
네이버페이 가맹점 연동 가이드

저작권

Copyright © 2011 NAVER Corp. All Rights Reserved.

이 문서는 네이버㈜의 지적 재산이므로 어떠한 경우에도 네이버㈜의 공식적인 허가 없이 이 문서의 일부 또는 전체를 복제, 전송, 배포하거나 변경하여 사용할 수 없습니다.

이 문서는 정보 제공의 목적으로만 제공됩니다. 네이버㈜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, 네이버㈜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.

네이버㈜는 이 문서의 내용을 예고 없이 변경할 수 있습니다.

문서 정보

문서 개요

이 문서는 네이버페이 가맹점이 구매 및 상품 정보, 찜 정보를 네이버페이와 연동하는 방법을 설명합니다.

독자

이 문서의 독자는 네이버페이 가맹점 사이트 개발자입니다.

문의처

이 문서의 내용에 오류가 있거나 내용과 관련한 의문 사항이 있으면 아래 연락처에 문의합니다.

기술지원: dl_techsupport@navercorp.com

문서 버전 및 이력

버전	일자	이력사항
1.0	2009-07-03	문서 배포
1.1	2009-08-05	문서 업데이트
1.2	2009-09-10	예제 오류 수정
1.3	2010-02-11	버튼 삽입 방식 변경, 배송 정보 연동 추가
1.4	2011-03-07	문의처 변경
1.5	2011-05-26	주문 정보 내용 추가(오픈마켓 하위 사업자 정보, 네이버마일리지 유입 경로)
1.6	2011-08-17	버튼 이미지 변경
1.7	2011-10-26	EC_MALL_PID 추가, 네이버쇼핑 CPA 코드 추가, SA 광고 CLICK ID 추가, 하위 사업자 정보 삭제, 예약 필드 삭제
1.8	2011-12-23	상품 반송 주소 정보 추가, SA CLICK ID 설명 및 예제 수정
1.9	2012-04-03	ITEM_OPTION_CODE 추가
2.0	2012-07-06	체크아웃 유입 경로별 혜택 기능 추가

2.1	2014-12-29	주문 정보 등록 시 주문 정보 내용에 SHIPPING_ADDITIONAL_PRICE 항목 추가
2.2	2015-03-20	"네이버 체크아웃"이 "네이버 페이"로 서비스 이름 변경
2.3	2015-05-21	"네이버 페이"에서 "네이버페이"로, "네이버 마일리지"에서 "네이버페이 포인트"로 서비스 이름 변경, 저작권 업데이트
2.4	2015-11-26	네이버페이 구매하기 버튼 변경 및 네이버 톡톡 기능 추가
2.5	2018-05-31	문화체육관광부의 도서공연비 소득공제 관련 필드 추가, 서비스 명칭 변경 반영, 기타 표기 개선

표기 규칙

UI 메시지 표기

이 문서에서 메뉴 명과 창 이름, 버튼 이름 등 UI(User Interface) 메시지를 표현할 때는 대괄호를 사용해서 '[구매하기]'와 같이 표기합니다. 단, 소스 코드에 사용된 기호는 이 표기 규칙에 해당하지 않습니다.

소스 코드 표기

이 문서에서 소스 코드는 회색 바탕에 검정색 글씨로 표기합니다.

```
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

1. 개요	9
1.1 네이버페이 연동 개요	10
1.2 네이버페이 연동 과정	11
1.2.1 전체 연동 과정	11
1.2.2 구매 정보 연동	11
1.2.3 상품 정보 연동	11
2. 네이버페이 버튼	13
2.1 네이버페이 버튼 삽입	14
2.2 네이버페이 버튼 동작	22
3. 네이버페이 연동	25
3.1 구매 정보 연동	26
3.1.1 네이버페이 [구매하기] 버튼 클릭	26
3.1.2 주문 정보 등록	26
3.1.3 주문서 전송	39
3.1.4 네이버페이 결제	39
3.2 상품 정보 연동	40
3.2.1 상품 정보 요청	40
3.2.2 상품 정보 제공	40
3.3 찜 정보 연동	46
3.3.1 네이버페이 [찜하기] 버튼 클릭	46
3.3.2 찜 정보 등록	46
3.3.3 찜 목록 전송	54
4. 유입 경로 스크립트 적용	57
4.1 네이버 공통 유입 경로 스크립트	58
4.2 유입 경로 스크립트 설정	59
4.3 가맹점 랜딩 페이지 설정	60

4.3.1	로그 수집 공통 함수	60
4.3.2	유입 경로 추적 함수	60

표 및 그림 목록

표 목록

표 2-1 네이버페이 버튼 모음 종류	14
표 2-2 모바일용 네이버페이 버튼 모음 종류	19
표 2-3 [구매하기] 버튼 클릭 시 오류 상황에 따른 팝업 경고 문구	22
표 3-1 주문 정보 등록 URL	26
표 3-2 주문 정보 내용	26
표 3-3 주문서 팝업 창 URL	39
표 3-4 주문서 내용	39
표 3-5 상품 정보 요청 내용	40
표 3-6 상품 정보 제공 내용	40
표 3-7 찜 정보 등록 URL	46
표 3-8 찜 정보 등록 내용	46
표 3-9 찜 목록 전송 팝업 창 URL	55
표 3-10 찜 목록 전송 내용	55

그림 목록

그림 1-1 네이버페이 가맹점 연동 구조도	11
-------------------------	----

1. 개요

이 장에서는 네이버페이 구매 정보 및 상품 정보 연동의 기본 개념을 설명한다.

1.1 네이버페이 연동 개요

네이버페이 가맹점 가입 심사 후 승인 완료되면, 가맹점은 네이버에서 제공하는 가이드에 따라 구매 정보 및 상품 정보 연동 시스템을 개발하여, 이용자가 가맹점 사이트에서 네이버페이를 구매할 수 있는 환경을 만들어야 한다.

네이버페이 버튼을 가맹점 사이트의 상품 상세 페이지와 장바구니 페이지에 삽입하고, 이용자가 네이버페이 버튼을 클릭하면, 가맹점은 구매 정보 또는 찜 정보를 네이버에 등록한다. 네이버페이에서 상품 정보를 요청하면 상품 정보를 전송한다.

가맹점이 사용하는 PG사와 네이버 담당자가 연동 시스템을 테스트하고, 연동 테스트가 완료되면 가맹점은 네이버페이 센터 가입 완료와 동시에 네이버페이 센터 서비스를 이용할 수 있다.

1.2 네이버페이 연동 과정

1.2.1 전체 연동 과정

가맹점은 다음과 같은 과정을 통해 구매 정보 및 상품 정보를 네이버페이와 연동시킨다.

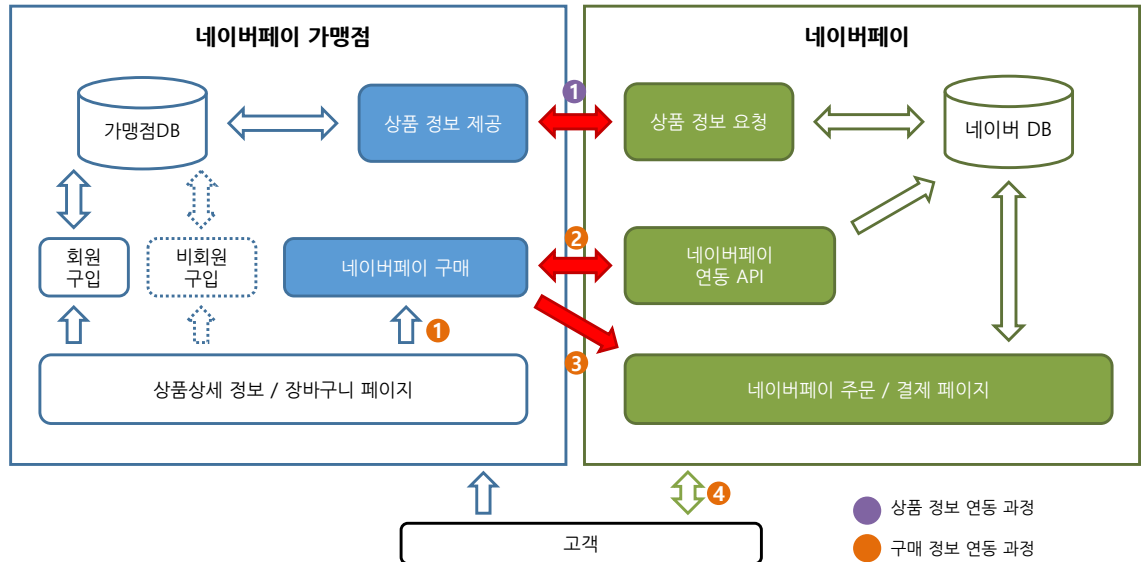


그림 1-1 네이버페이 가맹점 연동 구조도

1.2.2 구매 정보 연동

"그림 1-1"에서 주황색 원의 숫자는 구매 정보 연동 과정을 표시한 것이다. 구매 정보 연동 과정은 다음과 같다.

1. 이용자가 상품 정보 페이지나 장바구니 페이지에서 네이버페이 [구매하기] 버튼을 클릭하면, 해당 이벤트는 가맹점 사이트의 페이지로 전달된다.
2. 이벤트를 받은 페이지에서 네이버페이로 주문 정보를 등록하고 주문 ID를 받는다.
3. 네이버페이 주문서 페이지 팝업 창을 생성하고 주문 ID와 상점 ID, 총 주문 금액을 전송한다.
4. 이용자는 네이버페이 주문/결제 페이지에서 결제한다.

1.2.3 상품 정보 연동

"그림 1-1"에서 보라색 사각형의 숫자는 상품 정보 연동 과정을 표시한 것이다. 상품 정보 연동 과정은 다음과 같다.

1. 네이버페이가 상품의 정보를 요청할 경우, 가맹점은 상품 정보를 전송한다.

2. 네이버페이 버튼

이 장에서는 네이버페이 버튼의 삽입 방법과 네이버페이 [구매하기] 버튼 클릭 시 이벤트 처리 방법을 설명한다.

2.1 네이버페이 버튼 삽입

네이버페이 가맹점 가입 승인이 완료되면, 네이버페이는 상점 ID, 연동 인증키와 함께 네이버페이 버튼을 위한 버튼 인증키를 제공한다.

상품 상세 페이지에 삽입하는 네이버페이 버튼 모음에는 구매하기 버튼, 찜하기 버튼, 네이버 톡톡 버튼이 있고, 장바구니 페이지에 삽입하는 버튼 모음에는 구매하기 버튼만 있다. 버튼 모음은 종류와 색을 선택할 수 있다.

네이버 톡톡 버튼은 네이버 톡톡 연동 중일 때만 노출된다. 네이버 톡톡 미연동 상태에서는 구매하기 버튼과 찜하기 버튼만 노출된다. 이는 네이버페이 활성 상태와 비활성 상태 모두 동일하게 적용된다.

네이버페이로 구매가 불가능한 취급 불가 상품(설치 상품, 주문 제작 상품, 해외 배송 상품 등)에는 비활성 버튼 모음을 삽입해야 한다.

다음의 경우에는 네이버페이 버튼 모음을 삽입하지 않을 수 있다.

- 네이버페이로 구매가 불가능한 상품(개인 결제 창, 0원 상품 등)인 경우
- 상품 품질로 인해 판매가 불가능한 경우
 - 가맹점 사이트의 구매 버튼과 네이버페이 버튼 모음 모두 삽입하지 않는다. (권장)
 - 가맹점 사이트의 구매 버튼과 비활성 네이버페이 버튼 모음을 삽입한다.

2012년 7월 6일에 유입 경로별 혜택 기능이 추가되어 버튼 모음의 세로 길이가 이전보다 27px 증가했다.

표 2-1 네이버페이 버튼 모음 종류

종류	색	버튼 수 (크기)	활성 네이버 톡톡 연동	비활성
A	1	1 (236X88)		
				
2	2	(285X88)		
				

종류	색	버튼 수 (크기)	활성 네이버 톡톡 연동	비활성
B		1 1 (214X83)		
2		(261X83)		
C		1 1 (225X88)		
2		(272X88)		
2		1 (225X88)		
2		(272X88)		

2. 네이버페이 버튼

종류	색	버튼 수 (크기)	활성 네이버 톡톡 연동	비활성
3	1	(225X92)		
	2	(272X92)		
D	1	(210X83)		
	2	(251X83)		
2	1	(210X83)		

종류	색	버튼 수 (크기)	활성 네이버 톡톡 연동	비활성
		2 (251X83)		
		3 1 (210X87)		
		2 (251X87)		
E		1 1 (124X135)		
		2 (124X135)		

2. 네이버페이 버튼

종류	색	버튼 수 (크기)	활성 네이버 톡톡 연동	비활성
			<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>찜하기</p> <p>이벤트 선착순 참가 5,000원</p>	
2	1	(124X135)	<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>이벤트 선착순 참가 5,000원</p>	<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>이벤트 선착순 참가 5,000원</p>
			<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>찜하기</p> <p>이벤트 선착순 참가 5,000원</p>	
2		(124X135)	<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>찜하기</p> <p>이벤트 선착순 참가 5,000원</p>	<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>찜하기</p> <p>이벤트 선착순 참가 5,000원</p>
			<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>찜하기</p> <p>이벤트 선착순 참가 5,000원</p>	
3	1	(124X139)	<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>이벤트 선착순 참가 5,000원</p>	<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>이벤트 선착순 참가 5,000원</p>
			<p>NAVER 네이버ID로 간편구매</p> <p>N Pay 구매</p> <p>이벤트 선착순 참가 5,000원</p>	

종류 색 버튼 수 (크기)	활성 네이버 톡톡 연동	비활성
2 (124X139)		

표 2-2 모바일용 네이버페이 버튼 모음 종류

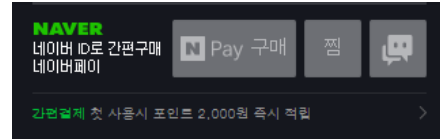
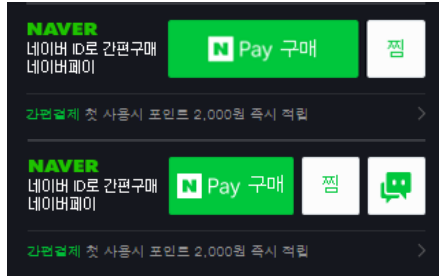
종류 버튼 수 (크기)	활성	비활성
MA 1 (290X85)		
2 (290X85)		
MB 1 (320X100)		

2. 네이버페이 버튼

종류 버튼 수 활성
(크기)

비활성

2
(320X100)



네이버페이 버튼 모음 이미지 삽입을 위한 소스 코드는 다음과 같다.

- PC 버전

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<title>네이버페이 버튼 사용법</title>
<script type="text/javascript"
src="http://pay.naver.com/customer/js/naverPayButton.js" charset="UTF-8"></script>
</head>
<body>

<script type="text/javascript" >
    naver.NaverPayButton.apply({
        BUTTON_KEY: "버튼 인증 키", // 네이버페이에서 제공받은 버튼 인증 키 입력
        TYPE: "E", // 버튼 모음 종류 설정
        COLOR: 1, // 버튼 모음의 색 설정
        COUNT: 2, // 버튼 개수 설정. 구매하기 버튼만 있으면(장바구니 페이지) 1, 찜하기 버튼도
        있으면(상품 상세 페이지) 2를 입력.
        ENABLE: "Y", // 품질 등의 이유로 버튼 모음을 비활성화할 때에는 "N" 입력
        " ": ""
    });
//]]&gt;&lt;/script&gt;

&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="129 677 243 692" data-label="Section-Header"><ul><li>• 모바일 버전</li></ul></div><div data-bbox="152 699 779 904" data-label="Text"><pre>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
&lt;head&gt;
&lt;meta http-equiv="Content-Type" content="text/html; charset=euc-kr"&gt;
&lt;title&gt;네이버페이 버튼 사용법&lt;/title&gt;
&lt;script type="text/javascript"
src="http://pay.naver.com/customer/js/mobile/naverPayButton.js" charset="UTF-
8"&gt;&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;script type="text/javascript" &gt;<![CDATA[
    naver.NaverPayButton.apply({
        BUTTON_KEY: "버튼 인증 키", // 네이버페이에서 제공받은 버튼 인증 키 입력
        TYPE: "MA", // 버튼 모음 종류 설정
        COLOR: 1, // 버튼 모음의 색 설정</pre></div><div data-bbox="67 944 297 959" data-label="Page-Footer"><p>20 · 네이버페이 가맹점 연동 가이드</p></div>
```

```
COUNT: 2, // 버튼 개수 설정. 구매하기 버튼만 있으면(장바구니 페이지) 1, 찜하기 버튼도  
있으면(상품 상세 페이지) 2를 입력.  
ENABLE: "Y", // 품절 등의 이유로 버튼 모음을 비활성화할 때에는 "N" 입력  
"": ""  
});  
//]]></script>  
</body>  
</html>
```

