
PAYCO 연동 가이드

- 간편결제, 바로구매

NHN PAYCO

저작권

Copyright © 2020 NHN PAYCO Corp. All Rights Reserved.

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

이 문서는 정보 제공의 목적으로만 제공됩니다. NHN PAYCO (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 PAYCO (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 PAYCO (주)는 이 문서의 내용을 예고 없이 변경할 수 있습니다.

문서 정보

문서 개요

이 문서는 가맹점 사이트에서 PAYCO로 결제할 수 있도록 PAYCO 연동 가이드에 따라 상품 정보 연동 시스템을 개발하는 방법에 대해 설명한다.

- PAYCO에서 제공하는 연동 API : PAYCO 연동 API 상세 명세
- 가맹점에서 구현 해야하는 API : 외부가맹점 API 상세

독자

이 문서의 독자는 PAYCO 연동 가맹점/사이트 개발자이다.

문의처

이 문서의 내용에 오류가 있거나 관련 문의사항은 아래의 연락처에 문의한다.

- PAYCO 기술지원: dl_payco_ts@nhn.com

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	7
1.1 들어가며	8
1.1.1 PAYCO 연동 방식	8
1.1.2 PAYCO 결제모드	8
1.1.3 사전 준비사항	8
1.2 PAYCO 결제 연동 방식	10
1.2.1 간편결제형(EASYPAY)	10
1.2.2 바로구매형(CHECKOUT)	11
1.3 PAYCO 결제연동 샘플	13
1.3.1 PAYCO 결제모드	13
2. PAYCO 버튼 적용 방법	16
2.1 PAYCO 버튼 삽입	17
2.1.1 PAYCO 버튼 종류	17
2.1.2 버튼 이미지 삽입을 위한 자바스크립트	17
2.2 PAYCO 버튼 클릭 시 처리 방법	19
3. PAYCO 연동하기	23
3.1 PAYCO 연동 API	24
3.1.1 연동 API 목록	24
3.1.2 PAYCO 연동 API 주소	25
3.1.3 PAYCO 연동 API 공통 규약	25
3.1.4 PAYCO 연동에 필요한 KEY 정보	26
3.2 PAYCO 연동 API 상세 명세	27
3.2.1 주문예약	27

3.2.2	주문 상품 상태 변경 (바로구매형만 해당)	32
3.2.3	결제승인 API	33
3.2.4	주문 및 결제 취소 가능여부 조회	40
3.2.5	주문 및 결제 취소 처리	42
3.2.6	결제 상세 조회 API	47
3.3	inApp 및 iframe 가이드	54
3.3.1	inApp 적용가이드	54
3.3.2	iframe 적용 이슈	58
3.4	외부가맹점 API 상세	59
3.4.1	추가 배송비 조회 (바로구매형만 해당)	59
3.4.2	결제인증 완료	61
3.4.3	무통장 입금 완료 통보	62
3.5	유의사항	67
3.6	코드 정의	68
3.6.1	주문유형(=결제유형)	68
3.6.2	주문상품상태	68
3.6.3	주문채널(=결제채널)	68
3.6.4	결제유형	69
3.6.5	결제수단	69
3.6.6	상품타입	69
3.6.7	과세타입	69
3.6.8	카드사코드	70
3.6.9	통신사코드	70
3.6.10	은행코드	70
3.6.11	에러코드	71
3.7	영수증	78
3.7.1	PAYCO 영수증 주소	78
3.7.2	PAYCO 영수증 노출정의	78

1. 개요

이 장에서는 PAYCO 연동의 기본 처리 절차를 간략하게 소개한다.

1.1 들어가며

1.1.1 PAYCO 연동 방식

항목 \ 주문연동	간편결제형 (EASYPAY)	바로구매형 (CHECKOUT)
개 요	간편결제수단(신용카드/휴대폰/간편계좌) 결제 인증/승인을 제공. 결제수단 목록에 "PAYCO" 결제수단 노출.	간편결제 기능에 추가로 PAYCO 회원의 배송지 주소록을 제공. (배송지 주소록이 PAYCO에서 통합관리됨) 주문버튼 영역에 "PAYCO구매하기" 버튼 노출.
Flow 요약	PAYCO 결제창을 띄우고 결제를 진행. 결제완료 시 결제 성공여부를 가맹점에 전달.	'PAYCO구매하기' 버튼 클릭 시 PAYCO 주문창 을 띄우고 결제를 진행. PAYCO 주문창에서 배송지를 입력 받아 결제정 보와 <u>배송지정보</u> 를 가맹점에 전달.

1.1.2 PAYCO 결제모드

'인증'과 '승인'을 분리하여 제공하는 방식.

PAYCO 결제창을 통하여, 결제수단 인증을 하며 이후 인증 토큰 값을 가맹점에 전달함.

가맹점에서는 해당 토큰을 이용하여 PAYCO 승인 API를 호출하여 결제 요청함.

1.1.3 사전 준비사항

- 주문연동방식(바로구매형, 간편결제형)에 대한 결정 필요
- 가맹점 방화벽 오픈 요청
 - PAYCO 결제서비스를 사용하기 위해서는 가맹점 측 서버 IP에 대해 방화벽 오픈요청을 해야 결제연동이 가능합니다.
(외부가맹점 URL이 80, 443 Port 외 사용 시 사전공유 부탁드립니다)
- 가맹점코드발급 요청
 - 테스트환경에서는 아래 개발환경 키 정보로 테스트를 진행합니다.
 - 실 운영환경에서는 PAYCO에서 발급해드린 키 정보를 사용합니다.
 - PAYCO 계약이 완료되면 가맹점 키 정보를 발급하여 전달해 드립니다.
- 가맹점코드 발급 시 가맹점사이트(PC, MOBILE) 및 약관URL 정보가 필요합니다.

구 분	sellerKey	cpId	productId
개발환경 (간편결제형)	S0FSJE	PARTNERTEST	PROD_EASY
개발환경 (바로구매형)	S0FSJE	PARTNERTEST	PROD_CHK (일반상품), DELIVERY_PROD (배송비상품 구분용도)
운영환경	발급받은 KEY	발급받은 KEY	발급받은 KEY

➤ 관련 사이트

구 분	URL
페이코 서비스사이트	http://www.payco.com
페이코 가맹점사이트	https://partner.payco.com

페이코 연동 관련 모든 문의 및 질의 사항은 PAYCO 기술지원으로 문의 바랍니다.

(PAYCO 기술지원: dl_payco_ts@nhn.com)

1.2 PAYCO 결제 연동 방식

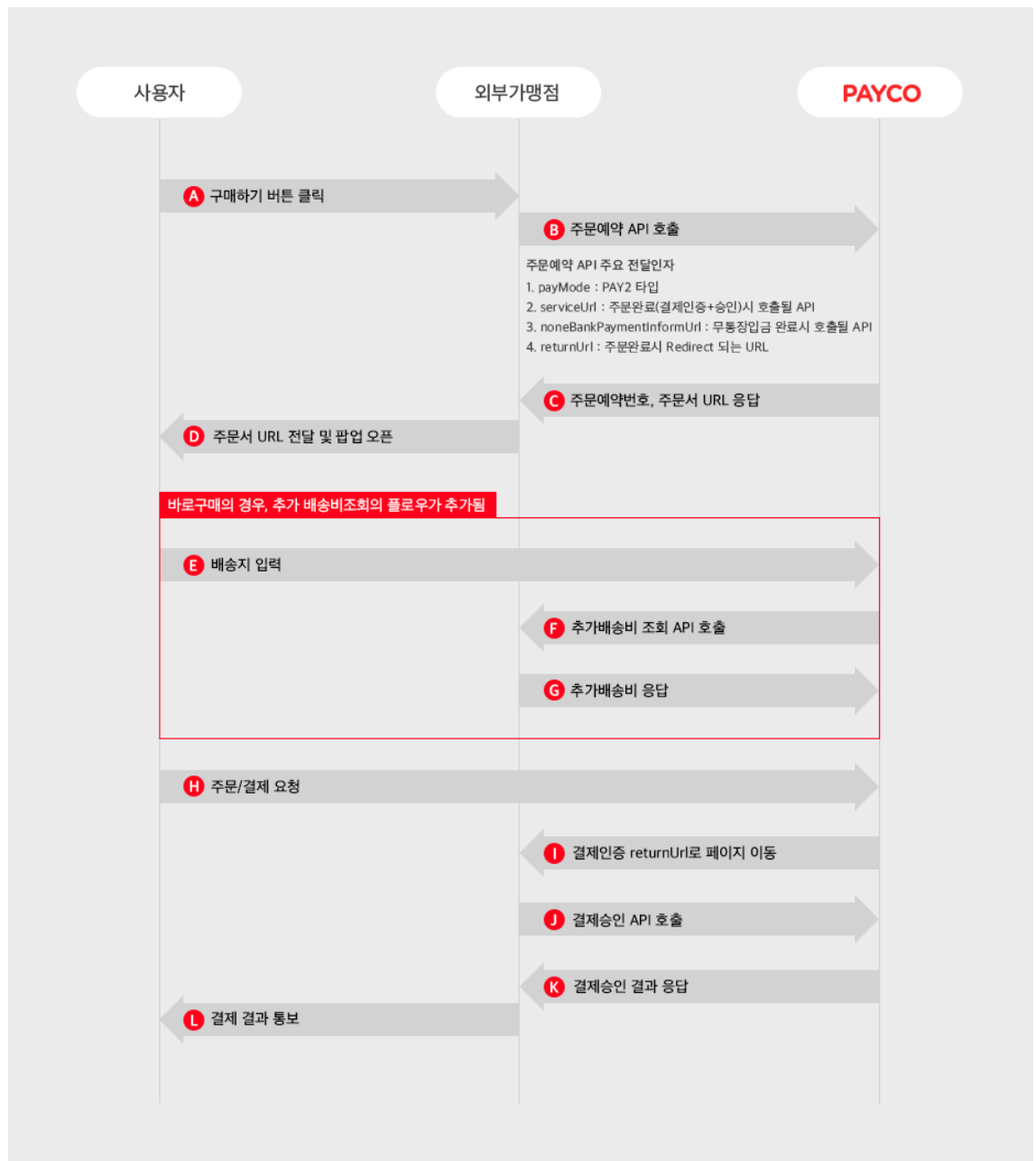
1.2.1 간편결제형(EASYPAY)



- ① 가맹점에서는 PAYCO에서 제공하는 버튼 Javascript plugin을 이용하거나 PAYCO가 제공하는 버튼 이미지를 사용한다. [\[2. PAYCO 구매하기 버튼 연동\]](#)
- ② 사용자가 PAYCO 버튼을 클릭하면 외부가맹점에서 해당 주문 건에 대해 PAYCO 주문예약 API를 호출한다. - (A~B) [\[3.2.1 주문예약 API\]](#)
- ③ PAYCO는 주문예약 API 요청을 받아 주문예약을 처리하고, 가맹점에게 주문예약 번호와 주문서 URL을 전달한다. - (C)
- ④ 가맹점은 전달받은 주문서 URL로 결제창을 띄운다. - (D)
- 모바일 환경의 경우 리다이렉트(redirect)로 처리한다.
- ⑤ 결제창에서 이용자가 결제인증을 완료하면 주문예약 시 가맹점으로부터 전달된 returnUrl로 결제인증 정보를 전달한다. - (I) [\[3.4.2 결제인증 완료\]](#)

- ⑥ 가맹점은 결제 인증정보를 참고하여 결제 완료처리가 필요한 경우 결제승인 API를 호출하여 승인결과를 전달받아 주문완료 처리한다. - (J~L) [\[3.2.3 결제승인 API\]](#)

1.2.2 바로구매형(CHECKOUT)



- ① 가맹점에서는 PAYCO에서 제공하는 구매하기 버튼 Javascript plugin을 이용하여 구매하기 버튼을 연동한다. [\[2. PAYCO 구매하기 버튼 연동\]](#)

1. 개요

- ② 사용자가 구매하기 버튼 클릭 시 가맹점에서 해당 주문건에 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, 가맹점에서는 주문완료 또는 결제완료 후 처리되어야 하는 내용을 PAYCO에서 제공하는 연동 API 명세에 맞추어 구현 해야한다. (하단 [외부가맹점 구현 API Spec](#) 참조)
- ✓ nonBankbookDepositInformUrl : 무통장 주문완료 또는 무통장 결제완료 시 PAYCO에서 이 URL(API)를 호출하므로, 가맹점에서는 무통장 주문완료 또는 무통장 결제완료 후 처리되어야 하는 내용을 PAYCO에서 제공하는 연동 API 명세에 맞춰 구현 해야한다.
- ✓ returnUrl : 주문 또는 결제완료 후 redirect될 외부가맹점의 페이지 URL.

