232002995447",
  "paymentMethodCode": "98",
  "paymentMethodName": "PAYCO 포인트",
  "paymentAmt": 10000,
  "taxfreeAmt": 0,
  "taxableAmt": 9090,
  "vatAmt": 910,
  "serviceAmt": 0,
  "tradeYmdt": "20170623135056",
  "pgAdmissionNo": "2017062320989065",
  "pgAdmissionYmdt": "20170623135055",
  "easyPaymentYn": "N"
}, {
  "cardSettleInfo": {
    "cardCompanyName": "BC카드",
    "cardCompanyCode": "CCBC",
    "cardNo": "490603*****3604",
    "cardInstallmentMonthNumber": "00",
    "cardAdmissionNo": "03628572",
    "cardInterestFreeYn": "N",
    "corporateCardYn": "N",
    "partCancelPossibleYn": "Y"
  },
  "paymentTradeNo": "201706232002995442",
  "paymentMethodCode": "31",
  "paymentMethodName": "신용카드",
  "paymentAmt": 69500,
  "taxfreeAmt": 0,
  "taxableAmt": 63181,
  "vatAmt": 6319,
  "serviceAmt": 0,
  "tradeYmdt": "20170623135056",
  "pgAdmissionNo": "17424902645974",
```



```
        "pgAdmissionYmdt": "20170623135055",  
        "easyPaymentYn": "Y"  
    }],  
    "orderCertifyKey":  
    "A7CZSz9A489nkOUWSMM3NZ1OAHPhKnPG2BBMh3D458YCC",  
    "paymentCompletionYn": "Y",  
    "paymentCompleteYmdt": "20170623135056"  
},  
"code": 0,  
"message": "success"  
}
```

3.2.8 결제수단 및 코드조회 API

- 가맹점에서 사용중인 결제수단 및 은행코드, 신용카드코드를 제공한다.
(사용중인 결제수단 : 가맹점 계약사항)
- URL : /outseller/support/search/info
- 표 3-18 결제수단 및 코드조회 API > INPUT

| Column | Type | Max Length | 필수 여부 | Description |
|-------------|--------|------------|-------|-------------|
| 1 sellerKey | String | 100 | Y | 가맹점 코드 |
| 2 cpId | String | 30 | Y | 상점ID |
| 3 productId | String | 50 | N | 상품ID |

표 3-19 결제수단 및 코드조회 API > OUTPUT

| 변수명 | Type | Max Lenth | 설명 | 예 |
|------------------------|--------|-----------|-----------|---|
| 1 usePaymentMethodList | Array | N/A | 사용 결제수단정보 | |
| 2 wholeCardList | Array | N/A | 전체 카드사정보 | |
| 3 wholeBankList | Array | N/A | 전체 은행사정보 | |
| 4 interestFreeUrl | String | 1000 | 무이자안내URL | |
| 5 introduceContents | String | 1000 | 공통소개문구 | |

⇒ usePaymentMethodList (사용 결제수단정보)

| 변수 | type | Max Length | 설명 | 예제 |
|---------------------|--------|------------|----------|----------|
| 1 paymentMethodCode | String | 10 | 결제수단코드 | 98 |
| 2 paymentMethodName | String | 15 | 결제수단명 | PAYCO포인트 |
| usableCardList | Array | N/A | 사용 카드사정보 | |
| usableBankList | Array | N/A | 사용 은행사정보 | |

⇒ usableCardList (사용 카드사정보)

| | 변수 | type | Max Length | 설명 | 예제 |
|---|-----------------|--------|------------|--------|----------|
| 1 | cardCompanyCode | String | 10 | 결제수단코드 | 98 |
| 2 | cardCompanyName | String | 15 | 결제수단명 | PAYCO포인트 |

⇒ usableBankList (사용 은행사정보)

| | 변수 | type | Max Length | 설명 | 예제 |
|---|----------|--------|------------|--------|----------|
| 1 | bankCode | String | 10 | 결제수단코드 | 98 |
| 2 | bankName | String | 15 | 결제수단명 | PAYCO포인트 |

⇒ wholeCardList (전체 카드사정보)

| | 변수 | type | Max Length | 설명 | 예제 |
|---|-----------------|--------|------------|--------|----------|
| 1 | cardCompanyCode | String | 10 | 결제수단코드 | 98 |
| 2 | cardCompanyName | String | 15 | 결제수단명 | PAYCO포인트 |

⇒ wholeBankList (전체 은행사정보)

| | 변수 | type | Max Length | 설명 | 예제 |
|---|----------|--------|------------|--------|----|
| 1 | bankCode | String | 10 | 결제수단코드 | 98 |

| | | | | | |
|---|----------|--------|----|-------|----------|
| 2 | bankName | String | 15 | 결제수단명 | PAYCO포인트 |
|---|----------|--------|----|-------|----------|

- 결제수단 및 코드조회 API > Sample

- 샘플 코드 INPUT

```
{
  "sellerKey": "S0FSJE",
  "cpId": "PARTNERTEST",
  "productId": "PROD_EASY"
}
```

- 샘플 코드 OUTPUT

```
{
  "result": {
    "usePaymentMethodList": [{
      "paymentMethodCode": "98",
      "paymentMethodName": "PAYCO 포인트"
    }, {
      "paymentMethodCode": "75",
      "paymentMethodName": "PAYCO 쿠폰"
    }, {
      "paymentMethodCode": "76",
      "paymentMethodName": "카드 쿠폰"
    }, {
      "paymentMethodCode": "77",
      "paymentMethodName": "가맹점 쿠폰"
    }, {
      "paymentMethodCode": "01",
      "paymentMethodName": "신용카드"
    }, {
      "paymentMethodCode": "02",
      "paymentMethodName": "무통장입금",
      "usableBankList": [{
        "bankCode": "23",
        "bankName": "SC제일은행"
      }, {
        "bankCode": "88",
        "bankName": "신한은행"
      }, {
        "bankCode": "20",
        "bankName": "우리은행"
      }, {
        "bankCode": "11",
        "bankName": "NH농협"
      }, {
        "bankCode": "04",
        "bankName": "KB국민은행"
      }, {
        "bankCode": "81",
        "bankName": "하나은행"
      }, {
        "bankCode": "05",
```

```

        "bankName": "외환은행"
      }, {
        "bankCode": "03",
        "bankName": "IBK기업은행"
      }, {
        "bankCode": "27",
        "bankName": "씨티은행"
      }, {
        "bankCode": "32",
        "bankName": "부산은행"
      }, {
        "bankCode": "39",
        "bankName": "경남은행"
      }, {
        "bankCode": "31",
        "bankName": "대구은행"
      }, {
        "bankCode": "34",
        "bankName": "광주은행"
      }, {
        "bankCode": "71",
        "bankName": "우체국"
      }
    ]
  }, {
    "paymentMethodCode": "04",
    "paymentMethodName": "계좌이체"
  }, {
    "paymentMethodCode": "52",
    "paymentMethodName": "휴대폰 결제 (자동결제)"
  }, {
    "paymentMethodCode": "31",
    "paymentMethodName": "신용카드",
    "usableCardList": [{
      "cardCompanyCode": "CCLG",
      "cardCompanyName": " 신한카드"
    }, {
      "cardCompanyCode": "CCKM",
      "cardCompanyName": "KB국민카드"
    }, {
      "cardCompanyCode": "CCBC",
      "cardCompanyName": "BC카드"
    }, {
      "cardCompanyCode": "CCSS",
      "cardCompanyName": "삼성카드"
    }, {
      "cardCompanyCode": "CCDI",
      "cardCompanyName": "현대카드"
    }, {
      "cardCompanyCode": "CCNH",
      "cardCompanyName": "NH농협카드"
    }, {
      "cardCompanyCode": "CCHN",
      "cardCompanyName": "하나카드"
    }, {
      "cardCompanyCode": "CCLO",
      "cardCompanyName": "롯데카드"
    }, {
      "cardCompanyCode": "CCCT",

```

```
        "cardCompanyName": "씨티카드"
      }, {
        "cardCompanyCode": "CCJB",
        "cardCompanyName": "전북카드"
      }, {
        "cardCompanyCode": "CCKJ",
        "cardCompanyName": "광주카드"
      }, {
        "cardCompanyCode": "CCSU",
        "cardCompanyName": "수협카드"
      }, {
        "cardCompanyCode": "CCPH",
        "cardCompanyName": "우리카드"
      }, {
        "cardCompanyCode": "CCSC",
        "cardCompanyName": "SC카드"
      }, {
        "cardCompanyCode": "CCCJ",
        "cardCompanyName": "제주카드"
      }
    ], {
      "paymentMethodCode": "35",
      "paymentMethodName": "계좌간편결제",
      "usableBankList": [{
        "bankCode": "23",
        "bankName": "SC제일은행"
      }, {
        "bankCode": "11",
        "bankName": "NH농협"
      }, {
        "bankCode": "04",
        "bankName": "KB국민은행"
      }, {
        "bankCode": "81",
        "bankName": "KEB 하나은행"
      }, {
        "bankCode": "03",
        "bankName": "IBK기업은행"
      }, {
        "bankCode": "27",
        "bankName": "씨티은행"
      }, {
        "bankCode": "02",
        "bankName": "산업은행"
      }, {
        "bankCode": "32",
        "bankName": "부산은행"
      }, {
        "bankCode": "39",
        "bankName": "경남은행"
      }, {
        "bankCode": "31",
        "bankName": "대구은행"
      }, {
        "bankCode": "37",
        "bankName": "전북은행"
      }, {
        "bankCode": "45",
```

```

        "bankName": "새마을금고"
      }, {
        "bankCode": "34",
        "bankName": "광주은행"
      }, {
        "bankCode": "35",
        "bankName": "제주은행"
      }, {
        "bankCode": "48",
        "bankName": "신협"
      }, {
        "bankCode": "07",
        "bankName": "수협은행"
      }, {
        "bankCode": "247",
        "bankName": "NH투자증권"
      }
    ]
  }, {
    "paymentMethodCode": "60",
    "paymentMethodName": "휴대폰 간편결제"
  }
],
"wholeCardList": [
  {
    "cardCompanyCode": "CCBC",
    "cardCompanyName": "BC카드"
  },
  {
    "cardCompanyCode": "CCSB01",
    "cardCompanyName": "IBK기업은행"
  },
  {
    "cardCompanyCode": "CCKM",
    "cardCompanyName": "KB국민카드"
  },
  {
    "cardCompanyCode": "CCNH",
    "cardCompanyName": "NH농협카드"
  },
  {
    "cardCompanyCode": "CCSB09",
    "cardCompanyName": "경남은행"
  },
  {
    "cardCompanyCode": "CCKJ",
    "cardCompanyName": "광주카드"
  },
  {
    "cardCompanyCode": "CCSB16",
    "cardCompanyName": "교보증권"
  },
  {
    "cardCompanyCode": "CCSB02",
    "cardCompanyName": "농축협"
  },
  {
    "cardCompanyCode": "CCSB03",
    "cardCompanyName": "농협농촌사랑"
  },
  {
    "cardCompanyCode": "CCSB07",
    "cardCompanyName": "대구은행"
  },
  {
    "cardCompanyCode": "CCSB10",
    "cardCompanyName": "동부증권"
  },
  {
    "cardCompanyCode": "CCAM",
    "cardCompanyName": "롯데아멕스카드"
  },
  {

```

