

독립몰 쇼핑몰 개발 담당자를 위한 주문 형 네이버페이 서비스 연동하기 V2.1



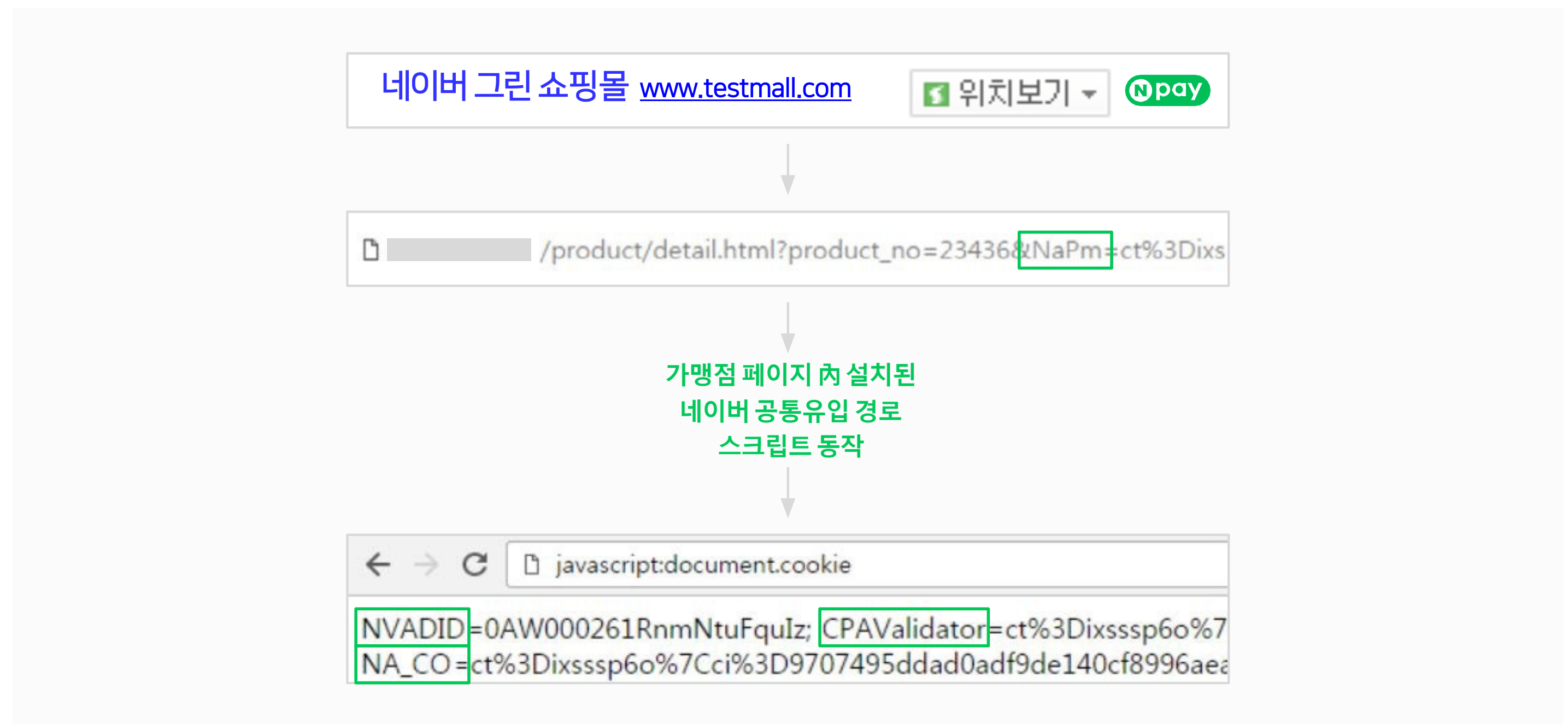
주문 V.2.1 개발

1. 네이버 공통 유입경로 스크립트 설치	3
2. 네이버페이 버튼 스크립트 설치	9
3. 주문등록	12
4. 상품정보XML	18
5. MOBILE 버전 개발	20

1. 네이버 공통 유입경로 스크립트 설치

네이버 공통 유입 경로 스크립트 동작방식(1)

네이버쇼핑, 네이버검색광고 등
네이버서비스에서 공통으로 사용
하는 스크립트이며,
유입경로 분석을 위해서 사용됨.



1. 네이버 공통 유입경로 스크립트 설치

네이버 공통 유입 경로 스크립트 동작방식(2)

자동생성되는 COOKIE 값

CPAValidator

네이버쇼핑 가맹점 중 파라미터 방식으로 CPA 트래킹을 사용.
주문등록 시 **CPA_INFLOW_CODE** 파라미터에 전달.

NVADID

네이버 검색광고주 일때 검색광고를 통해 쇼핑몰 유입시 쿠키 생성됨.
주문등록 시 **SA_CLICK_ID** 파라미터에 전달.

NA_CO

사이트, 쇼핑 등 검색을 통해 쇼핑몰 유입시 쿠키 생성됨.
주문등록 시 **NAVER_INFLOW_CODE** 파라미터에 전달.

1. 네이버 공통 유입경로 스크립트 설치

네이버 공통 유입 경로 스크립트 설치(1)

쇼핑몰의 모든 유입 페이지 설치하여, COOKIE 값이 생성/유지될 수 있도록 개발하며,
네이버페이 버튼 스크립트 보다 위에 있어야 함.