- ③ PAYCO는 주문예약 API 요청을 받아 주문예약을 처리하고 가맹점에게 주문예약 번호와 주문서 URL을 전달한다. - (C)

- ④ 가맹점은 전달받은 주문서 URL(간편결제의 경우 결제팝업)로 주문서/결제 팝업을 띄운다. - (D~E)

- ⑤ (바로구매형의 경우에만 해당) PAYCO 결제창에서 배송지 입력 후, 가맹점으로 추가배송비 조회

[\[3.4.1 추가배송비조회 API\]](#)

- ⑥ 팝업에서 사용자가 결제인증완료 시 주문예약 시 가맹점으로부터 전달받은 returnUrl로 결제인증 정보를 전달한다. - (G)

[\[3.4.2 결제인증 완료\]](#)

- ⑦ 가맹점은 결제인증정보를 참고하여 결제완료처리가 필요한 경우 결제승인 API를 호출하여 피드백을 전달받아 주문완료 처리한다. - (H~J)

[\[3.2.3 결제승인 API\]](#)

1.3 PAYCO 결제연동 샘플

1.3.1 PAYCO 결제모드

[결제인증과 승인 분리모드]



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

- request

```

curl -H "Content-Type: application/json" -d \
{
  "sellerKey": "S0FSJE",
  "payMode": "PAY2",
  "orderChannel": "PC",
  "orderMethod": "EASYPAY",
  "totalPaymentAmt": 5000,
  "inAppYn": "N",
  "orderTitle": "payco 결제 테스트 주문",
  "sellerOrderReferenceKey": "TEST86921076653",
  "returnUrl": "http://localhost/jsp/easypay/pay2/payco_return.jsp",
  "returnUrlParam": "{\"cart_no\":\"20191234567\"}",
  "extraData": "{
    \"cancelMobileUrl\":\"http://www.payco.com\",
    \"viewOptions\":{\"iframeYn\":\"N\",\"showMobileTopGnbYn\":\"Y\"}
  }",
  "orderProducts": [{
    "cpId": "PARTNERTEST",
    "productId": "PROD_EASY",
    "productPaymentAmt": 5000,
    "productAmt": 5000,
    "sortOrdering": 1,
    "orderQuantity": 1,
    "productName": "[GLOBE] TILT KIDS (NAVY/ORANGE)",
    "sellerOrderProductReferenceKey": "ITEM_100001"
  }]
}

' https://alpha-api-bill.payco.com/outseller/order/reserve
  
```

- response

```

{
  "result": {
    "reserveOrderNo": "201910172005189446",
    "orderSheetUrl": "https://alpha-bill.payco.com/easyLogin/201910172005189446"
  },
  "code": 0,
  "message": "success"
}
  
```

➤ 결제 승인 API 예시

● request

```
curl -H "Content-Type: application/json" -d '{
  "totalPaymentAmt": 5000,
  "sellerKey": "S0FSJE",
  "reserveOrderNo": "201910172005189459",
  "sellerOrderReferenceKey": "TEST86921076653",
  "paymentCertifyToken": "UPLIjLh2eQmPyM9QCJD9GEyPVQA4zZmnLJLo-
vydTLfBDT3ZcnyW00SnEv_OtP2o"
}' https://alpha-api-bill.payco.com/outseller/payment/approval
```

● response

```

{
  "result": {
    "sellerOrderReferenceKey": "TEST87108878575",
    "reserveOrderNo": "201910172005189459",
    "orderNo": "201910172002486459",
    "memberName": "강***",
    "memberEmail": "na****",
    "orderChannel": "PC",
    "totalOrderAmt": 5000,
    "totalDeliveryFeeAmt": 0,
    "totalRemoteAreaDeliveryFeeAmt": 0,
    "totalPaymentAmt": 5000,
    "receiptPaycoPointAmt": 0,
    "receiptPaycoPointTaxfreeAmt": 0,
    "receiptPaycoPointTaxableAmt": 0,
    "receiptPaycoPointVatAmt": 0,
    "receiptPaycoPointServiceAmt": 0,
    "orderProducts": [{
      "orderProductNo": "201910172002660790",
      "sellerOrderProductReferenceKey": "TV_2150246090",
      "orderProductStatusCode": "OPSPAED",
      "orderProductStatusName": "결제 완료",
      "cpId": "PARTNERTEST",
      "productId": "PROD_EASY",
      "productKindCode": "GENERAL_PRODUCT",
      "productPaymentAmt": 5000,
      "originalProductPaymentAmt": 5000
    }],
    "paymentDetails": [{
      "cardSettleInfo": {
        "cardCompanyName": "현대카드",
        "cardCompanyCode": "CCDI",
        "cardNo": "949019*****3333",
        "cardInstallmentMonthNumber": "00",
        "cardAdmissionNo": "05056844",
        "cardInterestFreeYn": "N",
        "corporateCardYn": "N",
        "partCancelPossibleYn": "N",
        "acquirersCorporationCode": "CCDI"
      },
      "paymentTradeNo": "201910172003575249",
      "paymentMethodCode": "31",
      "paymentMethodName": "신용카드",
      "paymentAmt": 5000,
      "taxfreeAmt": 0,
      "taxableAmt": 4545,
      "vatAmt": 455,
      "serviceAmt": 0,
      "tradeYmdt": "20191017153006",
      "pgAdmissionNo": "19540928475604",
      "pgAdmissionYmdt": "20191017153006",
      "easyPaymentYn": "Y"
    }],
    "orderCertifyKey": "S70BBn5uSRcJ3fnJuDJsBQvryuyY9BBDbxJbwimgapDGkCC",
    "paymentCompletionYn": "Y",
    "paymentCompleteYmdt": "20191017153006"
  },
  "code": 0,
  "message": "success"
}

```

2. PAYCO 버튼 적용 방법

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

2.1 PAYCO 버튼 삽입

2.1.1 PAYCO 버튼 종류

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

- 간편결제형(EASYPAY)

EASYPAY	A1	
	A2	

- 바로구매형(CHECKOUT)

CHECKOUT	A1	
	A2	

2.1.2 버튼 이미지 삽입을 위한 자바스크립트

- 버튼 이미지 삽입을 위한 자바 스크립트는 아래와 같다.
 - 제공하는 자바스크립트 URL을 추가한다.
 - PC : <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>
/* pc */
<script type="text/javascript" src="https://static-bill.nhnent.com/payco/checkout/js/payco.js" charset="UTF-8"></script>
/* mobile */
<script type="text/javascript" src="https://static-bill.nhnent.com/payco/checkout/js/payco_mobile.js" charset="UTF-8"></script>

<script type="text/javascript">
</head>
<body>
```

2. PAYCO 버튼 적용 방법

```
<div id="payco_type1"></div>
<script type="text/javascript">
//
    Payco.Button.register({
        SELLER_KEY : "S0FSJE",           // 가맹점 코드
        ORDER_METHOD : "EASYPAY",        // 주문유형
        BUTTON_TYPE : "A1",              // 버튼타입 선택 (A1,A2)
        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
        "" : ""
    });
}
//]]&gt;
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="94 463 321 478" data-label="Section-Header"><h3>• 자바스크립트 파라미터 설명</h3></div><div data-bbox="128 481 832 727" data-label="Table"><table><tr><th>컬럼명</th><th>설명</th><th>선택가능 값</th></tr><tr><td>SELLER_KEY</td><td>가맹점 코드</td><td>PAYCO 에서 발급받은 가맹점코드</td></tr><tr><td>ORDER_METHOD</td><td>주문 유형</td><td>EASYPAY : 간편결제형</td></tr><tr><td>BUTTON_TYPE</td><td>버튼 타입 선택</td><td>A1, A2 ...</td></tr><tr><td>BUTTON_HANDLER</td><td>구매하기 버튼 클릭 시 호출할 이벤트 Handler 함수 (callback)</td><td>자바스크립트 함수명<br/>ex) order(p1,p2) =&gt; order</td></tr><tr><td>BUTTON_HANDLER_ARG</td><td>구매하기 버튼 클릭 시 호출할 이벤트 Handler 함수 인자(필요 시)</td><td>ex) order(p1,p2) =&gt; [p1, p2]</td></tr><tr><td>DISPLAY_PROMOTION</td><td>프로모션 문구 출력 여부</td><td>Y, N (기본값 Y)</td></tr><tr><td>DISPLAY_ELEMENT_ID</td><td>버튼 노출 위치의 Element ID</td><td>자체적으로 정의한 값</td></tr></table></div><div data-bbox="67 944 284 959" data-label="Page-Footer"><p>18 · PAYCO 연동하기 개발 가이드</p></div>
```

2.2 PAYCO 버튼 클릭 시 처리 방법

1. 사용자가 PAYCO 버튼을 클릭하면 가맹점은 구매상품의 색상, 타입 등 필수 옵션 사항을 선택했는지 체크한다.
2. 사용자가 옵션을 선택했거나 선택해야 할 옵션이 없는 상품의 경우 주문정보 등록을 위한 처리를 진행한다 (PAYCO 주문예약 API 이용)
3. 주문정보 등록 후 응답받은 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 체크함 =====/

      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 주문예약 API를 이용하여 주문 정보를 등록
      3. 리턴받은 주문서 페이지로 이동 처리한다.
      */

      // ( test code ) 응답 받은 json data라고 가정.
      var jsonData = {
        "result" : {
          "reserveOrderNo" : "201910172005189459",
          "orderSheetUrl" : " https://alpha-bill.payco.com/easyLogin/201910172005189459"
        },
      },
```

2. PAYCO 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 : "S0FSJE",
        ORDER_METHOD : " EASYPAY",           // 주문유형 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
    });
```

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

</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

• PAYCO 제공 API 목록

API 명칭	간편결제 & 바로구매	참 고
<u>주문예약</u>	○	
<u>주문상태 변경</u>	▲	에스크로결제 사용 시 연동 필요
<u>결제승인</u>	○	
<u>결제취소 가능여부 조회</u>	▲	카드사로부터 응답을 받아 전달하는 것은 아니기 때문에 실제 취소 요청 시 가능여부와 다른 결과가 나올 수 있음. 필수 연동 API는 아니며, 사용하는 경우 결제 취소 요청 전 PAYCO 정책에 의한 취소불가 여부 판단용으로만 사용 필요.
<u>결제취소 처리</u>	▲	고객이 직접 취소요청 하거나 가맹점 관리자에서 취소처리를 하는 경우 연동 필요
결제상세 조회	▲	승인 완료된 결제 건에 대해 상세조회 API를 통해 승인완료 시 전달받은 Response 내역 확인 가능

✓ 해당 정보는 PAYCO에서 제공하는 API 목록입니다.

• 가맹점에서 구현해야 할 API 목록

API 명칭	간편결제 & 바로구매	참 고
결제인증 완료	○	
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

용 도	도메인
테스트용(개발환경)	https://alpha-api-bill.payco.com
서비스용(운영환경)	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.

field가 추가되더라도 오류가 발생하지 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.)

- 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