```
        "cardCompanyCode": "CCL0",
        "cardCompanyName": "롯데카드"
    }, {
        "cardCompanyCode": "CCSB11",
        "cardCompanyName": "미래에셋증권"
    }, {
        "cardCompanyCode": "CCSB08",
        "cardCompanyName": "부산은행"
    }, {
        "cardCompanyCode": "CCKD",
        "cardCompanyName": "산업카드"
    }, {
        "cardCompanyCode": "CCSS",
        "cardCompanyName": "삼성카드"
    }, {
        "cardCompanyCode": "CCSB15",
        "cardCompanyName": "새마을금고중앙회"
    }, {
        "cardCompanyCode": "CCSU",
        "cardCompanyName": "수협카드"
    }, {
        "cardCompanyCode": "CCSC",
        "cardCompanyName": "스탠다드차타드"
    }, {
        "cardCompanyCode": "CCSG",
        "cardCompanyName": "신세계한미"
    }, {
        "cardCompanyCode": "CCSH",
        "cardCompanyName": "신한카드"
    }, {
        "cardCompanyCode": "CCLG",
        "cardCompanyName": "신한카드"
    }, {
        "cardCompanyCode": "CCCU",
        "cardCompanyName": "신협카드"
    }, {
        "cardCompanyCode": "CCCT",
        "cardCompanyName": "씨티카드"
    }, {
        "cardCompanyCode": "CCSB06",
        "cardCompanyName": "씨티카드 (비자)"
    }, {
        "cardCompanyCode": "CCKE",
        "cardCompanyName": "외환카드"
    }, {
        "cardCompanyCode": "CCPH",
        "cardCompanyName": "우리카드"
    }, {
        "cardCompanyCode": "CCSB12",
        "cardCompanyName": "우체국"
    }, {
        "cardCompanyCode": "CCSB17",
        "cardCompanyName": "유안타증권"
    }, {
        "cardCompanyCode": "CCSB18",
        "cardCompanyName": "유진투자증권"
    }, {
        "cardCompanyCode": "CCUF",
```



```

        "cardCompanyName": "은련카드"
      }, {
        "cardCompanyCode": "CCSB14",
        "cardCompanyName": "저축은행중앙회"
      }, {
        "cardCompanyCode": "CCSB",
        "cardCompanyName": "저축카드"
      }, {
        "cardCompanyCode": "CCJB",
        "cardCompanyName": "전북카드"
      }, {
        "cardCompanyCode": "CCCJ",
        "cardCompanyName": "제주카드"
      }, {
        "cardCompanyCode": "CCSB04",
        "cardCompanyName": "조합농촌사랑"
      }, {
        "cardCompanyCode": "CCHN",
        "cardCompanyName": "하나SK카드"
      }, {
        "cardCompanyCode": "CCSB13",
        "cardCompanyName": "하나비자"
      }, {
        "cardCompanyCode": "CCHM",
        "cardCompanyName": "한미카드"
      }, {
        "cardCompanyCode": "CJCF",
        "cardCompanyName": "해외JCB"
      }, {
        "cardCompanyCode": "CMCF",
        "cardCompanyName": "해외마스터"
      }, {
        "cardCompanyCode": "CVSF",
        "cardCompanyName": "해외비자"
      }, {
        "cardCompanyCode": "CCHS",
        "cardCompanyName": "현대증권"
      }, {
        "cardCompanyCode": "CCDI",
        "cardCompanyName": "현대카드"
      }
    ],
    "wholeBankList": [{
      "bankCode": "02",
      "bankName": "산업은행"
    }, {
      "bankCode": "03",
      "bankName": "기업은행"
    }, {
      "bankCode": "04",
      "bankName": "국민은행"
    }, {
      "bankCode": "05",
      "bankName": "외환은행"
    }, {
      "bankCode": "07",
      "bankName": "수협"
    }, {
      "bankCode": "11",

```

```
    "bankName": "농협"
  }, {
    "bankCode": "20",
    "bankName": "우리은행"
  }, {
    "bankCode": "209",
    "bankName": "동양종합금융증권"
  }, {
    "bankCode": "218",
    "bankName": "현대증권"
  }, {
    "bankCode": "23",
    "bankName": "SC제일은행"
  }, {
    "bankCode": "230",
    "bankName": "미래에셋증권"
  }, {
    "bankCode": "238",
    "bankName": "대우증권"
  }, {
    "bankCode": "240",
    "bankName": "삼성증권"
  }, {
    "bankCode": "243",
    "bankName": "한국투자증권"
  }, {
    "bankCode": "247",
    "bankName": "NH투자증권"
  }, {
    "bankCode": "26",
    "bankName": "신한은행"
  }, {
    "bankCode": "261",
    "bankName": "교보증권"
  }, {
    "bankCode": "262",
    "bankName": "하이투자증권"
  }, {
    "bankCode": "263",
    "bankName": "HMC투자증권"
  }, {
    "bankCode": "264",
    "bankName": "키움증권"
  }, {
    "bankCode": "265",
    "bankName": "이트레이드증권"
  }, {
    "bankCode": "266",
    "bankName": "SK증권"
  }, {
    "bankCode": "267",
    "bankName": "대신증권"
  }, {
    "bankCode": "268",
    "bankName": "솔로몬투자증권"
  }, {
    "bankCode": "269",
    "bankName": "한화증권"
```

```

    }, {
      "bankCode": "27",
      "bankName": "씨티은행"
    }, {
      "bankCode": "270",
      "bankName": "하나대투증권"
    }, {
      "bankCode": "278",
      "bankName": "신한금융투자"
    }, {
      "bankCode": "279",
      "bankName": "동부증권"
    }, {
      "bankCode": "280",
      "bankName": "유진투자증권"
    }, {
      "bankCode": "287",
      "bankName": "메리츠종합금융증권"
    }, {
      "bankCode": "289",
      "bankName": "NH투자증권"
    }, {
      "bankCode": "290",
      "bankName": "부국증권"
    }, {
      "bankCode": "291",
      "bankName": "신영증권"
    }, {
      "bankCode": "292",
      "bankName": "LIG투자증권"
    }, {
      "bankCode": "31",
      "bankName": "대구은행"
    }, {
      "bankCode": "32",
      "bankName": "부산은행"
    }, {
      "bankCode": "34",
      "bankName": "광주은행"
    }, {
      "bankCode": "35",
      "bankName": "제주은행"
    }, {
      "bankCode": "37",
      "bankName": "전북은행"
    }, {
      "bankCode": "39",
      "bankName": "경남은행"
    }, {
      "bankCode": "45",
      "bankName": "새마을금고"
    }, {
      "bankCode": "48",
      "bankName": "신협"
    }, {
      "bankCode": "50",
      "bankName": "상호신용금고"
    }, {

```

```
        "bankCode": "54",
        "bankName": "HSBC은행"
    }, {
        "bankCode": "55",
        "bankName": "도이치은행"
    }, {
        "bankCode": "71",
        "bankName": "우체국"
    }, {
        "bankCode": "81",
        "bankName": "KEB 하나은행"
    }, {
        "bankCode": "88",
        "bankName": "신한은행"
    }, {
        "bankCode": "89",
        "bankName": "케이뱅크"
    }, {
        "bankCode": "90",
        "bankName": "카카오뱅크"
    }
    ],
    "interestFreeUrl": "https://alpha-
bill.payco.com/outseller/freeInterest/S0FSJE/PARTNERTEST",
    "introduceContents": ""
},
"code": 0,
"message": "success"
}
```

3.3 외부 가맹점 API 상세

외부가맹점은 아래 API를 제공해야 합니다.

API 스펙은 PAYCO에서 정의한 대로 개발되어 있어야 합니다

3.3.3 추가 배송비 조회

- API URL은 PAYCO 쪽에 미리 등록 요청이 필요 하다.
- 해당 API 는 체크아웃 형의 경우 배송지 정보 입력 시 호출 된다.
- 우편번호 기준으로 주문의 추가 배송비를 조회한다.
- 요청 정보는 "response"라는 이름의 parameter로 제공되며, parameter value를 UTF-8 urldecode하면 Json형태의 값을 받아 볼 수 있다.
- 결과는 Json으로 전달한다.
- < 추가배송비 과세처리방법 >
 - 주문예약정보 100% 과세금액인 경우 추가배송비도 과세처리
 - 주문예약정보 100% 비과세금액인 경우 추가배송비도 비과세처리
 - 주문예약정보 과세+비과세 섞인 경우 추가배송비 비과세처리

• 표 3-20 추가 배송비 조회 > INPUT

| 변수명 | Type | Max Lenth | 설명 | 예 |
|-----------------------------------|--------|-----------|-------------------------|--------------------|
| 1 zipcode | String | 50 | 우편번호 | 123-134 |
| 2 address1 | String | 300 | 기본주소 | 경기도 성남시 분당구 삼 평동 |
| 3 address2 | String | 300 | 상세주소 | 629 플레이뮤지엄 |
| 4 sellerOrderenceReferenceKey | String | 100 | 외부가맹점에서 관리 하는 주문연동Key | 201408212212505952 |
| 5 sellerOrderProductReferenceKeys | String | 100 | 외부가맹점에서 관리 하는 주문상품연동Key | 505952 |

• 표 3-21 추가 배송비 조회 > OUTPUT

| 변수명 | Type | Max Length | 설명 | 필수여부 | 예 |
|---------------------------------|--------|------------|-------------------------|------|---------|
| 1 resultCode | Number | | 결과코드 (0이면 성공, 그외는 Y 실패) | Y | 0 |
| 2 resultMessage | String | | 결과메시지 | N | 실패하였습니다 |
| 3 totalRemoteAreaDeliveryFeeAmt | Number | | 총 도서산간비용 | Y | 5500 |

• Sample

➤ 샘플 URL

