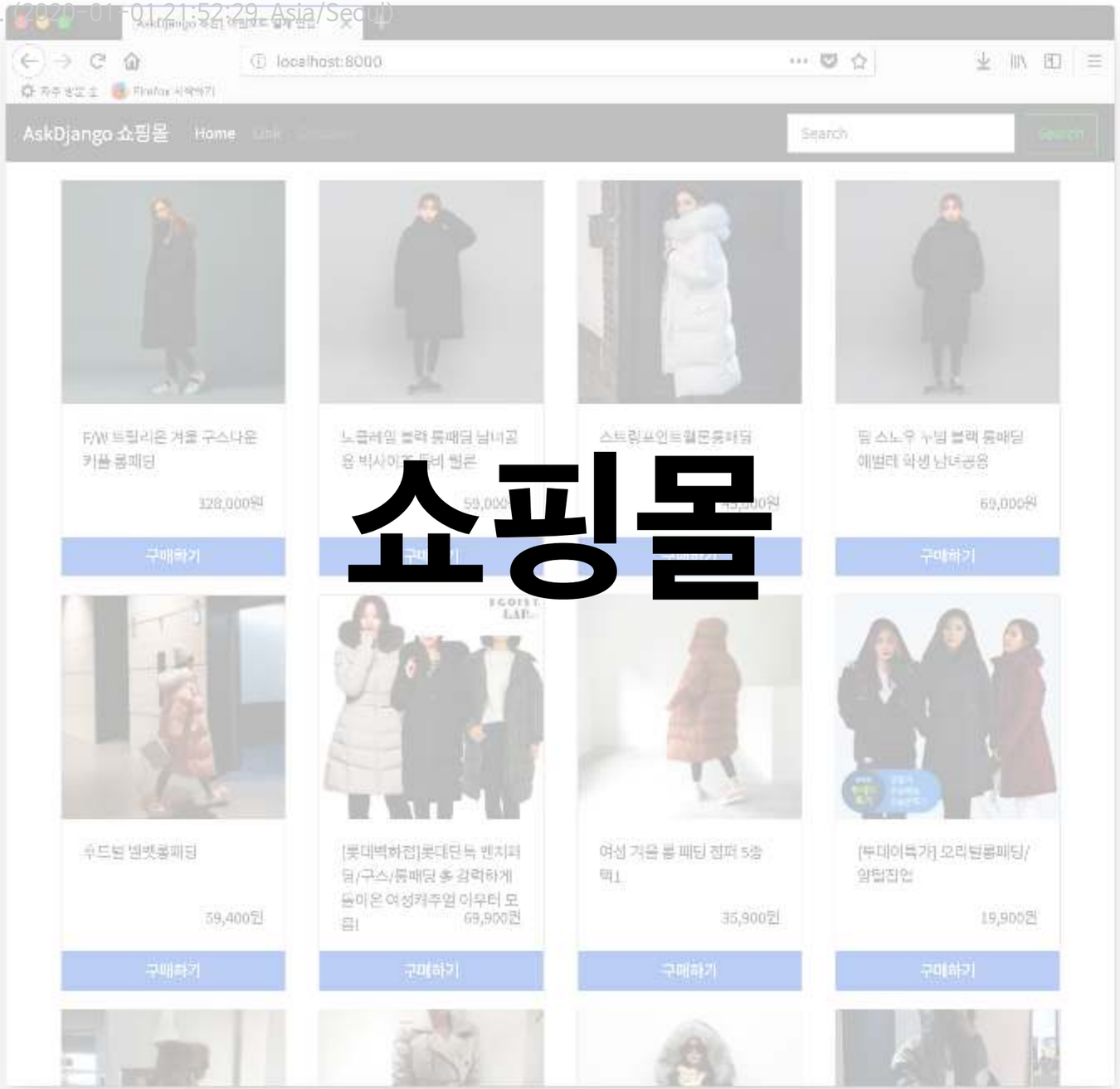
Ask Django

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

EP 1. Overview

이번 코스에서 살펴볼 내용

- 아임포트에 대한 이해
- 아임포트 자바스크립트 API를 장고 프로젝트에 적용하기
- 간단한 쇼핑몰 구현하기
- 장고 Model/ModelForm을 효율적으로 활용하여, 보다 견고한 결제 구현하기
- SMS 인증



쇼핑몰

우리가 만든 서비스에
결제를 붙이고 싶어요 !!!

그런데, 결제는 ...
왜 이리 붙이기가 어려운가 ...

전 세계가 다 그러한가?

NO !!!

우리만 어려움.

PG !!! 넌 내게 모욕감을 줬어.