```
{
  "pgErrorCode": "CC14",
  "code": 4505,
  "message": " PG 결제 처리가 실패했습니다.",
}
```

3.1.4 PAYCO 연동에 필요한 KEY 정보

- 아래 항목은 가맹점별로 발행 해드리는 항목으로 주문예약 및 취소 처리 시 필요한 값들입니다.

- sellerKey (가맹점 코드): PAYCO에서 발급
- cpId (상점 ID): PAYCO에서 발급
- productID (상품 ID): PAYCO에서 발급

구 분	sellerKey	cpId	productId
개발환경 (간편결제형)	S0FSJE	PARTNERTEST	PROD_EASY
개발환경 (바로구매형)	S0FSJE	PARTNERTEST	PROD_CHK (일반상품), DELIVERY_PROD (배송비상품 구분용도)
운영환경	발급받은 KEY	발급받은 KEY	발급받은 KEY

3.2 PAYCO 연동 API 상세 명세

PAYCO에서 제공하는 API들의 상세 명세를 기술한다.

3.2.1 주문예약

- PAYCO 주문서를 띄우기 전에 주문예약 API를 통해 필요한 정보를 전달한다.
- 가맹점에서는 주문예약 응답으로 받은 주문서 URL을 팝업으로 띄운다.
- 가맹점에서 관리하는 주문연동키(sellerOrderReferenceKey)는 주문별로 고유한 Key이어야 한다.

1) PAYCO 주문예약 후 결제팝업 오픈 시 팝업차단을 위한 개발 TIP!!

- orderSheetUrl로 PAYCO 팝업오픈 시 가맹점 팝업창(빈화면)에서 주문예약 후 리턴 받은 orderSheetUrl로 리다이렉트 하여 팝업차단을 우회할 수 있다.
- 팝업 오픈은 POST방식의 submit은 지양하도록 한다. (IE11 관리자모드 about:blank 새창 오픈)

- METHOD : POST
- URL : /outseller/order/reserve

• 주문예약 API > INPUT

Column	Type	Max Length	필수 여부	Description
sellerKey	String	100	Y	가맹점 코드 - PAYCO에서 전달받은 KEY 정보
sellerOrderReferenceKey	String	100	Y	외부가맹점 주문번호
sellerOrderReferenceKeyType	String	50	N	외부가맹점의 주문번호 타입 UNIQUE_KEY: 유니크 키 - 기본값 DUPLICATE_KEY: 중복 가능한 키 - 외부가맹점의 주문번호가 중복 가능한 경우 사용
currency	String	3	N	통화(default=KRW)
totalPaymentAmt	Number		Y	총 결제할 금액 - 면세금액, 과세금액, 부가세, 봉사료의 합

3. PAYCO 연동하기

				= totalTaxfreeAmt + totalTaxableAmt + totalVatAmt + totalServiceAmt
totalTaxfreeAmt	Number	N		면세금액(면세상품의 공급가액 합)
totalTaxableAmt	Number	N		과세금액(과세상품의 공급가액 합)
totalVatAmt	Number	N		부가세(과세상품의 부가세 합)
totalServiceAmt	Number	N		봉사료
orderTitle	String	300	Y	주문 타이틀 -여러 개 상품 구매 시 00외 0건 (ex. [단비샵]쿠폰할인30% + 즉시할인 10% 무조건 40% 할인 코트/패딩/원 피스 외 2건)
returnUrl	String	1000	Y	주문완료 후 Redirect 되는 Url (인증완료 후 redirect됨)
returnUrlParam	String	1000	N	주문완료 후 Redirect 되는 Url 시 함께 전달되어야 하는 파라미터 (Json 형태의 String)
nonBankbookDepositInformUrl	String	1000	N	무통장입금완료통보 URL
orderMethod	String	N/A	Y	<u>주문유형(=결제유형)</u> - 바로구매형: CHECKOUT - 가맹점 회원결제: EASYPAY_F (기본) - 가맹점 비회원결제: EASYPAY
orderChannel	String	N/A	Y	<u>주문채널</u> (default. PC) ex. PC, MOBILE
inAppYn	String	1	Y	인앱결제 여부(Y/N) (진입 경로가 가맹점 앱일경우 Y) 추가로 inApp 적용가이드 확인 필요
orderSheetUiType	String	N	N	주문서 UI 타입 선택 ※ PAYCO와 사전 협의 후 사용가능
payMode	String	N	Y	<u>결제모드</u> (default. PAY2)
orderProducts	Array		Y	주문상품 List
extraData	String	3000	N	부가정보 (Json 형태의 String) ※ PAYCO와 사전 협의 후 사용가능

refundUseYn	String	N	무통장입금 취소 시 PAYCO 환불 사용 여부 (Y:사용 / N:사용안함) ※ PAYCO와 사전 협의 후 사용가능
-------------	--------	---	--

⇒ **orderProducts** (주문상품 List)

Column	Type	Max Length	필수여부	Description
cpId	String	30	Y	상점 ID
productId	String	50	Y	상품 ID
productAmt	Number		Y	상품금액(상품단가 * 수량)
productPayment Amt	Number		Y	상품 결제금액(상품결제단가 * 수량)
orderQuantity	Number	10	Y	주문수량
option	String	100	N	상품 옵션
sortOrdering	Number	10	Y	상품노출순서
productName	String	160	Y	상품명
orderConfirmUrl	String	4000	N	주문완료 후 주문상품을 확인할 수 있는 url
orderConfirmMobileUrl	String	1000	N	주문완료 후 주문상품을 확인할 수 있는 모바일 url
productImageUrl	String	4000	N	상품 이미지 URL
sellerOrderProductReferenceKey	String	100	Y	외부가맹점에서 관리하는 주문상품 연동 키
taxationType	String	50	N	과세타입(default : TAXATION) <ul style="list-style-type: none"> • 면세상품 : DUTYFREE • 과세상품 : TAXATION • 결합상품 : COMBINE

✓ 주문 상품 List(orderProducts) 부가 설명

<간편결제형>

orderProducts 리스트에 상품(total)으로 하나만 등록한다.

복수개의 상품 주문예약 시 orderTitle(주문명) 및 productName(상품명) 은 '00외 0건'으로 전송한다.

3. PAYCO 연동하기

(총 결제할금액(totalPaymentAmt)과 상품별 결제금액 합(productPaymentAmt)은 일치하도록 전송)

<바로구매형>

상품별 리스트와 배송비 상품을 함께 등록한다.

예시1) 5개의 상품이 각각 배송비가 존재할 경우:

상품 리스트(5개) + 배송비상품(총배송비금액 1개) = 총 6개 리스트

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

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

⇒ **extraData (가맹점 추가정보)** (※ PAYCO와 사전 협의 후 사용가능)

Column	Type	Max Length	필수 여부	Description
payExpiryYmdt	String	14	N	해당 주문예약건 만료 처리 일시 (해당 일시 이후에는 결제 불가) 미설정시 디폴트 30 분입니다 Ex) "payExpiryYmdt":"20210510180000"
virtualAccountExpiryYmd	String	14	N	가상계좌만료일시 (YYYYMMDD or YYYYMMDDHHmmss) 미설정시 default 3 일이며, YYYYMMDD 일 경우 그날 24 시 만료
cancelMobileUrl	String	300	N	모바일 결제페이지에서 취소 버튼 클릭시 이동할 URL (미설정 시 returnUrl 이 우선순위)
viewOptions			N	화면 UI 옵션 / 아래 추가설명 참고
appUrl	String		N	iOS App 인 경우 필수 inApp 적용 가이드 참고

⇒ **viewOptions 화면UI 옵션**

Column	Type	Max Length	필수 여부	Description
iframeYn	String	1	N	iframe 호출 여부(default. N)
languageCode	String	100	N	주문서 언어 코드(default KR) - EN/KR

iframeCloseBtnYn	String	1	N	비로그인 시 회원 iframe 닫기버튼 노출 (default. N)
showLoginCloseBtnYn	String	1	N	비로그인 시 로그인창내 닫기버튼 노출여부(default. N)

주문예약 API > OUTPUT

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

Default format 부분 중 result 컬럼 부분만 표시하였음

• 주문예약 API > 샘플 코드

➤ 샘플 코드 INPUT

```
{
  "sellerKey": "S0FSJE",
  "sellerOrderReferenceKey": "mallseq1623672401879",
  "payMode": "PAY2",
  "currency": "KRW",
  "totalOrderAmt": "5500",
  "totalDeliveryFeeAmt": "",
  "totalPaymentAmt": "5500",
  "totalTaxableAmt": "",
  "totalVatAmt": "",
  "totalTaxfreeAmt": "",
  "totalServiceAmt": "",
  "orderTitle": "payco 결제 테스트 주문",
  "serviceUrl": "https://alpha-api-bill.payco.com/test/surl/ok",
  "serviceUrlParam": "",
  "returnUrl": "http://demo-bill.nhnent.com/alpha/easyComplete",
  "returnUrlParam": "{\\"sellerKey\\":\\"S0FSJE\\"}",
  "nonBankbookDepositInformUrl": "",
  "orderMethod": "EASYPAY",
  "orderChannel": "PC",
  "inAppYn": "N",
  "customUrlSchemeUseYn": "N",
  "individualCustomNoInputYn": "N",
  "orderSheetUiType": "RED",
  "appPaymentUseYn": "",
  "sellerOptions": "",
  "nonBankbookPaymentInformUrl": "https://alpha-api-bill.payco.com/test/surl/ok",
  "refundUseYn": "",
  "incomeTaxDeductionType": "",
  "extraData":
    "{\\"viewOptions\\":{\\"language\\":\\"\\",\\"languageCode\\":\\"\\",\\"iframeYn\\":\\"\\",\\"showLog
outYn\\":\\"\\",\\"showMobileTopGnbYn\\":\\"\\",\\"iframeCloseBtnYn\\":\\"\\",\\"paycoViewSkipYn\\":\\"\\",\\"shortUrlUseYn\\":\\"Y\\",\\"paymentMethodOptions\\":[{\\"paymentMethodCode\\":\\"31\\",\\"useCardList\\":null,\\"cardInstallmentMonthList\\":null,\\"cardInterestFreeQuota\\":null,\\"cardInstallmentMonth\\":\\"\\",\\"cardInterestFreeYn\\":\\"\\",\\"creditCardPolicy\\":null,\\"arsCardList\\":null,\\"fixedCardInstallmentMonth\\":\\"\\",\\"fixedCardInterestFreeYn\\":\\"\\",\\"fixedCardPointUseYn\\":\\"\\"}],\\"sameUserCheckYn\\":null,\\"userName\\":null,\\"userBirth\\":null,\\"userCi\\":null,\\"appUrl\\":null,\\"kftcPaymentAgencyFeeCredit\\":\\"600\\",\\"kftcPaymentAgencyFeeCheck\\":\\"500\\",\\"excludePaymentMethodCodes\\":null,\\"includePaymentMethodCodes\\":null,\\"userCellPhoneNo\\":null,\\"eventAmt\\":null}}",
}
```

3. PAYCO 연동하기

```
"orderProducts": [{
    "cpId": "PARTNERTEST",
    "productId": "PROD_EASY",
    "productAmt": "5500",
    "productPaymentAmt": "5500",
    "orderQuantity": "1",
    "option": "black",
    "sortOrdering": "1",
    "productName": "LG전자 VR6480VMNC",
    "orderConfirmUrl":
"http://www.popshoes.co.kr/app/product/detail/115312/0",
    "orderConfirmMobileUrl":
"http://www.popshoes.co.kr/app/product/detail/115312/0",
    "productImageUrl":
"http://image.toast.com/aaaade/BILLTEST2__DEMO/8728595019_20150803153443.jpg",
    "sellerOrderProductReferenceKey": "115312"
}]
}
```

➤ 샘플 코드 OUTPUT

```
{
  "result": {
    "reserveOrderNo": "202106142005468910",
    "orderSheetUrl": "https://alpha-
bill.payco.com/easyLogin/202106142005468910"
  },
  "code": 0,
  "message": "success"
}
```

3.2.2 주문 상품 상태 변경 (바로구매형만 해당)

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

• 주문 상품 상태 변경 API > INPUT

Column	Type	Max Length	필수 여부	Description
sellerKey	String	100	Y	가맹점 코드 (PAYCO에서 발급)
sellerOrderReferenceKey	String	100	Y	외부가맹점에서 관리하는 주문연동키
sellerOrderProductRefere nceKey	String	100	Y	외부가맹점에서 관리하는 주문상품연동키

orderProductStatus	String	N/A	Y	변경할 주문상품상태
--------------------	--------	-----	---	------------

• 주문상품 상태변경 API > OUTPUT

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

➔ Default format 중 result 컬럼 부분 없음

• Sample

➤ INPUT

