

# **INNOPAY 결제서비스 연동 매뉴얼**

---

**(신용카드, 계좌이체, 가상계좌, 간편결제, 계좌간편결제)**

PC / Mobile Web

2020. 12

Ver. 2.7

본 문서와 정보는 ㈜인피니소프트의 재산입니다. 또한, 모든 정보는 ㈜인피니소프트 소유 정보이며, ㈜인피니소프트의 사전 동의 없이 본 문서의 어떤 정보도 열람, 복사, 유용, 또는 타인과 공유되어서는 안됩니다. 본 문서의 정보는 변경될 수 있으며 변경 시 본 문서는 수정될 것입니다. 본 문서의 내용에 관한 어떠한 의견이라도 귀하의 프로젝트 또는 사업 담당자에게 전달하여 주십시오.

## 개 정 이 력

[illegible]

## 목 차

1. 결제 서비스.....	5
1.1. 결제 서비스의 개요 .....	5
1.2. 결제 서비스구성 .....	5
1.3. 결제 처리 구성도 .....	5
1.4. 결제 서비스의 특징 .....	5
1.5. 결제 관련 용어.....	6
2. 설치 및 설정.....	6
2.1. 결제 서비스 이용 절차 .....	7
2.2. 설치 요구 사항.....	7
2.3. 결제 프로그램.....	7
2.3.1. 웹서버 준비.....	7
2.3.2. js 파일 구성.....	7
2.3.3. 샘플 결제파일 구성 .....	8
2.3.4. 결제 파일 수정.....	8
3. 결제 페이지 구성 .....	8
3.1. 결제 요청 파라미터 설정.....	8
3.1.1. 화면처리방식 .....	10
3.1.2. 결제결과창유무.....	10
3.1.3. 모바일결제창 호출시 필요설정.....	9
3.2. 결제 요청 결과 파라미터.....	11
3.3. 결제 연동 예시.....	12
4. 결제 테스트.....	18
4.1. 결제서비스 별 접속 URL.....	18
5. 기타.....	18
5.1. 결제 취소 / 환불 처리.....	18
5.2. 결제 결과 URL 통보 (필수).....	19
6. 결제 결과 코드 및 메시지.....	20
6.1. 신용카드.....	20
6.2. 계좌이체.....	21
6.3. 가상계좌.....	21
6.4. 간편결제.....	16
6.5. 계좌간편결제.....	16
7. 코드표 .....	22
7.1. 카드사 코드.....	22
7.2. 은행 코드 .....	23

7.3. 가상계좌 은행 코드 .....	24
8. 기술지원 .....	24

## 1. 결제 서비스

### 1.1. 결제 서비스의 개요

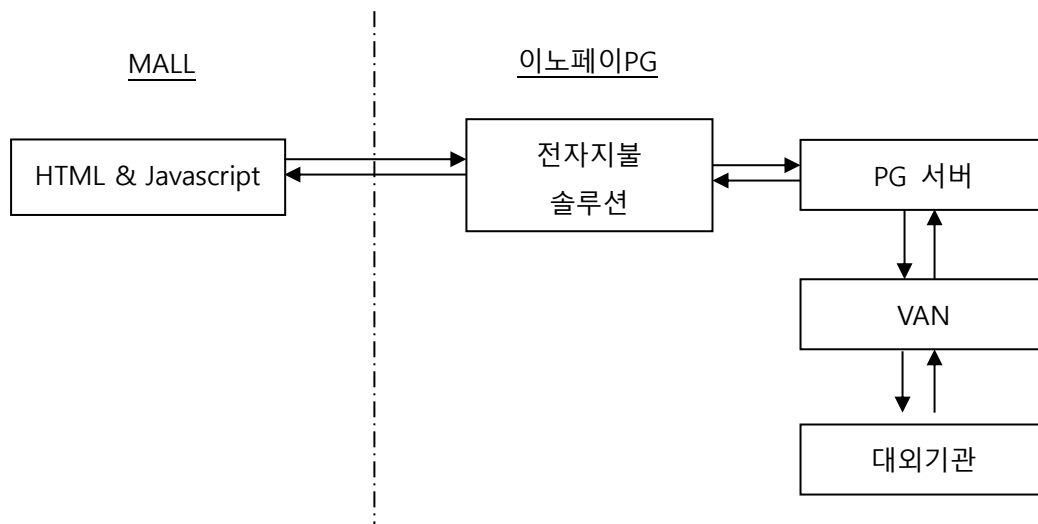
이노페이PG 웹 기반 전자 지불 결제 시스템은 안전하고 신뢰성 있는 인터넷 쇼핑을 도와주는 인터넷 전자 지불 솔루션 입니다.

### 1.2. 결제 서비스구성

구성	내용
결제 요청 처리	결제를 요청하는 WEB PAGE 샘플 제공
결제 결과 처리	지불 요청을 처리하고 결과를 반환하는 WEB PAGE 샘플 제공

### 1.3. 결제 처리 구성도

- ① 상점에서 이노페이PG에서 제공한 결제요청 샘플 WEB PAGE를 참조하여 결제요청 페이지를 개발합니다.
- ② 이노페이PG 전자지불 솔루션 연동 페이지 호출합니다.
- ③ PG서버 통해서 VAN 및 대외기관 승인요청에 대한 결과를 상점페이지로 전송합니다.