```
https://alpha-api-bill.payco.com/sample/outseller/getRemoteAreaDeliveryFee?response=%7B%22zipcode%22%3A%22463-400%22%2C%22sellerOrderenceReferenceKey%22%3A%22201408212212505952%22%2C%22sellerOrderProductReferenceKeys%22%3A%5B%221918597%22%5D%7D
```

➤ result

```
{
  "resultCode": 0,
  "resultMessage": "success",
  "totalRemoteAreaDeliveryFeeAmt": 1000
}
```

3.3.4 주문 완료 통보 (결제모드 PAY1 타입)

- 주문예약 API 연동 시 파라미터로 'serviceUrl'을 전달하게 된다.
- 'serviceUrl'은 주문완료통보 API의 URL로 외부가맹점이 제공하는 API 이다.
- PAYCO는 주문완료 또는 결제완료 시점에 외부가맹점의 serviceUrl API를 호출한다.
- server To server 로 호출이 된다.
- 오류 인 경우 'message' 라는 이름의 parameter에 오류 메시지가 전달된다.
- 주문 관련 정보는 'response' 라는 이름의 parameter로 제공되며, Json형태의 값을 받아볼 수 있다.

- 캐릭터셋은 UTF-8이며, POST로 전달한다.
- 자세한 항목은 아래 'serviceUrl'로 전달되는 정보를 참고한다.
- serviceUrl 결과로 성공은 'OK' 실패는 'ERROR'를 리턴 한다.
- OK외에 다른 문자열이 리턴 되면 주문 완료 창에서 아래의 문구가 노출 된다.

“결제는 완료되었으나 판매자 연동에 실패했습니다”

- [참고\) 가맹점에서는 주문 완료 통보 API에서 재고 체크를 하여 재고가 없는 경우 응답으로 실패 코드를 리턴 해야 한다.](#)
- 재고 부족 등으로 주문을 완료할 수 없는 상태인 경우에는 취소API를 호출하여 취소처리를 진행한다.
- 가맹점에서는 총 결제금액 (totalPaymentAmt)이 요청금액과 일치하는지 확인한다.
- 무통장입금인 경우 결제 완료여부가 N으로 전달되며, 입금 완료 시 동일한 포맷으로 nonBankbookDepositInformUrl로 입금 완료통보를 한다. (이때의 결제 완료여부는 Y)
- **추가 배송비**
 - 추가 배송비가 발생 한 경우 배송비 상품의 productPaymentAmt에 추가 배송비를 합산 한 금액이 전송된다.

• **표 3-22 ServiceUrl로 전달되는 정보**

- totalOrderAmt(총 주문금액) + totalDeliveryFeeAmt(총 배송비) + totalRemoteAreaDeliveryFeeAmt(총 도서산간비용) = 총 결제금액 (totalPaymentAmt)

| 변수명 | Type | Max Lenth | 설명 | 예 |
|---------------------------|--------|-----------|------------------------|----------------------------|
| 1 sellerOrderReferenceKey | String | 100 | 외부가맹점에서 발급하는 주문 연동 Key | 팝슈즈 예)20140821221250595 |
| 2 reserveOrderNo | String | 20 | 주문예약번호 | 201408211000000720 |
| 3 orderNo | String | 20 | 주문번호 | 20140821100000072 |
| 4 memberName | String | 15 | 주문자명(간편결제형만 일부 마스킹처리) | |

3. PAYCO 연동하기

| | | | | | | |
|----|----------------------------------------|-------------------|------|------------------------------------|-------------|--|
| 5 | memberEmail | String | 200 | 주문자EMAIL(일부 마스킹처리) | | |
| 6 | orderChannel | String | N/A | 주문채널 (PC/MOBILE) | PC | |
| 7 | totalOrderAmt | Number | | 총 주문금액 | 116500 | |
| 8 | totalDeliveryFeeAmt | Number | | 총 배송비 | 2500 | |
| 9 | totalRemoteAreaDeliveryFeeAmt | Number | | 총 도서산간비 (추가배송비) | 0 | |
| 10 | totalPaymentAmt | Number | | 총 결제금액 | 11900 | |
| 11 | paymentCompletionYn | String | 1 | 결제완료여부 (Y/N) | Y | |
| 12 | deliveryPlace | Object | N/A | 배송지 정보 | | |
| 13 | orderProducts | Array | N/A | 주문상품 List | | |
| 14 | paymentDetails | Array | N/A | 결제내역 List | | |
| 15 | serviceUrlParam | String | 1000 | 주문예약 API호출시 serviceUrlParam 항목 전달한 | | |
| 16 | orderCertifyKey | String | 100 | 주문인증키 | | |
| 17 | receiptAmt | Number | | 현금영수증 발급 대상 금액 | 0 | |
| 18 | receiptPaycoPointAmt | Number | | 현금영수증 발급 대상 포인트 결제금액 | Deprecat ed | |
| 19 | receiptPaycoPointTaxfreeAmt | Number | | 현금영수증 발급 대상 포인트 면세금액 | Deprecat ed | |
| 20 | receiptPaycoPointTaxableAmt | Number | | 현금영수증 발급 대상 포인트 과세금액 | Deprecat ed | |
| 21 | receiptPaycoPointVatAmt | Number | | 현금영수증 발급 대상 포인트 부가세 | Deprecat ed | |
| 22 | receiptPaycoPointServiceAmt | Number | | 현금영수증 발급 대상 포인트 봉사료 | Deprecat ed | |

deliveryPlace (배송지)

| 변수 | type | Max Length | 설명 | 예제 |
|----|------|------------|----|----|
|----|------|------------|----|----|

| | | | | | |
|---|------------------------|--------|-----|---------|-----------------|
| 1 | recipient | String | 15 | 수령인 | 홍길동 |
| 2 | englishReceipent | String | 50 | 수령인 영문명 | LEEJINHEE |
| 3 | address1 | String | 300 | 기본주소 | 경기도 성남시 분당구 삼평동 |
| 4 | address2 | String | 300 | 상세주소 | 629 플레이뮤지엄 |
| 5 | zipcode | String | 50 | 우편번호 | 463-400 |
| 6 | deliveryMemo | String | 30 | 배송메모 | 경비실에 맡겨주세요 |
| 7 | telephone | String | 15 | 연락처 | 010-1111-2222 |
| 8 | individualCustomUniqNo | String | 50 | 개인통관번호 | |

⇒ **orderProducts (주문상품List)**

| 변수 | Type | Max Length | 설명 | 예제 | |
|----|--------------------------------|------------|-----|----------------------------------------------------------------|-------------------|
| 1 | orderProductNo | String | 20 | 주문상품번호 | 20140821100000022 |
| 2 | cpId | String | 30 | 상점ID | PS_TOASTCO |
| 3 | productId | String | 50 | 상품ID | PS_TOASTCOMC1 |
| 4 | sellerOrderProductReferenceKey | String | 100 | 외부가맹점에서 발급한 주문상품연동Key | 19801895 |
| 5 | orderProductStatusCode | String | 50 | 주문상품 상태코드 | PAYED |
| 6 | orderProductStatusName | String | 10 | 주문상품 상태명 | 결제완료 |
| 7 | productKindCode | String | | 상품타입 | GENERAL_PRODUCT |
| 8 | productPaymentAmt | Float | | 상품결제금액 : 배송비 5000 상품의 경우 (원 상품결제금액 + 추가 배송비) 로 변경 될 수 있음 | |
| 9 | originalProductPaymentAmt | Float | | 원 상품결제금액 (배송비 상품의 경우 배송비 금액) | 2000 |

3. PAYCO 연동하기

주문 상품 List(orderProducts) 부가 설명

- 추가 배송비가 발생한 경우, productPaymentAmt 에 추가 배송비와 일반 배송비가 합산 된 금액으로 전달 된다.

- 예) 배송비 2000 , 추가 배송비 3000원일 경우

배송비 상품 orderProducts > originalProductPaymentAmt = 2000원

배송비 상품 orderProducts > productPaymentAmt = 5000원

⇒ paymentDetails (결제내역 List)

| 변수 | Type | Max Length | 설명 | 예제 |
|--------------------------------------|--------|------------|-----------------------------|--------------------|
| 1 paymentTradeNo | String | 20 | 결제 번호 | 201409191000001142 |
| 2 paymentMethodCode | String | 10 | 결제수단코드 | 01 |
| 3 paymentMethodName | String | 15 | 결제수단명 | 신용카드 |
| 4 paymentAmt | Number | | 결제금액 | 119000 |
| 5 taxfreeAmt | Number | | 면세금액 | 0 |
| 6 taxableAmt | Number | | 과세금액 | 0 |
| 7 vatAmt | Number | | 부가세 | 0 |
| 8 serviceAmt | Number | | 봉사료 | 0 |
| 9 tradeYmdt | String | 14 | 결제일시 (yyyyMMddHHmmss) | 20140822120201 |
| 10 pgAdmissionNo | String | 100 | 원천사승인번호 | 209380 |
| 11 pgAdmissionYmdt | String | 14 | 원천사승인일시 (yyyyMMddHHmmss) | 20140822120201 |
| 12 easyPaymentYn | String | 1 | 간편결제여부 (Y/N) | N |
| 13 cardSettleInfo | Object | N/A | 신용카드 결제 정보 | ↓ |
| 14 cellphoneSettleInfo | Object | N/A | 핸드폰 결제 정보 | ↓ |
| 15 realtimeAccountTransferSettleInfo | Object | N/A | 실시간계좌이체 결제정보 | ↓ |
| 16 nonBankbookSettleInfo | Object | N/A | 무통장입금 결제정보 | ↓ |

| | | | | |
|----|-------------------------|--------|-----|------|
| 17 | couponSettleInfo | Object | N/A | 쿠폰정보 |
|----|-------------------------|--------|-----|------|

⇒ **cardSettleInfo (카드결제정보)**

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------------------------|--------|------------|------------------------|-----------|
| 1 | cardCompanyName | String | 15 | 카드사명 | 하나SK |
| 2 | cardCompanyCode | String | 50 | 카드사코드 | CCSK |
| 3 | cardNo | String | 300 | 카드번호 | *****1809 |
| 4 | cardInstallmentMonthNumber | String | 2 | 할부개월(MM) | 00 |
| 5 | cardAdmissionNo | String | 20 | 카드사 승인번호 | 31213333 |
| 6 | cardInterestFreeYn | String | 1 | 무이자여부(Y/N) | Y |
| 7 | corporateCardYn | String | 1 | 법인카드여부 (개인 N ,법인 Y) | Y |
| 8 | partCancelPossibleYn | String | 1 | 부분취소가능유무(Y/N) | Y |

⇒ **cellphoneSettleInfo (핸드폰 결제 정보)**

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|-------------|--------|------------|--------|---------------|
| 1 | cellphoneNo | String | 32 | 휴대폰 번호 | 010-1111-2222 |

⇒ **realtimeAccountTransferSettleInfo (실시간계좌이체,바로이체 결제정보)**

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------|--------|------------|----------------------|----|
| 1 | bankName | String | 15 | 은행명 | 신한 |
| 2 | bankCode | String | 50 | 은행코드 | 88 |

⇒ **nonBankbookSettleInfo (무통장입금 결제정보)**

3. PAYCO 연동하기

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------------------|--------|------------|----------------------|---------------|
| 1 | bankName | String | 15 | 은행명 | 신한 |
| 2 | bankCode | String | 50 | 은행코드 | 88 |
| 3 | accountNo | String | 200 | 계좌번호 | 1901801980225 |
| 4 | paymentExpirationYmd | String | 8 | 입금만료일 | 20141208 |

⇒ couponSettleInfo (쿠폰정보)

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|----------------------|--------|------------|--------|-------|
| 1 | discountAmt | String | 15 | 쿠폰할인금액 | 10000 |
| 2 | discountConditionAmt | String | 50 | 쿠폰할인조건 | 50000 |

• Sample

➤ Callback Json 예제