```
{
  "sellerKey": "S0FSJE",
  "sellerOrderReferenceKey": " TESTORDERKEY00000001",
  "sellerOrderProductReferenceKey": "1170961",
  "orderProductStatus": "DELIVERY_START"
}
```

➤ OUTPUT

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

3.2.3 결제승인 API

- 사용자 결제인증 후 실제 승인을 처리하는 API
- 가맹점에서 주문완료 가능한 조건이 되는 경우 결제승인 API를 호출한다. (ex. 재고확인)
- 결제완료 시 PAYCO가 회원에게 PAYCO App에 주문완료 push알림 및 Email을 발송한다.
- 결제인증 후 30분 내 승인처리가 이루어져야 한다.
- 중복 승인 요청이 발생하지 않도록 방어코드 작성 필요.
- Timeout 30초 이상 설정 권장
- METHOD : POST
- URL : /outseller/payment/approval

• 결제승인 API > INPUT

3. PAYCO 연동하기

Column	Type	Max Length	필수 여부	Description
sellerKey	String	100	Y	가맹점 코드 - PAYCO에서 전달받은 KEY 정보
reserveOrderNo	String	20	Y	주문예약번호
sellerOrderReferenceKey	String	100	Y	외부가맹점에서 관리하는 주문연동키
paymentCertifyToken	String	100	Y	결제인증토큰 - 결제인증완료 후 returnUrl로 전달받은 정보
totalPaymentAmt	Number		Y	총 결제할 금액 - 면세금액, 과세금액, 부가세의 합 = totalTaxfreeAmt + totalTaxableAmt + totalVatAmt + totalServiceAmt
totalTaxfreeAmt	String		N	면세금액(면세상품의 공급가액 합)
totalTaxableAmt	String		N	과세금액(과세상품의 공급가액 합)
totalVatAmt	String		N	부가세(과세상품의 부가세 합)
totalServiceAmt	String		N	봉사료

• 결제승인 API > OUTPUT

- totalOrderAmt(총주문금액) + totalRemoteAreaDeliveryFeeAmt(총도서관간비용) = totalPaymentAmt(총결제금액)

Column	Type	Max Lenh	Description	Example
sellerOrderReferenceKey	String	100	외부가맹점 주문번호	20140821221250595
reserveOrderNo	String	20	주문예약번호	201408211000000720
orderNo	String	20	주문번호	20140821100000072
memberName	String	15	주문자명(간편결제형만 일부 마스킹처리)	
memberEmail	String	200	주문자 EMAIL (일부 마스킹처리)	
orderChannel	String	N/A	주문채널 (PC/MOBILE)	PC

totalOrderAmt	Number		총주문금액	116500
totalPaymentAmt	Number		총결제금액	119000
paymentCompletionYn	String	1	결제완료여부 (Y/N)	Y
orderProducts	Array	N/A	주문상품 List	
paymentDetails	Array	N/A	결제내역 List	- 복합결제의 경우 결제 수단별 array list로 구성되어 리턴되며, 결제 수단은 추후 추가될 수 있다.
orderCertifyKey	String	100	주문인증키	
receiptPaycoPointAmt	Number		현금영수증 발급 대상 포인트	0
			결제금액	
receiptPaycoPointTaxfreeAmt	Number		현금영수증 발급 대상 포인트	0
			면세금액	
receiptPaycoPointTaxableAmt	Number		현금영수증 발급 대상 포인트	0
			과세금액	
receiptPaycoPointVatAmt	Number		현금영수증 발급 대상 포인트	0
			부가세	
receiptPaycoPointServiceAmt	Number		현금영수증 발급 대상 포인트	0
			봉사료	

⇒ **deliveryPlace (배송지) – 바로구매형만 해당**

	Column	Type	Max Length	Description	Example
1	recipient	String	15	수령인	홍길동
2	englishRecipient	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

3. PAYCO 연동하기

7	telephone	String	15	연락처	010-1111-2222
8	individualCustomUniqNo	String	50	개인통관번호	

⇒ orderProducts (주문상품List)

Column	Type	Max Length	Description	Example
orderProductNo	String	20	주문상품번호	20140821100000022
cpId	String	30	상점ID	PS_TOASTCO
productId	String	50	상품ID	PS_TOASTCOMC1
sellerOrderProductReferenceKey	String	100	외부가맹점에서 발급한 주문상품연동Key	19801895
orderProductStatusCode	String	50	주문상품 상태코드	PAYED
orderProductStatusName	String	10	주문상품 상태명	결제완료
productKindCode	String		상품타입	GENERAL_PRODUCT
productPaymentAmt	Float		상품결제금액	5000
originalProductPaymentAmt	Float		원 상품결제금액 (배송비 상품의 경우 배송비 금액)	2000

⇒ paymentDetails (결제내역 List)

Column	Type	Max Length	Description	Example
paymentTradeNo	String	20	결제번호	201409191000001142
paymentMethodCode	String	10	결제수단코드	01
paymentMethodName	String	15	결제수단명	신용카드
paymentAmt	Number		결제금액	119000
taxfreeAmt	Number		면세금액	0
taxableAmt	Number		과세금액	0
vatAmt	Number		부가세	0
serviceAmt	Number		봉사료	0

tradeYmdt	String	14	결제일시 (yyyyMMddHHmmss)	20140822120201
pgAdmissionNo	String	100	원천사승인번호	209380
pgAdmissionYmdt	String	14	원천사승인일시 (yyyyMMddHHmmss)	20140822120201
easyPaymentYn	String	1	간편결제여부 (Y/N)	N
cardSettleInfo	Object	N/A	신용카드 결제 정보	↓
cellphoneSettleInfo	Object	N/A	핸드폰 결제 정보	↓
realtimeAccountTransfer SettleInfo	Object	N/A	실시간계좌이체 결제정보	↓
nonBankbookSettleInfo	Object	N/A	무통장입금 결제정보	↓
couponSettleInfo	Object	N/A	쿠폰정보	↓

⇒ cardSettleInfo (카드결제정보)

Column	Type	Max Length	Description	Example
cardCompanyName	String	15	카드사명	현대카드
cardCompanyCode	String	50	카드사코드	CCDI
cardNo	String	300	카드번호	912345*****1809
cardInstallmentMonthNumber	String	2	할부개월(MM)	00
cardAdmissionNo	String	20	카드사 승인번호	31213333
cardInterestFreeYn	String	1	무이자여부(Y/N)	Y
corporateCardYn	String	1	법인카드여부 (개인-N / 법인-Y)	Y
partCancelPossibleYn	String	1	부분취소가능유무(Y/N)	Y

⇒ cellphoneSettleInfo (핸드폰 결제 정보)

Column	Type	Max Length	Description	Example
--------	------	---------------	-------------	---------

3. PAYCO 연동하기

cellphoneNo	String	32	휴대폰번호	010-1111-2222
-------------	--------	----	-------	---------------

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

Column	Type	Max Length	Description	Example
bankName	String	15	은행명	신한
bankCode	String	50	은행코드	88

⇒ **nonBankbookSettleInfo** (무통장입금 결제정보)

Column	Type	Max Length	Description	Example
bankName	String	15	은행명	신한
bankCode	String	50	은행코드	88
accountNo	String	200	계좌번호	1901801980225
paymentExpirationYmd	String	8	입금만료일	20141208

⇒ **couponSettleInfo** (쿠폰정보)

Column	Type	Max Length	Description	Example
discountAmt	String	15	쿠폰할인금액	10000
discountConditionAmt	String	50	쿠폰할인조건	50000

• 결제승인 API > Sample

> 샘플 코드 INPUT

```
{
  "totalPaymentAmt": 5500.0,
  "sellerKey": "S0FSJE",
  "reserveOrderNo": "202106142005468927",
  "sellerOrderReferenceKey": "mallseq16236724018800",
  "paymentCertifyToken":
  "iI5wtXe94IxKLf3p7yqbfaKl8ziSqmD5owOmbi_3RjWkf6irkFh4fAGim3PVNgkm"
}
```

> 샘플 코드 OUTPUT

```
{
  "result": {
    "sellerOrderReferenceKey": "TEST33103947674",
    "reserveOrderNo": "201910172005189722",
  }
}
```

```

"orderNo": "201910172002486629",
"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": "201910172002661155",
  "sellerOrderProductReferenceKey": "TV_2150246090",
  "orderProductStatusCode": "OPSPAED",
  "orderProductStatusName": "결제완료",
  "cpId": "PARTNERTEST",
  "productId": "PROD_EASY",
  "productKindCode": "GENERAL_PRODUCT",
  "productPaymentAmt": 81500,
  "originalProductPaymentAmt": 81500
}],
"paymentDetails": [{
  "couponSettleInfo": {
    "discountAmt": 1000,
    "discountConditionAmt": 2000,
    "encCouponId": null
  },
  "paymentTradeNo": "201910172003575688",
  "paymentMethodCode": "75",
  "paymentMethodName": "PAYCO 쿠폰",
  "paymentAmt": 1000,
  "taxfreeAmt": 0,
  "taxableAmt": 910,
  "vatAmt": 90,
  "serviceAmt": 0,
  "tradeYmdt": "20191017181604",
  "pgAdmissionNo": "8549125",
  "pgAdmissionYmdt": "20191017181602",
  "easyPaymentYn": "N"
},
{
  "cardSettleInfo": {
    "cardCompanyName": "현대카드",
    "cardCompanyCode": "CCDI",
    "cardNo": "949019*****3333",
    "cardInstallmentMonthNumber": "00",
    "cardAdmissionNo": "42420914",
    "cardInterestFreeYn": "N",
    "corporateCardYn": "N",
    "partCancelPossibleYn": "N",
    "acquirersCorporationCode": "CCDI"
  },
  "paymentTradeNo": "201910172003575685",
  "paymentMethodCode": "31",
  "paymentMethodName": "신용카드",
  "paymentAmt": 78500,
  "taxfreeAmt": 0,
  "taxableAmt": 71363,
  "vatAmt": 7137,
  "serviceAmt": 0,
  "tradeYmdt": "20191017181604",
  "pgAdmissionNo": "19540928480633",
  "pgAdmissionYmdt": "20191017181604",

```

```
        "easyPaymentYn": "Y"
    },
    {
        "paymentTradeNo": "201910172003575689",
        "paymentMethodCode": "98",
        "paymentMethodName": "PAYCO 포인트",
        "paymentAmt": 2000,
        "taxfreeAmt": 0,
        "taxableAmt": 1818,
        "vatAmt": 182,
        "serviceAmt": 0,
        "tradeYmdt": "20191017181604",
        "pgAdmissionNo": "2019101722361099",
        "pgAdmissionYmdt": "20191017181603",
        "easyPaymentYn": "N"
    }
],
"serviceUrlParam": "{\\"sellerPgMid\\":\\"123456\\"}",
"orderCertifyKey":
"VkySn93UU51pJC4M0AaBBobH66M9uvOtJHsScSXugLGwCC",
"paymentCompletionYn": "Y",
"paymentCompleteYmdt": "20191017181604"
},
"code": 0,
"message": "success"
}
```

3.2.4 주문 및 결제 취소 가능여부 조회

- 취소처리 API를 호출하기 이전에 취소처리를 가능여부를 확인하기위해 사용한다.
***취소가능여부 조회 API 요청 시 카드사로 부터의 응답을 받아 전달하는 것은 아니기 때문에 실제 취소 요청 시 가능여부와 다른 결과가 나올 수 있습니다.**
***필수 연동 API는 아니며, 사용하시는 경우 결제 취소 요청 전 PAYCO 정책에 의한 취소불가 여부 판단용으로 사용하시기 바랍니다.**
- cancelPossibleYn이 N인 경우 PG취소가능금액(pgCancelPossibleAmt)만큼만 취소가 가능하다.
- **cancelPossibleYn 이 N인 경우**
 - 결제 건 중 PG결제 취소가 불가능한 내역이 있음
 - 부분 취소가 필요한 경우 "취소할 총 금액 <= PG 취소가능금액" 으로 요청 해야함
- **PG결제 취소 불가능한 경우**
 - 핸드폰 익월 취소요청
 - 무통장 입금 취소요청
 - 부분취소 불가능한 PG결제의 부분취소 요청
- METHOD : POST
- URL : /outseller/order/cancel/checkAvailability

• 주문 결제 취소 가능여부 조회 API > INPUT

Column	Type	Max Length	필수 여부	Description
sellerKey	String	100	Y	가맹점 코드(PAYCO에서 전달받은 KEY 정보)
sellerOrderReferenceKey	String	20	Y	외부가맹점에서 관리하는 주문번호
cancelTotalAmt	Number		Y	취소할 총 금액
orderProducts	Array	N/A	N	취소할 상품 List (부분취소인 경우 사용, 입력하지 않는 경우 전체 취소)

⇒ **orderProducts**

Column	Type	Max Length	필수 여부	Description
sellerOrderProductReferenceKey	String	100	Y	외부가맹점에서 관리하는 주문상품번호
cpId	String	30	Y	상점 ID
productId	String	50	Y	상품 ID
productAmt	Number		Y	취소할 주문상품 금액

• 주문 결제 취소 가능여부 조회 API > OUTPUT

Column	Type	Max Length	Description
cancelPossibleYn	String	1	취소가능여부
cancelImpossibleReason	String	500	취소불가이유
partCancelPossibleYn	String	1	부분취소가능여부
pgCancelPossibleAmt	Number		PG취소가능금액 (PG결제취소 불가인 결제수단을 제외한 취소가능금액)
cancelImpossibleReasonCode	Number		취소불가사유코드

→ 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 가능

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

- 샘플 OUTPUT - 취소 불가