- 이니시스 #home
  - 지원 샘플: ASP, ASP.NET, PHP, JSP ... :(
  - 이니시스 웹 연동 매뉴얼을 보시면 ;;;
- 그 외 여러 PG사들은 거의 10년 전에 만든 ASP/PHP/JSP 샘플만 제공
  - Python, Ruby, NodeJS 개발자들은 어쩔 ???
  - 결제부분만 **php-cli**를 통해 PHP로 처리하기도 ㅠ_ㅠ
- 아직 UTF-8이 아닌 euc-kr 인코딩을 쓰고 곳도 :(
- 게다가 PG사들마다 결제방법이 중구난방 :(

해외는 Stripe

가볍고 빠르게 적용할 수 있는 결제 서비스

- 설립 : 2010년
- 기업가치 : 약 10조원
- 고객사: 페이스북, 핀터레스트, 트위터, 킥스타터 등
- 2015년 기준, 미국인의 27%가 스트라이프 시스템을 이용한 결제경험
- 강점
  - 모바일앱, 웹사이트에 몇 줄의 소스코드만 추가하면 손쉽게 결제 연동
  - 개발자 친화적인 결제 시스템: 소스코드 공개
- 비즈니스 솔루션까지 확장 #관련기사

Developers first

```
// Require the Stripe library with a test secret key.
const stripe = require('stripe')('sk_test_BQokikJOvBiI2HlWgH4olfQ2');

// Create a payment from a test card token.
const charge = await stripe.charges.create({
  amount: 2000,
  currency: 'usd',
  source: 'tok_amex',
  description: 'My first payment'
});
```

물론 dj-stripe도 잘 관리되고 있어요. :)

국내에는 아임포트가 있어요.

- 아임포트가 각 PG사들에 대한 Gateway
- 즉 **단일 API**로 국내 수많은 PG사들을 커버합니다.
    - 개발환경에 중립적

카카오페이, 이니시스(웹표준결제), 나이스페이, 제이티넷, LG유플러스, 다날, 페이코, 시럽페이, 페이팔

- JavaScript 코드 하나로 PC/Mobile/In-APP 결제 모두 지원
    - 플랫폼 별로 약간의 추가 셋업이 필요하긴 합니다.
- 카드사가 요구하는 결제 프로세스를 100% 준수

아임포트, 서비스 비용 #공식페이지

- 기본기능에 대해서 FREE
  - 1개 PG사 연동
  - 기본 관리자 대시보드
  - 기본 기능 사용 제약 없음.
- 유료부분
  - 2개 이상 PG사 연동 (최초 1회만 10만원)
  - 결제 데이터 분석 (월 39,000원)

실제 연동 코드

```
<script src="//code.jquery.com/jquery-1.12.4.min.js" ></script>
<script src="//service.iamport.kr/js/iamport.payment-1.1.5.js"></script>
<script>
(function() {
    var IMP = window.IMP;
    var code = "iamport";  // FIXME: 가맹점 식별코드
    IMP.init(code);

    // 결제요청
    IMP.request_pay({
        pg : 'html5_inicis',
        pay_method : 'card',
        merchant_uid : 'merchant_' + new Date().getTime(),
        name : '주문명:결제테스트',
        amount : 14000,
        buyer_email : 'iamport@siot.do',
        buyer_name : '구매자이름',
        buyer_tel : '010-1234-5678',
        buyer_addr : '서울특별시 강남구 삼성동',
        buyer_postcode : '123-456',
        m_redirect_url : 'https://www.yourdomain.com/payments/complete'
    }, function(rsp) {
        if ( rsp.success ) {
        var msg = '결제가 완료되었습니다.';
        msg += '고유ID : ' + rsp.imp_uid;
        msg += '상점 거래ID : ' + rsp.merchant_uid;
        msg += '결제 금액 : ' + rsp.paid_amount;
        msg += '카드 승인번호 : ' + rsp.apply_num;
    }
    else {
        var msg = '결제에 실패하였습니다. 에러내용 : ' + rsp.error_msg
    }
    alert(msg);
});
})();
</script>
```