### 1.4. 결제 서비스의 특징

서비스	특징
신용카드	상점에서 상품을 구매하거나 서비스를 이용할 때 결제 대금을 신용카드를 이용하여 결제하는 서비스입니다.
신용카드 ISP (인터넷안전결제)	신용카드를 이용하여 상점에서 상품을 구매할 때 신용카드 결제 정보에 대한 보안과 인증 기능이 포함된 신용카드 결제 서비스입니다. - 비씨카드, 국민카드
신용카드 안심클릭	신용카드 결제 시 VISA에서 제공하는 3D 인증 및 보안 기능이 추

	가된 신용카드 결제 서비스입니다. - 신한카드, 삼성카드, 현대카드, 농협카드, 롯데카드, 하나카드 등
계좌이체	온라인 상거래에서 발생하는 각종 결제대금을 고객이 지정한 계좌에서 실시간으로 출금하여 결제할 수 있는 서비스입니다.
가상계좌	구매자는 쇼핑몰 등에서 상품을 주문한 후 상품대금을 입금할 1회용 가상계좌번호를 발급 받아 해당 가상계좌로 상품대금을 입금시 해당 쇼핑몰에 실시간으로 입금내역이 통보되어 해당 주문 건에 입금확인 및 수납처리 완료되는 결제 서비스 입니다.
간편결제	스마트 기기에 최초 1회 사용 시에 카드정보나 본인 명의 은행계좌 등을 등록하면 이후 사용 시에는 비밀번호나 생체인식만으로 빠르게 결제가 가능한 전자 결제 서비스입니다.
계좌간편결제	최초 1회 은행계좌 정보를 등록한 후 비밀번호 입력만으로 간편하게 결제 할 수 있는 결제 서비스입니다.

### 1.5. 결제 관련 용어

구분	내용
MID (Merchant ID)	<ul style="list-style-type: none"> <li>- 상점에 영문과 숫자로 이루어진 10자리의 상점 ID 발급합니다.</li> <li>- 상점은 ID로 상점을 구분하여 거래를 처리하므로 반드시 ID를 발급받아 이노페이PG 전자결제 서비스를 등록을 하셔야 합니다.</li> <li>- 최초 설치시 ID는 테스트용(testpay01m)으로 설정된 상태이므로 운영 전환시 반드시 실제 발급받은 ID를 사용하여야 정상적인 서비스가 가능합니다.</li> </ul>
MOID[OID]	<ul style="list-style-type: none"> <li>- 상점 판매 상품에 대한 주문(order) ID</li> <li>- 각 주문건 별로 고유한 값으로 관리하시는 것을 권장합니다.</li> </ul>
TID	<ul style="list-style-type: none"> <li>- 결제시 생성되는 거래(transaction) ID 입니다.</li> <li>- 결제테스트 진행 후 TID를 기록하여 결제취소 시 사용합니다.</li> </ul>
상점 서명키	<ul style="list-style-type: none"> <li>- 상점 서명키는 결제 데이터의 위변조를 방지하기 위해 MID를 발급시에 설정되는 키입니다.</li> <li>- 서명키는 <a href="https://admin.innopay.co.kr/mer/Login.do">https://admin.innopay.co.kr/mer/Login.do</a> 사이트에서 확인할 수 있습니다.</li> </ul> <p>메뉴: 가맹점정보 -&gt; 가맹점정보 -&gt; KEY 관리</p>

## 2. 설치 및 설정

- 이노페이PG 전자결제 솔루션의 다운로드, 설치, 설치 후 확인작업 등에 대하여 설명합니다
- 이노페이PG 전자결제 서비스 설치를 위한 설치 요구사항과 관련된 내용입니다.

## 2.1. 결제 서비스 이용 절차

구분	내용
회원사 등록 신청	- 결제 서비스를 이용하기 위해 회원사 등록 및 ID 발급을 신청합니다.
프로그램 및 설치 매뉴얼 다운로드	- 결제 페이지 구성을 위해 용도별 해당 프로그램 다운로드 및 설치/연동 매뉴얼을 다운로드 합니다.
서명키 다운로드	- 결제 데이터의 위.변조 방지를 위해 회원사 서명키를 다운로드 합니다.
설치 프로그램 수정	- 결제 서비스 연동을 위해 해당 설치 프로그램을 수정합니다.
설치 프로그램 테스트	- 프로그램의 정상 동작을 확인하기 위해 결제 테스트를 진행합니다.

## 2.2. 설치 요구 사항

구분	내용	요구 사항
웹서버	html	가맹점 웹서버에서 이노페이PG 연동 페이지 호출 필요합니다.
공통 사항	DBMS	- 결제 솔루션에는 DB 연동 기능이 포함되어 있지 않습니다. (데이터베이스 연동을 위한 지불 결과 데이터만 제공).
	브라우저	IE10 이상 및 Chrome, Safari, Opera 커스텀 브라우저(네이버 브라우저 등) 호환성 제공

## 2.3. 결제 프로그램

### 2.3.1. 웹서버 준비

구분	내용
설치 경로	웹 서버에서 {설치경로}/ 디렉터리의 파일들을 브라우저를 통해 접근할 수 있도록 설정합니다. 샘플결제 프로그램 소스를 {설치경로}/ 디렉토리에 업로드합니다.
설치 후 확인	브라우저에서 해당 URL 접속 http://가맹점서버도메인/ipaySample.htm 로 최초 결제 샘플 페이지를 접근할 수 있습니다.

### 2.3.2. js 파일 구성

번호	파일명	설명
1	jquery-2.1.4.min.js	https://pg.innopay.co.kr/ipay/js/jquery-2.1.4.min.js Jquery 자바스크립트 파일 ※ 상위버전에 대한 호환성은 보증하지 않으므로 문제가 발생할 경우 권장버전을 사용하시기 바랍니다.