```
{
  "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
}
```

3.2.5 주문 및 결제 취소 처리

- 기존에 생성된 주문 및 결제내역을 취소 처리하는 경우 사용한다.
PG취소가 가능한 금액으로 요청했을 때만 취소된다.
 - 쿠폰사용시에는 결제 부분취소가 불가능한 케이스가 존재함.
(부분취소금액 <= 쿠폰사용금액 조건인 경우 부분취소 불가)
 - 간편계좌의 경우 전체취소 가능하나 부분취소는 지원이 안되며

주문상태가 배송시작이거나 구매확정인 경우 또는 주문일시가 120일이 지난 건의 경우는 취소 불가.

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

• 주문 결제 취소 처리 API > INPUT

Column	Type	Max Length	필수 여부	Description
sellerKey	String	100	Y	가맹점 코드 - PAYCO에서 전달받은 KEY 정보
orderNo	String	20	Y	주문번호
cancelTotalAmt	Number		Y	취소할 총 금액 = totalCancelTaxfreeAmt + totalCancelTaxableAmt + totalCancelVatAmt + totalCancelServiceAmt
totalCancelTaxfreeAmt	Number		N	총 취소할 면세금액
totalCancelTaxableAmt	Number		N	총 취소할 과세금액
totalCancelVatAmt	Number		N	총 취소할 부가세
totalCancelServiceAmt	Number		N	총 취소할 봉사료
totalCancelPossibleAmt	Number		N	총 취소가능금액(현재기준) : 취소가능금액 검증
orderProducts	Array		N	취소할 상품 List(부분취소인 경우 사용, 입력하지 않는 경우 전체 취소)
orderCertifyKey	String		Y	결제 승인완료시 내려받은 토큰값
requestMemo	String		N	취소처리 요청메모

⇒ orderProducts

Column	Type	Max Length	필수 여부	Description
--------	------	------------	-------	-------------

3. PAYCO 연동하기

sellerOrderProductReferenceKey	String	100	Y	외부가맹점에서 관리하는 주문상품번호
cpId	String	30	Y	상점 ID
productId	String	50	Y	상품 ID
productAmt	Number		Y	취소할 주문상품 금액
cancelDetailContent	String	160	N	취소 상세 사유

• 주문 결제 취소 처리 API > OUTPUT

Column	Type	Max Length	Description
cancelTradeSeq	Number	19	취소내역번호
totalCancelPaymentAmt	Number		총 취소금액
remainCancelPossibleAmt	Number		잔여 취소가능금액
cancelReceiptPaycoPointAmt	Number		현금영수증 발급 대상 PAYCO 포인트 취소 금액
cancelReceiptPaycoPointTaxfreeAmt	Number		현금영수증 발급 대상 PAYCO 포인트 취소 면세금액
cancelReceiptPaycoPointTaxableAmt	Number		현금영수증 발급 대상 PAYCO 포인트 취소 과세금액
cancelReceiptPaycoPointVatAmt	Number		현금영수증 발급 대상 PAYCO 포인트 취소 부가세
cancelReceiptPaycoPointServiceAmt	Number		현금영수증 발급 대상 PAYCO 포인트 취소 봉사료
cancelPaymentDetails	Array	N/A	취소 결제 내역 List
cancelYmdt	String		취소 처리 시간

⇒ Default format 부분 중 result 컬럼 부분만 표시하였음

⇒ **cancelPaymentDetails**

Column	Type	Max	Description	Example
--------	------	-----	-------------	---------

Length				
paymentTradeNo	String	20	결제번호	201409191000001142
cancelPaymentTradeNo	String	20	취소거래내역 결제 번호	201409191000001142
paymentMethodCode	String	10	결제수단코드	31
paymentMethodName	String		결제수단명	신용카드
cancelPaymentAmt	Number		취소금액	119000
cancelTaxfreeAmt	Number		취소 면세금액	9000
cancelTaxableAmt	Number		취소 과세금액	100000
cancelVatAmt	Number		취소 부가세	10000
cancelServiceAmt	Number		취소 봉사료	1000
tradeYmdt	String	14	취소일시 (yyyyMMddHHmm ss)	20140822120201

- Sample 코드

⇒ INPUT

```
{
  "requestMemo": "결제취소 테스트입니다.",
  "orderNo": "201910172002486602",
  "orderCertifyKey": "84mnWFM8a5guzWEE934PHbwKDPBda3kTwfXGPrRAATC4CC",
  "cancelTotalAmt": 5000,
  "sellerKey": "S0FSJE",
  "totalCancelTaxableAmt": "4545",
  "totalCancelVatAmt": "455",
  "totalCancelPossibleAmt": "5000",
  "sellerOrderReferenceKey": "TEST343791010664",
  "totalCancelTaxfreeAmt": "0"
}
```

⇒ OUTPUT

```
{
  "result": {
    "orderNo": "201910172002486602",
    "cancelTradeSeq": 2000354240,
    "totalCancelPaymentAmt": 5000.0,
    "remainCancelPossibleAmt": 0.0,
    "cancelPaymentDetails": [{
      "paymentTradeNo": "201910172003575621",
      "paymentMethodCode": "31",

```

```
        "paymentMethodName": "신용카드",
        "cancelPaymentAmt": 5000.0,
        "tradeYmdt": "20191017174959",
        "cancelPaymentTradeNo": "201910172003575622",
        "cancelTaxableAmt": 4545.0,
        "cancelTaxfreeAmt": 0.0,
        "cancelVatAmt": 455.0,
        "cancelServiceAmt": 0.0
    }],
    "cancelReceiptPaycoPointAmt": 0.0,
    "cancelReceiptPaycoPointTaxfreeAmt": 0.0,
    "cancelReceiptPaycoPointTaxableAmt": 0.0,
    "cancelReceiptPaycoPointVatAmt": 0.0,
    "cancelReceiptPaycoPointServiceAmt": 0.0,
    "cancelYmdt": "20191017174959"
},
"code": 0,
"message": "success"
}
```

- **부분취소 중복을 막기 위해 totalCancelPossibleAmt 컬럼을 사용하는 예**

- 예를 들어 고객이 10만원 주문을 하고 2만원을 부분취소 하고 싶은데 두 번 요청되어 4만원이 취소되는 케이스 또는 어떠한 이유로 PAYCO 에서는 2만원 부분취소가 되었지만 가맹점에서는 취소가 발생하지 않았을 때 등의 예외상황이 발생했을 때를 대비하여 totalCancelPossibleAmt 컬럼의 값을 설정하여 보내주시면 중복취소를 막을 수 있습니다.

예시) 고객이 10만원 결제금액 중 2만원 부분취소 요청

-> PAYCO에는 2만원 취소성공, 그러나 어떠한 이유로 가맹점에서는 취소가 안되었음

-> 고객이 다시 2만원 취소 -> 이때 가맹점 에서는 취소된 상품이 하나도 없으므로

totalCancelPossibleAmt 값을 10만원으로 보냄 -> 가맹점은 totalCancelPossibleAmt 값이

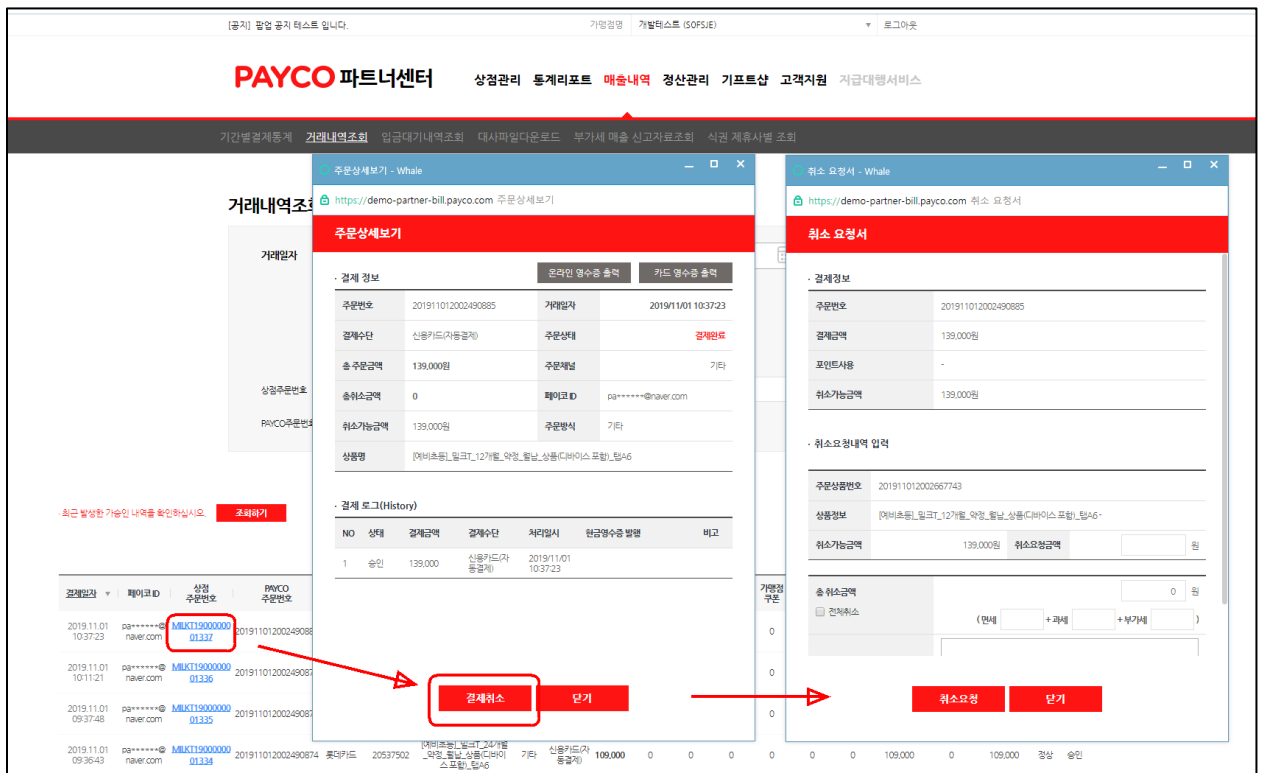
10만원이고 PAYCO는 2만원을 제외한 8만원이 취소가능금액 이므로 취소가능금액이 일치하지 않아 부분취소 불가.

- 각 가맹점은 PAYCO에서 제공하는 파트너센터에서 주문취소가 가능

- 개발환경 : <https://demo-partner.payco.com>

- 운영환경 : <https://partner.payco.com>

개발환경 예시



3.2.6 결제 상세 조회 API

- 정상적으로 승인 완료된 결제 건에 대해 상세조회 API를 통하여 승인완료 시 전달받은 OUTPUT 내역을 확인할 수 있다.
- METHOD : POST
- URL : /outseller/payment/approval/getDetailForVerify

결제 상세 API > INPUT

Column	Type	Max Length	필수 여부	Description
sellerKey	String	100	Y	가맹점키
reserveOrderNo	String	20	Y	PAYCO 주문예약번호
sellerOrderReferenceKey	String	100	Y	외부가맹점에서 관리하는 주문번호

⇒ reserveOrderNo 또는 sellerOrderReferenceKey 중 한 개의 값만 있어도 조회 가능

결제 상세 API > OUTPUT

totalOrderAmt(총 주문금액) = totalPaymentAmt(총 결제금액)

Column	Type	Max Lenth	Description	Example
sellerOrderReferenceKey	String	100	외부가맹점에서 발급하는 주문 연동 Key	20140821221250595
reserveOrderNo	String	20	주문예약번호	201408211000000720
orderNo	String	20	주문번호	20140821100000072
memberName	String	15	주문자명(간편결제형만 일부 마스킹처리)	
memberEmail	String	200	주문자EMAIL(일부 마스킹처리)	
orderChannel	String	N/A	주문채널 (PC/MOBILE)	PC
totalOrderAmt	Number		총 주문금액	116500
totalPaymentAmt	Number		총 결제금액	11900
paymentCompletionYn	String	1	결제완료여부 (Y/N)	Y
orderProducts	Array	N/A	주문상품 List	
paymentDetails	Array	N/A	결제내역 List	- 복합 결제의 경우 3.6.5 제수단 별 array list로 구성되어 리턴 되며, 결제 수단은 추후 추가될 수 있다.
orderCertifyKey	String	100	주문인증키	
receiptPaycoPointAmt	Number		현금영수증 발급 대상 포인트 결제금액	
receiptPaycoPointTaxfree Amt	Number		현금영수증 발급 대상 포인트 면세금액	
receiptPaycoPointTaxable Amt	Number		현금영수증 발급 대상 포인트 과세금액	

receiptPaycoPointVatAmt	Number	현금영수증 발급 대상 포인 트 부가세
receiptPaycoPointService Amt	Number	현금영수증 발급 대상 포인 트 봉사료

⇒ **orderProducts (주문상품List)**

Column	Type	Max Lenth	Description	Example
orderProductNo	String	20	주문상품번호	20140821100000022
cpId	String	30	상점ID	PS_TOASTCO
productId	String	50	상품ID	PS_TOASTCOMC1
sellerOrderProductReferenceKey	String	100	외부가맹점에서 발급한 주문상품연동Key	19801895
orderProductStatusCode	String	50	주문상품 상태코드	PAYED
orderProductStatusName	String	10	주문상품 상태명	결제완료
productKindCode	String		상품타입	GENERAL_PRODUCT
productPaymentAmt	Float		상품결제금액	5000
originalProductPaymentAmt	Float		원 상품결제금액	2000

⇒ **paymentDetails (결제내역 List)**

Column	Type	Max Lenth	Description	Example
paymentTradeNo	String	20	결제번호	20140919100000114 2
paymentMethodCode	String	10	결제수단코드	01
paymentMethodName	String	15	결제수단명	신용카드
paymentAmt	Number		결제금액	119000
taxfreeAmt	Number		면세금액	0
taxableAmt	Number		과세금액	0
vatAmt	Number		부가세	0

3. PAYCO 연동하기

serviceAmt	Number		봉사료	0
tradeYmdt	String	14	결제일시 (yyyyMMddHHmmss)	20140822120201
pgAdmissionNo	String	100	원천사승인번호	209380
pgAdmissionYmdt	String	14	원천사승인일시 (yyyyMMddHHmmss)	20140822120201
easyPaymentYn	String	1	간편결제여부 (Y/N)	N
cardSettleInfo	Object	N/A	신용카드 결제 정보	하단 참조
cellphoneSettleInfo	Object	N/A	핸드폰 결제 정보	하단 참조
realtimeAccountTransferSettleInfo	Object	N/A	실시간계좌이체 결제정보	하단 참조
nonBankbookSettleInfo	Object	N/A	무통장입금 결제정보	하단 참조
couponSettleInfo	Object	N/A	쿠폰정보	하단 참조

⇒ cardSettleInfo (카드결제정보)

Column	Type	Max Lenth	Description	Example
cardCompanyName	String	15	카드사명	현대카드
cardCompanyCode	String	50	카드사코드	CCDI
cardNo	String	300	카드번호	950000***** *1809
cardInstallmentMonthNumber	String	2	할부개월(MM)	00
cardAdmissionNo	String	20	카드사 승인번호	31213333
cardInterestFreeYn	String	1	무이자여부(Y/N)	Y
corporateCardYn	String	1	법인카드여부 (개인: N, 법인: Y)	Y
partCancelPossibleYn	String	1	부분취소가능유무(Y/N)	Y

⇒ **cellphoneSettleInfo** (핸드폰 결제 정보)

	Column	Type	Max Lenth	Description	Example
1	cellphoneNo	String	32	휴대폰번호	010-1111-2222

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

	Column	Type	Max Lenth	Description	Example
1	bankName	String	15	은행명	신한
2	bankCode	String	50	은행코드	88

⇒ **nonBankbookSettleInfo** (무통장입금 결제정보)

	Column	Type	Max Lenth	Description	Example
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)

	Column	Type	Max Lenth	Description	Example
1	discountAmt	String	15	쿠폰할인금액	10000
2	discountConditionAmt	String	50	쿠폰할인조건	50000
3	encCouponId	String	100	쿠폰 ID	44E8C452E1B0A5896 9211D9FC0E6C292

- 결제 상세조회 API > Sample

- ▶ 샘플 코드 INPUT