공통스크립트 파일 삽입

```
// http 프로토콜을 지원하는 웹페이지일 경우
<script type="text/javascript" src="http://wcs.naver.net/wcslog.js"></script>

// https 프로토콜을 지원하는 웹페이지일 경우
<script type="text/javascript" src="https://wcs.naver.net/wcslog.js"></script>
```

Account ID 설정

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

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

</script>
```

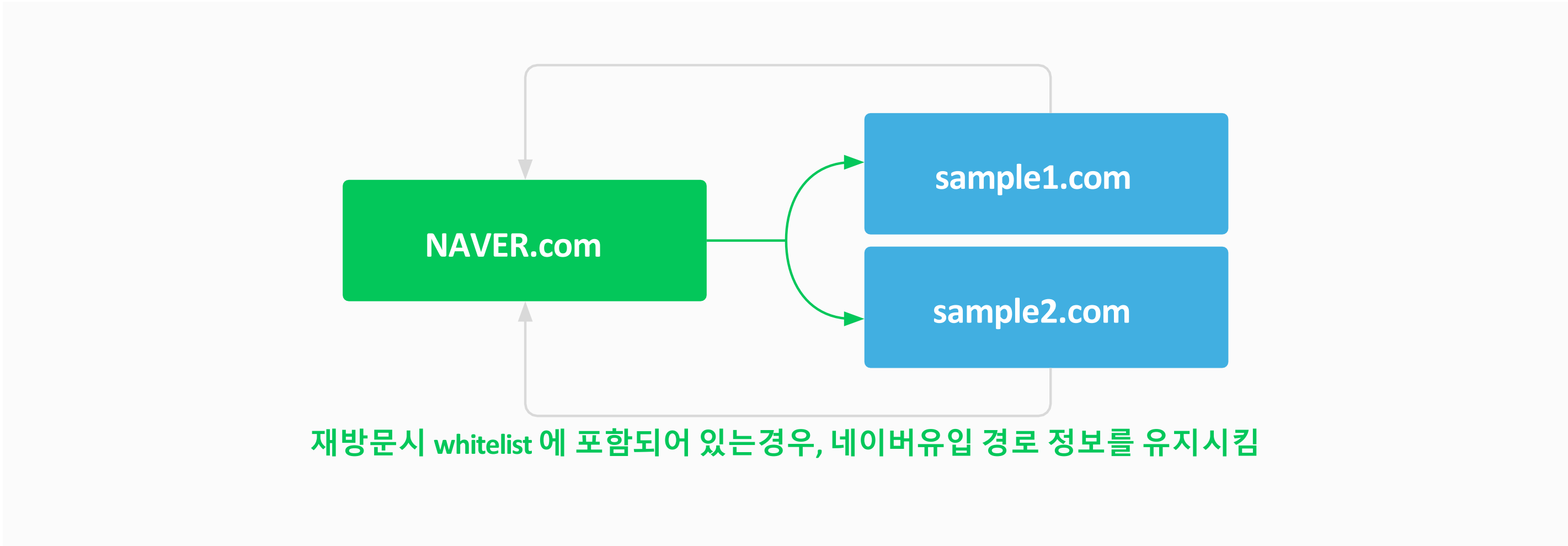
1. 네이버 공통 유입경로 스크립트 설치

네이버 공통 유입 경로 스크립트 설치(2)

네이버페이(구 체크아웃)
white list 설정

```
// 체크아웃 White list가 있을 경우  
wcs.checkoutWhitelist = ["sample1.com", "sample2.com" ];
```

예)
가맹점 도메인이 NAVER.com 이고,
다수 도메인을 사용할 때



1. 네이버 공통 유입경로 스크립트 설치

네이버 공통 유입 경로 스크립트 설치(3)

유입 경로 추적 함수 설정

```
wcs.inflow("naver.com");
```

최상위 도메인(TLD: Top Level Domain)을 인자로 넣어줍니다

예) www.naver.com -> "naver.com"

예) shop.naver.com -> "naver.com"

로그수집 공통함수 설정

```
// wcs_do 함수 호출  
wcs_do();
```

페이지 로딩이 끝난 시점에 로그가 전송될 수 있도록
</body>의 바로 앞에 wcs_do 함수를 호출.

1. 네이버 공통 유입경로 스크립트 설치

네이버 공통 유입 경로 스크립트 설치(4)

1

```
<head>
<script src="https://wcs.naver.net/wcslog.js"></script>
<script>
  if (!wcs_add) var wcs_add = {};
  wcs_add["wa"] = "s_111f1cad1a";
  wcs.checkoutWhitelist = ["aaa.com", "bbb.com"];
  wcs.inflow("naver.com");
</script>
</head>
```

2

