<참고>

- src/main/resource
  static: html, img등 정적 데이터를 담는 공간
  template: 웹페이지의 레이아웃 정보를 담는 공간
- src/main/webapp/WEB-INF/views
  jsp파일을 담는 공간
  *application.properties에 경로 설정
  **jsp파일을 직접실행하면 오픈 불가(404) controller를 통해서 실행해야함.
  ***controller 사용하지 않는 프로젝트의 경우 WEB-INF 안에 넣어주면 시스템에서 읽지 못한다. webapp내부에만 넣어줌
- pk설정

Shop_Member.java | MemberController.java | form.jsp

```
1  package com.example.app2.model.join;
2
3  import javax.persistence.Entity;
7
8  @Entity
9  public class Shop_Member {
10     @Id
11     private String id;//pk 설정
12
13     private String pwd;
14     private String name;
15     private String email;
16
17     @Enumerated(EnumType.STRING)
18     private MemberType type;
19
20     public Shop_Member() {
21     }
```

- repos 제네릭 설정값

Shop_Member.java | Shop_Member_Repos.java | MemberController.java | form.jsp | UserController.java

```
1  package com.example.app2.model.join;
2
3  import org.springframework.data.jpa.repository.JpaRepository;
4
5  public interface Shop_Member_Repos extends JpaRepository<Shop_Member, String> {
6  //<vo, pk의 type>
7  }
```

연습문제:: 쇼핑몰

- 일반 웹 방법
- 레스트풀 방법[백단과 앞단을 철저히 분리해서 처리하는 방법]
  뷰 단처리할 필요 없으므로 앞단에 보낼 때 json등의 형태로 보낸다.
  url을 명령어처럼 처리.

- json(javascript object notation 자바스크립트 객체 표기법) 사용

- {}: 객체 1개
  {“멤버변수 이름”: value} *String의 경우 반드시 더블코트 사용.
  ex. {“flag”:flag}
- []: 배열
  클라이언트에게 보내야하는 객체가 1개 이상일 경우
  ex. [{“id”:”aaa”,”pwd”:”111”,”name”:”yeji”},{“id”:”aaa”,”pwd”:”111”,”name”:”yeji”}]

1. project :: app4_t (-ing)

1) 회원가입 패키지

- form.jsp
  *cf. span과 div의 차이: span은 바로 옆에 붙여주고, div는 공간을 띄워준다.

```jsp
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR" import="com.example.app2.model.join.MemberType"%>
<!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>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    var id = $("#id").val();
    //ajax 요청
    $.post("/member/idCheck",
    {
      id: id
    },
    function(data,status){//응답이 오면 실행되는 함수.
        if(status=="success"){
            var html ="중복된 아이디";
            var res = $.parseJSON(data);//서버에서 무엇을 보내느냐에 따라 배열/객체가 될 수 있음
            if(res.flag==true){
                html="사용가능한 아이디";
            }
            $("#idCheck").html(html)
        }
//        alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
```

```jsp
</head>
<body>
<h3>join form</h3>S
<form action="/member/join" method="post">
id:<input id="id" type="text" name="id">
<input type="button" id="btn1" value="id check"><span id="idCheck"></span><br>
pwd:<input type="text" name="pwd"><br>
name:<input type="text" name="name"><br>
email:<input type="text" name="email"><br>
member type:<input type="radio" value="${MemberType.CONSUMER}" checked>CONSUMER
<input type="radio" value="${MemberType.SELLER}">SELLER<br/>
<input type="submit" value="join"><br>
</form>
</body>
</html>
```

join form

id: aaa [id check] 사용가능한 아이디
pwd:
name:
email:
member type: ◉ CONSUMER ○ SELLER
[join]

- MemberController.java
  - login(get,post)
  - logout

```java
@Controller
public class MemberController {
    @Autowired
    private Shop_MemberService service;

    @GetMapping("/member")
    public String form(){
        return "member/form";
    }

    @PostMapping("/member")
    public String join(Shop_Member sm){
        service.saveMember(sm);
        return "redirect:/member/login";
    }

    @GetMapping("/member/login")
    public void loginForm(){}

    @PostMapping("/member/login")
    public String login(Shop_Member sm, HttpServletRequest req){
        HttpSession session = req.getSession();
        Shop_Member a = service.getMember(sm.getId());
        boolean flag = false;
        String pwd = "";
        try{
            pwd = a.getPwd();
            if(pwd.equals(sm.getPwd())){
                flag=true;
            }else{
                System.out.println("패스워드 불일치");
            }
        }catch(EntityNotFoundException e){
            flag= true;
            System.out.println("아이디 불일치");
        }
```

```java
        String result="redirect:/member/login";
        if(flag){
            session.setAttribute("id", sm.getId());
            //구매자=>전체 상품 리스트 페이지
            //판매자=>내가 등록한 상품 페이지
            if(a.getType()==MemberType.CONSUMER){
                System.out.println("구매자로 로그인");
                result="redirect:/product/list";
            }else if(a.getType()==MemberType.SELLER){
                System.out.println("판매자로 로그인");
                result="redirect:/product/listbyseller/"+sm.getId();
            }
        }
        return result;
    }

    @RequestMapping("/member/logout")
    public String logout(HttpServletRequest req){
        HttpSession session = req.getSession(false);
        session.removeAttribute("id");
        session.invalidate();
        return "redirect:/member/login";
    }

    @RequestMapping("/member/idCheck")
    public String idCheck(@RequestParam("id") String id, Model m){
        System.out.println("id:"+id);
        Shop_Member a = service.getMember(id);
        boolean flag = false;
        try{
            a.getPwd();
        }catch (EntityNotFoundException e){
            flag=true;
        }
        m.addAttribute("flag", flag);
        return "member/idcheck";
    }
```

```java
    @GetMapping("/product/listbyseller/{seller}")
    public String listBySeller(@PathVariable("seller") String seller, Model m){
        Shop_Member sm = service.getMember(seller);
        List<Product> list = sm.getProducts();
        m.addAttribute("list", list);
        return "product/list";
    }

}
```