2.2 네이버페이 버튼 동작

이용자가 네이버페이 [구매하기] 버튼을 클릭하면, 가맹점은 이용자가 옵션, 배송료 지불 방법 등의 선택 사항을 모두 선택했는지 확인하고 그렇지 않은 경우 팝업 경고를 표시한다. 이용자가 선택 사항을 모두 선택했거나 이용자 선택 사항이 없으면, 주문 정보 등록을 위한 페이지로 이동한다.

이용자가 비활성화된 네이버페이 [구매하기] 버튼을 클릭하면, 가맹점은 팝업 경고를 표시한다.

팝업 경고의 문구는 다음과 같다.

표 2-3 [구매하기] 버튼 클릭 시 오류 상황에 따른 팝업 경고 문구

오류 상황	팝업 경고 문구
옵션이 있는 상품의 옵션 미선택	상품 옵션을 선택해 주세요.
복수 개의 옵션이 있는 상품의 일부 옵션 미선택	상품 옵션을 모두 선택해 주세요.
배송료 지불 방법 미선택	배송비를 선택해 주세요.
기타 선택 사항 미선택	{선택 사항}을 선택한 후 주문을 완료해 주세요.
네이버페이 버튼 모음 비활성화	죄송합니다. 네이버페이로 구매가 불가능한 상품입니다

다음은 상품 상세 페이지의 예이다.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<title>네이버페이 버튼 사용법</title>
<script type="text/javascript" src="http://pay.naver.com/customer/js/naverPayButton.js"
charset="UTF-8"></script>
<script type="text/javascript" >
function checkOption(selectBox, nonSelectedIndex, optionName)
{
    if (selectBox.selectedIndex == nonSelectedIndex) {
        alert("상품 옵션을 선택해 주세요. (" + optionName + ")");
        return false;
    }
    return true;
}

function checkShippingPrice(radio)
{
    var len = radio.length;
    if (len == undefined) {
        if (radio.checked) {
            return true;
        }
    } else {
        for(var i = 0; i &lt; len; i++) {
            if(radio[i].checked) {
                return true;
            }
        }
    }
    alert("배송비를 선택해 주세요.");
    return false;
}</pre>
</div>
<div data-bbox="67 943 297 959" data-label="Page-Footer">22 · 네이버페이 가맹점 연동 가이드</div>
```

```

}

function buy_nc(url)
{
    var check = checkOption(document.getElementById("opt").color, 0, "색상") &&
    checkOption(document.getElementById("opt").size, 0, "크기") &&
    checkShippingPrice(document.getElementById("opt").spt);

    if ( check ) {
        //네이버페이로 주문 정보를 등록하는 가맹점 페이지로 이동.
        //해당 페이지에서 주문 정보 등록 후 네이버페이 주문서 페이지로 이동.

        location.href=url;
    }

    return false;
}

function wishlist_nc(url)
{
    // 네이버페이로 찜 정보를 등록하는 가맹점 페이지 팝업 창 생성.
    // 해당 페이지에서 찜 정보 등록 후 네이버페이 찜 페이지로 이동.
    window.open(url, "", "scrollbars=yes,width=400,height=267");
    return false;
}

function not_buy_nc()
{
    alert("죄송합니다. 네이버페이로 구매가 불가능한 상품입니다.");
    return false;
}

//]]></script>
</head>

<body>

<div class="first">
<!-- 상품정보 -->
상품명 : 옷 <br/>
<form id="opt" action="" >
    색상 :
    <select id="color">
        <option value="no">-- 색상을 선택해 주세요 --</option>
        <option value="red">빨간색</option>
        <option value="red">파란색</option>
    </select><br/>
    크기 :
    <select id="size">
        <option value="no">-- 크기를 선택해 주세요 --</option>
        <option value="L">Large(L)</option>
        <option value="XL">Extra Large(XL)</option>
    </select><br/>
    배송비 :

    <input type="radio" name="spt" value="FREE"/>무료
    <input type="radio" name="spt" value="PAYED"/>선불
    <input type="radio" name="spt" value="ONDELIVERY"/>착불
</form>

<script type="text/javascript" >/*!CDATA[
naver.NaverPayButton.apply({

```

2. 네이버페이 버튼

```
        BUTTON_KEY: "207994C7-2944-4D98-A24C-387FBE19452C", // 네이버페이에서 할당받은 버튼
        KEY 를 입력하세요.

        TYPE: "E", // 템플릿을 확인하시고 원하는 타입의 버튼을 선택
        COLOR: 1, // 버튼의 색 설정
        COUNT: 2, // 버튼 개수 설정. 구매하기 버튼(장바구니 페이지)만 있으면 1, 찜하기 버튼(상품 상세
        페이지)과 함께 있으면 2 를 입력한다.