```
{
  "sellerOrderReferenceKey": "TEST848891082710",
  "reserveOrderNo": "201706232004537366",
  "orderNo": "201706232002275277",
  "memberName": "강**",
  "memberEmail": "na****",
  "orderChannel": "PC",
  "totalOrderAmt": 81500,
  "totalDeliveryFeeAmt": 0,
  "totalRemoteAreaDeliveryFeeAmt": 0,
  "totalPaymentAmt": 81500,
  "receiptPaycoPointAmt": 0,
  "receiptPaycoPointTaxfreeAmt": 0,
  "receiptPaycoPointTaxableAmt": 0,
  "receiptPaycoPointVatAmt": 0,
  "receiptPaycoPointServiceAmt": 0,
  "orderProducts": [{
    "orderProductNo": "201706232002428782",
    "sellerOrderProductReferenceKey": "ITEM_100001",
    "orderProductStatusCode": "OPSPAED",
    "orderProductStatusName": "결제완료",
    "cpId": "PARTNERTEST",
    "productId": "PROD_EASY",
    "productKindCode": "GENERAL_PRODUCT",
    "productPaymentAmt": 81500,
    "originalProductPaymentAmt": 81500
  }],
  "paymentDetails": [{
    "couponSettleInfo": {
```

```

        "discountAmt": 2000.0,
        "discountConditionAmt": 7000.0
    },
    "paymentTradeNo": "201706232002995627",
    "paymentMethodCode": "75",
    "paymentMethodName": "PAYCO 쿠폰",
    "paymentAmt": 2000,
    "taxfreeAmt": 0,
    "taxableAmt": 1820,
    "vatAmt": 180,
    "serviceAmt": 0,
    "tradeYmdt": "20170623150040",
    "pgAdmissionNo": "34540",
    "pgAdmissionYmdt": "20170623150039",
    "easyPaymentYn": "N"
}, {
    "paymentTradeNo": "201706232002995628",
    "paymentMethodCode": "98",
    "paymentMethodName": "PAYCO 포인트",
    "paymentAmt": 10000,
    "taxfreeAmt": 0,
    "taxableAmt": 9090,
    "vatAmt": 910,
    "serviceAmt": 0,
    "tradeYmdt": "20170623150040",
    "pgAdmissionNo": "2017062320989132",
    "pgAdmissionYmdt": "20170623150039",
    "easyPaymentYn": "N"
}, {
    "cardSettleInfo": {
        "cardCompanyName": "삼성카드",
        "cardCompanyCode": "CCSS",
        "cardNo": "941010*****2222",
        "cardInstallmentMonthNumber": "00",
        "cardAdmissionNo": "62153184",
        "cardInterestFreeYn": "N",
        "corporateCardYn": "N",
        "partCancelPossibleYn": "Y"
    },
    "paymentTradeNo": "201706232002995626",
    "paymentMethodCode": "31",
    "paymentMethodName": "신용카드",
    "paymentAmt": 69500,
    "taxfreeAmt": 0,
    "taxableAmt": 63181,
    "vatAmt": 6319,
    "serviceAmt": 0,
    "tradeYmdt": "20170623150040",
    "pgAdmissionNo": "17424902647103",
    "pgAdmissionYmdt": "20170623150039",
    "easyPaymentYn": "Y"
}],
    "serviceUrlParam":
    "{\\"serviceParam_productId\\":\\"JSP_SAMPLE_EASYPAY_BACK\\",\\"taxationType\\":\\"TAXATIO
N\\",\\"totalTaxfreeAmt\\":\\"0\\",\\"totalTaxableAmt\\":\\"74091\\",\\"totalVatAmt\\":\\"7409\\
"}",
    "orderCertifyKey": "YWUpcSiGuBjEsfJImlyAAmSn14nTAjU4R5l5BBfyL5SdQCC",
    "paymentCompletionYn": "Y",
    "paymentCompleteYmdt": "20170623150040"
}

```

- response 에 대해서는 성공 시 OK, 실패 시 ERROR 문자 외에 다른 값이 오면 처리 불가하므로

가맹점 로직이 성공한 경우 경고성 오류문구가 출력되지 않도록 유의

➤ **Result success**

OK

➤ **Result fail**

ERROR

3.3.5 결제 인증 완료 (결제모드 PAY2 타입)

- 주문예약 API 연동 시 파라미터로 'returnUrl'을 전달하게 된다.
- 'returnUrl'은 결제인증완료URL로 외부가맹점이 제공하는 페이지이다.
- 결제인증 **성공**시점 또는 **사용자취소**시 외부가맹점의 returnUrl로 이동한다
- 캐릭터셋은 UTF-8이며, GET로 전달한다.(option POST)
- 예약시 returnUrlParam 값에 셋팅된 정보가 존재시 해당값을 추가하여 returnUrl로 호출한다.
- 자세한 항목은 아래 'returnUrl'로 전달되는 정보'를 참고한다.
- returnUrl에서는 결제팝업창을 close한 다음 주문이 가능할 경우 결제승인처리API를 호출하여 주문이 완료 (결제완료) 될 수 있도록 한다.
- 가맹점에서는 총 결제금액 (totalPaymentAmt)과 외부가맹점번호가 일치하는지 확인한다.

• 표 3-23 returnUrl로 전달되는 정보

| 변수명 | Type | Max Length | 설명 | 예 |
|----------------------------------|--------|------------|----------------------|-----------------------------------------------|
| 1 reserveOrderNo | String | 20 | 주문예약번호 | 201408211000000720 |
| 2 sellerOrderReferenceKey | String | 100 | Y | 외부가맹점에서 관리하는 주문연동Key |
| 3 paymentCertifyToken | String | 100 | 결제인증토큰 - 결제승인시 필요 | 9FAFF0967482F8DE25AB9 2716C073842335235235 |
| 4 totalPaymentAmt | Number | | 총결제금액 | 119000 |
| 5 discountAmt | Number | | 쿠폰할인금액 | 5000 (PAYCO포인트 미포함) |
| 6 totalRemoteAreaDeliveryFee Amt | Number | | 총 도서산간비 (추가배송비) | 3000 |

| | | | | |
|---|------------|--------|---------------|---------------------------------------------------|
| 7 | pointAmt | Number | PAYCO포인트 사용금액 | |
| 8 | mainPgCode | Number | 주 결제수단 코드 | 31 |
| 9 | code | Number | 결과코드 | 0 : 성공 2222 : 사용자에 의한 취소(close) 그 외 코드 : 오류 |

➤ returnUrl Param