#listBtSeller함수는 Shop_MemberService를 사용하는 기능이기 때문에, MemberComtroller에 배치?

- login.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!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>Insert title here</title>
</head>
<body>
<h3>login form</h3>
<form action="/member/login" method="post">
id:<input type="text" name="id"><br/>
pwd:<input type="text" name="pwd"><br/>
<input type="submit" value="login">
</form>
<a href="/member">회원가입</a>
</body>
</html>
```

2) Product 패키지

- model
  - Product.java(dto) 생성

```java
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "product_sequence")
    @SequenceGenerator(name = "product_sequence", sequenceName = "seq_product")
    private Integer num;//클래스 타입으로 정의.

    @OneToMany(fetch=FetchType.LAZY, mappedBy="pnum",cascade=CascadeType.REMOVE)
    List<MyOrder> orders;

    @ManyToOne
    @JoinColumn(name="seller", nullable = false)//name=해당 테이블과 매핑할 현재클래스의 변수명
    private Shop_Member seller;

    private String name;
    private Integer price;
    private Integer amount;
    private String info;
    private String img;
```

*Shop_Member.java(dto) 수정

```java
    private String id;//pk 설정

    private String pwd;
    private String name;
    private String email;

    @Enumerated(EnumType.STRING)
    private MemberType type;

    @OneToMany(fetch=FetchType.LAZY, mappedBy="seller",cascade=CascadeType.REMOVE)
    private List<Product> products;
```

  - ProductFile.java 생성

```java
public class ProductFile {
    private Product p;
    private MultipartFile f;

    public ProductFile() {
    }
    public ProductFile(Product p, MultipartFile f) {
        super();
        this.p = p;
        this.f = f;
    }
```

  - Product_Repos

Product_Repo... | Product.java | MyOrder_Repo... | MyOrderServi... | MemberContro... | login.jsp

```java
package com.example.app2.model.product;

import java.util.List;

public interface Product_Repos extends JpaRepository<Product, Integer> {
    List<Product> findByNameLike(String name);
//  List<Product> findBySeller(Shop_Member seller);
}
```

- Product_Service(여기서 기능 모두 구현하고, controller에서는 함수만 호출하는 식으로) #save 반환값?

```java
@Service
public class Product_Service {
    @Autowired
    private Product_Repos rep;
    private String path = "C:\\Users\\Playdata\\Desktop\\workspace\\.metadata\\.plugins\\org.eclipse.wst.server.core\\tmp0\\webapps\\img\\";

    public void addProduct(ProductFile pf){//이미지명:제품 번호.jpg, 업로드 위치(db에 img값 저장형태:/img/이미지명)
        //이미지 파일 경로를 제외한 모든 상품정보 db에 저장
        Product p = rep.save(pf.getP());

        //이미지 파일 저장
        String fname = pf.getF().getOriginalFilename();
        System.out.println("fname:"+fname);
        String f_end = fname.substring(fname.lastIndexOf("."));
        System.out.println("f_end:"+f_end);
        String name = p.getNum()+f_end;
        System.out.println("name:"+name);
        File f = new File(path+name);
        try {
            pf.getF().transferTo(f);
        } catch (IllegalStateException | IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        //db에 이미지 파일 경로 세팅
        p.setImg("/img/"+name);
        editProduct(p);
    }
    public void editProduct(Product p){
        rep.save(p);
    }
    public void delProduct(int num){
        rep.deleteById(num);
    }
    public Product getByNum(int num){
        return rep.getOne(num);
    }
```

```java
    public List<Product> getAll(){
        return rep.findAll();
    }
    public List<Product> getByName(String name){
        return rep.findByNameLike("%"+name+"%");
    }
//  public List<Product> getBySeller(Shop_Member seller){
//      return rep.findBySeller(seller);
//  }
}
```

- ProductController.java

```java
@Controller
public class ProductController {
    @Autowired
    private Product_Service service;