        ENABLE: "Y", // 품절등과 같은 이유에 따라 버튼을 비활성화할 필요가 있을 경우
        BUY_BUTTON_HANDLER: buy_nc, // 구매하기 버튼 이벤트 Handler 함수 등록, 품절인 경우
        not_buy_nc 함수 사용

        BUY_BUTTON_LINK_URL: "http://mydomain.com/buy/url/", // 링크 주소 (필요한 경우만 사용)
        WISHLIST_BUTTON_HANDLER: wishlist_nc, // 찜하기 버튼 이벤트 Handler 함수 등록
        WISHLIST_BUTTON_LINK_URL: "http://mydomain.com/wishlist/popup/url/", // 찜하기 팝업
        링크 주소
        "": ""
    });
//]]></script>

</div>

</body>
</html>
```

3. 네이버페이 연동

이 장에서는 가맹점이 네이버페이와 구매 정보, 상품 정보, 찜 정보를 연동하는 방법을 설명하고 PHP, Java, ASP로 작성된 예시를 제공한다.

3.1 구매 정보 연동

3.1.1 네이버페이 [구매하기] 버튼 클릭

이용자가 상품 정보 페이지나 장바구니 페이지에서 네이버페이 [구매하기] 버튼을 클릭하면, 가맹점은 이용자 선택 사항이 모두 선택되었는지 확인한다. 이용자 선택 사항이 모두 선택되었거나 이용자 선택 사항이 없으면 주문 정보 등록을 위한 페이지로 이동한다. 주문 정보 등록 페이지로 이동하는 방법은 가맹점이 선택할 수 있으며, 네이버페이는 Ajax, 페이지 전환, 새 창 띄우기의 순서로 권장한다.

3.1.2 주문 정보 등록

가맹점은 네이버페이에 HTTPS로 주문 정보를 등록하고 주문 ID를 받는다. 주문 정보 등록시 기본 문자 세트는 UTF-8이고, 다른 문자 세트를 사용하는 경우 UTF-8로 변환하거나, 다른 URL을 이용한다.

표 3-1 주문 정보 등록 URL

환경	문자 세트	URL
테스트	UTF-8	https://test-pay.naver.com/customer/api/order.nhn
	EUC_KR	https://test-pay.naver.com/customer/api/CP949/order.nhn
서비스	UTF-8	https://pay.naver.com/customer/api/order.nhn
	EUC_KR	https://pay.naver.com/customer/api/CP949/order.nhn

response로 받는 주문 ID는 영문과 숫자로 이루어지며 최대 19자리이다.

request 시 post로 아래와 같은 정보를 전달한다.

표 3-2 주문 정보 내용

항목	필수 여부	설명
SHOP_ID	Y	상점 ID. 네이버페이에 가입 승인될 때 정해진다.
CERTI_KEY	Y	인증키. 네이버페이에 가입 승인될 때 정해진다.
ITEM_ID	Y	상품 ID
EC_MALL_PID	N	네이버쇼핑 EP의 ma_pid. 네이버쇼핑 가맹점이면 네이버쇼핑 EP의 ma_pid와 동일한 값을 입력해야 한다.
ITEM_NAME	Y	상품 이름
ITEM_COUNT	Y	상품 주문 개수
ITEM_UPRICE	Y	개별 상품 단가. 0보다 커야 한다.
ITEM_TPRICE	Y	해당 상품 총 가격. 상품 할인 행사가 있으면, ITEM_COUNT와 ITEM_UPRICE를 곱한 값보다 작은 값을 가질 수 있다.

항목	필수 여부	설명
ITEM_OPTION	Y	선택한 옵션 사항
ITEM_OPTION_CODE	N	주문 등록 시 각 상품의 옵션별 추가 정보를 저장하기 위한 상품 단위 관리 코드
SHIPPING_PRICE	Y	배송료. 무료이면 0, 선불 또는 착불이면 배송료(0 보다 커야 함). 착불이면서 배송료를 특정할 수 없는 경우에는 0
SHIPPING_TYPE	Y	배송료 지불 방법. 무료이면 "FREE", 선불이면 "PAYED", 착불이면 "ONDELIVERY"
SHIPPING_ADDITIONAL_PRICE	N	지역별 추가 배송료에 대한 안내. 글자 수 기준 최대 50 자 (예: 제주도 3,000 원 추가, 제주도 외 도서·산간 지역 5,000 원 추가).
TOTAL_PRICE	Y	총 주문 금액. ITEM_TPRICE 의 합과 선불 배송료를 더한 값과 같아야 한다.
BACK_URL	Y	네이버페이 주문서 페이지에서 [이전페이지]를 클릭했을 때 이동하는 페이지 URL. 상점 메인 화면으로 돌아가거나, 주문을 따로 저장했다면 주문서 페이지로 돌아가게 할 수 있다.
SALES_CODE	N	경로별 매출 코드. 영문자 기준 최대 300 자. 매출 코드가 필요한 경우에 입력한다.
MALL_MANAGE_CODE	N	몰 내부 관리 코드. 영문자 기준 최대 300 자. 내부 관리 코드가 필요한 경우에 입력한다.
CPA_INFLOW_CODE	N	네이버쇼핑 CPA 코드. 네이버쇼핑 가맹점 중 파라미터 방식을 이용한 CPA 과금을 원하는 가맹점은 이 값을 입력한다.
SA_CLICK_ID	N	SA CLICK ID. 네이버 검색광고 이용 가맹점 중 광고주 센터의 광고 효과 보고서를 통해 네이버페이 전환 데이터를 확인하길 원하는 가맹점은 SA 로부터 받은 추적 URL 파라미터 중 NVADID 를 입력한다.
NAVER_INFLOW_CODE	N	네이버 서비스 유입 경로 코드(네이버 서비스를 통해서 유입된 경우에 서비스를 구분하기 위해 사용되는 코드)를 입력한다.
MCST_CULTURE_BENEFIT_YN	N	문화체육관광부의 도서공연비 소득공제 정책을 적용할 수 있는 주문에 해당한다면 TRUE 를 입력한다.

ITEM_ID, EC_MALL_PID, ITEM_NAME, ITEM_COUNT, ITEM_UPRICE, ITEM_TPRICE, ITEM_OPTION, ITEM_OPTION_CODE는 여러 개일 수 있다. 예를 들어, 한 주문에 상품 종류가 두 개라면 각 파라미터는 두 개씩 존재한다. 이 때 각 파라미터에서 상품의 순서를 지켜야 한다.

고유 키(unique key)는 ITEM_ID가 아니라 (ITEM_ID, ITEM_OPTION)의 쌍이다. 따라서 동일한 ITEM_ID 의 상품의 ITEM_OPTION이 다르면 다른 파라미터로 전송해야 한다.

네이버쇼핑 가맹점은 네이버쇼핑 EP의 ma_pid와 동일한 값을 EC_MALL_PID에 입력해야 한다.

EC_MALL_PID로 네이버쇼핑 카테고리 정보를 추정해서 쿠폰, 네이버페이 포인트 등의 구매자 혜택을 지급하므로 이 값이 일치하지 않으면 구매자가 혜택을 받을 수 없다.

ITEM_OPTION의 값은 실제 주문서 페이지에 그대로 표시된다. 각 쇼핑물은 이용자가 선택한 옵션 사항을 텍스트로 표시할 수 있다. HTML 태그는 사용할 수 없다. ITEM_OPTION이 여러 종류일 경우 슬래시(/)로 구분하는 것을 권장한다. 예를 들어, 색상은 노랑이고 사이즈는 XL이면 ITEM_OPTION은 "노랑/XL" 또는 "색상:노랑/사이즈:XL"로 표기하는 것을 권장한다.

ITEM_OPTION_CODE는 각 가맹점이 주문 등록 시 각 상품의 옵션별 추가 정보를 저장하기 위한 상품 단위 관리 코드이다. 옵션별 상품 코드가 별도로 존재하거나 옵션별 상품 재고 관리가 필요하여 기존의 MALL_MANAGE_CODE만으로는 해당 상품 단위 관리가 어려울 때 이 값을 사용할 수 있다.

TOTAL_PRICE는 이용자가 최종 지불하는 총 금액으로서, ITEM_TPRICE의 총합과 같아야 한다. 배송료가 선불("PAYED")이면, ITEM_TPRICE의 총합에 SHIPPING_PRICE를 더한 값과 반드시 같아야 한다.

SALES_CODE는 유입경로 별 매출을 확인하기 위해 각 가맹점이 별도의 매출 코드 관리가 필요한 경우 입력한다.. 네이버페이의 각 주문에 대한 매출 코드를 출력하는 기능만을 담당하고 별도의 통계 기능 등은 제공하지 않는다.

MALL_MANAGE_CODE는 각 가맹점에서 네이버페이 주문서 호출 시 해당 정보를 자사 DB에 보관할 필요가 있는 경우, 해당 정보를 DB에 보관하고 보관에 대한 KEY 값을 입력한다. 네이버페이 주문 연동 정보가 실제 자사 주문 처리를 위해 필요한 DB와 다르거나 부족할 경우에 사용할 수 있다.

예를 들어 사진 인화 사이트의 경우, 구매자가 업로드한 파일 정보는 네이버페이 주문서 정보에 입력할 수 없다. 이때, 구매자가 업로드한 파일 정보를 각 가맹점이 내부적으로 저장한 후 그 정보에 대한 KEY 값을 네이버페이 주문 정보에 추가하면, 구매자의 결제 이후 각 주문에 대해 출력된 KEY 값을 이용해 자사 어드민에 저장된 DB와 매핑해서 주문 처리에 사용할 수 있다.

CPA_INFLOW_CODE는 네이버쇼핑 가맹점이 사용하는 항목이다. 네이버쇼핑 가맹점 중 파라미터 방식으로 CPA 트래킹을 하는 가맹점은 반드시 이 값을 입력해야 하며, 쿠키에서 CPAValidator 키에 해당하는 값을 입력한다. 자세한 내용은 네이버쇼핑에서 제공하는 "CPA 연동 가이드"를 참고한다.

SA_CLICK_ID는 네이버 검색광고 광고주가 사용하는 항목이다. 광고주 센터의 광고 효과 보고서를 통해 네이버페이 전환 데이터를 확인하길 원하는 가맹점은 반드시 이 값을 입력해야 하며, 자세한 내용은 네이버 검색광고에서 제공하는 "CTS 설치 가이드"를 참고한다. 단, 광고 효과 보고서 제공 시점은 네이버 검색광고의 정책을 따른다.

NAVER_INFLOW_CODE는 구매자가 네이버 서비스를 통해 상품 페이지에 유입된 경우에 유입 전의 네이버 서비스를 식별하여 유입 경로별 혜택을 적용하기 위한 항목이다. 쿠키에서 NA_CO 키에 해당하는 값을 입력한다.

다음은 PHP로 주문 정보를 등록하는 예이다.

```
<?
//item data를 생성한다.
class ItemStack {
    var $id;
    var $name;
    var $tprice;
    var $uprice;
    var $option;
    var $count;

    //option이 여러 종류라면, 선택된 옵션을 슬래시(/)로 구분해서 표시하는 것을 권장한다.
    function ItemStack($id, $_name, $_tprice, $_uprice, $_option, $_count) {
        $this->id = $_id;
        $this->name = $_name;
        $this->tprice = $_tprice;
        $this->uprice = $_uprice;
        $this->option = $_option;
        $this->count = $_count;
    }

    function makeQueryString() {
        $ret .= 'ITEM_ID=' . urlencode($this->id);
        $ret .= '&ITEM_NAME=' . urlencode($this->name);
        $ret .= '&ITEM_COUNT=' . $this->count;
        $ret .= '&ITEM_OPTION=' . urlencode($this->option);
        $ret .= '&ITEM_TPRICE=' . $this->tprice;
        $ret .= '&ITEM_UPRICE=' . $this->uprice;
        return $ret;
    }
};

$shopId = 'naver_pay';
$certiKey = 'naver_pay';
$shippingType = 'PAYED';

if ($shippingType == 'PAYED') {
    $shippingPrice = 2500;
} else {
    $shippingPrice = 0;
}

$backUrl = "http://www.naver.com";

$queryString = 'SHOP_ID=' . urlencode($shopId);
$queryString .= '&CERTI_KEY=' . urlencode($certiKey);
$queryString .= '&SHIPPING_TYPE=' . $shippingType;
$queryString .= '&SHIPPING_PRICE=' . $shippingPrice;
$queryString .= '&RESERVE1=&RESERVE2=&RESERVE3=&RESERVE4=&RESERVE5=';
$queryString .= '&BACK_URL=' . $backUrl;
$queryString .= '&SA_CLICK_ID=' . $_COOKIE["NVADID"]; //CTS
// CPA 스크립트 가이드 설치 업체는 해당 값 전달
$queryString .= '&CPA_INFLOW_CODE=' . urlencode($_COOKIE["CPAValidator"]);
$queryString .= '&NAVER_INFLOW_CODE=' . urlencode($_COOKIE["NA_CO"]);

$totalMoney = 0;
//DB와 장바구니에서 상품 정보를 얻어 온다.
//while(...) {
    $id = "item1";
    $name = "상품1";
    $uprice = 1000;
    $count = 1;
```