2	innopay-2.0.js	https://pg.innopay.co.가/ipay/js/innopay-2.0.js 이노페이 결제연동을 위한 자바스크립트 파일
---	----------------	---

### 2.3.3. 샘플 결제파일 구성

번호	파일명	설명
1	ipaySample.htm	결제 연동 샘플 페이지
2	Android 소스	안드로이드 결제 웹뷰 연동 샘플 소스
3	IOS 소스	IOS 결제 웹뷰 연동 샘플 소스

### 2.3.4. 결제 파일 수정

구분	내용
상점 서명키 변경 (ipaySample.htm)	// 상점서명키 (실 상점 MID 적용 시 꼭 해당 상점키로 바꿔주세요) 파라미터명 : MerchantKey
상점 MID 변경 (ipaySample.htm)	// 실 상점 MID 적용 시 꼭 해당 상점 MID로 바꿔주세요) 파라미터명 : MID

## 3. 결제 페이지 구성

이노페이PG 결제 프로그램을 통해 결제 요청 및 결제 결과 처리 시 설정되어야 할 값들을 설명합니다.

**※ 결제 요청 및 결과의 각 파라미터는 대소문자를 구분합니다.**

### 3.1. 결제 요청 파라미터 설정

#### 타입

숫자 : N

알파벳 : A

숫자+알파벳+한글 : AH

숫자+알파벳 : AN

알파벳+숫자+특수문자 : ANS



순번	파라미터	파라미터 내용	길이	타입	필수	비고
1	PayMethod	지불수단	10	A	필수	CARD:신용카드 BANK:계좌이체 VBANK:가상계좌 CARS:ARS결제요청 CSMS:SMS결제(인증) DSMS:SMS결제(비인증) CKEYIN:신용카드(비인증) EPAY:간편결제(인증) EBANK:계좌간편결제(인증)
2	MID	상점아이디	10	AN	필수	계약 시 부여된 Merchant ID 테스트 시에는 임의로 부여된 테스트 아이디로 테스트 가능.
3	MerchantKey	상점라이센스키	50	ANS	필수	상점MID 별 라이센스키 가맹점관리자에서 확인가능
4	GoodsName	상품명	40	AH	필수	영문: 40자리 한글: 20자리
5	Amt	거래금액	12	N	필수	상품가격 ' ' 는 입력하지 않음 예) 1004
6	Moid	상점주문번호 (고유값)	40	AN	필수	아래 특수문자 포함 불가 ~`~;{}[]<>.,!@#\$\$%^&*()_+ WW/?
7	BuyerName	구매자명	30	AH	필수	한글: 15자리
8	BuyerTel	구매자연락처	40	N	필수	숫자만 입력
9	BuyerEmail	구매자메일주소	60	AN	필수	공백/NULL 불가 이메일주소가 부정확하거나 임의의 값으로 요청된 경우 고객에 세 결제결과 메일이 전송되지 않음
10	ResultYN	결제결과창유무	1	A	필수	Y: PG결제 결과창 출력 N: 결제결과창 없이 상점 ReturnURL로 결제결과 전송
11	ReturnURL	결제결과 전송 URL	100	AH	필수	상점 결제결과 전송 URL을 http 포함 전송 상품 결제 성공 후 상점으로 결과 전송될 URL ex) http://www.aa.co.kr/return
12	FORWARD	결제창 연동방식	1	A	선택	X: 레이어팝업(기본값)

13	GoodsCnt	결제 상품 개수	2	N	선택	default : 1 가맹점 참고용
14	MallReserved	상점예비정보	100	AH	선택	가맹점 임의데이터를 파라미터로 보냄
15	OfferingPeriod	제공기간	20	N	선택	상품에 대한 별도제공기간이 있는 경우
16	ArsConnType	ARS 연동방식	2	N	선택	ARS결제 연동시 필수 01: 호전환 02: 가상번호(기본값) 03: 대표번호
17	DutyFreeAmt	면세금액	12	N	선택	복합과세/면세 가맹점용
18	EncodingType	인코딩타입	10	ANS	선택	euc-kr, utf-8(기본값)
19	MallIP	가맹점서버 IP	20	N	선택	가맹점에서 설정
20	UserIP	구매자 IP	20	N	선택	가맹점에서 설정
21	mallUserID	가맹점 고객ID	20	AN	선택	가맹점 참고용
22	User_ID	영업사원 ID	20	AN	선택	Innopay에 등록된 영업사원ID
23	VbankExpDate	입금예정일	8	N	선택	가상계좌 입금예정일 (YYYYMMDD)

- 상점에서는 결제요청 시 금액과 결제 후 리턴 되는 금액을 체크해야 합니다.
- 상점예비정보 예 : {Form변수}={Form값}&{Form변수}={Form값}&{Form변수}={Form값}& ...  
이 값을 인코딩 합니다.
- **ReturnURL 이 없는 경우 결제를 요청한 부모창의 URL 로 JSON 형식의 결제결과를  
전송합니다. (결제연동 예시 참조)**

### 3.1.1. 화면처리방식

결제 연동시 레이어팝업으로 결제요청 화면이 연동되며 결제완료 후 상점에서 입력한 ReturnURL 또는 결제를 요청한 페이지로 결제 완료 결과를 전송합니다.