```
<body>
..생략..
네이버페이 버튼 스크립트 설치(상품 상세페이지 or 장바구니 페이지)
..생략..
<script>
wcs_do();
</script>
</body>
```

2. 네이버페이 버튼 스크립트 설치

네이버페이 버튼 스크립트(1)

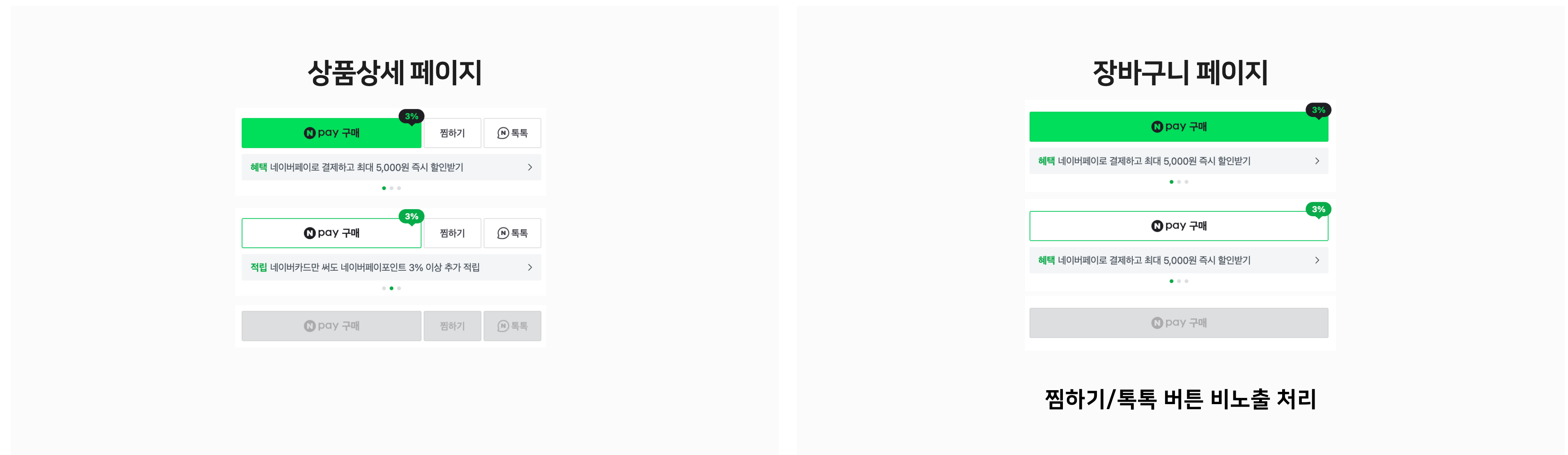
상품상세페이지 및 장바구니 페이지에 스크립트를 설치하여,
[구매하기], [찜] 기능이 동작되도록 개발.

공통스크립트 파일 삽입

샌드박스(테스트) 환경: `<script src="https://test-pay.naver.com/assets/button/latest/npay.button.js"></script>`

2. 네이버페이 버튼 스크립트 설치

네이버페이 버튼 스크립트(2)



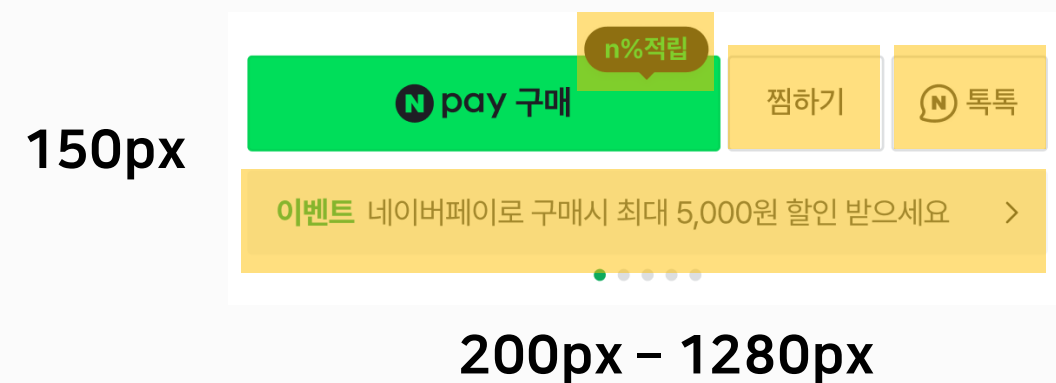
구매불가 및 품절상품의 경우 비활성화 or 비노출 방식으로 처리

2. 네이버페이 버튼 스크립트 설치

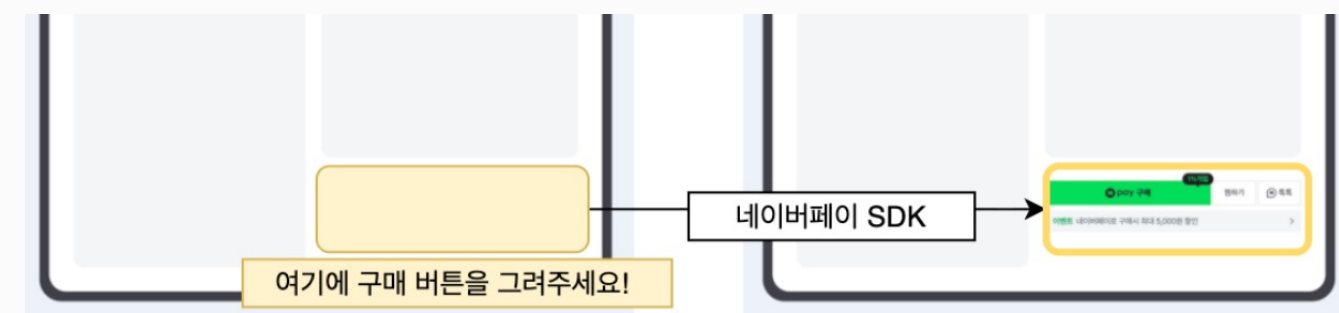
네이버페이 버튼 스크립트(3)

템플릿형 버튼

- 버튼 종류/색/수를 각각 지정하여 버튼을 호출하되, 가맹점에서 버튼의 구성요소를 선택할 수 있음
- 미리 네이버페이에서 지정한 디자인을 네이버페이가 직접 그려주는 방식
- 가맹점에서 구매 버튼이 그려질 영역을 지정하면 해당 영역에 맞게 반응형으로 지원



구매하기 버튼 외 모든 구성요소를
가맹점의 선택 사항으로 지원



2. 네이버페이 버튼 스크립트 설치

네이버페이 버튼 스크립트(4)

```
<!DOCTYPE html>
<html>
  <head>
    <!-- 네이버페이 버튼 SDK 로드 -->
    <script src="https://npay-order.pstatic.net/assets/button/latest/npay.button.js"></script>
  </head>
  <body>
    <!-- 템플릿 버튼 컨테이너 -->
    <div id="npay-button-container"></div>

