



[한국ICT인재개발원] 스프링 프레임워크

## 14. 결제모듈

前) 광고데이터 분석 1년

前) IT강의 경력 2년 6개월

前) 머신러닝을 활용한 데이터 분석 프로젝트반 운영 1년

前) 리그오브 레전드 데이터 분석 등...

現) 국비반 강의 진행중

결제모듈을 붙이면 우리가 만든 웹사이트 정보를 활용해  
모의결제를 해볼 수 있습니다.

추후 쇼핑몰 등을 제작한다고 가정하면  
배운 지식에 더해서 수익을 창출해 낼 수 있을지도 모릅니다.  
우선 가장 간단한 아임포트 결제모듈을 예시로 사용해보겠습니다.

로그인하기

관리자 콘솔 체험하기

이메일주소

비밀번호

[비밀번호를 잊으셨나요?](#)

로그인

[아임포트 계정이 없으신가요?](#)



I'mport;  
관리자페이지입니다

로그인

아이디(Email)

패스워드

로그인

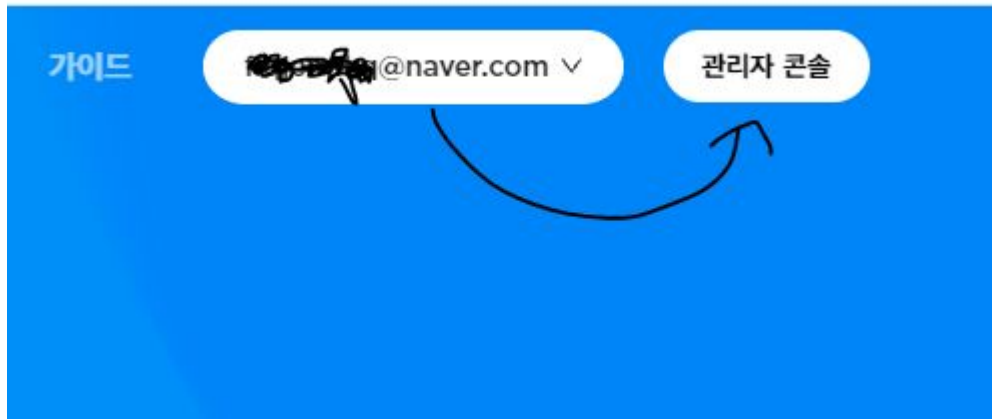
관리자 체험하기

← 비밀번호 찾기/변경

회원가입 →

결제모듈을 사용하기 위해서는 ajax를 통한 API 호출이 필요합니다.

먼저, 대표적인 모듈제공사이트인 아임포트에 가입하겠습니다.