- ReturnURL 사용시 : HTTP GET
- ReturnURL 미사용시 : JSON POST (결제연동 예시 참조)

### 3.1.2. 결제결과창유무

ResultYN을 Y로 설정하면 결제 요청에 대한 PG 처리결과를 브라우저에 출력합니다.

ResultYN을 N으로 설정하면 결제 요청에 대한 PG 처리결과를 브라우저에 출력하지 않고  
결제결과 파라미터만 상점으로 전송합니다.

### 3.1.3. 모바일결제창 호출시 필요설정

결제창의 iframe은 호출시에 호출되어지는 페이지의 크기를 계산하여 그에 맞게 프레임을 띄웁니다. iframe에 대한 width, height가 기존 크기에 비해 크거나 작다면 iframe이 호출되는 페이지의 모바일 디바이스의 화면을 맞춰주는 ViewPort를 설정해 주셔야 합니다.

※

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
```

### 3.2. 결제 요청 결과 파라미터

순번	서비스 구분	파라미터명	파라미터설명	길이	타입	비고
1	공통	PayMethod	지불수단	10	A	CARD:신용카드 BANK:계좌이체 VBANK:가상계좌 EPAY:간편결제 EBANK:계좌간편결제
2		MID	상점ID	10	AN	
3		TID	거래번호	30	AN	
4		mallUserID	가맹점 회원ID	20	AN	가맹점 참고용
5		Amt	금액	12	N	
6		name	결제자명	10	AH	
7		GoodsName	상품명	40	AH	
8		MOID[OID]	주문번호	40	AN	상점에서 전송한 주문번호
9		AuthDate	승인일자	12	N	yyMMddhhmmss 형식
10		AuthCode	승인번호	20	N	
11		ResultCode	결과코드	4	N	
12		ResultMsg	결과메시지	100	AH	
13		MallReserved	상점예비정보	100	ANS	
14		SUB_ID	서브몰 ID	20	AN	사용안함
15		BuyerEmail	구매자이메일주소	60	AN	
16	신용 카드	fn_cd	결제카드사코드	2	N	카드사코드표 참조
17		fn_name	결제카드사명	10	A	
		AcquCardCode	매입사코드	2	N	카드사코드표 참조

		AcquCardName	매입사카드명	10	A	
18		CardQuota	할부개월수	2	N	00:일시불
19	계좌	BankCd	은행코드	3	N	은행코드표 참조
20	이체	BankName	은행명	30	A	
21		ReceiptType	현금영수증 용도구분	1	N	0:미발행 1:발행(개인소득공제) 2:발행(사업자지출증빙)
22		BuyerAuthNum	현금영수증 구매자 인증번호	15	N	휴대폰번호 등
23	가상 계좌	ReceiptType	현금영수증 용도구분	1	N	0:미발행 1:발행(개인소득공제) 2:발행(사업자지출증빙)
24		VbankExpDate	가상계좌입금기 한	8	N	YYYYMMDD
25		VbankName	은행명	30	AH	
26		VbankNum	가상계좌번호	20	N	
27		VBankAccountName	입금자명	30	AH	
28		VBankCd	은행코드	3	N	은행코드표 참조
29		fn_cd	결제카드사코드	2	N	카드사코드표 참조
30	간편 결제	fn_name	결제카드사명	10	AH	
31		AcquCardCode	매입사코드	2	N	카드사코드표 참조
32		AcquCardName	매입사카드명	10	AH	
33		CardQuota	할부개월수	2	N	00:일시불
34		EPayCI	간편결제 코드	2	N	01:카카오 02:LPay 03:PAYCO 04:SSG 06:네이버
35	계좌	BankCd	은행코드	3	N	은행코드표 참조
36		BankName	은행명	30	AH	
37	간편 결제	ReceiptType	현금영수증 용도구분	1	N	0:미발행 1:발행(개인소득공제) 2:발행(사업자지출증빙)
38		BuyerAuthNum	현금영수증 구매자 인증번호	15	N	휴대폰번호 등

- 상점에서는 결제 요청 시 금액과 결제 후 리턴 되는 금액을 체크해야 합니다.

### 3.3. 결제 연동 예시

## 1. JavaScript Include

- html 페이지에 아래 javascript include

```
<script type="text/javascript" src="https://pg.innipay.co.kr/ipay/js/jquery-2.1.4.min.js"> </script>
```

```
<script type="text/javascript" src="https://pg.innipay.co.kr/ipay/js/innipay-2.0.js"> </script>
```

## 2. 결제요청 JavaScript 함수 호출

**innipay.goPay({**

```
    /// 필수 파라미터
    PayMethod: frm.PayMethod.value, // 결제수단(CARD,BANK,VBANK,CARS,CSMS,DSMS,EPAY,EBANK)
    MID: frm.MID.value, // 가맹점 MID
    MerchantKey:frm.MerchantKey.value, // 가맹점 라이선스키
    GoodsName:frm.GoodsName.value, // 상품명
    Amt:frm.Amt.value, // 결제금액(과세)
    BuyerName:frm.BuyerName.value, // 고객명
    BuyerTel:frm.BuyerTel.value, // 고객전화번호
    BuyerEmail:frm.BuyerEmail.value, // 고객이메일
    ResultYN:frm.ResultYN.value, // 결제결과창 출력유무(Y./N)
    Moid:'testpay01m01234567890', // 가맹점에서 생성한 주문번호 셋팅
    /// 선택 파라미터
    ReturnURL:frm.ReturnURL.value, // 결제결과 전송 URL