```
code=0&
reserveOrderNo=201706232004537278&
sellerOrderReferenceKey=TEST42827779108&
mainPgCode=31&
totalPaymentAmt=81500&
totalRemoteAreaDeliveryFeeAmt=0&
discountAmt=2000&
pointAmt=10000&
paymentCertifyToken=Sg3P_nctRELYNi7qppJiW50M%2FryDDz8u3Ueht5osiXgcKN53n8anjHORfbyFS%2FAN&
bid=R6ALSJWB6YLFK0F0954O25GJI
```

3.3.6 무통장 입금 완료 통보

- 주문예약 API 연동 시 파라미터로 nonBankbookDepositInformUrl 전달하게 된다.
- Payco는 무통장입금 결제완료 시점에 nonBankbookDepositInformUrl API를 호출한다.
- server To server 로 호출이 된다.
- 결제 관련 정보는 'response' 라는 이름의 parameter로 제공되며, parameter value를 UTF-8 UrlDecode하면 Json형태의 값을 받아볼수있다.
- 주문완료통보 API와 동일한 포맷으로 전달된다.
- 자세한 항목은 아래 nonBankbookDepositInformUrl 로 전달되는 정보'를 참고한다.
- nonBankbookDepositInformUrl 결과로 성공은 'OK' 실패는 'ERROR'를 리턴한다.

• 표 3-24 NonBankbookDepositInformUrl 로 전달되는 정보

| 변수명 | Type | Max Length | 설명 | 예 |
|---------------------------|--------|------------|------------------------|----------------|
| 1 sellerOrderReferenceKey | String | 100 | 외부가맹점에서 발급하는 주문 연동 Key | 팝슈즈 예)20140 |

3. PAYCO 연동하기

| | | | | | |
|----|-----------------------------------|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| | | | | | 8212212 505952 |
| 2 | reserveOrderNo | String | 20 | 주문예약번호 | 2014082 1100000 0720 |
| 3 | orderNo | String | 20 | 주문번호 - 빌링결제 이후 Payco에서 발급되는 Key이다. - reserverOrderNo 주문예약번호는 orderNo 를 발급받기 전까지의 임의의 Key라고 생각하면 된다. - orderNo 가 발급된 이후에는 Payco와 orderNo로 통신을 하게 된다. (또는 sellerOrderReferenceKey) - 주문예약 시 자동결제 완료여부를 N으로 한 경우, 결제수단이 무통장입금이 아니면 주문번 호가 전달되지 않는다. (결제 완료 API의 응답 값으로 전달 됨) | 2014082 1100000 072 |
| 4 | memberName | String | 15 | 주문자명(간편결제형만 일부 마스킹처리) | |
| 5 | memberEmail | String | 200 | 주문자EMAIL(일부 마스킹처리) | |
| 6 | orderChannel | String | N/A | 주문채널 (PC/MOBILE) | PC |
| 7 | totalOrderAmt | Number | | 총주문금액 | 116500 |
| 8 | totalDeliveryFeeAmt | Number | | 총배송비 | 2500 |
| 9 | totalRemoteAreaDeliveryFee Amt | Number | | 총도서산간비 | 0 |
| 10 | totalPaymentAmt | Number | | 총결제금액 | 119000 |
| 11 | paymentCompletionYn | String | 1 | 결제완료여부 (Y/N) | Y |
| 12 | deliveryPlace | Object | N/A | 배송지정보 | |
| 13 | orderProducts | Array | N/A | 주문상품 List | |
| 14 | paymentDetails | Array | N/A | 결제내역 List | |
| 15 | serviceUrlParam | String | 1000 | 주문예약 API호출시 전달한 항목 | serviceUrlParam |

| | | | | | |
|----|-----------------------------|--------|-----|----------------------|----------------|
| 16 | orderCertifyKey | String | 100 | 주문완료통보시 내려받은 인증값 | |
| 17 | receiptAmt | Number | | 현금영수증 발급 대상 금액 | 0 |
| 18 | receiptPaycoPointAmt | Number | | 현금영수증 발급 대상 포인트 결제금액 | Deprecat ed |
| 19 | receiptPaycoPointTaxfreeAmt | Number | | 현금영수증 발급 대상 포인트 면세금액 | Deprecat ed |
| 20 | receiptPaycoPointTaxableAmt | Number | | 현금영수증 발급 대상 포인트 과세금액 | Deprecat ed |
| 21 | receiptPaycoPointVatAmt | Number | | 현금영수증 발급 대상 포인트 부가세 | Deprecat ed |
| 22 | receiptPaycoPointServiceAmt | Number | | 현금영수증 발급 대상 포인트 봉사료 | Deprecat ed |

- $totalOrderAmt$ (총 주문금액) + $totalDeliveryFeeAmt$ (총 배송비) + $totalRemoteAreaDeliveryFeeAmt$ (총 도서산간비용) = 총 결제금액 ($totalPaymentAmt$)

⇒ 표 3-20 12번 컬럼 deliveryPlace (배송지)

| 변수 | Type | Max Length | 설명 | 예제 |
|--------------------------|--------|------------|--------|-----------------|
| 1 recipient | String | 15 | 수령인 | 홍길동 |
| 2 englishRecipient | String | 50 | 수령인영문명 | LEEJINHEE |
| 3 address1 | String | 300 | 기본주소 | 경기도 성남시 분당구 삼평동 |
| 4 address2 | String | 300 | 상세주소 | 629 플레이뮤지엄 |
| 5 zipcode | String | 50 | 우편번호 | 463-400 |
| 6 deliveryMemo | String | 30 | 배송메모 | 경비실에 맡겨주세요 |
| 7 telephone | String | 50 | 연락처 | 010-1111-2222 |
| 8 individualCustomUniqNo | String | 50 | 개인통관번호 | |

⇒ 표 3-20 13번 컬럼 orderProducts (주문상품List)

3. PAYCO 연동하기

| | 변수 | Type | Max Length | 설명 | 예제 |
|---|--------------------------------|--------|---------------|--------------------------|-----------------------|
| 1 | orderProductNo | String | 20 | 주문상품 번호 | 20140821100000 022 |
| 2 | cpId | String | 30 | 상점ID | PS_TOASTCO |
| 3 | productId | String | 50 | 상품ID | PS_TOASTCOMC 1 |
| 4 | sellerOrderProductReferenceKey | String | 100 | 외부가맹점에서 발급한 주문상품연동Key | 19801895 |
| 5 | orderProductStatusCode | String | 50 | 주문상품상태코드 | PAYED |
| 6 | orderProductStatusName | String | 10 | 주문상품상태명 | 결제완료 |

⇒ 표 3-20 14번 컬럼 paymentDetails (결제내역 List)

| | 변수 | Type | Max Length | 설명 | 예제 |
|----|-------------------|--------|---------------|-----------------------------|--------------------|
| 1 | paymentTradeNo | String | 20 | 결제번호 | 201409191000001142 |
| 2 | paymentMethodCode | String | 10 | 결제수단코드 | 01 |
| 3 | paymentMethodName | String | 15 | 결제수단명 | 신용카드 |
| 4 | paymentAmt | Number | | 결제금액 | 119000 |
| 5 | taxfreeAmt | Number | | 면세금액 | 0 |
| 6 | taxableAmt | Number | | 과세금액 | 0 |
| 7 | vatAmt | Number | | 부가세 | 0 |
| 8 | serviceAmt | Number | | 봉사료 | 0 |
| 9 | tradeYmdt | String | 14 | 결제일시 (yyyyMMddHHmmss) | 20140822120201 |
| 10 | pgAdmissionNo | String | 100 | 원천사승인번호 | 209380 |
| 11 | pgAdmissionYmdt | String | 14 | 원천사승인일시 (yyyyMMddHHmmss) | 20140822120201 |

| | | | | | |
|----|-----------------------|--------|-----|--------------|---|
| 12 | easyPaymentYn | String | 1 | 간편결제여부 (Y/N) | N |
| 13 | nonBankbookSettleInfo | Object | N/A | 무통장입금 결제정보 | |

⇒ paymentDetails 12번 컬럼 nonBankbookSettleInfo (무통장입금 결제정보)

| 변수 | Type | Max Length | 설명 | 예제 |
|----|----------------------|------------|-------|---------------|
| 1 | bankName | String 15 | 은행명 | 신한 |
| 2 | bankCode | String 50 | 은행코드 | 88 |
| 3 | accountNo | String 200 | 계좌번호 | 1901801980225 |
| 4 | paymentExpirationYmd | String 8 | 입금만료일 | 20141208 |

• Sample

➤ result Success

OK

➤ result fail

ERROR

➤ Input Json 예시

```
{
  "sellerOrderReferenceKey": "TEST63796357106",
  "reserveOrderNo": "201706232004537280",
  "orderNo": "201706232002275233",
  "memberName": "강**",
  "memberEmail": "na****",
  "orderChannel": "PC",
  "totalOrderAmt": 81500,
  "totalDeliveryFeeAmt": 0,
  "totalRemoteAreaDeliveryFeeAmt": 0,
  "totalPaymentAmt": 81500,
  "receiptPaycoPointAmt": 4,
  "receiptPaycoPointTaxfreeAmt": 0,
  "receiptPaycoPointTaxableAmt": 3,
  "receiptPaycoPointVatAmt": 1,
  "receiptPaycoPointServiceAmt": 0,
  "orderProducts": [{
    "orderProductNo": "201706232002428742",
    "sellerOrderProductReferenceKey": "ITEM 100001",
    "orderProductStatusCode": "OPSPAED",
    "orderProductStatusName": "결제완료",
    "cpId": "PARTNERTTEST",
```

3. PAYCO 연동하기