l'mport; 관리자페이지    사용매뉴얼 보기    REST API    이용약관    개인정보 처리방침

2차인증(Google Authenticator) 지원

시스템설정    결제승인내역    본인인증내역    현금영수증 발급내역    정기결제현황    네이버페이 상품구매평

### 결제승인내역 >

# PG사  
모두 보기 ▾

# 테스트모드 데이터 숨김  
모두 보기 ▾

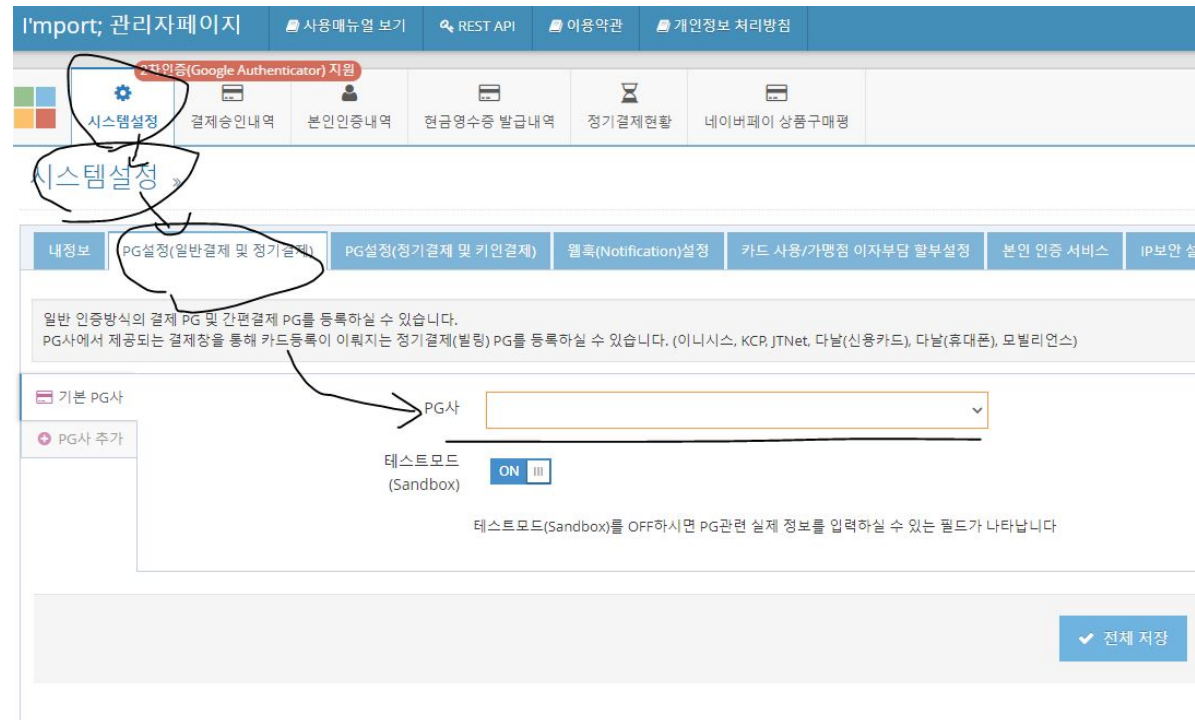
# 결제상태  
☒ 전체 (0)   ☒ 미결제 (0)   ☒ 결제완료 (0)   ☒ 전액취소 (0)   ☒ 부분취소 (0)   ☒ 결제실패 (0)

# 결제수단  
☒ 전체 (0)   ☒ 삼성페이 (0)   ☒ 신용카드 (0)   ☒ 실시간계좌이체 (0)   ☒ 가상계좌 (0)   ☒ 휴대폰소액 (0)   ☒ 문화상품

# 검색기간  
2021-10-13 00:00:00    2021-10-13 23:59:59    #

가입 후 로그인해서 우상단 관리자 콘솔을 클릭합니다.

재로그인시 관리자 페이지가 나옵니다.



관리자페이지에서 위와 같이 PG설정을 할 수 있습니다.

결제수단이 많이 나오는데 하나 골라주시면 되고,

하단 sandbox를 켜두셔야 테스트를 할 수 있습니다.

시스템설정 »

내정보	PG설정(일반결제 및 정기결제)	PG설정(정기결제 및 키인결제)	웹훅(Notification)설정	카드 사용/가맹점 이자부담 할부설정	본인 인증 서비스	IP보안 설정
-----	-------------------	-------------------	--------------------	---------------------	-----------	---------

가맹점 식별코드 :  [가맹점 식별코드란?](#)

REST API 키 :

REST API secret :

\* "재발급"이라는 글자가 복사되지 않도록 주의해주세요

TOTP 2차인증 :

QR

- \* 앱스토어에서 Google Authenticator 앱을 설치해주셔야 이용 가능합니다.
- \* 등록 이후부터는 로그인 시에 Google Authenticator 를 활용한 2차 인증이 진행됩니다.
- 1. Google Authenticator 설치
- 2. 오른쪽 상단의 + 버튼으로 등록 시작
- 3. 바코드 스캔 선택 후 좌측 QR코드 입력
- 4. 등록 후 화면에 표시된 6자리 숫자를 입력 후 등록버튼 클릭

PG사를 KG이니시스 표준결제창을 누르고 전체저장을 눌러주신 다음,  
내정보로 돌아오면 API호출용 키값이 주어집니다.

```
> pay_prj [spring_workspace master]
```

이제 간단한 프로젝트를 만들어서 예시를 보겠습니다.