    <script>
      // 네이버페이 버튼 SDK 초기화
      // 주의: orderRegistrationVersion 설정값에 따라 onBuyClick 반환 형식이 달라짐
      Npay.order.create({
        buttonKey: 'YOUR_BUTTON_KEY', // 필수: 가맹점 버튼 인증키
        containerId: 'npay-button-container', // 필수: 버튼이 렌더링될 컨테이너의 ID
        orderRegistrationVersion: '2.1', // 필수: 주문 등록 API 버전('1.0' 또는 '2.1')
        type: 'template', // 필수: 버튼 타입
        colorTheme: 'green', // 선택: 버튼 색상 테마(green / white), 기본값: green
        enable: true, // 선택: 버튼 활성화 여부, 기본값: true
        components: { // 선택: 표시할 컴포넌트 설정(미설정 시 컴포넌트 미노출)
          talkTalk: true, // 톡톡 버튼 노출 여부, 장바구니에서 사용x
          wishlist: true, // 찜하기 버튼 노출 여부, 장바구니에서 사용x
          benefitMessage: true, // 혜택 문구 노출 여부
          benefitCoachMark: true // 혜택 코치마크 노출 여부
        },
      },
```

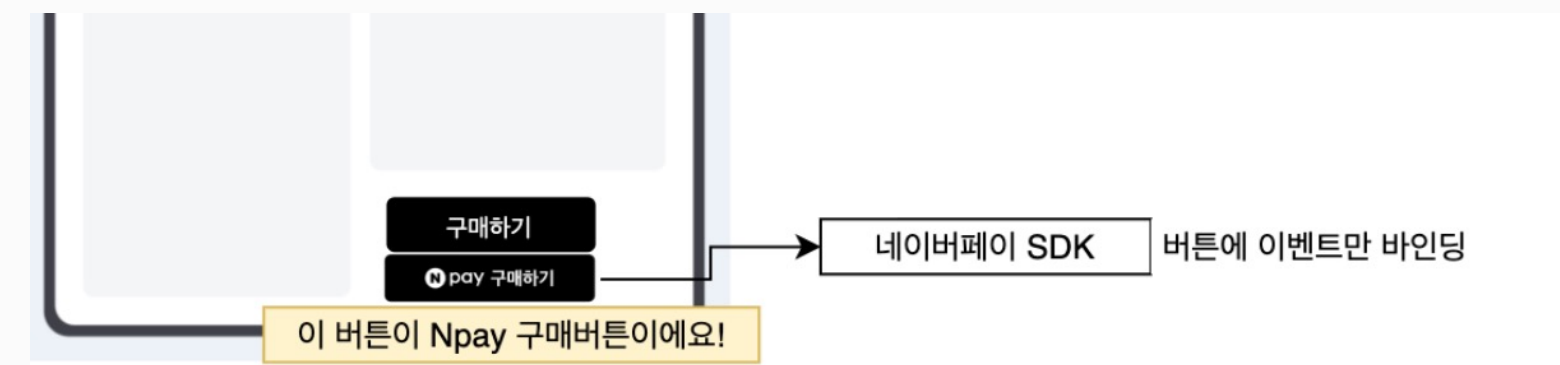
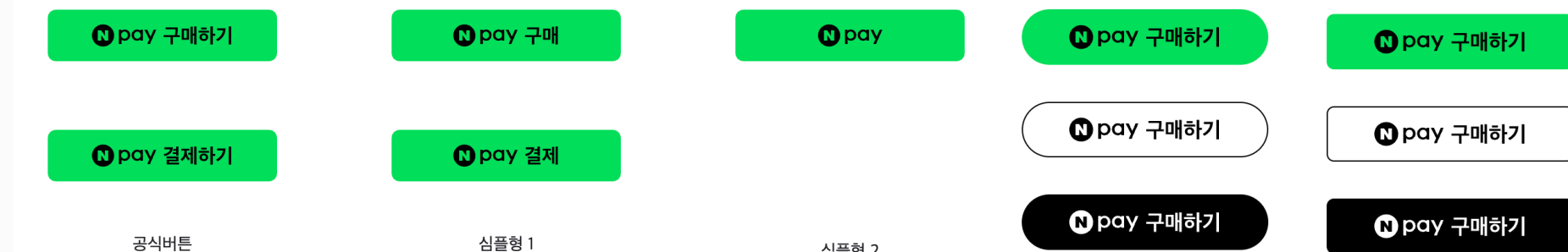
```
// 구매 버튼 클릭 핸들러
onBuyClick: async function () {
  // TODO: 주문 정보 등록 API 호출 후 결과 반환
  // -----
  // 현재 설정(orderRegistrationVersion: '2.1') 기준 반환값
  return {
    key: 'BUY_KEY', // 인증키(주문 정보 등록 결과)
    merchantNo: 'MERCHANT_NUMBER' // 가맹점 번호
  };
  // -----
  // orderRegistrationVersion을 '1.0'으로 변경한 경우 반환값 예시
  // return {
  //   key: 'ORDER_ID', // 주문 ID(주문 정보 등록 결과)
  //   merchantId: 'SHOP_ID' // 상점 ID(네이버페이 가입 승인 시 발급)
  // };
  // -----
},
// 찜하기 버튼 클릭 핸들러
onWishlistClick: async function () {
  // TODO: 실제 상품 정보로 교체
  return {
    merchantId: 'SHOP_ID', // 상점 ID(네이버페이 가입 승인 시 발급)
    payProductId: 'ITEM_ID' // 네이버페이 상품 ID(찜 등록 응답으로 수신)
  };
}
});
</script>
</body>
</html>
```

2. 네이버페이 버튼 스크립트 설치

네이버페이 버튼 스크립트(5)

커스텀형 버튼

- Npay 로고 및 네이밍 가이드만 준수한다면, 가맹점이 자체적으로 구매하기 버튼을 디자인할 수 있음
- 가맹점이 HTML 마크업을 구성하고, Npay 구매하기 SDK는 데이터 치환 및 이벤트를 바인딩



2. 네이버페이 버튼 스크립트 설치

네이버페이 버튼 스크립트(6)

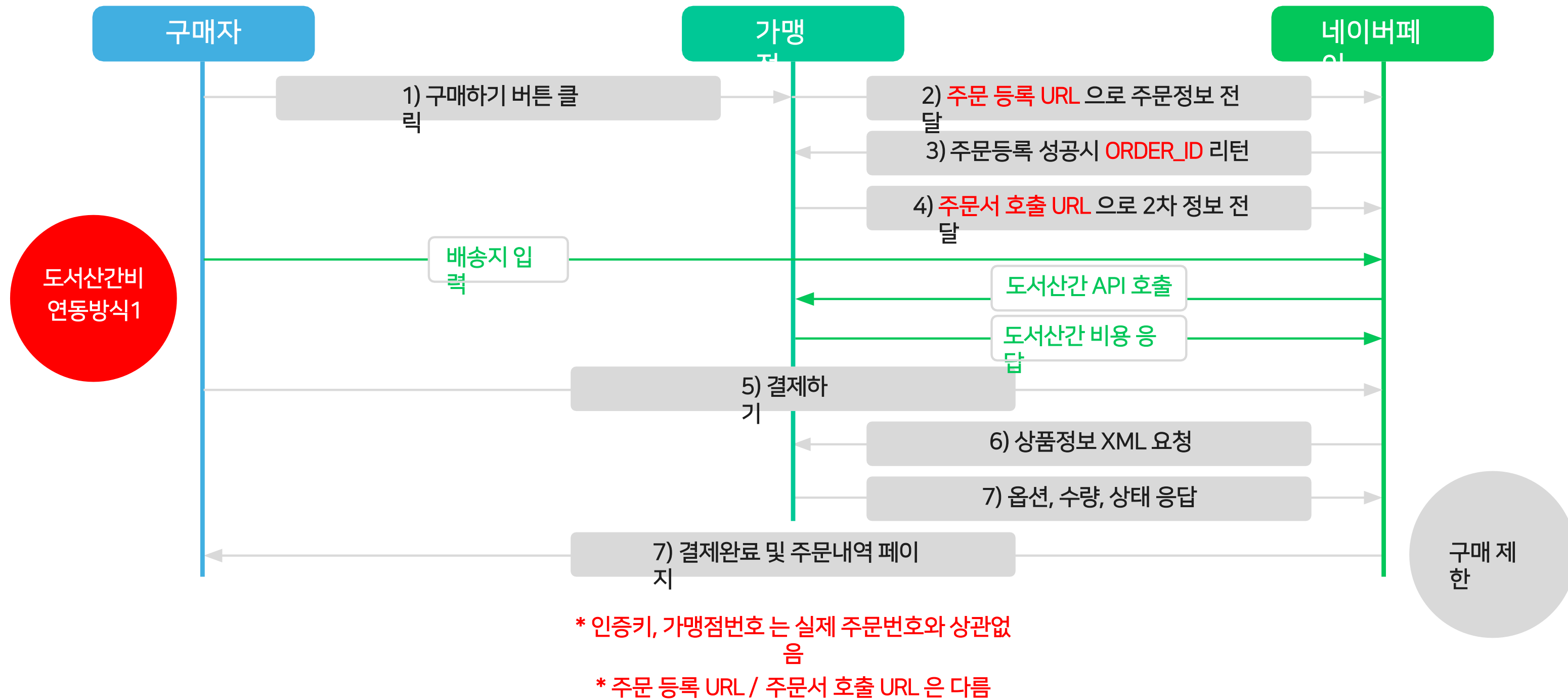
```
<!DOCTYPE html>
<html>
  <head>
    <!-- 네이버페이 버튼 SDK 로드 -->
    <script src="https://npay-order.pstatic.net/assets/button/latest/npay.button.js"></script>
  </head>
  <body>
    <!-- 템플릿 버튼 컨테이너 -->
    <div id="npay-button-container"></div>