```
        "productId": "PROD EASY",
        "productKindCode": "GENERAL PRODUCT",
        "productPaymentAmt": 81500,
        "originalProductPaymentAmt": 81500
    }],
    "paymentDetails": [{
        "paymentTradeNo": "201706232002995508",
        "paymentMethodCode": "98",
        "paymentMethodName": "PAYCO 포인트",
        "paymentAmt": 20000,
        "taxfreeAmt": 0,
        "taxableAmt": 18182,
        "vatAmt": 1818,
        "serviceAmt": 0,
        "tradeYmdt": "20170623141459",
        "pgAdmissionNo": "2017062320989082",
        "pgAdmissionYmdt": "20170623141459",
        "easyPaymentYn": "N"
    }, {
        "nonBankbookSettleInfo": {
            "bankName": "기업은행",
            "bankCode": "03",
            "accountNo": "48003470097101"
        },
        "paymentTradeNo": "201706232002995569",
        "paymentMethodCode": "02",
        "paymentMethodName": "무통장입금",
        "paymentAmt": 61500,
        "taxfreeAmt": 0,
        "taxableAmt": 55909,
        "vatAmt": 5591,
        "serviceAmt": 0,
        "tradeYmdt": "20170623143659",
        "pgAdmissionNo": "48003470097101",
        "pgAdmissionYmdt": "20170623023659",
        "easyPaymentYn": "N"
    }],
    "orderCertifyKey": "UEbmIQQhN0BBUeleS8yvPhFVuRLwb1PVXF4BB1RHsEKUYCC",
    "paymentCompletionYn": "Y",
    "paymentCompleteYmdt": "20170623143659"
}
```

3.4 유의사항

- ERROR 코드 9000은 시스템 점검시 내려주는 코드이므로 해당 코드에 대한 메시지는 사용자에게 노출처리 부탁드립니다.
- 결제수단 코드가 추가 될 수 있으니 참고부탁드립니다.
- 모바일웹의 경우 페이코 연동은 페이지 이동방식으로 작업 부탁드립니다.

주문예약API > 부가정보(extraData) > cancelMobileUrl에 취소버튼 클릭시 이동할 URL을 추가로 보내주세요.

3.5 코드 정의

3.5.1 주문 유형(=결제유형)

파트너센터 주문통계페이지 주문유형별 통계제공

| Code | 설명 |
|-----------|-------------------|
| CHECKOUT | 체크아웃 형 |
| EASYPAY | 간편결제형 (가맹점 비회원결제) |
| EASYPAY_F | 간편결제형 (가맹점 회원결제) |

3.5.2 주문상품상태

| Code | 설명 |
|-------------------|----------------|
| PAYMENT_WAITNG | 입금대기 |
| PAYED | 결제완료 (빌링 결제완료) |
| DELIVERY_READY | 배송 준비 중 |
| DELIVERING | 배송 중 |
| DELIVERY_COMPLETE | 배송완료 |
| DELIVERY_START | 배송시작(출고지시) |
| PURCHASE_DECISION | 구매확정 |
| CANCELED | 취소 |
| CANCEL_REQUEST | 취소요청 |
| EXCHANGE_WAIT | 교환접수중 |
| EXCHANGE_COMPLETE | 교환완료 |
| RETURN_WAIT | 반품접수중 |
| RETURN_COMPLETE | 반품완료 |

3.5.3 주문채널 (= 결제채널)

| Code | 설명 |
|--------|--------|
| PC | PC |
| MOBILE | MOBILE |

3.5.4 결제유형

| Code | 설명 |
|-------------------|------|
| ORIGINAL_APPROVAL | 원 결제 |
| WHOLE_CANCEL | 취소 |

3.5.5 결제수단

| Code | 설명 |
|------|----------------|
| 01 | 신용카드(일반) |
| 02 | 무통장입금(일반) |
| 04 | 계좌이체(일반) |
| 05 | 휴대폰(일반) |
| 31 | 신용카드 |
| 35 | 간편계좌 |
| 60 | 휴대폰 |
| 98 | PAYCO 포인트 |
| 75 | 페이코 쿠폰(자유이용쿠폰) |
| 76 | 카드 쿠폰 |
| 77 | 가맹점 쿠폰 |
| 96 | 충전금 환불 |

3.5.6 상품타입

| Code | 설명 |
|-----------------|------|
| GENERAL_PRODUCT | 일반상품 |
| DELIVERY_FEE | 배송비 |

3.5.7 과세타입

| Code | 설명 |
|----------|--------------|
| DUTYFREE | 면세 |
| COMBINE | 결합 (면세 + 과세) |
| TAXATION | 과세 |

3.5.8 카드사코드

: 카드발급사 기준

3. PAYCO 연동하기

| Code | 설명 |
|------|----------|
| CCBC | BC카드 |
| CCKM | KB국민카드 |
| CCNH | NH농협카드 |
| CCKJ | 광주카드 |
| CCAM | 롯데아멕스카드 |
| CCLO | 롯데카드 |
| CCKD | 산업카드 |
| CCSS | 삼성카드 |
| CCSU | 수협카드 |
| CCSG | 신세계한미 |
| CCLG | 신한카드 |
| CCCU | 신협카드 |
| CCCT | 씨티카드 |
| CCPH | 우리카드 |
| CCUF | 은련카드 |
| CCSB | 저축카드 |
| CCJB | 전북카드 |
| CCCJ | 제주카드 |
| CCKE | 하나(외환)카드 |
| CCHN | 하나카드 |
| CCHM | 한미카드 |
| CJCF | 해외JCB |
| CMCF | 해외마스터 |
| CVSF | 해외비자 |
| CCDI | 현대카드 |

3.5.9 통신사코드

| Code | 설명 |
|------|-----|
| CJH | CJH |
| KCT | KCT |
| KFT | KFT |

| Code | 설명 |
|------|-----|
| LGT | LGT |
| SKT | SKT |

3.5.10 은행코드

| Code | 설명 |
|------|----------|
| 11 | 농협 |
| 20 | 우리은행 |
| 23 | SC제일은행 |
| 26 | 신한은행 |
| 27 | 씨티은행 |
| 31 | 대구은행 |
| 32 | 부산은행 |
| 34 | 광주은행 |
| 35 | 제주은행 |
| 37 | 전북은행 |
| 39 | 경남은행 |
| 45 | 새마을금고 |
| 48 | 신용협동조합 |
| 50 | 상호신용금고 |
| 54 | HSBC은행 |
| 55 | 도이치은행 |
| 71 | 우체국 |
| 81 | 하나은행 |
| 88 | 신한은행 |
| 209 | 동양종합금융증권 |
| 218 | 현대증권 |
| 230 | 미래에셋증권 |
| 238 | 대우증권 |
| 240 | 삼성증권 |
| 243 | 한국투자증권 |
| 247 | 우리투자증권 |
| 261 | 교보증권 |

3. PAYCO 연동하기

| Code | 설명 |
|------|-----------|
| 262 | 하이투자증권 |
| 263 | HMC투자증권 |
| 264 | 키움증권 |
| 265 | 이트레이드증권 |
| 266 | SK증권 |
| 267 | 대신증권 |
| 268 | 솔로몬투자증권 |
| 269 | 한화증권 |
| 270 | 하나대투증권 |
| 278 | 신한금융투자 |
| 279 | 동부증권 |
| 280 | 유진투자증권 |
| 287 | 메리츠종합금융증권 |
| 289 | NH투자증권 |
| 290 | 부국증권 |
| 291 | 신영증권 |
| 292 | LIG투자증권 |
| 02 | 산업은행 |
| 03 | 기업은행 |
| 04 | 국민은행 |
| 05 | 외환은행 |
| 07 | 수협 |

3.5.11 에러코드

| Code | Message | Description |
|------|---------------------|-------------|
| 1001 | 파라미터 정보가 유효하지 않습니다. | |
| 1002 | 인증 정보가 유효하지 않습니다. | |
| 1003 | 상품주문상태가 유효하지 않습니다. | |

1004 유효하지 않은 주문번호입니다.

1005 유효하지 않은 주문번호입니다.

1006 주문상품내역이 없습니다.

1007 회원정보가 없습니다.

1008 유효하지 않은 주문상품번호입니다.

1009 판매자 정보가 없습니다.

1010 주문을 처리할수 있는 권한이 없습니다.

1011 PG 취소처리불가

1012 결제가능한 유효시간이 지났습니다.

1013 현금영수증 발급 내역이 존재하지 않습니다.

1014 해당 주문의 사용자정보와 일치하지 않습니다.

1015 결제 처리중입니다.

1016 인증된 회원정보가 없습니다.

1017 결제 가능한 카드가 아닙니다.

1018 결제할 주문정보가 없습니다.

1019 이미 결제 완료되었습니다.

1020 이미 발급처리된 OTC 거래요청번호입니다.

1021 허용되지 않은 IP 입니다.

1022 등록되지 않은 가맹점 코드(sellerKey)입니다.

-
- | | |
|------|---------------------------------------|
| 1023 | 등록된 정보와 가맹점 코드(sellerKey)가 일치하지 않습니다. |
|------|---------------------------------------|
-
- | | |
|------|---------------------|
| 1024 | 자동결제 정보가 존재하지 않습니다. |
|------|---------------------|
-
- | | |
|------|----------------|
| 1025 | 본인인증을 진행해 주세요. |
|------|----------------|
-
- | | |
|------|-----------------------------------|
| 1026 | 로그인 정보가 주문(결제)요청한 ID 와 일치하지 않습니다. |
|------|-----------------------------------|
-
- | | |
|------|---------------------|
| 1027 | 원거래 정보를 찾을 수가 없습니다. |
|------|---------------------|
-
- | | |
|------|-------------------------------------------------------|
| 1028 | 주문자와 결제자 정보가 일치 하지 않습니다. 주문자가 결제할 수 있도록 다시 결제 요청해 주세요 |
|------|-------------------------------------------------------|
-
- | | |
|------|-------------------------------|
| 1029 | 회원 정보를 확인할 수 없습니다.(token,eid) |
|------|-------------------------------|
-
- | | |
|------|------------------------|
| 1030 | 이미 주문서 발송 처리가 완료되었습니다. |
|------|------------------------|
-
- | | |
|------|------------------------|
| 1031 | 주문서 발송 최대 횟수를 초과하였습니다. |
|------|------------------------|
-
- | | |
|------|--------------------|
| 1032 | 보안 모듈이 설치되지 않았습니다. |
|------|--------------------|
-
- | | |
|------|------------------------------------------|
| 1033 | 유효한 PAYCO ID 가 아닙니다. PAYCO ID 를 확인해 주세요. |
|------|------------------------------------------|
-
- | | |
|------|------------------------|
| 1034 | 주문서 발송 최대 횟수를 초과하였습니다. |
|------|------------------------|
-
- | | |
|------|------------------|
| 1035 | 주문서 신고접수된 주문입니다. |
|------|------------------|
-
- | | |
|------|---------------------|
| 1036 | ARS 인증이 완료되지 않았습니다. |
|------|---------------------|