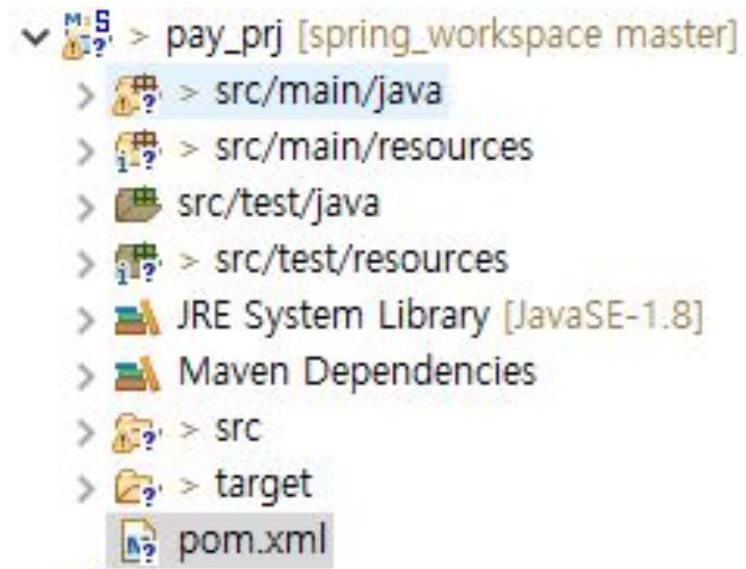
먼저 `pay_prj`를 만들어주세요.



```
10 <properties>
11     <java-version>1.8</java-version>
12     <org.springframework-version>5.0.7.RELEASE</org.springframework-version>
13     <org.aspectj-version>1.6.10</org.aspectj-version>
14     <org.slf4j-version>1.6.6</org.slf4j-version>
15 </properties>

    <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.5.1</version>
        <configuration>
            <source>1.8</source>
            <target>1.8</target>
            <compilerArgument>-Xlint:all</compilerArgument>
            <showWarnings>true</showWarnings>
            <showDeprecation>true</showDeprecation>
        </configuration>
    </plugin>
</configuration>
```

pom.xml내부는 위와같이 고쳐주세요.



우클릭 -> maven -> project update를 해서 1.8버전 적용을 해주세요.

```
40  
41 @GetMapping("/pay")  
42 public void pay() {  
43  
44 }  
45
```

```
▼ M5 > pay_prj [spring_workspace master]  
  ▼ > src/main/java  
    ▼ > org.ict.controller  
      > HomeController.java  
      - src/main/resources
```

```
▼ > src  
  ▼ > main  
    ▼ > webapp  
      resources  
      ▼ > WEB-INF  
        classes  
        > > spring  
        ▼ > views  
          home.jsp  
          pay.jsp
```

HomeController내부에 결제 페이지로 갈 수 있도록  
위와같이 MVC연결을 해주세요.

```
1 <%@ page language="java" c security_prj/src/main/webapp/WEB-INF/spring/security-context.xml
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <!-- 아임포트 모듈 -->
7 <script type="text/javascript" src="https://cdn.iamport.kr/js/iamport.payment-1.1.5.js"></script>
8 <!-- 제이쿼리 임포트(제이쿼리가 입력되어야 위의 모듈이 작동합니다.) -->
9 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
10 <meta charset="UTF-8">
11 <title>Insert title here</title>
12 </head>
```

body태그 전에는 우선 아임포트 모듈과 제이쿼리를 head태그 내에 같이 적용해둡니다.

```
<body>

<script>
function iamport(){
  IMP.init('XXXXXXXXXX'); //가맹점번호(본인 아이디에 입력된 번호로)
  IMP.request_pay({
    pg : 'html5_inicis', // KG이니시스
    pay_method : 'card', // 결제수단
    merchant_uid: "order_no_0001", // 상점에서 관리하는 주문 번호를 전달
    name : '주문명:결제테스트', //결제장에 올 상품명
    amount : 14000, //금액
    buyer_email : 'iamport@siot.do', //구매자 이메일
    buyer_name : '구매자이름', //구매자이름
    buyer_tel : '010-1234-5678', //구매자번호
    buyer_addr : '서울특별시 강남구 삼성동', //구매자주소
    buyer_postcode : '123-456', //구매자 우편번호
  }, function(rsp) {
    console.log(rsp);
    if ( rsp.success ) { // 결제 성공시 처리할 내역
      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 { // 결제 실패시 처리할 내역
      var msg = '결제에 실패하였습니다.';
      msg += '에러내용 : ' + rsp.error_msg;
    }
    alert(msg); // 여기서는 alert창만 띄우고 끝나지만 리다이렉트 등의 방법이 있음
  });
}
iamport();//실제로 실행 호출하기
</script>
```

코드가 상당히 길어보이지만 한 번만 작성하면 그 뒤로는 작성할 필요가 없습니다. 위와 같이 iamport라는 함수로 저장해두겠습니다.