    <script>
      // 네이버페이 버튼 SDK 초기화
      // 주의: orderRegistrationVersion 설정값에 따라 onBuyClick 반환 형식이 달라짐
      Npay.order.create({
        buttonKey: 'YOUR_BUTTON_KEY', // 필수: 가맹점 버튼 인증키
        containerId: 'npay-button-container', // 필수: 버튼이 렌더링될 컨테이너의 ID
        orderRegistrationVersion: '2.1', // 필수: 주문 등록 API 버전('1.0' 또는 '2.1')
        type: 'custom', // 필수: 버튼 타입
      },
```

```
// 구매 버튼 클릭 핸들러
onBuyClick: async function () {
  // TODO: 주문 정보 등록 API 호출 후 결과 반환
  // -----
  // 현재 설정(orderRegistrationVersion: '2.1') 기준 반환값
  return {
    key: 'BUY_KEY', // 인증키(주문 정보 등록 결과)
    merchantNo: 'MERCHANT_NUMBER' // 가맹점 번호
  };
  // -----
  // orderRegistrationVersion을 '1.0'으로 변경한 경우 반환값 예시
  // return {
  //   key: 'ORDER_ID', // 주문 ID(주문 정보 등록 결과)
  //   merchantId: 'SHOP_ID' // 상점 ID(네이버페이 가입 승인 시 발급)
  // };
  // -----
},
// 찜하기 버튼 클릭 핸들러
onWishlistClick: async function () {
  // TODO: 실제 상품 정보로 교체
  return {
    merchantId: 'SHOP_ID', // 상점 ID(네이버페이 가입 승인 시 발급)
    payProductId: 'ITEM_ID' // 네이버페이 상품 ID(찜 등록 응답으로 수신)
  };
}
});
</script>
</body>
</html>
```

3. 주문등록


주문 V2.1 프로세스



3. 주문등록

상품상세페이지

초코 램스울 스웨터 > 패션의류 > 패션의류



초코 램스울 스웨터
초코 램스울 스웨터

시중가격	84,000원
판매가격	69,000원
포인트	0점
배송비결제	주문시 결제

선택옵션

컬러: 아이보리

사이즈: L + 1,000원

컬러:아이보리 / 사이즈:L (+1,000원) 1

총 금액 : 70,000원

장바구니 바로구매

NAVER
네이버ID로 간편구매
네이버페이

N pay 구매

이벤트 네이버페이

```
<script type="text/javascript">
function buy_nc(url)
{
var check = checkOption(document.getElementById("opt").color, 0, "컬러") &amp;&amp;
checkOption(document.getElementById("opt").size, 0, "사이즈");
if ( check ) {
//주문 정보를 등록하는 가맹점 페이지로 이동.
//해당 페이지에서 주문 정보 등록 후 네이버페이 주문서 페이지로 이동.
location.href=url; // naverpay_order.php 으로 주문등록을 위한 상품 ID 등 전달.
}
return false;
}
function not_buy_nc()
{
alert("죄송합니다. 네이버페어로 구매가 불가능한 상품입니다.");
return false;
}
}]]&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="https://test-pay.naver.com/customer/js/naverPayButton.js" charset="UTF-8"&gt;&lt;/script&gt;
&lt;script type="text/javascript" &gt;<![CDATA[
naver.NaverPayButton.apply({
BUTTON_KEY: " 버튼인증키",
TYPE: "A",
COLOR: 1,
COUNT: 2,
ENABLE: "Y",
BUY_BUTTON_HANDLER: buy_nc,
WISHLIST_BUTTON_HANDLER : wishlist_nc,
BUY_BUTTON_LINK_URL:"naverpay_order.php", // 링크 주소 (필요한 경우만 사용) "";"
});
]]&gt;&lt;/script&gt;</pre></div><div data-bbox="47 900 371 947" data-label="Page-Footer"><p>16 독립몰 쇼핑몰 개발 담당자를 위한 주문형 네이버페이 서비스 연동하기 V2.1</p></div><div data-bbox="885 900 958 935" data-label="Page-Footer"><p>NAVER FINANCIAL</p></div>
```

3. 주문등록

단일상품 주문등록 샘플

1

