

# 개발자 매뉴얼

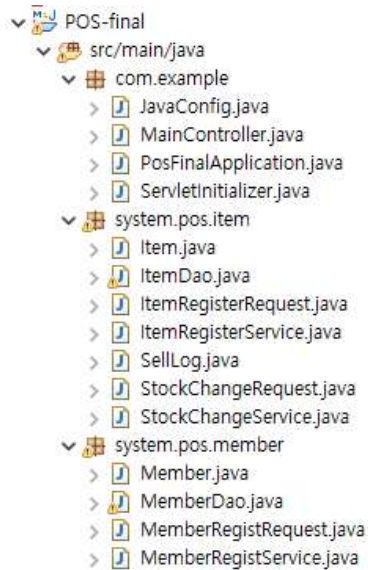
소프트웨어학과

2016301078

최원석

1. java파일.....3쪽	6. login.....8쪽
- com.example	- login
- system.pos.item	- loginchek
- system.pos.member	7. member.....8쪽
2. jsp파일.....4,5쪽	- delete
- login	- finishDelete
- member	- errorPage
- menu	- regist
- stat	- finishRegist
- stock	- view
3. com.example.....5,6쪽	8. menu.....8,9쪽
- JavaConfig	- stockMenu
- MainController	- statMenu
4. system.pos.item.....6,7쪽	- memberMenu
- Item	- sell
- ItemDao	- addSell
- ItemRegisterRequst	- finishSell
- ItemRegisterService	- errorPage
- SellLog	9. stat.....9쪽
- StockChangeRequest	- all
- StockCahngeService	- earning5
5. system.pos.member.....7쪽	10. stock.....9쪽
- Member	- change/ finishChange.
- MemberDao	- delete/finishDelete
- MemberRegistRequest	- receive/finishReceive
- MemberRegistService	- regist/finishRegist
	- search/finishSearch
	- view
	- errorpage
	11. pos_main.....9쪽

# 1. java 파일



java 파일은 위와 같다.

## com.example

- JavaConfig: Bean과 Datasource를 정의한다.
- MainController: jsp 페이지를 불러올 때 처리할 내용들은 담고 있다.

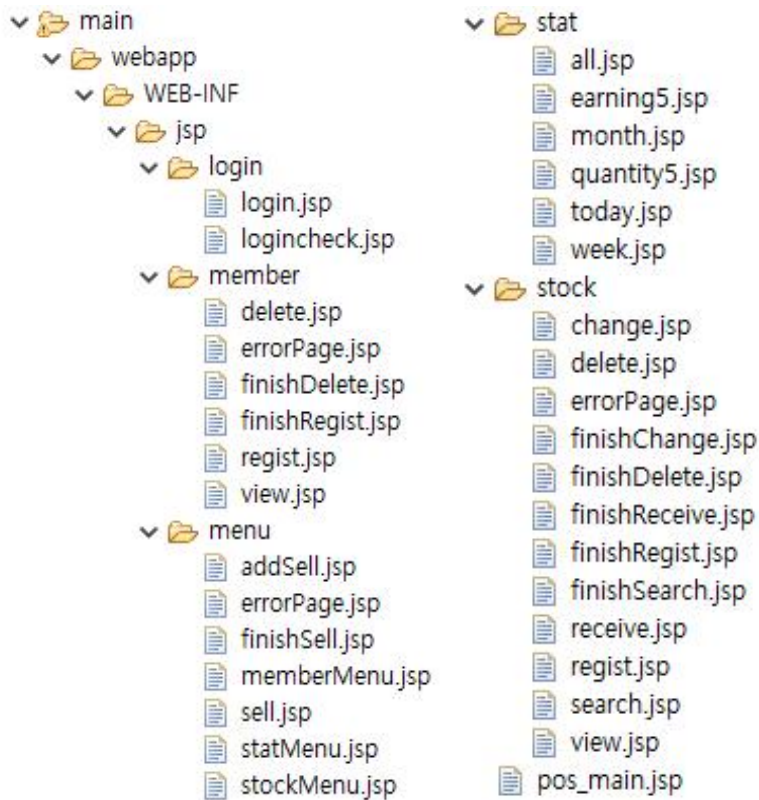
## system.pos.item

- Item: 상품의 정보(상품 코드, 상품 명, 가격, 재고, 입고 날짜)를 저장할 클래스. 생성자와 getter를 가진다
- ItemDao: 데이터베이스에 정보를 저장하거나 읽어오는 파일이다.
- ItemRegisterRequest: 상품 등록을 위해 상품 정보를 일시적으로 저장하는 클래스  
getter와 setter를 가진다.
- ItemRegisterService: itemRegisterRequest객체에 저장된 정보를 처리하는 클래스
- SellLog: 판매 아이템의 정보를 일시적으로 저장하는 클래스
- StockChangeRequest: 재고가 변경될 때 변경될 상품 정보를 일시적으로 저장할 클래스  
getter와 setter를 가진다.
- StockChangeService: StockChangeRequest에 저장된 정보를 처리하는 클래스.

## System.pos.member

- Member: 멤버의 정보( 직급, 이름, id, password, 등록일)을 저장할 클래스. 생성자와 getter을 가진다.
- MemberDao: 데이터베이스에서 Member의 정보를 읽어오거나 저장하는 클래스
- MemberRegistRequest: 멤버 등록을 위해 정보를 일시적으로 저장하는 클래스. getter와 setter를 가진다.
- MemberRegistService: MeberRegistRequest에 저장된 정보를 처리하는 클래스

## 2. jsp 파일



jsp 파일은 위와 같다.