```

    $tprice = $uprice * $count;
    $option = "";
    $item = new ItemStack($id, $name, $tprice, $uprice, $option, $count);
    $totalMoney += $tprice;
    $queryString .= '&'.$item->makeQueryString();
}

$totalPrice = (int)$totalMoney + (int)$shippingPrice;
$queryString .= '&TOTAL_PRICE='.$totalPrice;

echo($queryString."<br>\n");

$req_addr = 'ssl://test-pay.naver.com';
$req_url = 'POST /customer/api/order.nhn HTTP/1.1'; // utf-8
// $req_url = 'POST /customer/api/CP949/order.nhn HTTP/1.1'; // euc-kr
$req_host = 'test-pay.naver.com';
$req_port = 443;
$nc_sock = @fsockopen($req_addr, $req_port, $errno, $errstr);
if ($nc_sock) {
    fwrite($nc_sock, $req_url."\r\n" );
    fwrite($nc_sock, "Host: ".$req_host.":".$req_port."\r\n" );
    fwrite($nc_sock, "Content-type: application/x-www-form-urlencoded; charset=utf-8\r\n");
    //fwrite($nc_sock, "Content-type: application/x-www-form-urlencoded; charset=CP949\r\n");
    fwrite($nc_sock, "Content-length: ".strlen($queryString)."\r\n");
    fwrite($nc_sock, "Accept: */*\r\n");
    fwrite($nc_sock, "\r\n");
    fwrite($nc_sock, $queryString."\r\n");
    fwrite($nc_sock, "\r\n");

    // get header
    while(!feof($nc_sock)){
        $header=fgets($nc_sock,4096);
        if($header=="\r\n"){
            break;
        } else {
            $headers .= $header;
        }
    }

    // get body
    while(!feof($nc_sock)){
        $bodys.=fgets($nc_sock,4096);
    }

    fclose($nc_sock);

    $resultCode = substr($headers,9,3);

    if ($resultCode == 200) {
        // success
        $orderId = $bodys;
    } else {
        // fail
        echo $bodys;
    }
}
else {
    echo "$errstr ($errno)<br>\n";
    exit(-1);
    //에러처리
}

//리턴받은 order_id로 주문서 page를 호출한다.
echo ($orderId."<br>\n");

$orderUrl = "https://test-pay.naver.com/customer/order.nhn";

?>
<html>
<body>

```

```

<form name="frm" method="get" action="<?=$orderId?>">
<input type="hidden" name="ORDER_ID" value="<?=$orderId?>">
<input type="hidden" name="SHOP_ID" value="<?=$shopId?>">
<input type="hidden" name="TOTAL_PRICE" value="<?=$totalPrice?>">
</form>
</body>
<script>

<? if ($resultCode == 200) { ?>
document.frm.target = "_top";
document.frm.submit();
<? } ?>
</script>
</html>

```

다음은 Java로 주문 정보를 등록하는 예이다.

```

package test.com.nhncorp.bp.external.mall;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.io.Writer;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.List;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.Cookie;

public class JavaOrderSample
{
    private static final String ENCODING = "UTF-8";
    private static final String NCKEY_ITEMID = "ITEM_ID";
    private static final String NCKEY_ITEMNAME = "ITEM_NAME";
    private static final String NCKEY_COUNT = "ITEM_COUNT";
    private static final String NCKEY_TPRICE = "ITEM_TPRICE";
    private static final String NCKEY_UPRICE = "ITEM_UPRICE";
    private static final String NCKEY_OPTION = "ITEM_OPTION";
    private static final String NCKEY_SHIPPINGTYPE = "SHIPPING_TYPE";
    private static final String NCKEY_SHIPPINGPRICE = "SHIPPING_PRICE";
    private static final String NCKEY_TOTALPRICE = "TOTAL_PRICE";
    private static final String NCKEY_SHOPID = "SHOP_ID";
    private static final String NCKEY_CERTIKEY = "CERTI_KEY";
    private static final String NCKEY_BACKURL = "BACK_URL";
    private static final String NCKEY_RESERVE1 = "RESERVE1";
    private static final String NCKEY_RESERVE2 = "RESERVE2";
    private static final String NCKEY_RESERVE3 = "RESERVE3";
    private static final String NCKEY_RESERVE4 = "RESERVE4";
    private static final String NCKEY_RESERVE5 = "RESERVE5";
    private static final String NCKEY_SA_CLICK_ID = "SA_CLICK_ID"; //CTS
    private static final String NCKEY_CPA_INFLOW_CODE = "CPA_INFLOW_CODE";
    private static final String NCKEY_NAVER_INFLOW_CODE = "NAVER_INFLOW_CODE";

    public URL _url;
    private SSLSocketFactory _sslSockFactory;

    public JavaOrderSample()

```

```

{
    _url = null;
}

public JavaOrderSample(URL url)
{
    _url = url;
    _initHttps();
}

public JavaOrderSample(String url) throws MalformedURLException
{
    this(new URL(url));
}

public void setUrl(String url) throws MalformedURLException
{
    _url = new URL(url);
}

public static class ItemStack
{
    private String _itemId;
    private String _itemName;
    private int _itemTPrice;
    private int _itemUPrice;
    private String _selectedOption;
    private int _count;

    /**
     * @param itemId Mall Item Code
     * @param itemName 상품
     * @param itemPrice 상품 개별 가격
     * @param selectedOption 선택된 옵션. - 여러 옵션이 선택되었을 경우 '/'로 구분하는 것을 권장
     * @param count 해당 상품 구매 갯수.
     */
    public ItemStack(String itemId, String itemName, int itemTPrice, int itemUPrice,
String selectedOption, int count)
    {
        _itemId = itemId;
        _itemName = itemName;
        _itemTPrice = itemTPrice;
        _itemUPrice = itemUPrice;
        _selectedOption = selectedOption;
        _count = count;
    }

    public String getItemId()
    {
        return _itemId;
    }

    public String getItemName()
    {
        return _itemName;
    }

    public int getItemTotalPrice()
    {
        return _itemTPrice;
    }

    public int getItemUnitPrice()
    {
        return _itemUPrice;
    }

    public String getSelectedOption()
    {
        if (_selectedOption == null) return "";
        return _selectedOption;
    }
}

```



```

    }

    public int getCount()
    {
        return _count;
    }
}

private void _urlEncode(StringBuffer sb, String key, String value)
{
    try {
        sb.append(URLEncoder.encode(key, ENCODING));
        sb.append('=');
        sb.append(URLEncoder.encode(value, ENCODING));
    }
    catch(UnsupportedEncodingException e) {
        // 일어나지 않음
        throw new Error(e);
    }
}

private String _makeQueryString(String shopId, String certificationKey, ItemStack[]
items, int shippingPrice, String shippingType, String backURL, String saClickId, String
cpaInflowCode, String naverInflowCode)
{
    // 주문 금액 = 각 상품 금액 + 배송비(단 선불일 경우)
    int totalPrice = shippingPrice>0?shippingPrice:0;
    StringBuffer sb = new StringBuffer();
    _urlEncode(sb, NCKEY_SHOPID, shopId);
    sb.append('&');
    _urlEncode(sb, NCKEY_CERTIKEY, certificationKey);
    sb.append('&');
    for (ItemStack is : items) {
        totalPrice += is.getItemTotalPrice();
        _urlEncode(sb, NCKEY_ITEMID, is.getItemId());
        sb.append('&');
        _urlEncode(sb, NCKEY_ITEMNAME, is.getItemName());
        sb.append('&');
        _urlEncode(sb, NCKEY_TPRICE, String.valueOf(is.getItemTotalPrice()));
        sb.append('&');
        _urlEncode(sb, NCKEY_UPRICE, String.valueOf(is.getItemUnitPrice()));
        sb.append('&');
        _urlEncode(sb, NCKEY_COUNT, String.valueOf(is.getCount()));
        sb.append('&');
        _urlEncode(sb, NCKEY_OPTION, is.getSelectedOption());
        sb.append('&');
    }

    _urlEncode(sb, NCKEY_SHIPPINGTYPE, shippingType);
    sb.append('&');
    _urlEncode(sb, NCKEY_SHIPPINGPRICE, String.valueOf(shippingPrice));
    sb.append('&');
    _urlEncode(sb, NCKEY_TOTALPRICE, String.valueOf(totalPrice));
    sb.append('&');
    _urlEncode(sb, NCKEY_BACKURL, backURL);
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE1, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE2, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE3, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE4, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE5, "");
    sb.append('&'); //CTS
    _urlEncode(sb, NCKEY_SA_CLICK_ID, saClickId); //CTS
    // CPA 스크립트 가이드 설치업체는 해당 값 전달
    sb.append('&');
    _urlEncode(sb, NCKEY_CPA_INFLOW_CODE, cpaInflowCode);
    sb.append('&');
}

```

```

        _urlEncode(sb, NCKEY_NAVER_INFLOW_CODE, naverInflowCode);
        System.out.println(sb.toString());

        return sb.toString();
    }

    /* test 환경에서는 인증서 오류가 날 수도 있다. 이 코드를 이용해 인증서 오류를 회피한다. */
    private void _initHttps()
    {
        TrustManager[] trustAllCerts = new TrustManager[] {
            new X509TrustManager() {
                public void checkClientTrusted(X509Certificate[] chain, String authType)
                throws CertificateException {}
                public void checkServerTrusted(X509Certificate[] chain, String authType)
                throws CertificateException {}
                public X509Certificate[] getAcceptedIssuers()
                {
                    return new X509Certificate[0];
                }
            }
        };
        try {
            SSLContext sslContext = SSLContext.getInstance("SSL");
            sslContext.init(null, trustAllCerts, new SecureRandom());
            _sslSockFactory = sslContext.getSocketFactory();
        }
        catch (Exception e) {
            RuntimeException re = new RuntimeException(e);
            re.setStackTrace(e.getStackTrace());
            throw re;
        }
    }

    /**
     * @param items 주문 상품 목록.
     * @param shippingPrice 배송비.
     * @param shippingType 배송비결제 구분. "FREE": 무료. "PAYED": 선불. "ONDELIVERY": 착불
     * @return 주문키
     * @throws IOException
     */
    public String sendOrderInfoToNC(String shopId, String certificationKey, ItemStack[]
    items, int shippingPrice, String shippingType, String backURL, String nvadId) throws
    IOException
    {
        HttpURLConnection conn = (HttpURLConnection)_url.openConnection();

        /* test 환경에서는 인증서 오류가 날 수도 있다. 이 코드를 이용해 인증서 오류를 회피한다. */
        if (conn instanceof HTTPSURLConnection) {
            ((HTTPSURLConnection)conn).setSSLSocketFactory(_sslSockFactory);
            ((HTTPSURLConnection)conn).setHostnameVerifier(
                new HostnameVerifier() {
                    @Override
                    public boolean verify(String hostname, SSLSession session)
                    {
                        return true;
                    }
                }
            );
        }
        conn.setDoInput(true);
        conn.setDoOutput(true);
        conn.setUseCaches(false);
        conn.setRequestMethod("POST");
        conn.addRequestProperty("Content-Type", "application/x-www-form-urlencoded;
        charset=UTF-8");

        Writer writer = new OutputStreamWriter(conn.getOutputStream(), ENCODING);
        writer.write(_makeQueryString(shopId, certificationKey, items, shippingPrice,
        shippingType, backURL, nvadId));
        writer.flush();
    }

```

```

        writer.close();

        int respCode = conn.getResponseCode();
        if (respCode != 200) {
            throw new RuntimeException(String.format("NC Response fail : %d %s", respCode,
conn.getResponseMessage()));
        }

        BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        String orderKey = reader.readLine();
        return orderKey;
    }

    private String getCookieValue(HttpServletRequest request, String name) {
        if (name == null || request == null) {
            return "";
        }

        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
            for (int i = 0; i < cookies.length; i++) {
                if (name.equals(cookies[i].getName())) {
                    return cookies[i].getValue();
                }
            }
        }
        return "";
    }

    public static void main(String[] args) throws IOException
    {
        //주문 상품 내역으로 items 데이터를 생성한다.
        List<ItemStack> items = new ArrayList<ItemStack>();
        items.add(new ItemStack("a1", "아이템1", 2000, 1000, "", 3));
        items.add(new ItemStack("a2", "아이템2", 2000, 1000, "XL/빨강", 2));

        int shippingPrice = 2500;
        String shippingType = "PAYED";
        String backURL = "http://www.naver.com";
        String nvadId="11111+aa12345678901234"; //CTS
        //servlet인 경우 쿠키값을 넣어야 함
        String nvadId = getCookieValue((HttpServletRequest)request, "NVADID");
        // CPA 스크립트 가이드 설치 업체는 해당 값 전달
        String cpaInflowCode = getCookieValue((HttpServletRequest)request, "CPAValidator");
        String naverInflowCode = getCookieValue((HttpServletRequest)request, "NA_CO");
        JavaOrderSample sample = new JavaOrderSample("https://test-
pay.naver.com/customer/api/order.nhn");
        String orderKey = sample.sendOrderInfoToNC("naver_pay", "naver_pay",
items.toArray(new ItemStack[0]), shippingPrice, shippingType, backURL, nvadId,
cpaInflowCode, naverInflowCode);
        //여기서 얻은 orderKey로 NC 결제창에 넘겨 결제를 진행한다.
        System.out.println(String.format("OrderKey = %s", orderKey));
    }
}

```

다음은 ASP로 주문 정보를 등록하는 예이다.

```

<%
Class ItemStack
    Private mId
    Private mName
    Private mCount
    Private mUnitPrice
    Private mTotalPrice
    Private mItemOption

    Public Property Get Id()
        Id = mId
    End Property

```

```

    Public Property Let Id(val)
        mId = val
    End Property

    Public Property Get Name()
        Name = mName
    End Property

    Public Property Let Name(val)
        mName = val
    End Property

    Public Property Get Count()
        Count = mCount
    End Property

    Public Property Let Count(val)
        mCount = val
    End Property

    Public Property Get UnitPrice()
        UnitPrice = mUnitPrice
    End Property

    Public Property Let UnitPrice(val)
        mUnitPrice = val
    End Property

    Public Property Get TotalPrice()
        TotalPrice = mTotalPrice
    End Property

    Public Property Let TotalPrice(val)
        mTotalPrice = val
    End Property

    Public Property Get ItemOption()
        ItemOption = mItemOption
    End Property

    Public Property Let ItemOption(val)
        mItemOption = val
    End Property

    Public Property Get QueryString()
        QueryString = "ITEM_ID=" + Server.URLEncode(mId)
        QueryString = QueryString + "&ITEM_NAME=" + Server.URLEncode(mName)
        QueryString = QueryString + "&ITEM_COUNT=" + Server.URLEncode(mCount)
        QueryString = QueryString + "&ITEM_UPRICE=" + Server.URLEncode(mUnitPrice)
        QueryString = QueryString + "&ITEM_TPRICE=" + Server.URLEncode(mTotalPrice)
        QueryString = QueryString + "&ITEM_OPTION=" + Server.URLEncode(mItemOption)
    End Property
End Class

Class Order
    Private mShopId
    Private mCertiKey
    Private mShippingPrice
    Private mShippingType
    Private mTotalPrice
    Private mBackURL
    Private mSaClickId //CTS
    Private mCpaInflowCode
    Private mNaverInflowCode

    Public Property Get ShopId()
        ShopId = mShopId
    End Property

    Public Property Let ShopId(value)
        mShopId = value
    End Property

```

```

Public Property Get CertiKey()
    CertiKey = mCertiKey
End Property

Public Property Let CertiKey(value)
    mCertiKey = value
End Property

Public Property Get ShippingPrice()
    ShippingPrice = mShippingPrice
End Property

Public Property Let ShippingPrice(value)
    mShippingPrice = value
End Property

Public Property Get ShippingType()
    ShippingType = mShippingType
End Property

Public Property Let ShippingType(value)
    mShippingType = value
End Property

Public Property Get TotalPrice()
    TotalPrice = mTotalPrice
End Property

Public Property Let TotalPrice(value)
    mTotalPrice = value
End Property

Public Property Get BackURL()
    BackURL = mBackURL
End Property

Public Property Let BackURL(value)
    mBackURL = value
End Property

Public Property Get SaClickId() //CTS
    SaClickId = mSaClickId
End Property

Public Property Let SaClickId(value) //CTS
    mSaClickId = value
End Property

Public Property Get CpaInflowCode()
    CpaInflowCode = mCpaInflowCode
End Property

Public Property Let CpaInflowCode(value)
    mCpaInflowCode = value
End Property

Public Property Get NaverInflowCode()
    NaverInflowCode = mNaverInflowCode
End Property

Public Property Let NaverInflowCode(value)
    mNaverInflowCode = value
End Property

Public Property Get QueryString()
    QueryString = "SHOP_ID=" + Server.URLEncode(mShopId)
    QueryString = QueryString + "&CERTI_KEY=" + Server.URLEncode(mCertiKey)
    QueryString = QueryString + "&SHIPPING_PRICE=" +
Server.URLEncode(mShippingPrice)
    QueryString = QueryString + "&SHIPPING_TYPE=" +
Server.URLEncode(mShippingType)

```

```

        QueryString = QueryString + "&TOTAL_PRICE=" + Server.URLEncode(mTotalPrice)
        QueryString = QueryString + "&BACK_URL=" + Server.URLEncode(mBackURL)
        QueryString = QueryString + "&RESERVE1="
        QueryString = QueryString + "&RESERVE2="
        QueryString = QueryString + "&RESERVE3="
        QueryString = QueryString + "&RESERVE4="
        QueryString = QueryString + "&RESERVE5="
        QueryString = QueryString + "&SA_CLICK_ID=" + mSaClickId //CTS
        // CPA 스크립트 가이드 설치업체는 해당 값 전달
        QueryString = QueryString + "&CPA_INFLOW_CODE=" +
Server.URLEncode(mCpaInflowCode)
        QueryString = QueryString + "&NAVER_INFLOW_CODE=" +
Server.URLEncode(mNaverInflowCode)
    End Property

End Class

Dim myOrder, items(0), queryString

' loop
Set items(0) = new ItemStack
items(0).Id = "items2"
items(0).Name = "아이템2"
items(0).Count = 1
items(0).UnitPrice = "100"
items(0).TotalPrice = "100"
items(0).ItemOption = ""
' end loop

Set myOrder = new Order
myOrder.ShopId = "naver_pay"
myOrder.CertiKey = "naver_pay"
myOrder.ShippingPrice = 1500
myOrder.ShippingType = "PAYED"
myOrder.TotalPrice = 1600
myOrder.BackURL = "http://www.naver.com"
myOrder.SaClickId = Request.Cookies("NVADID") //CTS
// CPA 스크립트 가이드 설치 업체는 해당 값 전달
myOrder.CpaInflowCode = Request.Cookies("CPAValidator")
myOrder.NaverInflowCode = Request.Cookies("NA_CO")

queryString = myOrder.QueryString
For Each item In items
    queryString = queryString + "&" + item.QueryString
Next

Response.Write(queryString+"<br>"&chr(10))

Dim objXMLHTTP, orderId
Set objXMLHTTP = server.CreateObject("MSXML2.ServerXMLHTTP")
objXMLHTTP.setOption 2,13056 'ignore SSL errors
objXMLHTTP.Open "POST", "https://test-pay.naver.com/customer/api/order.nhn", false
'objXMLHTTP.Open "POST", "https://test-pay.naver.com/customer/api/CP949/order.nhn", false
objXMLHTTP.setRequestHeader "Content-Type", "application/x-www-form-
urlencoded; charset=UTF-8"
'objXMLHTTP.setRequestHeader "Content-Type", "application/x-www-form-urlencoded; charset=
CP949 "

objXMLHTTP.Send queryString

If objXMLHTTP.status = 200 Then
    orderId = objXMLHTTP.responseText
Else
    '에러처리
End if

Response.Write(orderId)

orderUrl = "https://test-pay.naver.com/customer/order.nhn"
%>

```

```

<html>
<body>
<form name="frm" method="get" action="<%=orderId%>">
<input type="hidden" name="ORDER_ID" value="<%=orderId%>">
<input type="hidden" name="SHOP_ID" value="<%=myOrder.ShopId%>">
<input type="hidden" name="TOTAL_PRICE" value="<%=myOrder.TotalPrice%>">
</form>
</body>
<script>

<% If objXMLHTTP.status = 200 Then %>
document.frm.target = "_top";
document.frm.submit();
<% End if

Set objXMLHTTP = Nothing
%>
</script>
</html>

```

3.1.3 주문서 전송

가맹점은 네이버페이 주문서 팝업 창을 생성하여 GET 인자로 아래의 정보를 전달한다.

표 3-3 주문서 팝업 창 URL

환경	URL
테스트	https://test-pay.naver.com/customer/order.nhn
서비스	https://pay.naver.com/customer/order.nhn

표 3-4 주문서 내용

항목	필수 여부	설명
ORDER_ID	Y	주문 ID. 주문 정보 등록 결과 네이버페이에서 받는다.
SHOP_ID	Y	상점 ID. 네이버페이에 가입 승인될 때 정해진다.
TOTAL_PRICE	Y	총 주문 금액

3.1.4 네이버페이 결제

결제 과정은 네이버페이 주문/결제 페이지에서 이루어진다. 이후, 네이버페이에서 상품 정보를 업데이트하기 위해 상품 정보를 요청할 수 있다.

3.2 상품 정보 연동

3.2.1 상품 정보 요청

네이버페이는 비주기적으로 상품 정보를 업데이트하기 위해, 가맹점 사이트에 HTTP로 상품 정보 요청을 보낸다. 가맹점은 요청받은 상품 정보를 XML로 전송하는 페이지를 생성해야 한다.

네이버페이가 상품 정보를 요청하는 형식은 다음과 같다.

```
http://가맹점도메인/가맹점페이지?ITEM_ID=XXX&ITEM_ID=XXX&ITEM_ID=XXX
```

표 3-5 상품 정보 요청 내용

항목	필수 여부	설명
ITEM_ID	Y	주문 정보 등록 시 네이버에 등록한 ITEM_ID. ITEM_ID는 여러 개일 수 있다.

3.2.2 상품 정보 제공

가맹점은 XML로 response에 각각의 ITEM_ID에 대해 다음의 상품 정보를 전송한다.

필수 항목은 모두 element로 명시해야 하며, URL 값은 반드시 유효해야 한다.

표 3-6 상품 정보 제공 내용

항목	필수 여부	설명
item@id	Y	상품 ID
item/mall_pid	N	네이버쇼핑 EP의 ma_pid. 네이버쇼핑 가맹점이면 네이버쇼핑 EP의 ma_pid와 동일한 값을 입력해야 한다.
item/name	Y	상품 이름
item/url	Y	상품 정보 URL
item/description	Y	상품 설명
item/image	Y	상품 사진 URL
item/thumb	Y	상품 썸네일 URL
item/price	Y	상품의 정상 가격
item/quantity	Y	상품의 재고량
item/options	N	선택한 옵션 사항
item/category	Y	가맹점 사이트에서 상품의 카테고리

항목	필수 여부	설명
item/returnInfo	N	상품별 반품 주소. 상품별로 반품 주소가 필요한 경우에 사용한다. 이 값을 지정하지 않으면 가맹점 주소가 사용되며, 가맹점 주소가 없으면 가맹점 대표의 주소가 사용된다.

문자 세트는 각 환경에 맞게 설정하고, content-type은 application/xml로 한다.