```
<order>
  <merchantId>naver_pay</merchantId> <certiKey>11112A2A-F985-2222-TEST-4D837D879A12</certiKey>
  <product>
    <id>1000000072</id>
    <ecMallProductId>1000000072</ecMallProductId>
    <name><![CDATA[테스트상품]]></name>
    <basePrice>15000</basePrice>
    <taxType>TAX</taxType>
    <infoUrl><![CDATA[http://aaa.com/goods/goods_view.php?goodsNo=1000000072]]></infoUrl>
    <imageUrl><![CDATA[http://aaa.com/data/commonimg/ico_noimg_35.gif]]></imageUrl>
  <single>
    <quantity>1</quantity>
  </single>
  <shippingPolicy>
    <groupId>78</groupId> // 묶음배송, 개별배송 처리
    <method>DELIVERY</feePayType>
    <feeType>CONDITIONAL_FREE</feeType> // 조건부무료배송 처리
    <conditionalFree>
      <basePrice>20000</basePrice>
    </conditionalFree>
  </shippingPolicy>
</order>
```

2

```
  <feePayType>PREPAYED</feePayType>
  <feePrice>5000</feePrice>
  <surchargeByArea>
    <apiSupport>true</apiSupport>
  </surchargeByArea>
  </shippingPolicy>
</product>
<backUrl><![CDATA[http://aaa.com/goods/goods_<!-- $_COOKIE["CPAValidator"];
  $_COOKIE["NA_CO"];
  $_COOKIE["NVADID"]; -->]]></backUrl>
<interface>
  <salesCode></salesCode>
  <cpaInflowCode></cpaInflowCode>
  <naverInflowCode></naverInflowCode>
  <saClickId></saClickId>
  <merchantCustomCode1>1606031026193103</merchantCustomCode1>
  <merchantCustomCode2>2420</merchantCustomCode2>
</interface>
</order>
```

도서산간
비연동방
식 1

3. 주문등록(도서산간비 처리)

도서산간비 연동을 위한 2가지 방법

1. 도서산간 API 활용