    // FORWARD:"", // 결제창 연동방식 (X:레이어, 기본값)
    // GoodsCnt:"", // 상품갯수 (가맹점 참고용)
    // MallReserved:"", // 가맹점 데이터
    // OfferingPeriod:"", // 제공기간
    // ArsConnType:'02', ///* ARS 결제 연동시 필수 01:호전환, 02(가상번호), 03:대표번호
    */
    // DutyFreeAmt:"", // 결제금액(복합과세/면세 가맹점의 경우 금액설정)
    // EncodingType:'utf-8', // 가맹점 서버 인코딩 타입 (utf-8, euc-kr)
    // MallIP:"", // 가맹점 서버 IP
    // UserIP:"", // 고객 PC IP
    // mallUserID:"", // 가맹점 고객 ID
    // User_ID:"", // Innipay 에 등록된 영업사원 ID
    // Currency:"", // 통화코드가 원화가 아닌 경우만 사용(KRW/USD)

});
```

## 3. 결제결과 수신 JavaScript 함수 호출

- ReturnURL 파라미터가 없는 경우 아래 함수로 결제결과 수신이 가능합니다.

- 결제요청 페이지를 호출한 부모창에서 구현해야 합니다. (innipay\_result, 함수명 변경불가)

```
function innipay_result(data){
```

```
    var a = JSON.stringify(data);
```

```
    // Sample
```

```
    var mid = data.MID;
```

```
    // 가맹점 MID
```

```

var tid = data.TID; // 거래고유번호
var amt = data.Amt; // 금액
var moid = data.MOID; // 주문번호
var authdate = data.AuthDate; // 승인일자
var authcode = data.AuthCode; // 승인번호
var resultcode = data.ResultCode; // 결과코드(PG)
var resultmsg = data.ResultMsg; // 결과메세지(PG)
var errorcode = data.ErrorCode; // 에러코드(상위기관)
var errormsg = data.ErrorMsg; // 에러메세지(상위기관)
var EPayCl = data.EPayCl // 간편결제 종류 (1: 카카오페이, 2: L페이 3: 페이코 4: SSG페이
// 결제결과 처리 프로세스 구현하세요
alert("[ "+resultcode+" ] "+resultmsg);
}

```

#### 4. 모바일 APK 웹뷰(WebView) 연동 (안드로이드)

- 해당 APK [PayDemoActivity.java](#) PAY\_URL 항목  
Final String PAY\_URL = <https://pg.innopay.co.kr/ipay/appLink.jsp>

#### 4. 모바일 APK 웹뷰(WebView) 연동 (IOS)

##### \* Swift

- 해당 IOS App [AfterPayController.swift](#) 항목  
var url = "<https://pg.innopay.co.kr/ipay/appLink.jsp>"

##### \* Objective-C

- 해당 IOS App [AfterPayController.m](#) 항목  
NSString \*url = @"<https://pg.innopay.co.kr/ipay/appLink.jsp>"

#### 5. 모바일 웹뷰 결제 결과 파라미터 상수 선언 (안드로이드)

```

public static final String RESULT_PAY_METHOD = "PayMethod";
public static final String RESULT_MID = "MID";
public static final String RESULT_TID = "TID";
public static final String RESULT_MALL_USER_ID = "mallUserID";
public static final String RESULT_AMT = "Amt";
public static final String RESULT_NAME = "name";
public static final String RESULT_GOODS_NAME = "GoodsName";
public static final String RESULT_OID = "OID";
public static final String RESULT_MOID = "MOID";
public static final String RESULT_AUTH_DATE = "AuthDate";
public static final String RESULT_AUTH_CODE = "AuthCode";

```

```

public static final String RESULT_RESULT_CODE = "ResultCode";
public static final String RESULT_RESULT_MSG = "ResultMsg";
public static final String RESULT_MALL_RESERVED = "MallReserved";
public static final String RESULT_FN_CD = "fn_cd";
public static final String RESULT_FN_NAME = "fn_name";
public static final String RESULT_CARD_QUOTA = "CardQuota";
public static final String RESULT_BUYER_EMAIL = "BuyerEmail";
public static final String RESULT_BUYER_AUTH_NUM = "BuyerAuthNum";
public static final String RESULT_ERROR_CODE = "ErrorCode";
public static final String RESULT_ERROR_MSG = "ErrorMsg";

```

## 6. 모바일 결제 성공시 해당 apk 파일로 결과값 전송 (앱 소스 내 함수 호출) (안드로이드)

```

public void finishActivity( String p_strFinishMsg )
{
    Intent intent = new Intent();