-
- | | |
|------|-------------------|
| 1037 | ARS 인증 완료처리되었습니다. |
|------|-------------------|
-
- | | |
|------|--------------------------|
| 1038 | 입력된 생년월일 정보가 일치하지 않습니다.. |
|------|--------------------------|
-
- | | |
|------|------------------------------------------|
| 1039 | 생년월일 정보 확인 불가]페이코 홈페이지에서 본인인증을 진행해 주세요.. |
|------|------------------------------------------|
-

1040 쿠폰 사용 불가합니다.

1100 조회가능한 기간이 아닙니다.

1101 최대 1 년 범위내에서만 조회 가능합니다.

1102 최대 6 개월 이내만 조회 가능합니다.

1103 최대 3 개월 이내만 조회 가능합니다.

1104 최대 1 개월 범위내이내만 조회 가능합니다.

1105 최대 7 일 범위내에서만 조회 가능합니다.

1106 최대 1 일 이내만 조회 가능합니다.

1200 이미 판매자로 등록한 PAYCO ID NO 입니다.

1201 존재하지 않는 Member Key 입니다.

1202 휴면 계정 입니다.

1203 재로그인 후 이용해 주세요.

1301 기취소건입니다.

1302 테스트가맹점 설정 오류.

1303 PAYCO 포인트 잔액 부족으로 주문을 취소할 수
없습니다. PAYCO 고객센터 : 1544-6891

가맹점의 취소가능금액이 부족하여 취소할수 없습니다.

1304 가맹점에서 페이코 파트너센터> 충전금 관리 메뉴에서
일정금액 충전하시면 취소 가능합니다.

1305 진행중인 취소요청건이 있습니다.

1402 죄송합니다. 알수없는 오류가 발생하여 결제가
불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망).

1403 비정상적인 결제시도가 확인되었습니다. 고객센터로
문의해주세요.

1450 죄송합니다. 알수없는 오류가 발생하여 결제가
불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망).

1451 죄송합니다. 알수없는 오류가 발생하여 결제가
불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망).

1452 죄송합니다. 알수없는 오류가 발생하여 결제가
불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망).

1453 SMS 본인인증이 필요합니다.

1454 ARS 본인인증이 필요합니다.

1455 해당 상품은 PAYCO 로 구매할 수 없는 상품 군으로
분류되어 결제를 진행할 수 없습니다.
이용에 불편을 드려 죄송합니다.
자세한 사항은 고객센터로 문의해 주시기 바랍니다..

1456 죄송합니다. 알수없는 오류가 발생하여 결제가
불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망).

1457 죄송합니다. 알수없는 오류가 발생하여 결제가
불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망)

죄송합니다. 알수없는 오류가 발생하여 결제가
1458 불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망)

죄송합니다. 알수없는 오류가 발생하여 결제가
1459 불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망)

죄송합니다. 알수없는 오류가 발생하여 결제가
1460 불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망)

죄송합니다. 알수없는 오류가 발생하여 결제가
1461 불가능합니다. 잠시후 다시 이용해 주시기
바랍니다.(동일현상 반복시 고객센터로 문의 요망)

1501 프로모션 참여가능 불가

1580 등록된 간편결제 정보가 없습니다.

1601 혜택참여 불가

1700 충전 가능한 포인트를 확인해주세요.

1702 처리중입니다

1703 결제 중 오류가 발생했습니다.

1704 충전 중 오류가 발생했습니다. 잠시후 다시 이용해
주시기 바랍니다.

1705 충전 완료되었습니다. 자세한 내용은 목록에서
확인해주세요.

1800 실명을 확인해주세요

1801 송금 수신 데이터가 존재하지 않습니다

3. PAYCO 연동하기

1802 입금처리 오류로 송금이 취소되었습니다.

1803 송금 한도 체크가 실패되었습니다.잠시후 다시 이용해 주시기 바랍니다.

1804 미처리된 송금건이 존재합니다.

1805 출금하는 계좌로 송금하실 수 없습니다.

1806 송금 한도가 초과되었습니다

1807 PAYCO 에서 지원하지 않는 계좌형식입니다.

1808 파라미터 정보가 유효하지 않습니다

1809 보내는 사람 입력란에 특수문자(공백포함)는 입력하실 수 없습니다.

1810 송금 요청 유효시간이 만료되었습니다.

1811 송금요청은 하루에 30 번까지 가능합니다

1890 점검중입니다.

1900 기기정보 확인 중입니다. 잠시후 다시 이용해 주시기 바랍니다.(동일현상 반복시 고객센터로 문의 요망)

2001 빌링 API 오류

2222 주문이 취소되었습니다.

3001 페이코 회원 API 오류

3002 PUSH API 오류

3003 한게임 회원 API 오류

3004 회원정보 갱신 오류

3005 본인인증된 회원만 간편결제를 이용하실 수 있습니다.

3006 파트너 회원 API 오류

3007 Appguard API 오류

3008 한게임 포털 API 오류

3100 외부 API 응답없음

3101 NPC API 오류

3201 COS API 오류

3202 마일리지 API 연동 실패

3203 포인트 시스템 점검 중

3301 데이터베이스 처리 실패

3302 데이터베이스 제약 조건 위반

3401 결제취소 처리 실패

3401 취소처리 불가합니다

3401 당월 취소처리만 가능합니다.

3401 거래일시 기준 6 개월 이내 취소처리 가능합니다.

3401 거래일시 기준 1 년 이내 취소처리 가능합니다.

3401 부분취소 불가합니다.

3401 2 회 이상 부분취소 처리 불가합니다.

3401 취소가능한 금액이 없습니다

3401 가상계좌 정보가 없습니다.

3. PAYCO 연동하기

3401 유효한 결제내역이 없습니다.

3402 요청된 금액으로 결제취소 불가능합니다.(쿠폰 부분취소 불가) {쿠폰금액}원 이상의 취소금액을 입력하십시오.

3501 가상계좌 발급 처리 실패

3601 외부판매자 API 연동 실패

3602 이벤트 처리 실패

3603 결제내역이 없어 이벤트 적립 확정 처리 불가

3701 포토인프라 연동 실패

3702 FDS 로그연동 실패

3801 정산 API 연동 실패

3802 Failed to encrypt or decrypt

3803 한도 API 연동 실패

3804 KCP API 연동 실패

4001 주문예약 실패

4002 주문서 처리 실패

4003 결제는 완료되었으나 판매자 연동에 실패했습니다.

4004 유효하지 않은 데이터입니다.

4005 이미 주문 완료되었습니다.

4006 현금영수증 발급취소 실패

4007 주문서 혜택 returnUrl 이 올바르지 않습니다.

4008 이미 존재하는 앱인증 결제건입니다.

4501 데이터가 존재하지 않습니다.

4502 가능한 PG 가 없습니다.

4503 결제 불가능한 금액입니다.

4504 결제 불가능한 결제수단입니다.

4505 PG 결제 처리가 실패했습니다.

4506 PG 결제 처리 중 오류가 발생했습니다

4507 결제 처리 중 오류가 발생했습니다.

4508 PG 처리 중 오류가 발생했습니다

4509 시도 가능한 횟수를 초과했습니다.

4510 인증 숫자를 잘못 입력하셨습니다.

4511 비밀번호 오류가 발생했습니다.

4512 신용카드 정보 삭제시 오류가 발생했습니다

4513 페이코티머니에 가입되지 않은 정보입니다.

4514 본인인증에 실패하였습니다. (은행 정기점검)

4515 유효하지 않은 계좌입니다.

4516 잠시후 다시 시도부탁드립니다 (TIMEOUT)

4517 연속 10 회 오류로 계좌확인을 이용할 수 없습니다.

4520 SC 은행 간편결제 회원이 아닙니다

4521 간편결제 등록에 실패했습니다

3. PAYCO 연동하기

4522 계좌 비밀번호를 잘못입력하셨습니다.

4523 계좌 비밀번호 오류횟수가 초과되었습니다.

4524 ATM 서비스를 이용하실 수 없는 계좌입니다.

4525 간편결제 삭제에 실패했습니다

4530 계좌 등록을 위한 계좌정보 추가 인증은 1 일 3 회까지
가능합니다. 익일에 다시 시도해 주세요

4531 추가 인증 유효시간이 초과됐습니다. 다시 진행해
주시기 바랍니다.

4532 인증 유효시간이 초과됐습니다. 다시 진행해 주시기
바랍니다.

인증 처리 중 일시적인 오류가 발생했습니다. 잠시후
4539 다시 이용해 주시기 바랍니다.(동일현상 반복시
고객센터로 문의 요망)

4540 기등록된 카드정보입니다.

4601 간편결제 정보가 초기화 되었습니다. 다시 등록해주세요

4602 가상계좌가 존재하지 않습니다.

4603 입금금액이 원거래와 다릅니다.

4604 잠금 상태의 결제수단으로는 결제가 불가능 합니다.

4701 결제 비밀번호 처리 중 오류가 발생했습니다

4702 비밀번호를 잘못 입력하셨습니다.

4703 결제 비밀번호 5 회 오류 : 결제 비밀번호를 재설정
해주세요.

4704 비밀번호형식이 올바르지 않습니다..

4705 결제비밀번호를 등록하지 않은 사용자 계정입니다.

4706 인증미완료.

4707 결제 비밀번호 입력 시간이 초과되었습니다. 다시 시도해 주세요.

4708 이미 결제 비밀번호가 등록되어 있습니다.

5001 송금처리가 실패되었습니다.

6001 Fail Cache Evict.

6002 Fail Cache Init.

6100 수기취소 불가한 주문입니다.

6101 부분취소 불가능한 신용카드입니다.

6101 부분취소 불가능한 신용카드입니다.

7001 PG Package 에 해당하는 클래스가 존재하지 않습니다.

7002 [승인] PG 와의 통신 중에 에러가 발생하였습니다.

7003 [결제] PG 와의 통신 중에 에러가 발생하였습니다.

7004 [취소] PG 와의 통신 중에 에러가 발생하였습니다.

7005 [계좌발급] PG 와의 통신 중에 에러가 발생하였습니다

9000 서비스 임시 점검중입니다.

9980 Not found URI

9990 Couldn't parse request

9999 알 수 없는 오류가 발생하였습니다.

3. PAYCO 연동하기

- ERROR 코드 9000은 시스템 점검시 내려주는 코드이므로
해당 코드에 대한 메시지는 사용자에게 노출처리 부탁드립니다.

3.6 영수증

3.6.1 PAYCO 영수증 주소

| 용도 | 도메인 |
|-------|---------------------------------------------------------------------------|
| 테스트 용 | https://alpha-bill.payco.com/ |
| 서비스 용 | https://bill.payco.com |

3.6.2 PAYCO 영수증 노출정의

- **가맹점용 URL** : /seller/receipt/{가맹점키}/{외부가맹점주문번호}/{PAYCO주문번호}
/seller/receipt{sellerKey}/{sellerOrderReferenceKey}/{orderNo}
- **고객용 URL** : /outseller/receipt/{PAYCO주문번호}
/outseller/receipt/{orderNo}
- 가맹점용 URL은 PAYCO 로그인 없이 이용가능 ,
고객용 URL은 PAYCO 로그인 후 본인인 경우 이용 가능
- 기본URL은 주문번호의 일반 결제건에 대한 영수증을 노출한다.
- 일반결제건의 종류가 카드인 경우 신용카드매출전표, 현금영수증 신청인 경우 현금영수증, 그 외는 구매영수증 노출
- 정해진 영수증 노출을 원할시 receiptKind 파라미터를 추가하여 호출한다.
 - 현금영수증 : URL?receiptKind=cash
 - 온라인영수증 : URL?receiptKind=online
 - 신용카드매출전표: URL?receiptKind=card

➢ 가맹점용 샘플 URL

<https://alpha-bill.payco.com/seller/receipt/DEMO/201501122000791783/15011?receiptKind=card>

➢ 고객용 샘플 URL

<https://alpha-bill.payco.com/outseller/receipt/201501122000791783?receiptKind=card>

3.7 부가설명

3.7.1 inApp 적용가이드

- IOS > 쿠키저장소에 Policy를 NSHTTPCookieAcceptPolicyAlways로 설정이 필요함

➤ sample

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    NSHTTPCookieStorage *cookieStorage=[NSHTTPCookieStorage sharedHTTPCookieStorage];
    [cookieStorage setCookieAcceptPolicy:NSHTTPCookieAcceptPolicyAlways];
    .....
    return YES;
}
```