```
<surchargeByArea>  
  <apiSupport>true</apiSupprot>  
</surchargeByArea>
```

요청

[http://www.aaa.co.kr/shop/additionalFee.php?productId\[0\]=P001&productId\[1\]=P002&zipcode=463050&address1=6rK96r iw64-EIOyEseuCqOyLnCDrtoTri7nqtawg7ISc7ZiE64-Z](http://www.aaa.co.kr/shop/additionalFee.php?productId[0]=P001&productId[1]=P002&zipcode=463050&address1=6rK96r iw64-EIOyEseuCqOyLnCDrtoTri7nqtawg7ISc7ZiE64-Z)

응답

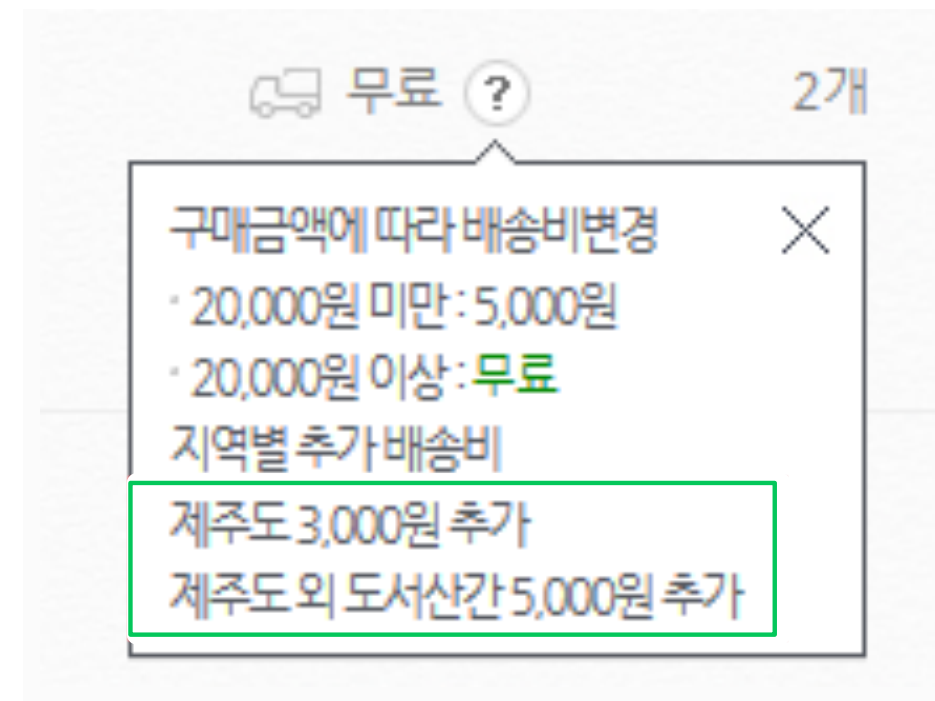
```
<additionalFees>  
  <additionalFee>  
    <id>P001</id>  
    <surprice>3000</surprice>  
  </additionalFee>  
  <additionalFee>  
    <id>P002</id>  
    <surprice>3000</surprice>  
  </additionalFee>  
</additionalFees>
```

3. 주문등록(도서산간비 처리)

도서산간비 연동을 위한 2가지 방법

2. 네이버페이 권역별 추가배송비 활용

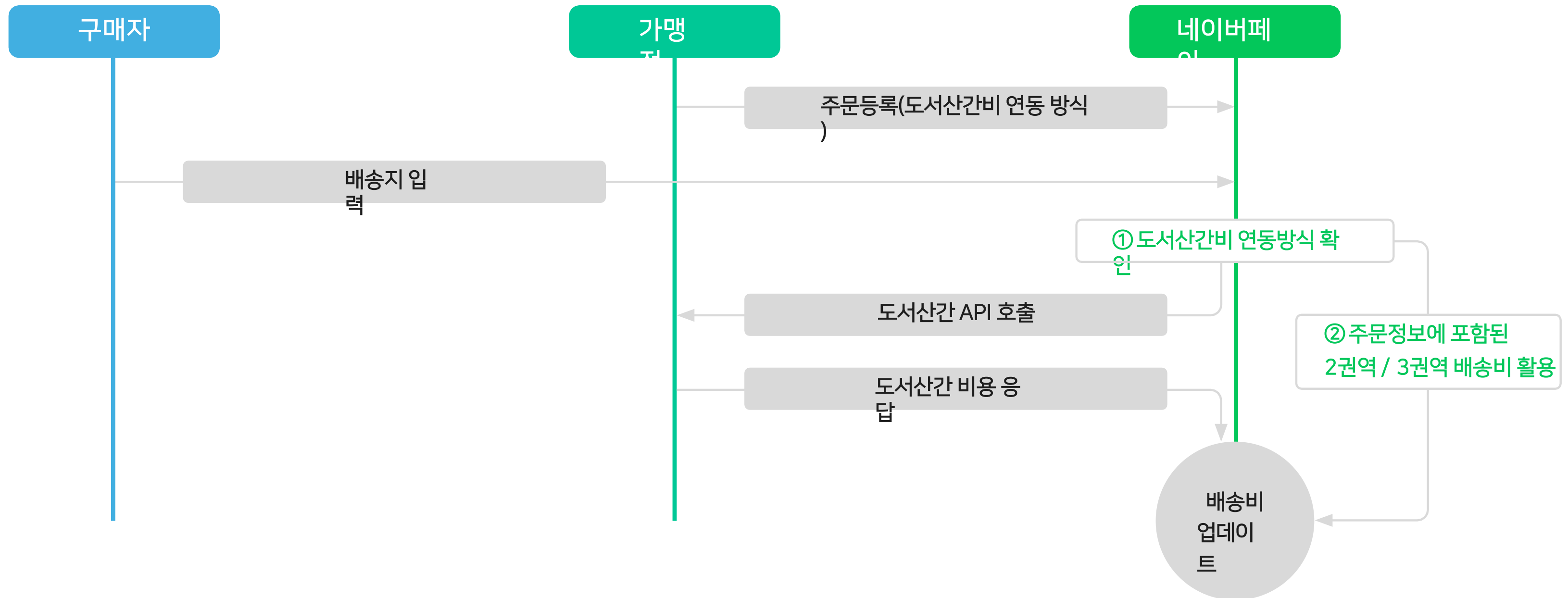
```
<surchargeByArea>  
  <splitUnit>3</splitUnit>  
  <area2Price>3000</area2Price>  
  <area3Price>5000</area3Price>  
</surchargeByArea>
```



3. 주문등록(도서산간비 처리)

도서산간비 연동을 위한 2가지 방법

- ① 도서산간비 API 활용
- ② 네이버페이 권역별 배송비 활용



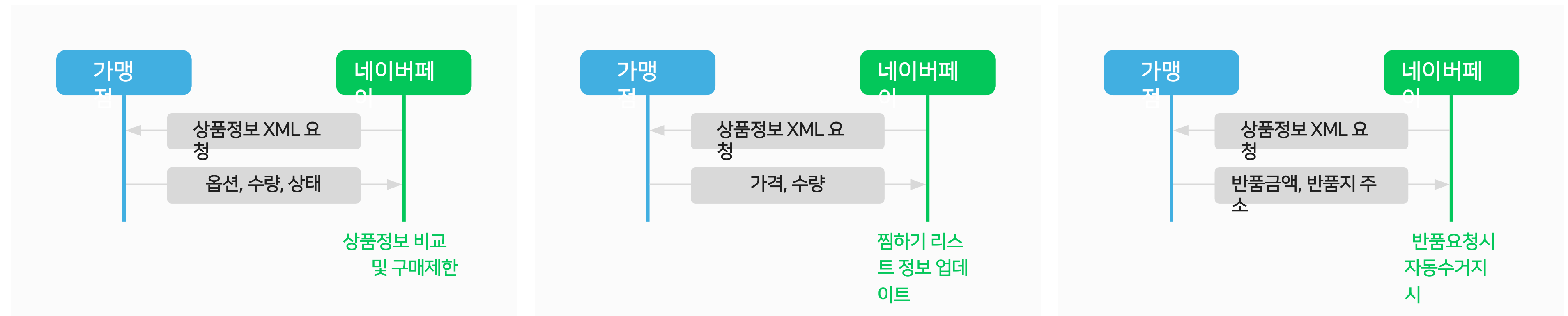
4. 상품정보 XML

상품정보XML 응답

네이버페이는 비주기적으로 상품정보를 업데이트하기 위해서 가맹점에서 지정해준 URL 으로 요청을 보내며, 가맹점에서는 요청받은 상품 정보를 XML 로 응답.

요청형식

- 본상품 [http://www.aaa.co.kr/shop/productInfo.php?product\[0\]\[id\]=1111&product\[1\]\[id\]=2222](http://www.aaa.co.kr/shop/productInfo.php?product[0][id]=1111&product[1][id]=2222)
- 옵션 및 추가상품 [http://www.aaa.co.kr/shop/productInfo.php?product\[0\]\[id\]=111&product\[0\]\[optionManageCodes\]=S_R,L_B&product\[0\]\[supplementIds\]=333&supplementSearch=true&optionSearch=true](http://www.aaa.co.kr/shop/productInfo.php?product[0][id]=111&product[0][optionManageCodes]=S_R,L_B&product[0][supplementIds]=333&supplementSearch=true&optionSearch=true)



4. 상품정보 XML

상품정보XML 응답 - 단일상품 응답형식

1

```
<?xml version="1.0" encoding="utf-8"?>
<products>
  <product>
    <id>1234</id>
    <ecMallProductId>1234</ecMallProductId>
    <name><![CDATA[테스트상품]]></name>
    <basePrice>15000</basePrice>
    <taxType>TAX</taxType>
    <infoUrl><![CDATA[http://aaa.com/goods/goods_view.php?goodsNo=1234]]></infoUrl>
    <imageUrl><![CDATA[http://aaa.com/data/commonimg/ico_noimg_35.gif]]></imageUrl>
    <status>ON_SALE</status> //거래상태 여부 체크
    <stockQuantity>100</stockQuantity> //재고체크
    <supplementSupport>>false</supplementSupport>
    <optionSupport>>false</optionSupport>
    <returnShippingFee>5000</returnShippingFee> //상품별 반품비용
    <exchangeShippingFee>10000</exchangeShippingFee> //상품별 교환비용
    <returnInfo>
      <zipcode>13561</zipcode>
      <address1><![CDATA[경기도 성남시 분당구 불정로 6]]></address1>
      <address2><![CDATA[그린팩토리]]></address2>
    </returnInfo>
  </product>
</products>
```

2