```
{
  "sellerKey": "S0FSJE",
  "reserveOrderNo": "201911132005203337",
  "sellerOrderReferenceKey": "TEST76338386107"
}
```

- ▶ 샘플 코드 OUTPUT

```
{
  "result": {
    "sellerOrderReferenceKey": "TEST76338386107",
    "reserveOrderNo": "201911132005203337",
    "orderNo": "201911132002493306",
    "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": "201911132002671327",
      "sellerOrderProductReferenceKey": "TV_2150246090",
      "orderProductStatusCode": "OPSPAED",
      "orderProductStatusName": "결제완료",
      "cpId": "PARTNERTEST",
      "productId": "PROD_EASY",
      "productKindCode": "GENERAL_PRODUCT",
      "productPaymentAmt": 81500,
      "originalProductPaymentAmt": 81500
    }],
    "paymentDetails": [{
      "cardSettleInfo": {
        "cardCompanyName": "신한카드",
        "cardCompanyCode": "CCLG",
        "cardNo": "510737*****9999",
        "cardInstallmentMonthNumber": "00",
        "cardAdmissionNo": "42459836",
        "cardInterestFreeYn": "N",
        "corporateCardYn": "N",
        "partCancelPossibleYn": "Y",
        "acquirersCorporationCode": "CCLG"
      },
      "paymentTradeNo": "201911132003595921",
      "paymentMethodCode": "31",
      "paymentMethodName": "신용카드",
      "paymentAmt": 71500,
      "taxfreeAmt": 0,
      "taxableAmt": 65000,
      "vatAmt": 6500,
      "serviceAmt": 0,
      "tradeYmdt": "20191113164547",
      "pgAdmissionNo": "19567928801329",
      "pgAdmissionYmdt": "20191113164547",
      "easyPaymentYn": "Y"
    }], {
      "paymentTradeNo": "201911132003595924",
```

```
        "paymentMethodCode": "98",
        "paymentMethodName": "PAYCO 포인트",
        "paymentAmt": 10000,
        "taxfreeAmt": 0,
        "taxableAmt": 9091,
        "vatAmt": 909,
        "serviceAmt": 0,
        "tradeYmdt": "20191113164547",
        "pgAdmissionNo": "2019111322440409",
        "pgAdmissionYmdt": "20191113164546",
        "easyPaymentYn": "N"
    }],
    "orderCertifyKey":
"qppOzNtPvT4mLZYkJQYvlpFxOIfpTnratvLVBB6BBHMicCC",
    "paymentCompletionYn": "Y",
    "paymentCompleteYmdt": "20191113164547"
},
"message": "success",
"code": 0
}
```

3.3 inApp 및 iframe 가이드

3.3.1 inApp 적용가이드

- 안드로이드(AOS)
- 쿠키 및 자바스크립트 처리를 위한 웹뷰 설정이 필요함
(*상세내용 https://github.com/nhn/Payco_TechSupport.app_aos 참고)

➤ sample

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

// WebView가 자바스크립트를 사용할 수 있도록 설정
getSettings().setJavaScriptEnabled(true);
getSettings().setJavaScriptCanOpenWindowsAutomatically(true);

// Android LOLIPOP 이상에서만 작동하도록함
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {

// http에서 https를 호출할 수 있도록 설정
webView.getSettings().setMixedContentMode(WebSettings.MIXED_CONTENT_ALWAYS_ALLOW);

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

- 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);
}
```

- 'tel:' URL scheme이 호출되는 경우 전화걸기가 실행될 수 있도록 처리가 필요함
- 'about:blank;' 호출되는 경우 브라우저가 열리지 않고 무시하도록 처리가 필요함

● 아이폰(iOS)

- inApp 주문예약시 추가설정

(*상세내용 https://github.com/nhn/Payco_TechSupport.app_ios 참고)

주문예약 > extraData > appUrl

Column	Type	Max Length	필수여부	Description
appUrl	string	4000	Y	inApp 결제 시 PAYCO(앱결제시)/ISP모바일 등 앱에서 결제를 처리한 뒤 복귀할 앱 url scheme (ex. testapp://open)

- iOS의 경우 appUrl 이 전달되어야 가맹점 App으로 정상 복귀할 수 있습니다.

- 스킴 추가(Optional)

- iOS PAYCO 결제 연동은 기본적으로 유니버설링크 기반으로 동작합니다. 다만 앱업데이트나 iOS 업데이트 과정에서 유니버설링크가 비활성화 되는 경우가 있으며 이런 상황에서 PAYCO 결제 브릿지 웹뷰 내부에서는 URLScheme 방식으로 우회하여 결제를 처리합니다.
- PAYCO 결제 브릿지 웹뷰 내부에서 URLScheme 처리를 위해 가맹점 앱 내부에서 canOpenURL(:)로 앱 설치 여부를 판단하는 코드가 있을 경우 Info.plist에 아래와 같이 앱 스킴이 추가되어야 합니다.

```
<key>LSApplicationQueriesSchemes</key>
<array>
  <string>payco</string>
  ...
</array>
```

- 쿠키 저장소 정책 변경

- 원활한 결제 Flow 제공을 위해 쿠키 저장소 정책을 HTTPCookie.AcceptPolicy.always로 설정 필요

➤ Sample(swift)

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {

    let cookieStorate: HTTPCookieStorage = HTTPCookieStorage.shared
    cookieStorate.cookieAcceptPolicy = .always
    ...
    return true
}
```

➤ **Sample(Objective-C)**

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

• 웹뷰 델리게이트 메소드 구현

➤ **Sample(swift)**

```
//WKWebView를 사용할 경우, WKNavigationDelegate의 메소드 예시
func webView(_ webView: WKWebView, decidePolicyFor navigationAction:
WKNavigationAction, decisionHandler: @escaping (WKNavigationActionPolicy) -> Void) {
    ...
    var policy: WKNavigationActionPolicy = .allow
    if let url: URL = navigationAction.request.url,
        let scheme: String = url.scheme {
        switch scheme {
            case "payco":

                policy = .cancel

                if #available(iOS 10.0, *) {
                    UIApplication.shared.open(url, options: [:]) { (success) in
                        decisionHandler(policy)
                    }
                } else {
                    UIApplication.shared.openURL(url)
                    decisionHandler(policy)
                }
                break

            default:
                decisionHandler(policy)
                break
        }
    }
    ...
}

//UIWebView를 사용할 경우, UIWebViewDelegate의 메소드 예시
func webView(_ webView: UIWebView, shouldStartLoadWith request: URLRequest,
navigationType: UIWebView.NavigationType) -> Bool
{
    ...
    if let url: URL = request.url,
        let scheme: String = url.scheme {
        var result : Bool = true
        switch scheme {
            case "payco":
                result = false

                if #available(iOS 10.0, *) {
                    UIApplication.shared.open(url, options: [:], completionHandler:
nil)

                } else {
                    UIApplication.shared.openURL(url)
                }
                break
        }
    }
}
```