신용카드

사이트/APP 회원관리  
하나로 끝!

대한민국

미국

중국

KG 이니시스

안전하고 편리한 이니시스결제입니다.

이용약관 ☐ 전체동의

전자금융거래 이용약관 ☐ 동의

개인정보 수집 및 이용안내 ☐ 동의

개인정보 제공 및 위탁안내 ☐ 동의

약관보기

pay 온 국민이 다 쓰는 카카오페이

N Pay 네이버페이로 결제하세요

현대카드

삼성카드

KB국민

비씨(페이북)

신한카드

NH농협

하나카드

씨티카드

롯데카드

UnionPay

그외카드

카드사별 무이자 할부 가능 개월 수 상이  
무이자 할부 제외 대상: 법인, 체크, GIFT, 선불, 은행계열 카드

무 무이자정액

PAYCO

L.pay

그외결제

ID 찾기? 비밀번호 찾기?  
어떻게 찾지?

인증으로 찾지~  
“이니시스 통합인증서비스”

KG 이니시스

상품명 주문명:결제테스트

상품가격 14,000 원

결제금액 14,000 원

다 음

이제 해당 페이지에 접속하면 결제창이 뜹니다.



결제를 꼭 진행하면 위와같이 결제가 완료됩니다.  
당일 23시 50분에 결제취소되니 부담없이 테스트해볼 수 있습니다.



```
117 <!-- https://mvnrepository.com/artifact/com.zaxxer/H
118 <dependency>
119     <groupId>com.zaxxer</groupId>
120     <artifactId>HikariCP</artifactId>
121     <version>4.0.3</version>
122 </dependency>
123
124 <!-- https://mvnrepository.com/artifact/org.mybatis/
125 <dependency>
126     <groupId>org.mybatis</groupId>
127     <artifactId>mybatis</artifactId>
128     <version>3.5.6</version>
129 </dependency>
130
131 <!-- https://mvnrepository.com/artifact/org.mybatis/
132 <dependency>
133     <groupId>org.mybatis</groupId>
134     <artifactId>mybatis-spring</artifactId>
135     <version>2.0.6</version>
136 </dependency>
137
138 <!-- https://mvnrepository.com/artifact/org.bgee.log
139 <dependency>
140     <groupId>org.bgee.log4jdbc-log4j2</groupId>
141     <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
142     <version>1.16</version>
143 </dependency>
144
```

```
11 <bean id="hikariConfig" class="com.zaxxer.hikari.Hikar
12     <property name="driverClassName"
13         value="net.sf.log4jdbc.sql.jdbcapi.DriverS
14     <property name="jdbcUrl"
15         value="jdbc:log4jdbc:oracle:thin:@localhos
16     <property name="username" value="mytest" />
17     <property name="password" value="mytest" />
18 </bean>
19
20 <bean id="dataSource"
21     class="com.zaxxer.hikari.HikariDataSource" destroy
22     <constructor-arg ref="hikariConfig" />
23 </bean>
24
25 <bean id="sqlSessionFactory" class="org.mybatis.spring
26     <property name="dataSource" ref="dataSource" />
27 </bean>
28
29 <mybatis-spring:scan base-package="org.ict.mapper"/>
30
```

다음으로 결제 데이터를 테이블에 담아보겠습니다.

먼저 프로젝트에 DB접근이 가능하도록 세팅해주시고, 마이바티스 연결도 해주세요.  
헛갈리면 05번 교안 세팅을 참조하세요.