```
<?xml version="1.0" encoding="UTF-8">
<response>
  <item id="1"> <!-- 상품 ID -->
    <name><![CDATA[상품 이름]]></name>
    <url>http://www.mymall.com/show_item?itemId=itemId</url> <!-- 상품 설명 페이지 url -->
    <description><![CDATA[상품 설명]]></description>
    <image>http://www.mymall.com/images/itemId.jpg</image> <!-- 상품 image url -->
    <thumb>http://www.mymall.com/images/itemId.jpg</thumb> <!-- 상품 썸네일 image url -->
    <price>10000</price> <!-- 정상 가격 -->
    <quantity>10</quantity> <!-- 재고량 -->
    <options> <!-- 상품의 옵션이 없으면 이 내용은 없어도 된다. -->
      <option name="색상"> <!-- 옵션 종류 이름 -->
        <select><![CDATA[빨강]]></select> <!-- 해당 옵션의 선택 사항 -->
        <select><![CDATA[파랑]]></select>
        <select><![CDATA[노랑]]></select>
      </option>
      <option name="크기">
        <select><![CDATA[L]]></select>
        <select><![CDATA[XL]]></select>
      </option>
    </options>
    <category> <!-- 카테고리 -->
      <first>...</first> <!-- 최상위 카테고리 이름 -->
      <second>...</second> <!-- 차상위 카테고리 이름 -->
      <!-- 최대 4단계 카테고리까지 표시 가능하다. (first ~ fourth) -->
    </category>
    <returnInfo> <!-- 상품별 반품 주소 -->
      <zipcode>13591</zipcode><!-- 우편번호 -->
      <address1>경기도 성남시 분당구 서현동</address1> <!-- 기본 주소. 동(읍/면/리)까지 입력 -->
      <address2>266-1 분당퍼스트타워 13층</address2> <!-- 상세 주소. 번지 및 아파트 동호수까지 입력 -->
      <sellername>NBP판매자</sellername> <!-- 수령인 이름 -->
      <contact1>1566-3388</contact1> <!-- 연락처1 -->
      <contact2></contact2> <!-- 연락처2. 이 값은 생략할 수 있다. -->
    </returnInfo>
  </item>
  <!-- 여러 개의 상품 정보를 요청하면 그 개수만큼의 item element가 필요하다. -->
</response>
```

다음은 PHP로 상품 정보를 전송하는 예이다.

```
<?
$query = $_SERVER['QUERY_STRING'];
$vars = array();
foreach(explode('&', $query) as $pair) {
    list($key, $value) = explode('=', $pair);
```

```

        $key = urldecode($key);
        $value = urldecode($value);
        $vars[$key][] = $value;
    }

    $itemIds = $vars['ITEM_ID'];

    if (count($itemIds) < 1) {
        exit('ITEM_ID 는 필수입니다.');
```

```

    }

    header('Content-Type: application/xml;charset=euc-kr');
    echo ('<?xml version="1.0" encoding="euc-kr"?>');
    ?>

    <response>
    <?

    //while(...):
        $id = "itemid";
        $name = "아이템명";
        $description = "간지나는아이템";
        $price = 1000;
        $quantity = 1;
    ?>

    <item id="<?=$id?>">
        <name><![CDATA[<?=$name?>]]></name>
        <url>http://<?=$_SERVER["HTTP_HOST"]?>/test.html</url>
        <description><![CDATA[<?=$description?>]]></description>
        <image>http://<?=$_SERVER["HTTP_HOST"]?>/test.jpg</image>
        <thumb>http://<?=$_SERVER["HTTP_HOST"]?>/test.jpg</thumb>
        <price><?=$price?></price>
        <quantity><?=$quantity?></quantity>
        <category>
            <first id="MJ01">대분류</first>
            <second id="ML01">중분류</second>
            <third id="MN01">소분류</third>
        </category>

        <options>
            <option name="크기">
                <select> <![CDATA[ X ]]> </select>
                <select> <![CDATA[ XL ]]> </select>
            </option>
        </options>
    </item>
    <?
    //end while;

    echo('</response>');
    ?>

```

다음은 Java로 상품 정보를 전송하는 예이다.

```

package test.com.nhncorp.bp.external.mall;

import java.io.IOException;
import java.io.Writer;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@SuppressWarnings("serial")
public class JavaItemInfoServlet extends HttpServlet
{
    public void _writeItemInfo(String itemId, Writer writer) throws IOException
    {

```

```

//itemId로 DB에서 정보 조회
String description = "";
String url = "http://www.mymall.com/show_item?itemId=" + itemId;
String itemName = "아이템이름";
String itemImage = "http://www.mymall.com/images/" + itemId + ".jpg";
String thumbImage = "http://www.mymall.com/thumbs/" + itemId + ".jpg";
String[] options = new String[2];
options[0] = "색상";
options[1] = "크기";
String[][] optionList = new String[2][];
optionList[0] = new String[3];
optionList[0][0] = "빨강";
optionList[0][1] = "파랑";
optionList[0][2] = "노랑";
optionList[1] = new String[2];
optionList[1][0] = "L";
optionList[1][1] = "XL";

int price = 10000;
int quantity = 10;

String majorCategoryId = "MJ01";
String majorCategoryName = "대분류";

String middleCategoryId = "ML01";
String middleCategoryName = "중분류";

String minorCategoryId = "MN01";
String minorCategoryName = "소분류";
// 여기까지 DB에서 조회한 정보.

// ! 각 String에 xml의 attribute나 body 혹은 CDATA block안에 들어갈 수 없는 문자가 있다면
적절하게 escaping 해야 한다.

// Write구조 대신 javax.xml.transform.TransformerFactory 과 org.w3c.dom.Document 를
이용해 DOM구조를 XML로 출력하는 방법을 사용할 수도 있다.
writer.write(String.format("<item id=\"%s\">\r\n", itemId));
writer.write(String.format("<name><![CDATA[%s]]>\r\n", itemName));
writer.write(String.format("<url><![CDATA[%s]]>\r\n", url));
writer.write(String.format("<description><![CDATA[%s]]>\r\n",
description));
writer.write(String.format("<image><![CDATA[%s]]>\r\n", itemImage));
writer.write(String.format("<thumb><![CDATA[%s]]>\r\n", thumbImage));

if (options != null) {
    writer.write("<options>\r\n");
    for (int i=0;i<options.length;++i) {
        writer.write(String.format("<option name=\"%s\">\r\n", options[i]));
        for (int j=0;j<optionList[i].length;++j) {
            writer.write(String.format("<select><![CDATA[%s]]>\r\n",
optionList[i][j]));
        }
        writer.write("</option>\r\n");
    }
    writer.write("</options>\r\n");
}
writer.write(String.format("<price>%d</price>\r\n", price));
writer.write(String.format("<quantity>%d</quantity>\r\n", quantity));

writer.write("<category>\r\n");
writer.write(String.format("<first id=\"%s\"><![CDATA[%s]]>\r\n",
majorCategoryId, majorCategoryName));
writer.write(String.format("<second id=\"%s\"><![CDATA[%s]]>\r\n",
middleCategoryId, middleCategoryName));
writer.write(String.format("<third id=\"%s\"><![CDATA[%s]]>\r\n",
minorCategoryId, minorCategoryName));

```

```

        writer.write("</category>\r\n");
        writer.write("</item>\r\n");
    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, java.io.IOException
    {
        String[] itemIds = req.getParameterValues("ITEM_ID");
        resp.setContentType("application/xml; charset=UTF-8");
        Writer writer = resp.getWriter();
        writer.write("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n");
        writer.write("<response>\r\n");
        for (String itemId : itemIds) {
            _writeItemInfo(itemId, writer);
        }
        writer.write("</response>");
        writer.flush();
        writer.close();
    }

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, java.io.IOException
    {
        doPost(req, resp);
    }
}

```

다음은 ASP로 상품 정보를 전송하는 예이다.

```

<% Response.ContentType = "text/xml; charset=euc-kr" %><?xml version="1.0" encoding="euc-kr"?>
<response>

<%

Dim itemId, url, name, description, image, thumb, options(), price, quantity

for i=1 to Request.QueryString("ITEM_ID").Count

    itemId = Request.QueryString("ITEM_ID")(i)
    url = "http://www.mymall.com/show_item?itemid=" + itemId
    name = "아이템이름"
    description = ""
    image = "http://www.mymall.com/images/" + itemId + ".jpg"
    thumb = "http://www.mymall.com/thumbs/" + itemId + ".jpg"
    price = 10000
    quantity = 10
    majorId = "MJ01"
    majorName = "대분류"
    middleId = "ML01"
    middleName = "중분류"
    minorId = "MN01"
    minorName = "소분류"

%>
<item id="<%=itemId%>">
    <name> <![CDATA[ <%=name%> ]]> </name>
    <url><%=url%></url>
    <description> <![CDATA[ <%=description%> ]]> </description>
    <image><%=image%></image>
    <thumb><%=thumb%></thumb>
    <price><%=price%></price>
    <quantity><%=quantity%></quantity>
    <category>
        <first id="<%=majorId%>"><![CDATA[ <%=majorName%> ]]> </first>
        <second id="<%=middleId%>"><![CDATA[ <%=middleName%> ]]> </second>
        <third id="<%=minorId%>"><![CDATA[ <%=minorName%> ]]> </third>
    </category>
    <options>

```

```
        <option name="크기">
            <select> <![CDATA[ X ]]> </select>
            <select> <![CDATA[ XL ]]> </select>
        </option>
    </options>

</item>
<%
    next
%>
</response>
```

3.3 찜 정보 연동

3.3.1 네이버페이 [찜하기] 버튼 클릭

이용자가 상품 정보 페이지에서 네이버페이 [찜하기] 버튼을 클릭하면, 가맹점은 팝업 창을 생성하여 찜 정보 등록을 위한 페이지로 이동한다.

3.3.2 찜 정보 등록

가맹점은 네이버페이에 HTTPS로 찜한 상품 정보를 등록하고 네이버페이 상품 ID를 받는다.

찜 정보 등록 시 기본 문자 세트는 UTF-8이고, 다른 문자셋을 사용하는 경우 UTF-8로 변환하거나, 다른 URL을 이용한다.

표 3-7 찜 정보 등록 URL

환경	문자셋	URL
테스트	UTF-8	https://test-pay.naver.com/customer/api/wishlist.nhn
	EUC-KR	https://test-pay.naver.com/customer/api/CP949/wishlist.nhn
서비스	UTF-8	https://pay.naver.com/customer/api/wishlist.nhn
	EUC-KR	https://pay.naver.com/customer/api/CP949/wishlist.nhn

response로 받는 네이버페이 상품 ID는 영문과 숫자로 이루어지며 최대 19자리이다.

request 시 post로 아래와 같은 정보를 전달한다. URL 값은 반드시 유효해야 한다.

표 3-8 찜 정보 등록 내용

항목	필수 여부	설명
SHOP_ID	Y	상점 ID. 네이버페이에 가입 승인될 때 정해진다.
CERTI_KEY	Y	인증키. 네이버페이에 가입 승인될 때 정해진다.
ITEM_ID	Y	상품 ID
EC_MALL_PID	N	네이버쇼핑 EP의 ma_pid. 네이버쇼핑 가맹점이면 네이버쇼핑 EP의 ma_pid와 동일한 값을 입력해야 한다.
ITEM_NAME	Y	상품 이름
ITEM_DESC	N	상품 설명
ITEM_UPRICE	Y	개별 상품 단가. 0보다 커야 한다.
ITEM_IMAGE	Y	상품 사진 URL.
ITEM_THUMB	N	상품 썸네일 URL.

항목	필수 여부	설명
ITEM_URL	Y	상품 정보 URL.

다음은 PHP로 찜 정보를 등록하는 예이다.