```

        default:
            break
    }
}

//UIWebView를 사용할 경우, UIWebViewDelegate의 메소드 예시
func webView(_ webView: UIWebView, shouldStartLoadWith request: URLRequest,
navigationType: UIWebView.NavigationType) -> Bool
{
    var shouldStartLoadWith: Bool = true

    if let url: URL = request.url,
        let scheme: String = url.scheme {

        switch scheme {
            case "payco":

                if #available(iOS 10.0, *) {
                    shouldStartLoadWith = false
                    UIApplication.shared.open(url, options: [:], completionHandler:
nil)

                } else {
                    shouldStartLoadWith = false
                    UIApplication.shared.openURL(url)

                }
                break

            default:
                break
        }

    }

    return shouldStartLoadWith
}

```

➤ Sample(Objective-C)

```

//WKWebView를 사용할 경우, WKNavigationDelegate의 메소드 예시
- (void)webView:(WKWebView *)aWebView
decidePolicyForNavigationAction:(WKNavigationAction *)navigationAction
decisionHandler:(void (^)(WKNavigationActionPolicy))decisionHandler
{
    ...
    WKNavigationActionPolicy sPolicy = WKNavigationActionPolicyAllow;
    NSURL *sURL = navigationAction.request.URL;
    NSString *sScheme = sURL.scheme;

    if ([@"payco" isEqualToString:sScheme]) {

        sPolicy = WKNavigationActionPolicyCancel;

        if (@available(iOS 10.0, *)) {

            [[UIApplication sharedApplication] openURL:sURL options:@{
completionHandler:^(BOOL sSuccess) {

                decisionHandler(sPolicy);

            }}];
        }
        else {

```

```
        [[UIApplication sharedApplication] openURL:sURL];

        decisionHandler(sPolicy);
    }
}
...
}

//UIWebview를 사용할 경우, UIWebViewDelegate의 메소드 예시
- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:(NSURLRequest
*)request navigationType:(UIWebViewNavigationType)navigationType
{
    NSURL *sURL                = request.URL;
    NSString *sScheme           = sURL.scheme;

    if ([@"payco" isEqualToString:sScheme]) {
        if (@available(iOS 10.0, *)) {
            [[UIApplication sharedApplication] openURL:sURL options:@{
completionHandler:^(BOOL sSuccess) {}]];
            return NO;
        }
        else {
            [[UIApplication sharedApplication] openURL:sURL];
            return NO;
        }
    }

    return YES;
}
```

3.3.2 iframe 적용 이슈

- iframe 사용 시 아래의 특이사항으로 인하여 사용을 권장하지 않습니다.
 - *PC에서는 사용 가능하며, Mobile/App은 사용 불가
 - *iframe 적용이 꼭 필요한 경우 별도 문의 주시기 바랍니다. (PAYCO 기술지원: dl_payco_ts@nhn.com)
- iOS 14 출시 이후 Apple 정책으로 인한 iOS 14 이용자 대상 모바일 웹결제 불가
(추후 Chrome 브라우저도 보안 강화 시 동일이슈 발생 가능성 있음)

3.4 외부가맹점 API 상세

외부가맹점은 아래 API를 제공해야 합니다.

API 스펙은 PAYCO에서 정의한대로 개발되어 있어야 합니다

3.4.1 추가 배송비 조회 (바로구매형만 해당)

- API URL은 PAYCO 측으로 미리 등록 요청이 필요하다.
 - 해당 API는 바로구매형인 경우에만 배송지 정보 입력 시 호출된다.
 - 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 3-1 추가 배송비 조회 > INPUT

Column	Type	Max Lenth	Description	Example
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

3. PAYCO 연동하기

- 표 3-2 추가 배송비 조회 > OUTPUT

Column	Type	Max Length	Description	필수 여부	Example
1 resultCode	Number		결과코드 (0이면 성공, Y 그외는 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}
```

3.4.2 결제인증 완료

- 주문예약 API 연동 시 파라미터로 'returnUrl'을 전달하게 된다.
- 'returnUrl'은 결제인증완료 URL로 외부가맹점이 제공하는 페이지이다.
- 결제인증 **성공**시점 또는 **사용자취소 시** 외부가맹점의 returnUrl로 이동한다.
- 캐릭터셋은 UTF-8이며, GET로 전달한다. (별도의 요청이 있는 경우 POST로도 가능)
- 예약 시 returnUrlParam 값에 셋팅된 정보가 존재시 해당값을 추가하여 returnUrl로 호출한다.
- 자세한 항목은 아래 'returnUrl 로 전달되는 정보'를 참고한다.
- 가맹점에서는 총결제금액(totalPaymentAmt)과 외부가맹점 주문번호가 일치하는지 확인한다.
- returnUrl에서는 주문이 가능한 경우 결제승인 처리 API를 호출하여 주문이 완료(결제완료)될 수 있도록 하고 결제팝업창을 Close한 다음 가맹점 정책에 따라 주문완료 페이지로 이동한다.

• returnUrl로 전달되는 정보

Column	Type	Max Length	Description	Example
reserveOrderNo	String	20	주문예약번호	202106142005468927
sellerOrderReferenceKey	String	100	Y	외부가맹점 주문번호
paymentCertifyToken	String	100	결제인증토큰 - 결제승인시 필요	9FAFF0967482F8DE25AB927 16C073842335235235
totalPaymentAmt	Number		총결제금액	5500
totalRemoteAreaDeliveryFee Amt	Number		총도서관간비	0
discountAmt	Number		쿠폰할인금액	1000 (PAYCO포인트 미포함)
pointAmt	Number		PAYCO 포인트 사용금액	
mainPgCode	Number		주 결제수단 코드	31
code	Number		결과코드	0 : 성공 2222 : 사용자에게 의한 취소(close) 그 외 코드: 오류

➤ returnUrl Param

```
code=0&
reserveOrderNo=202106142005468927&
sellerOrderReferenceKey=mallseq16236724018800&
mainPgCode=31&
totalPaymentAmt=5500&
totalRemoteAreaDeliveryFeeAmt=0&
discountAmt=1000&
pointAmt=0&
paymentCertifyToken=il5wtXe94lxKlf3p7yqbfaKl8ziSqmd5owOmbi_3RjWkf6irkFh4fAGim3PVNgkm&
sellerKey=S0FSJE
```

3.4.3 무통장 입금 완료 통보

- 주문예약 API 연동 시 파라미터로 nonBankbookDepositInformUrl 전달 하게된다.
- PAYCO는 무통장입금 계좌에 입금이 완료되는 시점에 nonBankbookDepositInformUrl API를 호출한다
- Server to Server로 호출된다.
- 결제 관련 정보는 'response' 라는 이름의 parameter로 제공되며, parameter value를 UTF-8 UrlDecode하면 Json형태의 값을 받아볼 수 있다.
- 주문완료통보 API와 동일한 포맷으로 전달된다.
- 자세한 항목은 아래 nonBankbookDepositInformUrl 로 전달되는 정보'를 참고한다.
- nonBankbookDepositInformUrl 결과로 성공은 'OK' 실패는 'ERROR'를 리턴한다.
OK, ERROR 이외의 문자 또는 공백이 전달되지 않도록 유의한다.

• NonBankbookDepositInformUrl로 전달되는 정보

Column	Type	Max Length	Description	Example
sellerOrderReferenceKey	String	100	외부가맹점에서 관리하는 주문번호	
reserveOrderNo	String	20	주문예약번호	2014082 1100000 0720
orderNo	String	20	주문번호 - 결제 이후 PAYCO에서 발급되는 Key이다. - reserveOrderNo 주문예약번호는 orderNo를 발급받기 전까지의 임의의 Key 개념임	2014082 1100000 072

- orderNo가 발급된 이후에는 PAYCO와 orderNo로 통신을 하게 된다. (결제 완료 API의 응답 값으로 전달됨)				
memberName	String	15	주문자명(간편결제형만 일부 마스킹처리)	
memberEmail	String	200	주문자EMAIL(일부 마스킹처리)	
orderChannel	String	N/A	주문채널 (PC/MOBILE)	PC
totalOrderAmt	Number		총주문금액	116500
totalPaymentAmt	Number		총결제금액	119000
paymentCompletionYn	String	1	결제완료여부 (Y/N)	Y
orderProducts	Array	N/A	주문상품 List	
paymentDetails	Array	N/A	결제내역 List	
serviceUrlParam	String	1000	주문예약 API호출시 전달한 serviceUrlParam 항목	
orderCertifyKey	String	100	주문완료통보시 내려받은 인증값	
receiptAmt	Number		현금영수증 발급 대상 금액	0

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

⇒ **orderProducts (주문상품List)**

Column	Type	Max Length	Description	Example
orderProductNo	String	20	주문상품번호	20140821100000 022
cpId	String	30	상점ID	PS_TOASTCO
productId	String	50	상품ID	PS_TOASTCOMC 1
sellerOrderProductReferenceKey	String	100	외부가맹점에서 발급한 주문상품연동Key	19801895
orderProductStatusCode	String	50	주문상품상태코드	PAYED

3. PAYCO 연동하기

orderProductStatusName	String	10	주문상품상태명	결제완료
------------------------	--------	----	---------	------

⇒ paymentDetails (결제내역 List)

Column	Type	Max Length	Description	Example
paymentTradeNo	String	20	결제번호	201409191000001142
paymentMethodCode	String	10	결제수단코드	01
paymentMethodName	String	15	결제수단명	신용카드
paymentAmt	Number		결제금액	119000
taxfreeAmt	Number		면세금액	0
taxableAmt	Number		과세금액	0
vatAmt	Number		부가세	0
serviceAmt	Number		봉사료	0
tradeYmdt	String	14	결제일시 (yyyyMMddHHmmss)	20140822120201
pgAdmissionNo	String	100	원천사승인번호	209380
pgAdmissionYmdt	String	14	원천사승인일시 (yyyyMMddHHmmss)	20140822120201
easyPaymentYn	String	1	간편결제여부 (Y/N)	N
nonBankbookSettleInfo	Object	N/A	무통장입금 결제정보	

⇒ nonBankbookSettleInfo (무통장입금 결제정보)

	Column	Type	Max Length	Description	Example
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": "TEST4934699724",
  "reserveOrderNo": "201910172005189733",
  "orderNo": "201910172002486633",
  "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": "201910172002661159",
    "sellerOrderProductReferenceKey": "TV_2150246090",
    "orderProductStatusCode": "OPSPAED",
    "orderProductStatusName": "결제완료",
    "cpId": "PARTNERTEST",
    "productId": "PROD_EASY",
    "productKindCode": "GENERAL_PRODUCT",
    "productPaymentAmt": 81500,
    "originalProductPaymentAmt": 81500
  }],
  "paymentDetails": [{
    "nonBankbookSettleInfo": {
      "bankName": "우리은행",
      "bankCode": "20",
      "accountNo": "47963199118490"
    },
    "paymentTradeNo": "201910172003575713",
    "paymentMethodCode": "02",
    "paymentMethodName": "무통장입금",
    "paymentAmt": 81500,
    "taxfreeAmt": 0,
    "taxableAmt": 74091,
    "vatAmt": 7409,
    "serviceAmt": 0,
    "tradeYmdt": "20191017185256",
    "pgAdmissionNo": "47963199118490",
    "pgAdmissionYmdt": "20191017065256",
    "easyPaymentYn": "N"
  }],
  "orderCertifyKey": "cz4BMUyFJQSVzLBbSh9L9ZWKbaBBemGY4046ATXUdtpwCC",
  "paymentCompletionYn": "Y",
  "paymentCompleteYmdt": "20191017185256"
}
```

3. PAYCO 연동하기