```
<sellername><![CDATA[네이버페이]]></sellername>
<contact1>01011112222</contact1>
<contact2>01011112222</contact2>
</returnInfo>
<shippingPolicy>
  <groupId>78</groupId>
  <method>DELIVERY</feePayType>
  <feeType>CONDITIONAL_FREE</feeType>
  <feePayType>PREPAYED</feePayType>
  <feePrice>5000</feePrice>
  <conditionalFree>
    <basePrice>20000</basePrice>
  </conditionalFree>
  <surchargeByArea>
    <apiSupport>>true</apiSupport>
  </surchargeByArea>
</shippingPolicy>
</product>
</products>
```

5. MOBILE 버전 개발

MOBILE 에서 주문/찜 요청 URL 구분

네이버페이 버튼 스크립트

PC/MOBILE 동일

운영 환경 (프로덕션): <https://npay-order.pstatic.net/assets/button/latest/npay.button.js>

테스트 환경 (샌드박스): <https://test-pay.naver.com/assets/button/latest/npay.button.js>

주문등록

PC / MOBILE 동일: <https://test-api.pay.naver.com/o/customer/api/order/v20/register>

네이버페이 버튼 스크립트

PC: <https://test-order.pay.naver.com/customer/buy/인증키/가맹점번호>

MOBILE: <https://test-m.pay.naver.com/o/customer/buy/인증키/가맹점번호>

찜등록

PC / MOBILE 동일: <https://test-pay.naver.com/customer/api/wishlist.nhn>

네이버페이 버튼 스크립트

PC: <https://test-pay.naver.com/customer/wishlistPopup.nhn>

MOBILE: <https://test-m.pay.naver.com/mobile/customer/wishList.nhn>

5. MOBILE 버전 개발

MOBILE 에서 주문서/찜하기 페이지 호출 방식

MOBILE 환경에서 주문서 페이지 및 찜하기는 팝업이 아닌 화면전환의 형태로 연동해 주시기 바랍니다.

팝업으로 호출



PC(팝업) `window.open(url, "", "scrollbars=yes,width=XXX,height=XXX");`

페이지 전환으로 호출

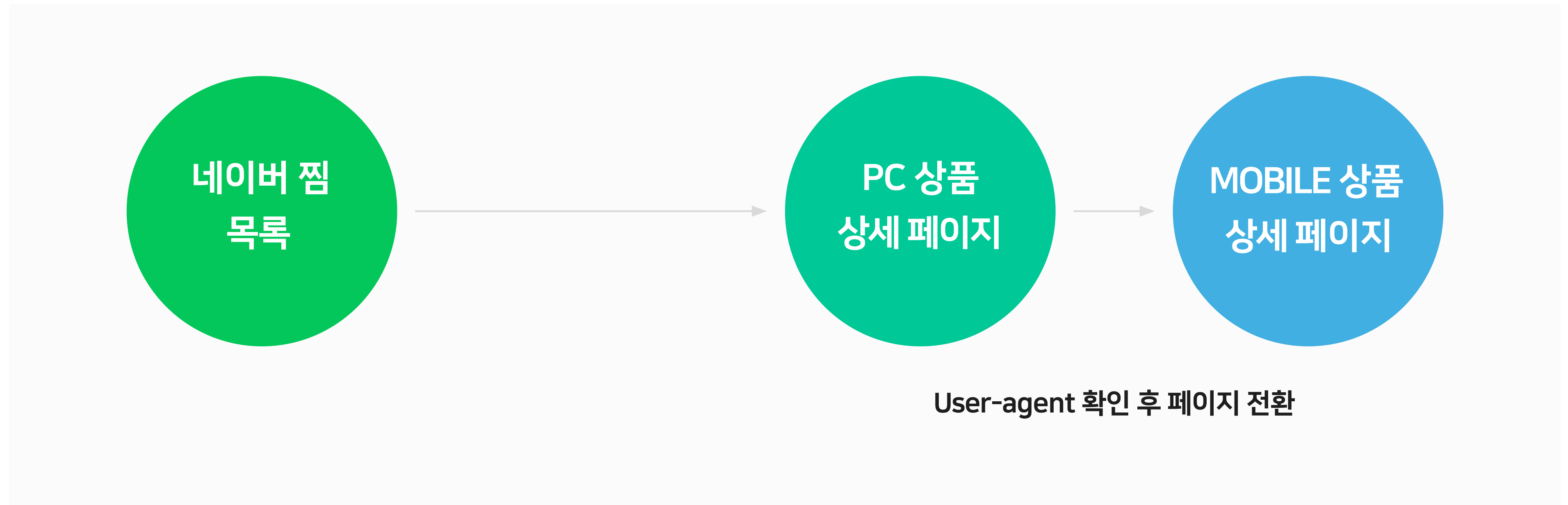


MOBILE(화면전환) `location.href = url;`

5. MOBILE 버전 개발

사이트맵에서 상품상세페이지로 유입

MOBILE(USER-AGENT로 구분)에서 접근했으면 모바일 상품상세 페이지로 Redirect 처리.



5. MOBILE 버전 개발

사이트맵에서 상품상세페이지로 유입

USER-AGENT 확인

PHP `$_SERVER['HTTP_USER_AGENT']`

JAVASCRIPT `navigator.userAgent`

- Mozilla/5.0 (Linux; **Android** 4.1.1; Nexus 7 Build/JRO03D) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Safari/535.19
- Mozilla/5.0 (iPhone; U; CPU **iPhone** OS 4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5

참고. 검수요청

개발 및 자체테스트

- 테스트환경에서 개발
- 상품별 주문등록 정보 확인
- 다양한 주문상황에 따른 주문등록 정보 확인
예) 수량, 옵션, 배송정책(선불, 착불, 무료)

개발 및 자체테스트

기술지원팀 dl_techsupport@navercorp.com 으로 아래 기본정보를 첨부하여 검수요청.

- 가맹점명
- 가맹점ID
- 주문연동버전 (1.0 or 2.1)
- 테스트페이지URL 및 개발환경에 따른 테스트방법
- 상품정보XML URL

E.O.D

-
기타궁금한점은 [네이버페이 가맹점 고객센터 톡톡](#)이나
1:1 문의하기로 연락주시면 친절히 답변 드리겠습니다.