그리고 json처리를 위한 jackson-databind도 잊지말고 추가합니다.



```
13<body>
14
15 <h1>품질좋은 상품목록</h1>
16
17<div class="itemSection">
18  <div class="itemCard">
19    <div class="itemTitle">
20      <h2>월장을 위한 근육보충제</h2>
21    </div>
22    <div class="itemContent">
23      <h2>맛없지만 단백질 보충이 됩니다.</h2>
24    </div>
25    <div class="itemPrice">
26      <p data-price="40000">40000원</p>
27    </div>
28    <div class="itemButton">
29      <button class="orderBtn">주문하기</button>
30    </div>
31  </div>
32
33  <div class="itemCard">
34    <div class="itemTitle">
35      <h2>개발자를 위한 키보드</h2>
36    </div>
37    <div class="itemContent">
38      <h2>타건감이 뛰어난 키보드</h2>
39    </div>
40    <div class="itemPrice">
41      <p data-price="200000">200000원</p>
42    </div>
43    <div class="itemButton">
44      <button class="orderBtn">주문하기</button>
45    </div>
46  </div>
47</div>
```

그리고, pay.jsp를 위와같이 고쳐줍니다. 옆 라인을 보고 순차적으로 적어주세요.

```
48
49<script>
50 // 미리 받아와야할 정보를 변수를 전역변수처럼 쓰기위해 선언해두기
51 var itemPrice = 0;//가격
52 var itemTitle = "";//물건이름
53 var merchant_uid = "";// 주문번호
54 // 위임처리로 어떤 상품을 클릭했을때 그 상품에 대한
55 $(".itemSection").on("click", ".orderBtn", function(){
56     itemPrice = $(this).parent().siblings(".itemPrice").children().attr("data-price");
57     itemTitle = $(this).parent().siblings(".itemTitle").children().text();
58     d = new Date();// 현 시간
59     merchant_uid = "order" + d.getTime();// 현시간을 유닉스시간으로 변경해 order에 붙임
60     // 입력정보를 가져오고 나서 함수 호출
61     iamport();
62 });
63
```

이제 자바스크립트 코드에서는, 위임을 통해 각 상품 구매버튼을 눌렀을때  
상품의 가격과 이름, 그리고 주문번호를 얻고난 다음 결제모듈창을 커도록 처리합니다.  
역시 옆 라인을 보고 순차적으로 적어주세요.  
클릭시 맞는 정보가 들어오는지 디버깅은 필수입니다!

```
64 function iamport(){
65     IMP.init('imp23066347'); //가맹점번호(본인 아이디에 입력된 번호로)
66     IMP.request_pay({
67         pg : 'html5_inicis', // KG이니시스
68         pay_method : 'card', // 결제수단
69         merchant_uid: merchant_uid, // 주문번호는 order+유닉스시간으로 처리해 겹치지 않게
70         name : itemTitle, //결제창에 뜰 상품명
71         amount : itemPrice, //금액
72         // 신상정보는 나중에 로그인한 사용자정보로 대체
73         buyer_email : 'iamport@siot.do', //구매자 이메일
74         buyer_name : '구매자이름', //구매자이름
75         buyer_tel : '010-1234-5678', //구매자번호
76         buyer_addr : '서울특별시 강남구 삼성동', //구매자주소
77         buyer_postcode : '123-456', //구매자 우편번호
78     }, function(rsp) {
79         console.log(rsp);
80         if ( rsp.success ) { // 결제 성공시 ajax로 DB에 데이터를 전송해 입력
```

하단의 iamport 함수 선언부는 두 부분으로 나뉘어서 설명드리겠습니다.  
우선 상품별로 바뀌어야 하는 주문번호, 결제상품명, 금액만 예시코드로 작성하고  
추후 로그인한 사용자의 신상정보를 이용해 주문주소까지 직접 처리해주세요.

```
78     }, function(rsp) {
79         console.log(rsp);
80         if ( rsp.success ) {// 결제 성공시 ajax로 DB에 데이터를 전송해 입력
81             $.ajax({
82                 type: 'post',
83                 url: '/order',
84                 headers: {
85                     "Content-Type": "application/json",
86                     "X-HTTP-Method-Override": "POST"
87                 },
88                 dataType: "text",
89                 data: JSON.stringify({
90                     itemname : itemTitle,
91                     amount: itemPrice,
92                     merchant_uid : merchant_uid
93                 }),
94                 success: function(){
95                     alert(itemTitle + " 결제완료!");
96                 }
97             });
98         } else {// 결제 실패시 처리할 내역
99             var msg = '결제에 실패하였습니다.';
100             msg += '에러내용 : ' + rsp.error_msg;
101             alert(msg);
102         }
103     });
104 }
105 </script>
```

그리고, 80번라인의 **rsp.success**가 바로 결제요청이 성공했을때 **true**를 반환합니다. 결제 성공시 비동기식 요청으로 물건이름, 가격, 주문번호를 DB에 넣도록 요청합니다.