줄이면

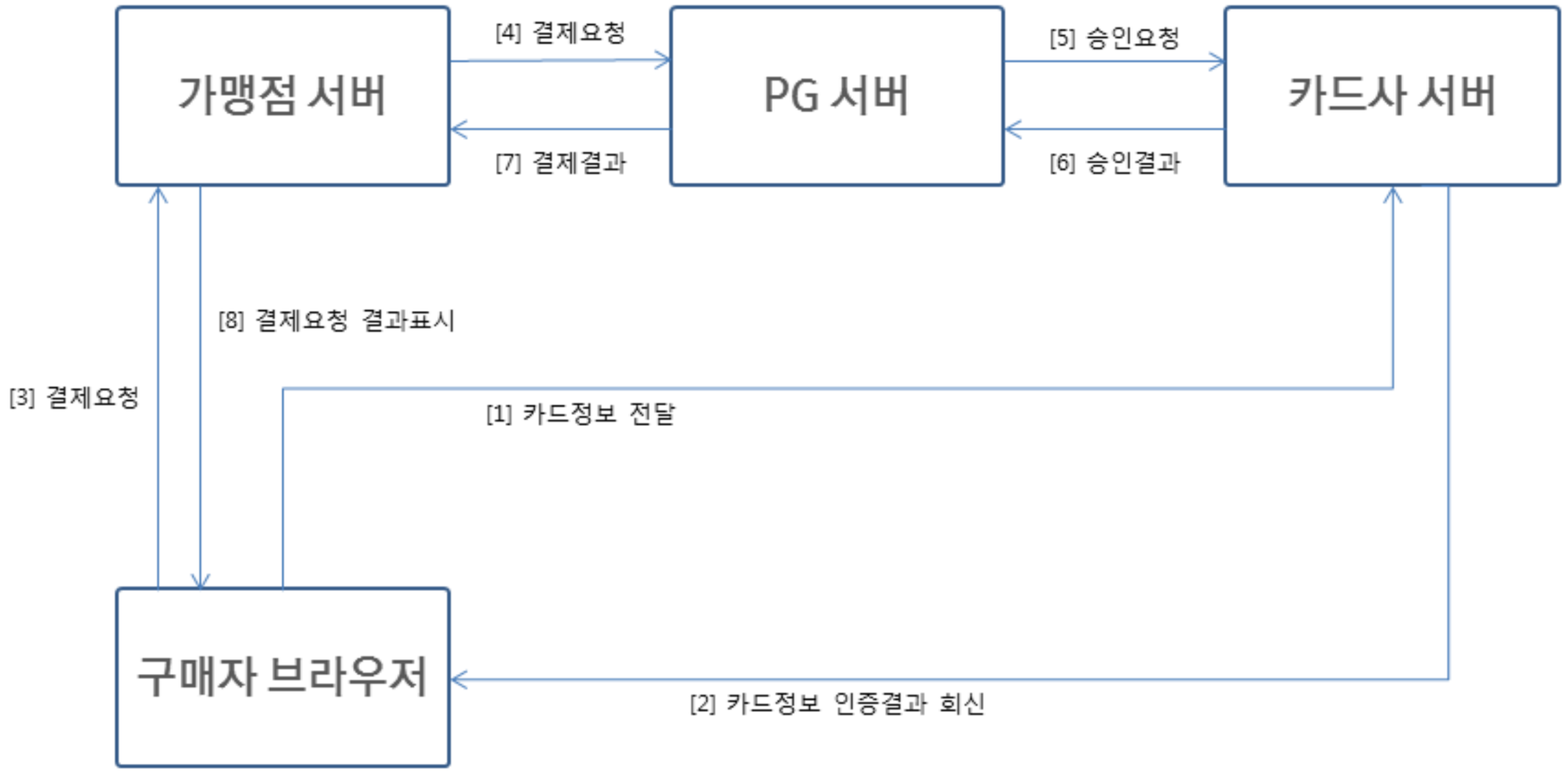
```
<script src="//code.jquery.com/jquery-1.12.4.min.js" ></script>
<script src="//service.iamport.kr/js/iamport.payment-1.1.5.js"></script>
<script>
(function() {
    var IMP = window.IMP;
    var code = "iamport";  // FIXME: 가맹점 식별코드
    IMP.init(code);

    // 결제요청
    IMP.request_pay({
        name : '주문명:결제테스트',
        amount : 14000
    }, function(rsp) {
        if ( rsp.success ) {
            var msg = '결제가 완료되었습니다.';
            msg += '고유ID : ' + rsp.imp_uid;
            msg += '상점 거래ID : ' + rsp.merchant_uid;
            msg += '결제 금액 : ' + rsp.paid_amount;
            msg += '카드 승인번호 : ' + rsp.apply_num;
        }
        else {
            var msg = '결제에 실패하였습니다. 에러내용 : ' + rsp.error_msg
        }
        alert(msg);
    });
})();
</script>
```