```
<?
//item data를 생성한다.
class ItemStack {
    var $id;
    var $name;
    var $uprice;
    var $image;
    var $thumb;
    var $url;

    function ItemStack($_id, $_name, $_uprice, $_image, $_thumb, $_url) {
        $this->id = $_id;
        $this->name = $_name;
        $this->uprice = $_uprice;
        $this->image = $_image;
        $this->thumb = $_thumb;
        $this->url = $_url;
    }

    function makeQueryString() {
        $ret .= 'ITEM_ID=' . urlencode($this->id);
        $ret .= '&ITEM_NAME=' . urlencode($this->name);
        $ret .= '&ITEM_UPRICE=' . $this->uprice;
        $ret .= '&ITEM_IMAGE=' . urlencode($this->image);
        $ret .= '&ITEM_THUMB=' . urlencode($this->thumb);
        $ret .= '&ITEM_URL=' . urlencode($this->url);
        return $ret;
    }
};

$shopId = 'naver_pay';
$certiKey = 'naver_pay';

$queryString = 'SHOP_ID=' . urlencode($shopId);
$queryString .= '&CERTI_KEY=' . urlencode($certiKey);
$queryString .= '&RESERVE1=&RESERVE2=&RESERVE3=&RESERVE4=&RESERVE5=';

//DB 에서 상품 정보를 얻어온다.
//while(...) {
    $uid = "item1";
    $name = "상품1";
    $uprice = 1000;
    $image = "http://mymall.com/image/item1.jpg";
    $thumb = "http://mymall.com/image/item1.jpg";
    $url = "http://mymall.com/item?id=item1";
    $item = new ItemStack($uid, $name, $uprice, $image, $thumb, $url);
    $queryString .= '&'.$item->makeQueryString();
//}

echo($queryString."<br>\n");

$req_addr = 'ssl://test-pay.naver.com';
$req_url = 'POST /customer/api/wishlist.nhn HTTP/1.1'; // utf-8
// $req_url = 'POST /customer/api/CP949/wishlist.nhn HTTP/1.1'; // euc-kr
$req_host = 'test-pay.naver.com';
$req_port = 443;
$nc_sock = @fsockopen($req_addr, $req_port, $errno, $errstr);
if ($nc_sock) {
    fwrite($nc_sock, $req_url."<br>\n");
    fwrite($nc_sock, "Host: ".$req_host."<br>\n");
}
```

```

        fwrite($nc_sock, "Content-type: application/x-www-form-urlencoded; charset=utf-8\r\n"); // utf-8
        //fwrite($nc_sock, "Content-type: application/x-www-form-urlencoded; charset=CP949\r\n"); // euc-kr
        fwrite($nc_sock, "Content-length: ".strlen($queryString)." \r\n");
        fwrite($nc_sock, "Accept: */*\r\n");
        fwrite($nc_sock, "\r\n");
        fwrite($nc_sock, $queryString." \r\n");
        fwrite($nc_sock, "\r\n");

        // get header
        while(!feof($nc_sock)){
            $header=fgets($nc_sock,4096);
            if($header=="\r\n"){
                break;
            } else {
                $headers .= $header;
            }
        }

        // get body
        while(!feof($nc_sock)){
            $bodys.=fgets($nc_sock,4096);
        }

        fclose($nc_sock);

        $resultCode = substr($headers,9,3);

        if ($resultCode == 200) {
            // success
            $itemId = $bodys;
        } else {
            // fail
            echo $bodys;
        }
    }
} else {
    echo "serrstr ($errno)<br>\n";
    exit(-1);
    //에러처리
}

//리턴받은 itemId로 주문서 page를 호출한다.
echo ($itemId."<br>\n");

$wishlistPopupUrl = "https://test-pay.naver.com/customer/wishlistPopup.nhn";

?>
<html>
<body>
<form name="frm" method="get" action="<?=$wishlistPopupUrl?>">
<input type="hidden" name="SHOP_ID" value="<?=$shopId?>">
<input type="hidden" name="ITEM_ID" value="<?=$itemId?>">
</form>
</body>
<script>

<? if ($resultCode == 200) { ?>
document.frm.target = "_top";
document.frm.submit();
<? } ?>
</script>
</html>

```

다음은 Java로 찜 정보를 등록하는 예이다.

```

package test.com.nhncorp.bp.external.mall;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

```



```

import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.io.Writer;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

public class JavaZzimSample
{
    private static final String ENCODING = "UTF-8";
    private static final String NCKEY_ITEMID = "ITEM_ID";
    private static final String NCKEY_ITEMNAME = "ITEM_NAME";
    private static final String NCKEY_UPRICE = "ITEM_UPRICE";
    private static final String NCKEY_ITEMIMAGE = "ITEM_IMAGE";
    private static final String NCKEY_ITEMTHUMB = "ITEM_THUMB";
    private static final String NCKEY_ITEMURL = "ITEM_URL";
    private static final String NCKEY_SHOPID = "SHOP_ID";
    private static final String NCKEY_CERTIKEY = "CERTI_KEY";
    private static final String NCKEY_RESERVE1 = "RESERVE1";
    private static final String NCKEY_RESERVE2 = "RESERVE2";
    private static final String NCKEY_RESERVE3 = "RESERVE3";
    private static final String NCKEY_RESERVE4 = "RESERVE4";
    private static final String NCKEY_RESERVE5 = "RESERVE5";

    public URL _url;
    private SSLSocketFactory _sslSockFactory;

    public JavaZzimSample()
    {
        _url = null;
    }

    public JavaZzimSample(URL url)
    {
        _url = url;
        _initHttps();
    }

    public JavaZzimSample(String url) throws MalformedURLException
    {
        this(new URL(url));
    }

    public void setUrl(String url) throws MalformedURLException
    {
        _url = new URL(url);
    }

    public static class ItemStack
    {
        private String _itemId;
        private String _itemName;
        private int _itemUPrice;
        private String _itemImage;
        private String _itemThumb;
        private String _itemUrl;

        /**

```

```
    * @param itemId Mall Item Code
    * @param itemName 상품
    * @param itemPrice 상품 개별 가격
    */
    public ItemStack(String itemId, String itemName, int itemUPrice, String itemImage,
String itemThumb, String itemUrl)
    {
        _itemId = itemId;
        _itemName = itemName;
        _itemUPrice = itemUPrice;
        _itemImage = itemImage;
        _itemThumb = itemThumb;
        _itemUrl = itemUrl;
    }

    public String getItemId()
    {
        return _itemId;
    }

    public String getItemName()
    {
        return _itemName;
    }

    public int getItemUnitPrice()
    {
        return _itemUPrice;
    }

    public String getItemImage()
    {
        return _itemImage;
    }

    public String getItemThumb()
    {
        return _itemThumb;
    }

    public String getItemUrl()
    {
        return _itemUrl;
    }
}

private void _urlEncode(StringBuffer sb, String key, String value)
{
    try {
        sb.append(URLEncoder.encode(key, ENCODING));
        sb.append('=');
        sb.append(URLEncoder.encode(value, ENCODING));
    }
    catch(UnsupportedEncodingException e) {
        //일어나지 않음
        throw new Error(e);
    }
}

private String _makeQueryString(String shopId, String certificationKey, ItemStack[]
items)
{
    StringBuffer sb = new StringBuffer();
    _urlEncode(sb, NCKEY_SHOPID, shopId);
    sb.append('&');
    _urlEncode(sb, NCKEY_CERTIKEY, certificationKey);
    sb.append('&');
    for (ItemStack is : items) {
        _urlEncode(sb, NCKEY_ITEMID, is.getItemId());
        sb.append('&');
        _urlEncode(sb, NCKEY_ITEMNAME, is.getItemName());
    }
}
```

```

        sb.append('&');
        _urlEncode(sb, NCKEY_UPRICE, String.valueOf(is.getItemUnitPrice()));
        sb.append('&');
        _urlEncode(sb, NCKEY_ITEMIMAGE, String.valueOf(is.getItemImage()));
        sb.append('&');
        _urlEncode(sb, NCKEY_ITEMTHUMB, String.valueOf(is.getItemThumb()));
        sb.append('&');
        _urlEncode(sb, NCKEY_ITEMURL, String.valueOf(is.getItemUrl()));
        sb.append('&');
    }
    _urlEncode(sb, NCKEY_RESERVE1, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE2, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE3, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE4, "");
    sb.append('&');
    _urlEncode(sb, NCKEY_RESERVE5, "");
    System.out.println(sb.toString());

    return sb.toString();
}

/* test 환경에서는 인증서 오류가 날 수도 있다. 이 코드를 이용해 인증서 오류를 회피한다. */
private void _initHttps()
{
    TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            public void checkClientTrusted(X509Certificate[] chain, String authType)
            throws CertificateException {}
            public void checkServerTrusted(X509Certificate[] chain, String authType)
            throws CertificateException {}
            public X509Certificate[] getAcceptedIssuers()
            {
                return new X509Certificate[0];
            }
        }
    };
    try {
        SSLContext sslContext = SSLContext.getInstance("SSL");
        sslContext.init(null, trustAllCerts, new SecureRandom());
        _sslSockFactory = sslContext.getSocketFactory();
    }
    catch(Exception e) {
        RuntimeException re = new RuntimeException(e);
        re.setStackTrace(e.getStackTrace());
        throw re;
    }
}

/**
 * @param items 주문 상품 목록.
 * @return 주문키
 * @throws IOException
 */
public String[] sendZzimToNC(String shopId, String certificationKey, ItemStack[]
items) throws IOException
{
    HttpURLConnection conn = (HttpURLConnection)_url.openConnection();

    /* test 환경에서는 인증서 오류가 날 수도 있다. 이 코드를 이용해 인증서 오류를 회피한다. */
    if (conn instanceof HttpsURLConnection) {
        ((HttpsURLConnection)conn).setSSLSocketFactory(_sslSockFactory);
        ((HttpsURLConnection)conn).setHostnameVerifier(
            new HostnameVerifier() {
                @Override
                public boolean verify(String hostname, SSLSession session)
                {
                    return true;
                }
            }
        );
    }
}

```

```

    }
    );
}
conn.setDoInput(true);
conn.setDoOutput(true);
conn.setUseCaches(false);
conn.setRequestMethod("POST");
conn.addRequestProperty("Content-Type", "application/x-www-form-urlencoded; charset=UTF-8");

Writer writer = new OutputStreamWriter(conn.getOutputStream(), ENCODING);
writer.write(_makeQueryString(shopId, certificationKey, items));
writer.flush();
writer.close();

int respCode = conn.getResponseCode();
if (respCode != 200) {
    throw new RuntimeException(String.format("NC Response fail : %d %s", respCode,
conn.getResponseMessage()));
}

BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
String retStr = reader.readLine();
return retStr.split(",");
}

public static void main(String[] args) throws IOException
{
    //주문상품 내역으로 items 데이터를 생성한다.
    List<ItemStack> items = new ArrayList<ItemStack>();
    items.add(new ItemStack("a1", "아이템1", 1000, "http://mymall.com/images/a1.jpg",
"http://mymall.com/thumbs/a1.jpg", "http://mymall.com/iteminfo?item_id=a1"));
    items.add(new ItemStack("a2", "아이템2", 1000, "http://mymall.com/images/a2.jpg",
"http://mymall.com/thumbs/a2.jpg", "http://mymall.com/iteminfo?item_id=a2"));

    JavaZzimSample sample = new JavaZzimSample("https://test-
pay.naver.com/customer/api/wishlist.nhn");
    String[] prodSeqs = sample.sendZzimToNC("naver_pay", "naver_pay", items.toArray(new
ItemStack[0]));
    //여기서 얻은prodSeqs로 zzim popup을 띄운다.
    System.out.println(Arrays.toString(prodSeqs));
}
}

```

다음은 ASP로 찜 정보를 등록하는 예이다.