    @GetMapping("/product")
    public String form(){
        return "product/form";
    }

    @PostMapping("/product")
    public String add(ProductFile pf){
        service.addProduct(pf);
        String seller = pf.getP().getSeller().getId();
        return "redirect:/product/listbyseller/"+seller;
    }

    @GetMapping("/product/details/{num}")
    public String details(@PathVariable("num")int num,Model m){
        Product p = service.getByNum(num);
        m.addAttribute("p", p);
        return "product/details";
    }
    //구매자: 전체 상품 페이지
    @RequestMapping("/product/list")
    public String list(Model m){
        List<Product> list = service.getAll();
        m.addAttribute("list",list);
        return "product/allproductlist";
    }
}
```

*MemberController.java 수정(판매자 로그인시 아이디 함께 보내기)

MemberController.java | form.jsp | DemoWebController.java | ProductController.java

```
34
35      @GetMapping("/member/login")
36      public void loginForm(){}
37
38      @PostMapping("/member/login")
39      public String login(Shop_Member sm, HttpServletRequest req){
40          HttpSession session = req.getSession();
41          Shop_Member a = service.getMember(sm.getId());
42          boolean flag = false;
43          String pwd = "";
44          try{
45              pwd = a.getPwd();
46              if(pwd.equals(sm.getPwd())){
47                  flag=true;
48              }else{
49                  System.out.println("패스워드 불일치");
50              }
51          }catch(EntityNotFoundException e){
52              flag= true;
53              System.out.println("아이디 불일치");
54          }
55          String result="redirect:/member/login";
56          if(flag){
57              session.setAttribute("id", sm.getId());
58              //구매자=>전체 상품 리스트 페이지
59              //판매자=>내가 등록한 상품 페이지
60              if(a.getType()==MemberType.CONSUMER){
61                  result="redirect:/product/list";
62              }else if(a.getType()==MemberType.SELLER){
63                  result="redirect:/product/listbyseller/"+sm.getId();
64              }
65          }
66          return result;
67      }
```

<숙제 :: Order 패키지에 구매자 기능 추가하기>

- Product 패키지
  seller 로그인 => 내 상품 목록 => 제품명 클릭=> 상세페이지
  =>수정(이름,가격,수량)/삭제
- Order 패키지
  <VO>
  Order
  주문번호 num
  주문한 제품번호 FK(product(num)) pnum
  *해당 제품의 주문들 리스트
  주문수량 amount
  결제금액 total_price
  주문타입(즉시결제/장바구니) total_price
  주문자id FK(shop_member(id)) con_id
  *주문한 사람 id

  기능
  주문자 로그인=>전체상품목록=>상품명 클릭=>주문페이지(수량입력,
  결제금액)=>즉시결제/장바구니
  즉시결제=>내 즉시결제 내역 목록으로 이동
  장바구니=>내 장바구니 목록으로 이동

  전체상품목록
  : 이름으로 검색/판매자로 검색 메뉴 존재

1) MyOrder(dto)
   - 오류

- *매핑에러: pnum과 매핑할 Product 리스트
- *dto이름 Order => My Order로

MyOrder.java

```java

@Entity
public class MyOrder {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "my_order_sequence")
    @SequenceGenerator(name = "my_order_sequence", sequenceName = "seq_my_order")
    private Integer num;

    @ManyToOne
    @JoinColumn(name = "pnum", nullable = false)
    private Product pnum;
    private Integer order_num;
    private Integer total_price;

    @Enumerated(EnumType.STRING)
    private OrderType type;

    @ManyToOne
    @JoinColumn(name = "con_id", nullable = false)
    private Shop_Member con_id;

    public MyOrder() {
```

2) MyOrder_Repos

MyOrder.java | MyOrderRepository.java

```java
package com.example.app2.model.order;

import org.springframework.data.jpa.repository.JpaRepository;

public interface MyOrderRepository extends JpaRepository<MyOrder, Integer> {

}
```

3) MyOrder_service

MyOrder.java | MyOrderRepository.java | MyOrderService.java

```java
package com.example.app2.model.order;

import org.springframework.beans.factory.annotation.Autowired;

@Service
public class MyOrderService {

    @Autowired
    private MyOrderRepository rep;

    public void addOrder(MyOrder o){
        rep.save(o);
    }

}
```

MyOrder.java | MyOrderRepository.java | MyOrderService.java | OrderController.java

```java
package com.example.app2.controllers;

import org.springframework.beans.factory.annotation.Autowired;

@Controller
public class OrderController {
    @Autowired
    private MyOrderService service;
    @GetMapping("/order")
    public String form(){
        return "order/form";
    }

    @PostMapping("/order")
    public String order(MyOrder o){
        service.addOrder(o);
        System.out.println(o.getCon_id().getOrders());
        return "order/form";
    }
}
```

4) (여기부터) 구매자가 볼 전체 상품 리스트
5) 상품명 클릭하면 주문 페이지

- tostring에서 fk키 빼기 => stackoverflow