    if ( p_strFinishMsg != null )
    {
        try {
            JSONObject resultData = new JSONObject(p_strFinishMsg);
            intent.putExtra( RESULT_PAY_METHOD, resultData.optString(RESULT_PAY_METHOD) );
            intent.putExtra( RESULT_MID, resultData.optString(RESULT_MID) );
            intent.putExtra( RESULT_TID, resultData.optString(RESULT_TID) );
            intent.putExtra( RESULT_MALL_USER_ID, resultData.optString(RESULT_MALL_USER_ID) );
            intent.putExtra( RESULT_AMT, resultData.optString(RESULT_AMT) );
            intent.putExtra( RESULT_NAME, resultData.optString(RESULT_NAME) ); intent.putExtra( RESULT_GOODS_NAME,
            resultData.optString(RESULT_GOODS_NAME) );
            intent.putExtra( RESULT_OID, resultData.optString(RESULT_OID) );
            intent.putExtra( RESULT_MOID, resultData.optString(RESULT_MOID) );
            intent.putExtra( RESULT_AUTH_DATE, resultData.optString(RESULT_AUTH_DATE) );
            intent.putExtra( RESULT_AUTH_CODE, resultData.optString(RESULT_AUTH_CODE) );
            intent.putExtra( RESULT_RESULT_CODE, resultData.optString(RESULT_RESULT_CODE) );
            intent.putExtra( RESULT_RESULT_MSG, resultData.optString(RESULT_RESULT_MSG) );
            intent.putExtra( RESULT_MALL_RESERVED, resultData.optString(RESULT_MALL_RESERVED) );
            intent.putExtra( RESULT_FN_CD, resultData.optString(RESULT_FN_CD) );
            intent.putExtra( RESULT_FN_NAME, resultData.optString(RESULT_FN_NAME) );
            intent.putExtra( RESULT_CARD_QUOTA, resultData.optString(RESULT_CARD_QUOTA) );
            intent.putExtra( RESULT_BUYER_EMAIL, resultData.optString(RESULT_BUYER_EMAIL) );
            intent.putExtra( RESULT_BUYER_AUTH_NUM, resultData.optString(RESULT_BUYER_AUTH_NUM) );
            intent.putExtra( RESULT_ERROR_CODE, resultData.optString(RESULT_ERROR_CODE) );
            intent.putExtra( RESULT_ERROR_MSG, resultData.optString(RESULT_ERROR_MSG) );
        } catch (Exception e){
            Log.e(TAG, e.getMessage());
        }
        setResult( RESULT_OK, intent );
    }
    else
    {

```

```

        setResult( RESULT_CANCELED );
    }
    finish();
}

```

## 5. 모바일 웹뷰 결제 결과 파라미터 상수 선언 (IOS)

### \* Swift

```

let RESULT_PAY_METHOD = "PayMethod";
let RESULT_MID = "MID";
let RESULT_TID = "TID";
let RESULT_MALL_USER_ID = "mallUserID";
let RESULT_AMT = "Amt";
let RESULT_NAME = "name";
let RESULT_GOODS_NAME = "GoodsName";
let RESULT_OID = "OID";
let RESULT_MOID = "MOID";
let RESULT_AUTH_DATE = "AuthDate";
let RESULT_AUTH_CODE = "AuthCode";
let RESULT_RESULT_CODE = "ResultCode";
let RESULT_RESULT_MSG = "ResultMsg";
let RESULT_MALL_RESERVED = "MallReserved";
let RESULT_FN_CD = "fn_cd";
let RESULT_FN_NAME = "fn_name";
let RESULT_CARD_QUOTA = "CardQuota";
let RESULT_BUYER_EMAIL = "BuyerEmail";

```

### \* Objective-C

```

NSString *RESULT_PAY_METHOD = @"PayMethod";
NSString *RESULT_MID = @"MID";
NSString *RESULT_TID = @"TID";
NSString *RESULT_MALL_USER_ID = @"mallUserID";
NSString *RESULT_AMT = @"Amt";
NSString *RESULT_NAME = @"name";
NSString *RESULT_GOODS_NAME = @"GoodsName";
NSString *RESULT_OID = @"OID";
NSString *RESULT_MOID = @"MOID";
NSString *RESULT_AUTH_DATE = @"AuthDate";
NSString *RESULT_AUTH_CODE = @"AuthCode";
NSString *RESULT_RESULT_CODE = @"ResultCode";
NSString *RESULT_RESULT_MSG = @"ResultMsg";
NSString *RESULT_MALL_RESERVED = @"MallReserved";
NSString *RESULT_FN_CD = @"fn_cd";

```



```

NSString *RESULT_FN_NAME = @"fn_name";
NSString *RESULT_CARD_QUOTA = @"CardQuota";
NSString *RESULT_BUYER_EMAIL = @"BuyerEmail";

```

## 6. 모바일 결제 성공시 해당 apk 파일로 결과값 전송 (앱 소스 내 함수 호출) (안드로이드)

### \* Swift

```

func userContentController(_ userContentController: WKUserContentController, didReceive message:
WKScriptMessage) {
    switch message.name {
    case "payResult": //인증결제 후 브릿지를 통하여 리턴받는 JS 함수명
        if let body = message.body as? String, body == "close"{
            let vcName = self.storyboard?.instantiateViewController(withIdentifier: "beforePay")
            vcName?.modalTransitionStyle = .coverVertical
            self.dismiss(animated: true, completion: nil)
            self.present(vcName!, animated: true, completion: nil)

        }

        let json = convertToDictionary(text: message.body as! String) //딕셔너리로 변경
        print("[W(message.name)] = W(message.body) Wn convert Json [W(json)]")
        let alertController = UIAlertController(title: "success", message: message.body as! String,
        preferredStyle: .alert)
        alertController.addAction(UIAlertAction(title: "확인", style: .default , handler: { action in
            self.delegate?.dataReceived(data: json!) //델리게이트를 통한 데이터 전달
            self.dismiss(animated: true, completion: nil)))
        self.present(alertController, animated: true, completion: nil)
    default:
        let alertController = UIAlertController(title: "error", message: "정의된 userContentController 가
없습니다.", preferredStyle: .alert)
        alertController.addAction(UIAlertAction(title: "확인", style: .default))
        self.present(alertController, animated: true, completion: nil)
        break
    }
}

```

### \* Objective-C