- Android > 쿠키 및 자바스크립트 처리를 위한 웹뷰 설정이 필요함

➤ sample

```
getSettings().setAppCacheEnabled(true);
getSettings().setDomStorageEnabled(true);
getSettings().setJavaScriptEnabled(true);
getSettings().setJavaScriptCanOpenWindowsAutomatically(true);

// Android LOLIPOP 이상에서만 작동하도록함
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    // https -> http 호출 허용.
    webView.getSettings().setMixedContentMode(WebSettings.MIXED_CONTENT_ALWAYS_ALLOW);

    // 서드파티 쿠키 허용.
    CookieManager cookieManager = CookieManager.getInstance();
    cookieManager.setAcceptCookie(true);
    cookieManager.setAcceptThirdPartyCookies(webView, true);
}
setWebChromeClient(new WebChromeClient());
```

- Android > 가맹점 앱에서 Payco 앱이 실행될 수 있도록 intent처리가 필요함

➤ sample

```
@Override
public boolean shouldOverrideUrlLoading(WebView view, String url) {

    ClientType type = ClientType.getClientType(url);
    if (type.equals(ClientType.BLANK) || type.equals(ClientType.JAVASCRIPT)) {
        return true;
    }

    if (type.equals(ClientType.WEB)) {
        return false;
    }

    // ARS 인증을 위한 전화 연결
    if (url.startsWith("tel:")) {
        startCallPhoneIntent(url);
        return true;
    }
}
```



```
// 앱 실행
return handleAppUrl(url);
}
```

- Android > 추가적용사항
- 'tel:' URL scheme이 호출되는 경우 전화걸기가 실행 될 수 있도록 처리가 필요함
- 'about:blank' 또는 'about:blank;' 호출되는 경우 브라우저가 열리지 않고 무시하도록 처리가 필요함

- inApp 주문예약시 추가설정

주문예약 > inAppYn

| Column | Type | MaxLength | 필수여부 | Description |
|---------|--------|-----------|------|--------------------------------------------------|
| inAppYn | string | 1 | N | inApp 결제일때 Y * inAppYn 이 Y일 때 appUrl이 추가되어야 함 |

주문예약 > extraData > appUrl

| Column | Type | MaxLength | 필수여부 | Description |
|--------|--------|-----------|------|---------------------------------------------------------------------------------|
| appUrl | string | 4000 | N | 인앱 결제시 PAYCO(앱결제시)/ISP모바일 등 앱에서 결제를 처리한 뒤 복귀할 앱 url scheme (ex. testapp://open) |

- 일반 신용카드 결제수단을 사용하는 경우 별도 문의 필요

3.7.2 Iframe 적용가이드

필수 사항

- Iframe 적용시 최상단 포커스이동을 위해 **onload="scroll(0,0)"** 옵션 추가 및 height 600px로 처리
- scroll 이슈관련 script 추가
- extraData의 viewOptions > iframeYn='Y'로 예약처리 필수

➤ sample

```
<iframe src="..." ..... height="600px" onload="scroll(0,0);" />

<script type="text/javascript">
function onResponseMessage (evt) {
    if (evt.data == "payco message") {
        scroll(0,0);
    }
}

if (window.addEventListener) {
    window.addEventListener("message", onResponseMessage, false);
} else {
    window.attachEvent("onmessage", onResponseMessage);
}
</script>
```

3.8 부록

3.8.1 판매자 부가정보(게임사 결제용)

- 판매자 부가정보가 필요한 경우 주문예약 API[[표 3-4](#)]에 추가되어야 합니다.
- 게임결제시 부가정보의 clientIp 와 memberId 는 필수값입니다.

| Column | Type | Max Length | 필수 여부 | Description |
|-----------------|------|------------|-------|---------------------------|
| 1 sellerOptions | | | N | 판매자 부가정보(Json 형태의 String) |

sellerOptions 의 부가정보 > INPUT

| | Column | Type | Max Length | 필수여부 | Description |
|---|----------|--------|------------|------|----------------------------------------|
| 1 | clientIp | String | 20 | N | 클라이언트 IP |
| 2 | memberId | String | 50 | N | 서비스의 회원 ID - 가맹점(게임)의 유저 ID 를 의미합니다 |

3.8.2 판매자 부가정보(가맹점 이벤트용)

- 가맹점 이벤트 적용시에만 해당 됨(사전 PAYCO 와 협의된 가맹점에 한함)
- 판매자 부가정보가 필요한 경우 주문예약 API 에 추가되어야 합니다.

| Column | Type | Max Length | 필수 여부 | Description |
|-----------------|------|------------|-------|---------------------------|
| 1 sellerOptions | | | N | 판매자 부가정보(Json 형태의 String) |

sellerOptions 의 부가정보 > INPUT

| Column | Type | Max Length | 필수여부 | Description |
|-------------------------|--------|------------|------|---------------------------------------------------------------------------------------------|
| 1 orderSheetMessageType | String | 50byte | N | 주문서 메시지 타입 FIRST_PAYMENT : 생애 첫 결제자 메시지 노출 VALID_FIRST_PAYMENT : 실제 첫 결제자 판단 및 메시지 노출 |
| 2 orderSheetMessage | String | 100byte | N | 주문서 메시지 내용 |

- orderSheetMessageType = FIRST_PAYMENT 일 경우 [결제승인](#)시 아래의 결과가 추가적으로 전달됩니다.
결제완료 결과에 포함되어 반환 > OUPUT

| 변수 | Type | Max Length | 설명 | 예제 |
|------------------|--------|------------|------------------|--------|
| 1 firstPaymentYn | String | 50 | PAYCO 생애 첫 결제 여부 | Y or N |

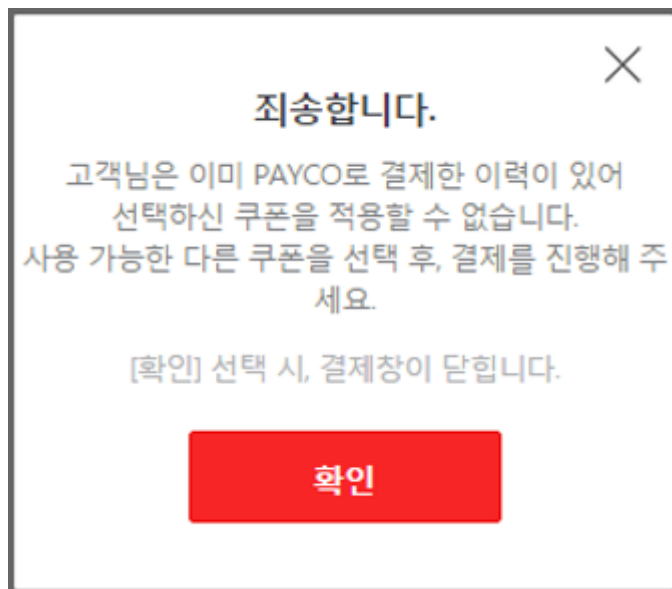
3.8.3 판매자 부가정보(가맹점 혜택 조건 검증)

- 사전 협의된 가맹점 혜택 조건 검증용
- 주문예약 API 에 추가되어야 합니다.

> INPUT

| Column | Type | Max Length | 필수 여부 | Description |
|--------------------|--------|------------|-------|----------------------------------|
| 1 benefitPolicySeq | String | 50 | N | 가맹점 혜택 정책번호 (사전 PAYCO와 협의된 값) |

- 사전에 협의된 조건에 따라 결과가 표시 됩니다.
> OUTPUT 예제 (생애첫 결제 조건 적용시 조건과 맞지 않는 경우 아래와 같은 alert 이 표시됨)



3.8.4 판매자 부가정보(추가배송비)

- 추가배송비 조회 사용여부를 선택할 수 있습니다.
- 추가배송비 조회용 URL을 파라미터로 제어할 수 있습니다.
- 추가배송비 조회 사용시 returnParam 을 사용할 수 있습니다.
- 판매자 부가정보를 사용합니다.(주문예약 -> sellerOptions)

주문예약 > sellerOptions > INPUT

| Column | Type | Max Length | 필수 여부 | Description |
|---------------------------------------|--------|------------|-------|-------------------------------------------------|
| 1 remoteAreaDeliveryFeeYn | String | 1 | N | 추가배송비조회 사용여부 Y : 사용 N : 사용안함 |
| 2 remoteAreaDeliveryFeeSearchUrl | String | 1000 | N | 추가배송비조회URL |
| 3 remoteAreaDeliveryFeeSearchUrlParam | String | 3000 | N | 추가배송비조회URL 파라미터 (추가배송비 조회 URL에 POST 로 전달된다.) |

- 주문 예약 API > 샘플 코드

샘플 코드 INPUT

```
{
  "sellerKey": "S0FSJE",
  "sellerOrderReferenceKey": "TESTORDERKEY000001111",
  "totalDeliveryFeeAmt": "0",
  "totalPaymentAmt": "3000",
  "serviceUrl": "https://test.com/test/surl/ok",
  "serviceUrlParam": "{\"serviceUrlParam1\":\"data1\",\"serviceUrlParam2\":300}",
  "returnUrl": "http://test.com/result.nhn",
  "returnUrlParam": "{\"returnUrlParam1\":\"data1\", \"returnUrlParam2\":300}",
  "orderMethod": "EASYPAY F",
  "orderProducts": [{
    "cpId": "PARTNERTEST",
    "productId": "PROD_EASY",
    "productAmt": "3000",
    "productPaymentAmt": "3000",
    "sortOrdering": 1,
    "productName": "BAG",
    "orderQuantity": 1,
    "sellerOrderProductReferenceKey": "TESTPRODKEY00000001"
  }],
  "extraData": "{\"payExpiryYmdt\": \"20181231000000\"}",
  "sellerOptions": {
    "\"remoteAreaDeliveryFeeYn\":\"Y\", \"remoteAreaDeliveryFeeSearchUrl\":\"https://test.com/test/getRemoteAreaDeliveryFeeWithParam\", \"remoteAreaDeliveryFeeSearchUrlParam\": \"{{{\"param1\\\\\":\\\\\"data1\\\\\",\\\\\"param2\\\\\":300}}}\""}
  }
}
```

3.8.5 판매자 부가정보(결제 사용자 체크)

- 주문자와 결제자가 같은 경우만 결제를 진행할 수 있습니다.
- 주문자의 정보와 PAYCO에 로그인한 결제자의 정보가 다른 경우 결제가 진행되지 않습니다.

주문예약 > extraData

| Column | Type | Max Length | 필수 여부 | Description |
|-------------------|--------|---------------|----------|-----------------------------------------------------|
| 1 sameUserCheckYn | String | 1 | N | 동일 사용자 체크사용여부 Y : 사용 N : 사용안함 |
| 2 username | String | 100 | N | 사용자 이름 (sameUserCheckYn = Y인 경우 필수) |
| 3 userBirth | String | 300 | N | 사용자 생년월일 (yyyyMMdd) (sameUserCheckYn = Y인 경우 필수) |