결제 과정

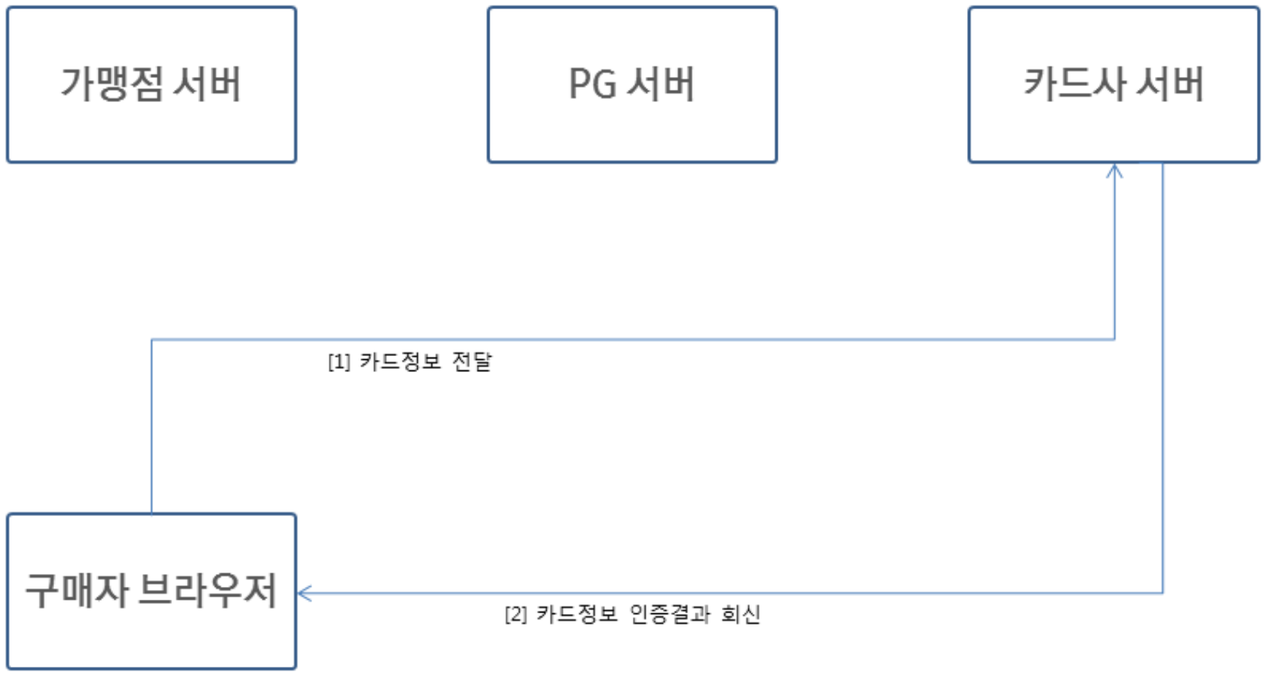
- 한국에서는 카드사, 일부 PG사(적격PG로 선정된 곳)를 제외하고는 **카드정보를 저장할 수 없도록 규정**
- 카드정보가 가맹점서버, PG사 서버를 **직접 거쳐가지 않도록** 시스템을 구성합니다.