```

- (void)userContentController:(WKUserContentController *)userContentController
didReceiveScriptMessage:(WKScriptMessage *)message{
    if ([message.name isEqualToString:@"payResult"]) {
        NSData *json = [self convertToDictionary:message.body];
        NSLog(@"%s : %@", __PRETTY_FUNCTION__, message.name , message.body) ;
        UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"success"
        message:message.body

```

```

preferredStyle:UIAlertControllerStyleAlert];
    UIAlertAction* defaultAction = [UIAlertAction actionWithTitle:@"확인" style:UIAlertActionStyleDefault
                                                                    handler:^(UIAlertAction * action) {
                                                                    [self.delegate dataReceived:json];
                                                                    [self dismissViewControllerAnimated:true
                                                                    completion:nil];
                                                                    }];

    [alertController addAction:defaultAction];
    [self presentViewController:alertController animated:YES completion:nil];
}else{
    UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"error"
                                                                    message:@"정의된
userContentController 가 없습니다."
                                                                    preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction* defaultAction = [UIAlertAction actionWithTitle:@"확인" style:UIAlertActionStyleDefault
                                                                    handler:^(UIAlertAction * action) {
                                                                    [self dismissViewControllerAnimated:true
                                                                    completion:nil];
                                                                    }];

    [alertController addAction:defaultAction];
    [self presentViewController:alertController animated:YES completion:nil];
}
}

```

## 4. 결제 테스트

- 설치 후 상점 프로세스와의 연동이나 실제 서비스 전환 등을 시행하기 이전에 테스트를 하셔야 합니다.
- 테스트시 오류가 발생하는 경우 당사로 문의해 주십시오.(dev@infinisoft.co.kr)

### 4.1. 결제서비스 별 접속 URL

서비스	접속 URL
결제 테스트	<a href="https://pg.innopay.co.kr/ipay/ipaySample.htm">https://pg.innopay.co.kr/ipay/ipaySample.htm</a>
모바일 결제 테스트	이노페이 웹뷰 샘플 소스

## 5. 기타

### 5.1. 결제 취소 / 환불 처리

(1) 상점관리 시스템을 이용한 취소 방법

- 이노페이PG 에서 제공하는 상점관리 사이트(<https://admin.innopay.co.kr/mer/Login.do>) 에

접속하여 로그인합니다.

- 상단 메뉴의 결제내역, 좌측 메뉴의 거래조회를 클릭합니다. 조회기간 설정 후 조회버튼을 클릭합니다

가맹점	테스트가맹점	결제서비스	선택하세요
거래일	2018/02/07 31 ~ 2018/02/07 31	당일	1주일
구매자명		주문번호	
거래금액		거래상태	선택하세요
TID		결제구분	선택하세요
상품명			

[조 회](#)
[EXCEL](#)
[합계조회](#)

승인	0 건	0 원	취소	0 건	0 원	합계	0 건	0 원	결제대기	0 건	주문취소	0 건	주문타입아웃	0 건
----	-----	-----	----	-----	-----	----	-----	-----	------	-----	------	-----	--------	-----

1 < 1 > 50

가맹점상호	MID	거래일	취소일	거래금액	결제서비스	결제구분	구매자명
-------	-----	-----	-----	------	-------	------	------

데이터가 존재하지 않습니다.

- 취소거래에 해당하는 TID를 클릭합니다. 출력되는 팝업 페이지에 취소 작업자, 취소 사유, 취소비밀번호를 입력하고 취소 버튼을 클릭합니다.

\* 취소 비밀번호는 상점관리사이트에서 가맹점정보관리 메뉴를 이용하여 설정이 가능합니다.

Home > 가맹점정보 > 가맹점정보관리

## 기타

거래취소 비밀번호	..... 재입력	등록
-----------	-----------	----

비밀번호를 등록할 경우 거래 취소시 위에 등록한 비밀번호를 입력해야 합니다.