```

<%
Class Itemstack
    Private mId
    Private mName
    Private mUnitPrice
    Private mItemImage
    Private mItemThumb
    Private mItemUrl

    Public Property Get Id()
        Id = mId
    End Property

    Public Property Let Id(val)
        mId = val
    End Property

    Public Property Get Name()
        Name = mName
    End Property

    Public Property Let Name(val)
        mName = val
    End Property

```

```

Public Property Get UnitPrice()
    UnitPrice = mUnitPrice
End Property

Public Property Let UnitPrice(val)
    mUnitPrice = val
End Property

Public Property Get ItemImage()
    ItemImage = mItemImage
End Property

Public Property Let ItemImage(val)
    mItemImage = val
End Property

Public Property Get ItemThumb()
    ItemThumb = mItemThumb
End Property

Public Property Let ItemThumb(val)
    mItemThumb = val
End Property

Public Property Get ItemUrl()
    ItemUrl = mItemUrl
End Property

Public Property Let ItemUrl(val)
    mItemUrl = val
End Property

Public Property Get QueryString()
    QueryString = "ITEM_ID=" + Server.URLEncode(mId)
    QueryString = QueryString + "&ITEM_NAME=" + Server.URLEncode(mName)
    QueryString = QueryString + "&ITEM_UPRICE=" + Server.URLEncode(mUnitPrice)
    QueryString = QueryString + "&ITEM_IMAGE=" + Server.URLEncode(mItemImage)
    QueryString = QueryString + "&ITEM_THUMB=" + Server.URLEncode(mItemThumb)
    QueryString = QueryString + "&ITEM_URL=" + Server.URLEncode(mItemUrl)
End Property
End Class

Class Zzim
    Private mShopId
    Private mCertiKey

    Public Property Get ShopId()
        ShopId = mShopId
    End Property

    Public Property Let ShopId(value)
        mShopId = value
    End Property

    Public Property Get CertiKey()
        CertiKey = mCertiKey
    End Property

    Public Property Let CertiKey(value)
        mCertiKey = value
    End Property

    Public Property Get QueryString()
        QueryString = "SHOP_ID=" + Server.URLEncode(mShopId)
        QueryString = QueryString + "&CERTI_KEY=" + Server.URLEncode(mCertiKey)
        QueryString = QueryString + "&RESERVE1="
        QueryString = QueryString + "&RESERVE2="
        QueryString = QueryString + "&RESERVE3="
        QueryString = QueryString + "&RESERVE4="
        QueryString = QueryString + "&RESERVE5="
    End Property

```

```
End Class

Dim myZzim, items(0), queryString
Set myZzim = new Zzim
myZzim.ShopId = "naver_pay"
myZzim.CertiKey = "naver_pay"

Set items(0) = new ItemStack
items(0).Id = "items2"
items(0).Name = "아이템2"
items(0).UnitPrice = "100"
items(0).ItemImage = "http://mymall.com/image/items2.jpg"
items(0).ItemThumb = "http://mymall.com/thumb/items2.jpg"
items(0).ItemUrl = "http://mymall.com/item?itemid=items2"

queryString = myZzim.QueryString
For Each item In items
    queryString = queryString + "&" + item.QueryString
Next

Dim objXMLHTTP, sXML
set objXMLHTTP = server.CreateObject("MSXML2.ServerXMLHTTP")
objXMLHTTP.setOption 2,13056 'ignore SSL errors
objXMLHTTP.Open "POST", "https://test-pay.naver.com/customer/api/wishlist.nhn", false
'objXMLHTTP.Open "POST", "https://test-pay.naver.com/customer/api/CP949/wishlist.nhn",
false
objXMLHTTP.setRequestHeader "Content-Type", "application/x-www-form-
urlencoded;charset=utf-8"
'objXMLHTTP.setRequestHeader "Content-Type", "application/x-www-form-
urlencoded;charset=CP949"
objXMLHTTP.Send queryString

sXML = objXMLHTTP.responseText

If objXMLHTTP.status = 200 Then
    itemId = objXMLHTTP.responseText
Else
    '에러처리
End if

Response.Write(itemId)

wishlistPopupUrl = "https://test-pay.naver.com/customer/wishlistPopup.nhn"
%>
<html>
<body>
<form name="frm" method="get" action="<%=wishlistPopupUrl%>">
<input type="hidden" name="SHOP_ID" value="<%=myZzim.ShopId%>">
<input type="hidden" name="ITEM_ID" value="<%=itemId%>">
</form>
</body>
<script>

<% If objXMLHTTP.status = 200 Then %>
document.frm.target = "_top";
document.frm.submit();
<% End if

Set objXMLHTTP = Nothing
%>
</script>
</html>
```

3.3.3 찜 목록 전송

네이버페이 찜 목록 전송 팝업 창을 생성하여 GET 인자로 아래의 정보를 전송한다.

표 3-9 찜 목록 전송 팝업 창 URL

환경	URL
테스트	https://test-pay.naver.com/customer/wishlistPopup.nhn
테스트(모바일)	https://test-m.pay.naver.com/customer/wishList.nhn
서비스	https://pay.naver.com/customer/wishlistPopup.nhn
서비스(모바일)	https://m.pay.naver.com/customer/wishlistList.nhn

표 3-10 찜 목록 전송 내용

항목	필수 여부	설명
SHOP_ID	Y	상점 ID. 네이버페이에 가입 승인될 때 정해진다.
ITEM_ID	Y	네이버페이 상품 ID. 찜 정보를 등록하고 response 로 받는다.

4. 유입 경로 스크립트 적용

이 장에서는 네이버 서비스 유입 경로 코드를 얻는 추적 스크립트를 삽입하고 코드값을 얻는 방법을 설명한다.

4.1 네이버 공통 유입 경로 스크립트

네이버 공통 유입 경로 스크립트는 네이버 네이버쇼핑, 네이버페이 포인트, 네이버페이 서비스에서 공통으로 사용하는 스크립트이다. 따라서 한 번 설치하면 다른 서비스에 추가로 가입할 때 다시 설치하지 않아도 된다. 네이버 공통 유입 경로 스크립트는 가맹점에 메일로 전달된다.

4.2 유입 경로 스크립트 설정

가맹점은 유입 경로 스크립트에 Account ID(네이버 공통 인증키)를 설정해야 한다. Account ID는 네이버 공통 유입 경로 스크립트를 적용한 사이트를 식별할 수 있도록 네이버에서 가맹점에 발급하는 ID로, 쇼핑몰 ID와는 다르다. 네이버 공통 유입 경로 스크립트를 적용한 사이트는 네이버 네이버쇼핑, 네이버페이 포인트, 네이버페이 서비스 모두에 동일한 ID를 사용한다.

참고

Account ID가 없는 경우에도 wcs.inflow 함수는 정상적으로 호출되어야 한다. 자세한 내용은 4.3.2 유입 경로 추적 함수를 참고한다.

Account ID는 다음과 같은 페이지에 접속하여 확인할 수 있다.

- 가맹점 센터(네이버페이 센터) > 내 정보 > 가맹점 가입 정보 > 쇼핑몰 정보
- 파트너 어드민 > 가맹점 회원 관리 > 가맹점 정보 관리 > 정보 관리

다음과 같이 유입 경로 스크립트의 wcs_add["wa"] 변수에 발급받은 Account ID를 입력한다.

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

if(!wcs_add) var wcs_add = {};
wcs_add["wa"] = "AccountId";

</script>
```

4.3 가맹점 랜딩 페이지 설정

랜딩 페이지는 상품 상세 페이지, 검색 결과 페이지, 기획전 페이지, 쇼핑몰 메인 페이지 등 쇼핑몰에서 발행하는 모든 페이지를 의미한다. 가맹점은 랜딩 페이지에 로그 수집 공통 함수와 유입 경로 추적 함수를 사용해야 한다.

4.3.1 로그 수집 공통 함수

Description

wcs_do 함수는 로그 수집 서버로 로그를 전송한다.

랜딩 페이지에 wcslog.js 파일을 포함시키고, 페이지 로딩이 끝난 시점에 로그가 전송될 수 있도록 </body>의 바로 앞에 wcs_do 함수를 호출해야 한다.

Syntax

```
wcs_do(oEtc);
```

Parameters

(Object) oEtc

[in] 선택 입력 사항으로, 주문 정보 등 추가 정보가 있을 때 입력한다. key, value의 형태로 입력하며, key에는 String, value에는 String, Number, JSON 형식의 데이터를 입력할 수 있다.

Return

없음

Example Code

```
//wcslog.js 파일 삽입
<script type="text/javascript" src="http://wcs.naver.net/wcslog.js"></script>

(중략)

<script type="text/javascript">
// 추가 정보 입력

// wcs_do 함수 호출
wcs_do();
</script>
```

4.3.2 유입 경로 추적 함수

Description

wcs.inflow 함수는 유입 경로를 추적하여 관련 정보를 쿠키에 기록한다.

랜딩 페이지에 wcslog.js 파일을 포함시키고, 네이버페이 버튼을 삽입하기 전에 wcs.inflow 함수를 호출해야 한다. 네이버페이 버튼 삽입에 대한 자세한 내용은 "2.1 네이버페이 버튼 삽입"을 참고한다.

랜딩 페이지에서 wcs.inflow 함수를 호출할 때 Account ID를 입력하는 것을 권장하나, Account ID가 없는 경우에도 wcs.inflow 함수는 정상적으로 호출되어야 NA_CO 쿠키가 정상적으로 생성된다.

네이버페이 가맹점은 whitelist가 존재한다면 wcs.inflow 함수를 호출하기 전에 반드시 wcs.checkoutWhitelist에 해당 URL을 배열 형태로 입력해야 한다. whitelist란 이용자가 가맹점의 랜딩 페이지(A)에서 다른 사이트(B)로 이동했다가 다시 가맹점의 랜딩 페이지(A)로 이동한 경우에도 유입 경로 정보를 유지해야 하는 사이트(B)의 도메인을 의미한다.

Syntax

```
wcs.inflow(sDomain);
```

Parameters

(String) sDomain

[in] 선택 입력 사항으로, 쿠키 도메인을 입력한다. 한 사이트가 여러 서브 도메인으로 관리되는 경우 최상위 도메인(TLD: Top Level Domain)을 입력한다. 이 값을 입력하지 않으면 기본값으로 랜딩 페이지의 도메인이 사용된다. 이 값이 제대로 설정되지 않으면 쿠키의 크로스도메인 문제로 인해 유입 추적 정보가 쿠키에 제대로 기록되지 않을 수 있다.

Return

없음

Example Code

```
// wcslog.js 파일 삽입
<script type="text/javascript" src="http://wcs.naver.net/wcslog.js"></script>

(중략)

<script type="text/javascript">

// 네이버페이 whitelist가 있을 경우
wcs.checkoutWhitelist = ["aaa.com", "bbb.com"];

// 유입 추적 함수 호출
wcs.inflow("xxx.com");

</script>
```