전체 프로세스



단계 1) 인증

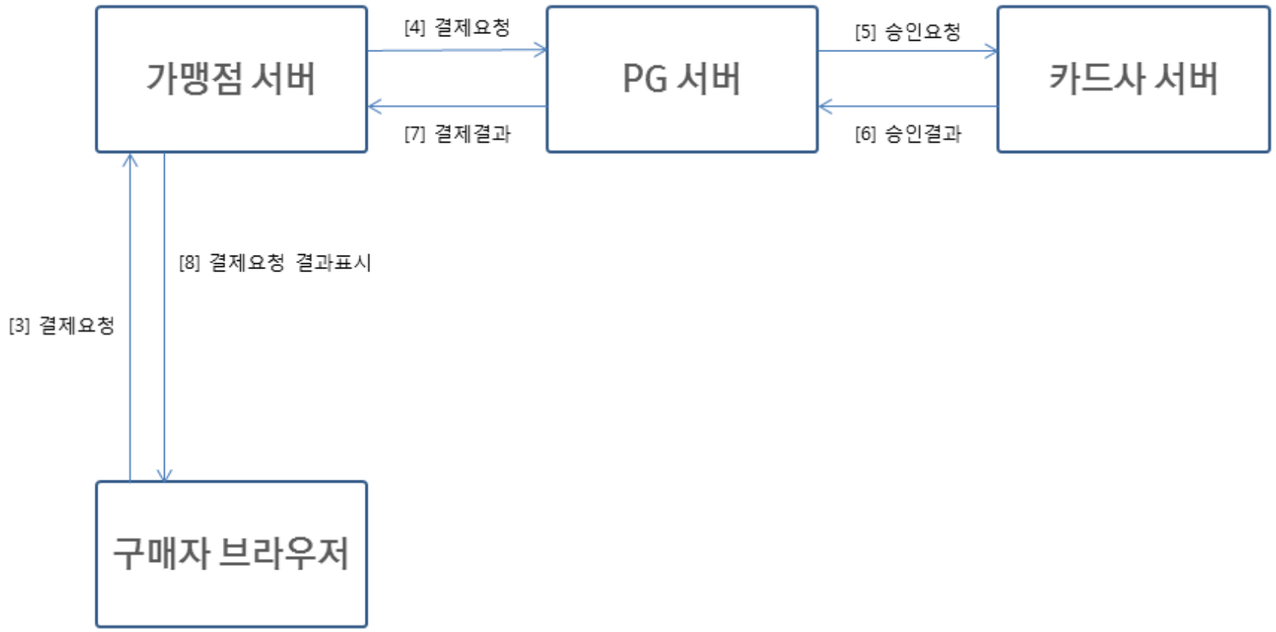
1. 구매자는 카드사 웹페이지를 통해 카드번호/유효기간/비밀번호를 입력하여 카드 확인
    - 이때, 카드정보의 보호(???)를 위해 별도의 프로그램 설치를 요구 ㅠ_ㅠ
2. 카드정보가 확인되면, 카드사는 1회성 인증키를 PG사에 제공되며, 이 인증키가 실제 결제에 사용



단계 2) 결제

가맹점 서버와 PG사 서버 간 통신

1. 가맹점 서버: 인증단계에서 사용된 주문번호, 구매자 연락처 등의 정보를 PG사 서버로 전송
2. PG사는 카드사로부터 전달된 인증키와 함께 해당 카드사로 승인요청을 진행
    - 이 과정에서 **실제 카드사로부터 결제승인에 대한 결과를 알 수 있습니다.** 결제승인 혹은 결제실패(한도초과/분실카드 등의 사유)가 판단됩니다.



안심클릭[1]/ISP[2] 결제모듈

PC에서는 계열에 상관없이 동일한 인터페이스
하지만, 모바일에서는 연동방식의 차이가 있음.

- 안심클릭: 카드번호, CVC, 안심클릭 비밀번호를 입력하여 카드정보를 인증
  - 별도의 앱이 필요없기에, 모바일 브라우저나 WebView 앱결제가 동일한 처리
- ISP Internet Secure Payment: 공개키 기반의 전자인증서를 통해 사전에 등록된 카드정보를 인증
  - ISP앱 설치/인증서복사를 요구
  - ISP을 거쳤다가, 다시 원래 결제프로세스로 이동
  - 다시 앱으로 돌아오기 위해, URL Scheme 정의/처리가 필요

---

[1] 안심클릭 계열: 신한, 현대, 하나(외환), 롯데, 삼성, NH

[2] ISP 계열: KB국민, 우리, BC 수협, 전북, 광주, 제주, 신협, 새마을금고 등

앱카드

- ISP와 유사한 역할을 하는 인증용 앱을 별도로 운영하고 이 앱을 통해 카드정보를 인증하도록 제공
  - 다시 앱으로 돌아오기 위해, URL Scheme 정의/처리가 필요

비인증결제

- key-in 결제: 전화를 통한 비대면 상황에서 카드정보 + 개인정보를 활용해 결제
- 빌링 결제: 정기결제

여신금융업법 16조(신용카드회원등에 대한 책임)에 따르면, [3]

회원(구매자)의 고의 또는 중과실로 카드정보 또는 비밀번호가 유출되어 부정사용된 것을 카드사가 입증해내지 못하면 신용카드사는 회원(구매자)에게 책임을 지울 수 없으며 보상처리를 하도록 규정되어있습니다.

이에 카드사/PG사는 방어적인 것이 현실.

간소화된 인증방식의 결제 서비스 사용을 최소한으로 제한

---

[3] #한국결제특징

아임포트가 Dirty Process를 대신 수행하고 ...

- 개발자는 아임포트가 제공하는 표준화된 API로 간편히 연동.

  1. PG창 결제창 호출 -> 결제 결과 수신
  2. 결제 조회/검증 (REST API)

- 본 코스를 통해, 장고와 유기적으로 잘 연동하는 법에 대해서 익혀봅시다.

결제하고 싶은데, 복잡하다 @_@;;;

아임포트를 통한 결제연동에 대해 차근차근 살펴봅시다. :D

이제 결제가 쉬워집니다.