(2) API 연동을 이용한 방법

- INNOPAY\_취소통합\_연동매뉴얼 을 참조하시기 바랍니다.

## 5.2. 결제 결과 URL 통보 (필수)

- 웹 연동 결제의 경우 브라우저 오류, 네트워크 오류, 사용자 강제종료 등 다양한 사유로 인해 결제결과가 상점에 정상적으로 전송되지 못할 수 있습니다.

이런 문제를 보완하기 위해 결제요청시 상점으로 결제결과가 전송되는 방식 이외에 별도로 등록된 상점URL로 각 결제수단별 결제결과를 전송하는 프로그램을 제공합니다.

- 결제결과 통보 URL 설정

Home > 가맹점정보 > 가맹점정보관리 화면의 **결제데이터 통보** 항목에서 각 결제수단별로

설정이 가능합니다.

- 결제결과 수신 페이지 연동은 이노페이PG\_노티\_매뉴얼 을 참조하시기 바랍니다.

※ 가상계좌 및 ARS / SMS결제를 연동하는 경우 반드시 결제결과 수신페이지를 개발하여 연동해야 합니다.

## 6. 결제 결과 코드 및 메시지

### 6.1. 신용카드

번호	결과코드	메시지
1	3001	카드 결제 성공
2	3002	VAN실패 응답
3	3011	카드번호 오류
4	3012	카드가맹점 정보 미확인
5	3013	카드 가맹점 개시 안됨
6	3014	카드가맹점 정보 오류
7	3021	유효기간 오류
8	3022	할부개월오류
9	3023	할부개월 한도 초과
10	3031	무이자할부 카드 아님
11	3032	무이자할부 불가 개월수
12	3033	무이자할부 가맹점 아님
13	3034	무이자할부 구분 미설정
14	3041	금액 오류(1000원 미만 신용카드 승인 불가)
15	3051	해외카드 미등록 가맹점
16	3052	통화코드 오류
17	3053	확인 불가 해외카드
18	3054	환률전환오류
19	3055	인증시 달러승인 불가
20	3056	국내카드 달러승인불가
21	3057	인증 불가카드
22	3061	국민카드 인터넷안전결제 적용 가맹점
23	3062	신용카드 승인번호 오류
24	3071	매입요청 가맹점 아님
25	3072	매입요청 TID 정보 불일치
26	3073	기매입 거래
27	3081	카드 잔액 값 오류

28	3091	제휴카드 사용불가 가맹점
29	3095	카드사 실패 응답

※ 일부 오류 코드이며 카드사별로 별도의 오류코드로 전송될 수 있습니다.

#### 6.2. 계좌이체

번호	결과코드	메시지
1	4000	계좌이체 결제 성공
2	4001	금결원오류응답
3	4002	회원사 서비스 불가 은행
4	4003	출금일자 불일치
5	4004	출금요청금액 불일치
6	4005	거래번호(TID) 불일치
7	4006	회신 정보 불일치
8	4007	계좌이체 승인번호 오류
9	4008	은행 시스템 서비스 중단
10	4041	금액오류(500원 이하 이체불가)

#### 6.3. 가상계좌

번호	결과코드	메시지
1	4100	가상계좌 발급 성공
2	4101	가상계좌 최대거래금액 초과
3	4102	가상계좌 입금예정일 오류
4	4103	가상계좌 입금예정시간 오류
5	4104	가상계좌 정보 오류
6	4110	가상계좌 입금 성공
7	4141	금액오류(500원 이하 이체불가)
8	4142	가상계좌 연동업체코드 미설정

#### 6.4. 간편결제

번호	결과코드	메시지
1	3001	카드 결제 성공
2	3002	VAN실패 응답
3	3011	카드번호 오류
4	3012	카드가맹점 정보 미확인
5	3013	카드 가맹점 개시 안됨
6	3014	카드가맹점 정보 오류

7	3021	유효기간 오류
8	3022	할부개월오류
9	3023	할부개월 한도 초과
10	3031	무이자할부 카드 아님
11	3032	무이자할부 불가 개월수
12	3033	무이자할부 가맹점 아님
13	3034	무이자할부 구분 미설정
14	3041	금액 오류(1000원 미만 신용카드 승인 불가)
15	3051	해외카드 미등록 가맹점
16	3052	통화코드 오류
17	3053	확인 불가 해외카드
18	3054	환률전환오류
19	3055	인증시 달러승인 불가
20	3056	국내카드 달러승인불가
21	3057	인증 불가카드
22	3061	국민카드 인터넷안전결제 적용 가맹점
23	3062	신용카드 승인번호 오류
24	3071	매입요청 가맹점 아님
25	3072	매입요청 TID 정보 불일치
26	3073	기매입 거래
27	3081	카드 잔액 값 오류
28	3091	제휴카드 사용불가 가맹점
29	3095	카드사 실패 응답

## 7. 코드표

### 7.1. 카드사 코드

코드	카드사명	코드	카드사명
01	비씨	15	우리
02	국민	16	하나
03	외환	21	광주
04	삼성	22	전북
06	신한	23	제주
07	현대	24	산은
08	롯데	32	우체국
11	씨티	33	저축은행

12	NH농협	35	새마을금고
13	수협		

## 7.2. 은행 코드

코드	은행	코드	은행
001	한국은행	081	하나은행
002	산업은행	088	신한은행
003	기업은행	093	한국주택금융공사
004	국민은행	094	서울보증보험
005	외환은행	095	경찰청
007	수협중앙회	099	금융결제원
008	수출입은행	209	동양종합금융증권
011	농협중앙회	218	현대증권
012	농협회원조합	238	미래에셋대우(미래에셋,대우)
020	우리은행	240	삼성증권
023	SC제일은행	243	한국투자증권
027	한국씨티은행	247	우리투자증권
031	대구은행	261	교보증권
032	부산은행	262	하이투자증권
034	광주은행	263	에이치엠씨투자증권
035	제주은행	264	키움증권
037	전북은행	265	이트레이드증권
039	경남은행	266	에스케이증권
045	새마을금고연합회	267	대신증권
048	신협중앙회	268	솔로몬투자증권
050	상호저축은행	269	한화증권
052	모건스탠리은행	270	하나대투증권
054	HSBC은행	278	굿모닝신한증권
055	도이치은행	279	동부증권
056	에이비엔암로은행	280	유진투자증권
057	제이피모간체이스은행	287	메리츠증권
058	미즈호코퍼레이트은행	289	엔에이치투자증권
059	미쓰비시도쿄UFJ은행	290	부국증권
060	BOA	291	신영증권
071	정보통신부 우체국	292	엘아이지투자증권
076	신용보증기금		

077	기술신용보증기금		
-----	----------	--	--

### 7.3. 가상계좌 은행 코드

코드	은행	코드	은행
003	기업은행	032	부산은행
004	국민은행	034	광주은행
005	외환은행	037	전북은행
007	수협중앙회	039	경남은행
011	농협중앙회	045	새마을금고연합회
020	우리은행	071	정보통신부 우체국
023	SC제일은행	081	하나은행
027	한국씨티은행	088	신한은행
031	대구은행		

## 8. 기술지원

Tel : 1688-1250

Email : dev@infinisoft.co.kr