```
}
```

3.5 유의사항

➤ 주문예약(reserve) 및 인증토큰 유효시간

구 분	유효시간	설 명
주문예약	30분	payExpiryYmdt 파라미터 설정이 없다면 기본 30분
결제인증(토큰)	30분	결제인증 시 발급된 토큰의 유효 시간. (인증완료 후 해당 인증정보로 승인요청이 가능한 시간은 30분입니다)
무통장입금 완료 통보		주문예약 시 virtualAccountExpiryYmd 파라미터를 YYmmdd 형식으로 보냈을 경우 해당일 24시 까지

➤ 서버 타임아웃(Timeout) 설정

- PAYCO API 호출 시 네트워크 지연 등으로 인해 응답시간이 길어질 수 있으므로
가맹점 서버 Timeout 설정은 최소 **30초 이상으로 권장** 드립니다.

➤ 결제 최소금액

- 신용카드의 경우 최소결제금액은 100원이며, PAYCO 포인트는 1원입니다.

➤ PAYCO 방화벽 정책

- 공인 아이피가 아닌 사설아이피는 등록 불가. (10, 172, 192, 168로 시작하는 IP)

3.6 코드 정의

3.6.1 주문유형(=결제유형)

파트너센터 주문통계페이지 주문유형별 통계제공

Code	설명
EASYPAY	간편결제형 (가맹점 비회원결제)
EASYPAY_F	간편결제형 (가맹점 회원결제)

3.6.2 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

3.6.3 주문채널(=결제채널)

Code	설명
PC	PC
MOBILE	MOBILE

3.6.4 결제유형

Code	설명
ORIGINAL_APPROVAL	원 결제
WHOLE_CANCEL	취소

3.6.5 결제수단

* 결제수단 코드는 필요에 따라 추가될수 있으니 참고하시기 바랍니다.

■ 간편결제

Code	설명
31	신용카드
35	간편계좌
51	신용카드(자동결제)
60	휴대폰
98	PAYCO 포인트
70	식권 쿠폰
75	페이코 쿠폰
76	카드 쿠폰
77	가맹점 쿠폰
78	상품 쿠폰
96	충전금 환불

3.6.6 상품타입

Code	설명
GENERAL_PRODUCT	일반상품
DELIVERY_FEE	배송비

3.6.7 과세타입

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

3.6.8 카드사코드

: 카드발급사 기준

Code	설명
CCBC	BC카드
CCKM	KB국민카드
CCNH	NH농협카드
CCKJ	광주카드
CCLO	롯데카드
CCSS	삼성카드
CCSU	수협카드
CCLG	신한카드
CCCT	씨티카드
CCPH	우리카드
CCJB	전북카드
CCCJ	제주카드
CCHN	하나카드
CCDI	현대카드
CCKA	카카오뱅크카드

3.6.9 통신사코드

Code	설명
CJH	CJH
KCT	KCT
KFT	KFT
LGT	LGT
SKT	SKT

3.6.10 은행코드

Code	은행명	Code	은행명
02	산업은행	89	케이뱅크
03	기업은행	90	카카오뱅크
04	국민은행	103	SBI 저축은행

05	외환은행	209	유안타증권
07	수협	218	KB 증권
11	농협	227	KTB 증권
12	단위농협	230	미래에셋증권
20	우리은행	238	미래에셋대우
23	SC 제일은행	240	삼성증권
26	신한은행	243	한국투자증권
27	씨티은행	247	NH 투자증권
31	대구은행	261	교보증권
32	부산은행	262	하이투자증권
34	광주은행	263	현대차투자증권
35	제주은행	264	키움증권
37	전북은행	265	이베스트투자증권
39	경남은행	266	SK 증권
45	새마을금고	267	대신증권
48	신협	269	한화투자증권
50	저축은행	270	하나금융투자
54	HSBC 은행	278	신한금융투자
55	도이치은행	279	DB 금융투자
57	JP 모간	280	유진투자증권
60	BOA	287	메리츠종합금융증권
61	BNP 파리바	288	카카오페이증권
62	중국공상	289	NH 투자증권
64	산림조합중앙회	290	부국증권
67	중국건설	291	신영증권
71	우체국	292	케이프투자증권
81	하나은행	294	펀드온라인코리아
88	신한은행	432	한화생명

3.6.11 에러코드

Code	Message
1000	회원 정보가 없습니다.
1001	파라미터 정보가 유효하지 않습니다.
1002	인증 정보가 유효하지 않습니다.
1003	상품주문상태가 유효하지 않습니다.

3. PAYCO 연동하기

1004	유효하지 않은 주문번호입니다.
1005	유효하지 않은 주문번호입니다.
1006	주문상품내역이 없습니다.
1007	회원정보가 없습니다.
1008	유효하지 않은 주문상품번호입니다.
1009	서비스 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.
1021	허용되지 않은 IP 입니다.
1022	등록되지 않은 가맹점 코드(sellerKey)입니다.
1023	등록된 정보와 가맹점 코드(sellerKey)가 일치하지 않습니다.
1024	등록된 자동결제정보와 요청정보가 일치하지 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)
1040	쿠폰 사용 불가합니다.
1041	PAYCO 결제 이력이 있어 프로모션 상품 구매가 불가합니다.
1042	환불 처리 오류
1043	죄송합니다. 해당카드는 롯데면세점에서 결제가 불가합니다. 다른 카드를 이용해 주세요.
1051	미성년자에게는 제공되지 않는 기능입니다.
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.
1107	파라미터 정보가 유효하지 않습니다.
1108	이미 주문 완료된 데이터가 존재합니다.
1111	이미 해지된 건입니다.
1112	이미 등록된 건입니다.
1201	존재하지 않는 Member Key 입니다.
1202	휴면 계정 입니다.
1207	휴대폰본인인증 정보가 유효하지 않습니다.
1301	기취소건입니다.
1303	PAYCO 포인트 잔액 부족으로 주문을 취소할 수 없습니다. PAYCO 고객센터 : 1544-6891
1304	가맹점의 취소가능금액이 부족하여 취소할수 없습니다. 가맹점에서 페이코 파트너센터> 충전금 관리 메뉴에서 일정금액 충전하시면 취소 가능합니다.
1305	진행중인 취소요청건이 있습니다.
1306	PAYCO 환불 불가(PAYCO 환불계좌 정보가 없음)
1307	부분취소 불가(쿠폰 또는 포인트 사용)
1308	부분취소 불가(복수개의 쿠폰 사용)
1309	취소 완료된 결제(취소 가능 잔액 : 0 원)
1310	승인처리 되지 않은 주문건입니다.
1402	이용 한도 초과로 결제가 불가합니다.
1403	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1440	계좌 인증이 필요합니다.
1441	죄송합니다. 알 수 없는 오류 발생으로 송금이 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1450	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1451	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1452	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1453	SMS 본인인증이 필요합니다.
1454	ARS 본인인증이 필요합니다.
1455	죄송합니다. 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. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1457	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1458	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)

3. PAYCO 연동하기

1459	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1461	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1462	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1462	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1464	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1465	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1466	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1467	죄송합니다. 알 수 없는 오류 발생으로 결제가 불가능합니다. 잠시 후 다시 이용해 주시기 바랍니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
1468	단말기 본인인증이 필요합니다.
1501	프로모션 참여가능 불가
1502	PAYCO 제휴 쿠폰 적용 조건을 충족하지 못해 적용이 불가능합니다. PAYCO 고객센터로 문의 해주세요(1544-6891)
1580	등록된 간편결제 정보가 없습니다.
1591	신용정보 조회 중 오류가 발생하였습니다. 잠시 후 다시 시도해 주세요. 문제가 계속될 경우 고객센터(1544-6891)로 연락부탁드립니다.
1601	혜택참여 불가
1701	이 주문에 대한 영수증은 가맹점에서 직접 발급되며, 구매처에서 영수증을 확인하실 수 있습니다.
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.
1706	로그인 동의를 진행해주세요.
1890	점검중입니다.
2222	주문이 취소되었습니다.
3001	잠시 후 다시 시도해주세요.
3301	데이터베이스 처리 실패
3302	데이터베이스 제약 조건 위반
3401	결제취소 처리 실패
3402	취소요청금액이 쿠폰금액보다 작아 취소가 불가능합니다.
3403	취소요청금액이 원거래 금액과 달라 취소가 불가능합니다.
3501	가상계좌 발급 처리 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.
4000	잔여 금액 부족
4002	결제 금액 한도 초과
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.
4602	가상계좌가 존재하지 않습니다.
4604	안심서비스 사용 중인 결제수단으로는 결제할 수 없습니다. 안심서비스 해제 후 이용해 주세요.
4605	유효기간이 만료된 카드입니다. 다른 카드를 이용해 주세요.
4701	결제 비밀번호 처리 중 오류가 발생했습니다.
4702	결제 비밀번호를 잘못입력하셨습니다.
4703	결제 비밀번호 5 회 오류 : 결제 비밀번호를 재설정 해주세요.
4705	결제 비밀번호를 등록해야 서비스를 이용할 수 있습니다.
4706	인증미완료
4707	결제 비밀번호 입력 시간이 초과되었습니다. 다시 시도해 주세요.
4708	이미 결제 비밀번호가 등록되어 있습니다.
4709	지문인증을 위한 연동이 실패했습니다. 비밀번호로 인증해주세요.
4801	계좌 잔액 부족
4802	유효하지 않은 계좌
4803	결제 잔액 부족
4804	명의자 정보 불일치
4805	미등록 결제수단
4806	미지원 결제수단
4807	거래불가 결제수단

3. PAYCO 연동하기

4808	결제수단 정보 불일치
4809	인증 오류 횟수 초과
4810	결제 한도 초과
4811	가맹점 운영 시간 종료
4812	서비스 미지원 가맹점
4813	결제사 시스템 점검
4814	시스템 일시 장애
4815	죄송합니다. 알 수 없는 오류로 해당 계좌의 서비스 이용이 불가능합니다. (동일현상 반복시 PAYCO 고객센터로 문의 요망)
6100	수기취소 불가능한 주문입니다.
6101	부분취소 불가능한 신용카드입니다.
6102	문화누리카드는 부분취소가 불가능합니다.
6103	제로페이 상품권은 부분취소가 불가능합니다.
6201	QR 인증 대기중입니다.
6202	QR 결제 진행 중입니다.
6203	QR 결제 진행 중입니다.
7002	[승인] PG와의 통신 중에 에러가 발생하였습니다.
7003	[결제] PG와의 통신 중에 에러가 발생하였습니다.
7004	[취소] PG와의 통신 중에 에러가 발생하였습니다.
7005	[계좌발급] PG와의 통신 중에 에러가 발생하였습니다.
8801	요청하신 거래 처리가 어렵습니다.(방금 오픈뱅킹 활용에 동의하신 경우 10분 뒤 거래 가능합니다) 기타 문의사항은 고객센터(1544-6891)로 문의하세요.
8802	잔액 조회 실패입니다.기타 문의사항은 고객센터(1544-6891)로 문의하세요.
12100	유효하지 않은 포인트 링크 계좌입니다.
12101	포인트 링크 계좌가 등록되어 있지 않습니다.
12102	이미 충전처리된 건입니다.
14000	제로페이 QR 데이터가 올바르지 않습니다.
14001	제로페이 동의처리가 필요합니다.
14002	제로페이 사용가능한 결제수단이 없습니다.
14003	제로페이 한도가 초과되었습니다.
14004	제로페이 취소 불가능한 시간입니다.
14005	200 원 이상부터 결제할 수 있습니다.결제 금액을 확인해 주세요
14006	최대 200 만원까지 결제할 수 있습니다.결제 금액을 확인해 주세요
14005	제로페이 결제토큰 유효시간 만료
14006	제로페이 미등록 가맹점.
9000	서비스 임시 점검중입니다.
9102	파일처리 실패

9103	암(복)호화 실패
9980	Couldn't parse request.
9991	서비스 점검
9992	마일리지 적립 정책이 없습니다.
10006	결제 실패
10007	존재하지 않는 주문입니다.
30001	PLP 실패
30002	PLP 미등록된 사용자입니다.
30003	등록되지 않은 가맹점 코드(companyCode)입니다.
30010	충전계좌가 안심서비스 이용 중입니다. 안심서비스를 해제하거나 충전계좌를 변경해주세요.
9990	Not found URI
9999	알 수 없는 오류가 발생하였습니다.

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

3.7 영수증

3.7.1 PAYCO 영수증 주소

용 도	도메인
테스트용(개발환경)	https://alpha-bill.payco.com
서비스용(운영환경)	https://bill.payco.com

3.7.2 PAYCO 영수증 노출정의

- **고객용 URL** : /outseller/receipt/{PAYCO주문번호}
/outseller/receipt/{orderNo}
- 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

➢ **고객용 샘플 URL**

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