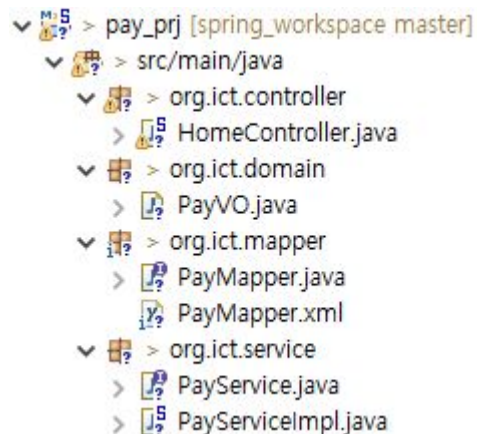
커스터마이징시 테이블을 설계에 맞춰 생성하고 맞는 데이터를 보내세요.

```
CREATE TABLE pay_tbl(  
    itemname varchar(100) not null,  
    amount number not null,  
    merchant_uid varchar(100) not null primary key  
);
```

먼저 테이블을 생성합니다.

예시코드에서는 물건이름, 가격, 주문번호만 준비합니다.

추후 유저정보나 주문시각, 주문주소 등을 저장하도록 설계해주세요.



```
1 package org.ict.domain;
2
3 import lombok.Data;
4
5 @Data
6 public class PayVO {
7
8     private String itemname;
9     private Long amount;
10    private String merchant_uid;
11 }
12
```

이제 이 비동기요청을 처리할 수 있도록 컨트롤러, 서비스, 매퍼를 세팅합니다.

그리고 테이블 정보를 묶어둘 VO 내부는 위와 같이 생성합니다.



```
PayMapper.java ✖
1 package org.ict.mapper;
2
3 import org.ict.domain.PayVO;
4
5 public interface PayMapper {
6
7     public void insertPay(PayVO vo);
8 }
9
```

PayMapper.xml ✖

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper
3     PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <!-- 위의 스키마 작성 후, 어떤 인터페이스의 메서드를 쿼리문과
6 연결해 줄지 아래와 같이 인터페이스 파일부터 mapper태그로 연결합니다. -->
7 <mapper namespace="org.ict.mapper.PayMapper">
8
9     <insert id="insertPay">
10         INSERT INTO pay_tbl (itemname, amount, merchant_uid)
11         VALUES
12             (#{itemname}, #{amount}, #{merchant_uid})
13     </insert>
14
15 </mapper>
```

매퍼쪽에 insert구문을 생성해서 주문시 비동기로 받는 정보를 입력하도록 합니다.

PayService.java

```
1 package org.ict.service;
2
3 import org.ict.domain.PayVO;
4
5 public interface PayService {
6
7     public void insertPay(PayVO vo);
8
9 }
```

PayServiceImpl.java

```
1 package org.ict.service;
2
3 import org.ict.domain.PayVO;
4
5 @Service
6 public class PayServiceImpl implements PayService {
7
8     @Autowired
9     private PayMapper mapper;
10
11     @Override
12     public void insertPay(PayVO vo) {
13
14         mapper.insertPay(vo);
15     }
16
17 }
```

서비스쪽에서도 동일하게 처리해줍니다.



```
1 package org.ict.controller;
2
3 import java.lang.annotation.Target;
4
5 /**
6  * Handles requests for the application home page.
7  */
8 @Controller
9 public class HomeController {
10
11     @Autowired
12     private PayService service;
13
14     private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
15
16     * Simply selects the home view to render by returning its name.
17     public String home(Locale locale, Model model) {}
18
19     @GetMapping("/pay")
20     public void pay() {
21
22     }
23
24     @ResponseBody
25     @PostMapping(value="/order", consumes="application/json",
26                 produces= {MediaType.TEXT_PLAIN_VALUE})
27     public ResponseEntity<String> orderInsert(@RequestBody PayVO vo) {
28         logger.info("VO검증" + vo);
29         service.insertPay(vo);
30         return new ResponseEntity<String>("success", HttpStatus.OK);
31     }
32
33 }
```

추가

검증

추가

마지막으로 컨트롤러에서 서비스를 호출하도록 해 주고, /order 주소에 대한 post처리를 할 수 있는 메서드를 작성합니다.

← → ↻ ⓘ localhost:8181/pay  
출퇴근부 룰 분석 관련 스포츠 분석 관련 rating

## 품질좋은 상품목록

헬창을 위한 근육보충제

맛없지만 단백질 보충이 됩니다.

40000원

주문하기

개발자를 위한 키보드

타건감이 죽여주는 키보드

200000원

주문하기

## 품질좋은 상품목록

헬창을 위한 근육보충제

맛없지만 단백질 보충이 됩니다.

40000원

주문하기

개발자를 위한 키보드

타건감이 죽여주는 키보드

200000원

주문하기

신용카드

KG 이니시스

안전하고 편리한 이니시스결제입니다.

이용약관

☐ 전체동의

전자금융거래 이용약관

☐ 동의

개인정보의 수집 및 이용안내

☐ 동의

개인정보 제공 및 위탁안내

☐ 동의

약관보기

pay

은 국민이 다 쓰는 카카오페이

N Pay

네이버페이로 결제하세요

현대카드

삼성카드

KB국민

비씨(페이북)

신한카드

NH농협

하나카드

씨티카드

롯데카드

UnionPay

그외카드

카드사별 무이자 할부 가능 개월 수 상이  
무이자 할부 제외 대상: 법인, 체크, GIFT, 선불, 은행계열 카드

무이자정액

PAYCO

L.pay

그외결제

사이트/APP 회원관리  
하나로 끝!

한국 미국 중국

페이북

BC카드 7만원 이용하면  
7만원 적립

카드사별 무이자 할부 가능 개월 수 상이  
무이자 할부 제외 대상: 법인, 체크, GIFT, 선불, 은행계열 카드

카드 하나로  
MULTI  
플레이!

다음

주문하기 버튼을 누르면 제대로 결제창이 열리나 확인부터 해 주세요.

신용카드

사이트/APP 회원관리  
하나로 끝!

KG 이니시스

안전하고 편리한 이니시스결제입니다.

이용약관

전자금융거래 이용약관

☒ 동의

개인정보의 수집 및 이용안내

☒ 동의

개인정보 제공 및 위탁안내

☒ 동의

약관보기

결제방법

신용카드(안전결제[ISP])

판매자

(주)케이지이니시스

구매자

구매자 \* 례

이메일

입력하신 이메일로 결제내역을 보내드립니다.

결제 내용을 확인 후, 동의하시면 우측 하단의 '결제' 버튼을 눌러주세요.

BC

페이북

BC카드 7만원 이용하면  
7만원 적립

KG 이니시스

상품명

개발자를 위한 키보드

상품가격

2,000 원

할부구분

일시불

결제금액

2,000 원

카드 하나로  
MULTI  
플레이!

결 제

추후 구매자나 이메일은 회원정보로 바뀌어서 넣어주시면 됩니다.  
2000원인 이유는 결제금액이 커지면 부담되어 테스트 금액을 조절했습니다.

27



```
SELECT * FROM pay_tbl;
```

ITEMNAME	AMOUNT	MERCHANT_UID
1 개발자를 위한 키보드	2000	order1634143970186

결제완료 후 DB를 조회하면 위와같이 주문내역이 들어가있습니다.

추후 여러분들이 커스터마이징 한다면

상품 상세페이지에서 상품갯수를 입력해 주문하도록 만드는 부분과

회원이 로그인해서 주문할 경우 주문자명이나 주소등을 회원으로 처리해주는 부분 등이 있습니다.

지금 작성한 결제모듈은 잘 작동하지만, 번조에 취약한 형태입니다.

번조 방지 기능은 우선 아임포트 사이트에서 공식 문서를 보고 확인해주세요.

회원기능 등을 고려해 프로젝트에 적용할 수 있도록 커스터마이징해주시면 됩니다.