- pos\_main.jsp: 작업을 선택할 메인페이지

### login

- login.jsp: 로그인 페이지
- logincheck.jsp: 로그인시 아이디 또는 비밀번호가 잘못된 경우 출력하는 페이지

### member

- delete.jsp: 멤버를 삭제하기 위한 페이지
- finishDelete.jsp: 멤버 삭제가 완료된 후 출력하는 페이지
- regist.jsp: 멤버를 등록하기 위한 페이지
- finishRegist.jsp: 멤버 등록이 완료된 후 출력하는 페이지
- view.jsp: 전체 멤버를 보여주는 페이지
- errorPage.jsp: 멤버 삭제시 없는 아이디를 입력할 경우 출력되는 페이지

### menu

- sell.jsp: 제품 판매를 위한 페이지
- addSell.jsp: sell페이지에서 제품을 추가할 때, 추가된 제품을 출력해 보여주는 페이지
- finishSell.jsp: sell 페이지에서 구매완료 버튼을 누르면 DB처리 후 나타나는 페이지
- errorPage.jsp: sell에서 없는 제품 코드 입력시 나타나는 페이지
- stockMenu.jsp: 재고 관리 메뉴 페이지(입고, 제품 등록/삭제/확인, 재고 수정/검색)
- statMenu.jsp: 통계 메뉴 페이지(전체 매출, 당일/일주일/한달 매출, 판매개수 Top5, 판매액 Top5)
- memberMenu: 멤버 메뉴 페이지 (멤버 확인, 멤버 등록, 멤버 삭제)

## stat

- all.jsp: 현재까지의 모든 판매 목록을 보여주는 페이지
- today.jsp: 당일 매출을 보여주는 페이지
- week.jsp: 당일로부터 일주일간의 매출을 보여주는 페이지
- month.jsp: 당일로부터 한달간의 매출을 보여주는 페이지
- quantity5.jsp: 가장 많은 수량이 판매된 제품 5개를 보여주는 페이지
- earning5.jsp: 가장 많은 판매 금액을 기록한 제품 5개를 보여주는 페이지

## stock

- receive.jsp: 입고를 위한 페이지
- finishReceive.jsp: 입고 처리와 완료된 후 나타나는 페이지
- regist.jsp: 제품 등록을 위한 페이지
- finishRegist.jsp: 제품 등록이 완료된 후 나타나는 페이지
- delete.jsp: 제품 삭제를 위한 페이지
- finishDelete.jsp: 제품 삭제가 완료된 후 나타나는 페이지
- view.jsp: 현재 등록되어있는 모든 제품을 보여주는 페이지
- change.jsp: 제품의 재고를 변경하기 위한 페이지
- finishChange.jsp: 재고 변경이 완료된 후 나타나는 페이지
- search.jsp: 제품 검색을 위한 페이지
- finishSearch.jsp: 제품 검색이 완료되면 찾는 제품 정보를 출력해주는 페이지
- errorPage.jsp: 입고, 제품 삭제, 제품 검색 시 없는 제품 코드를 입력하면 나타나는 페이지

## 3. com.example

### 1) JavaConfig

- **public DataSource dataSource()** : DataSource 정의 및 초기화  
POS 데이터 베이스 연동, 계정은 root 비밀번호는 1234이다.
- @Bean 등록

### 2) MainController

- 각 페이지 호출시 필요한 작업 진행.
- **public String login()**: 첫 페이지이자 로그인 페이지
- **public String logincheck()**: 입력받은 id와 password를 통해 해당 멤버 정보가 일치하면 pos\_main화면 출력, 그렇지 않을 경우 로그인 확인 창 출력
- **public String addSell()**: sell 페이지에서 받은 code와 count값으로 itemDao를 통해 품목 정보를 가져오고 품목 정보가 없으면 errorPage 출력, 있을 경우 sellList와 countList에 각각 저장
- **public String finishSell()**: sellList에 저장된 코드들을 통해 해당 상품 정보를 가져와서 StockChangeRequest에 정보를 저장해 StockChangeService를 통해 재고를 수정하고 ItemRegisterRequest에 정보를 저장해 ItemRegisetService를 통해 판매제품을 DB에 저장한다.
- **public String stockReveiveFinish()**: receive페이지에서 입력받은 code와 stock을 통해 재고 수정 재고 수정이 완료되면 finishReceive페이지를 출력한다.  
등록 되어있지 않은 코드 입력시 errorPage를 출력
- **public String stockDeleteFinish()**: delete 페이지에서 입력받은 code를 통해 제품을 삭제 재고 삭제가 완료되면 finishDelete페이지를 출력한다.  
등록 되어있지 않은 코드 입력시 errorPage를 출력
- **public String stockRegistFinish()**: regist 페이지에서 입력받은 codr, name, price를 req에 임시 저장 후 itemRegSvc의 regist함수를 통해 해당 정보를 DB에 저장.

- **public String stockChangefinish():** change페이지에서 입력받은 code와 stock를 itemDao를 통해 해당 상품 정보를 가져와 req에 저장하고 stockChangeSvc의 change 함수를 통해 재고를 변경한다. 없는 제품 코드를 입력한 경우 errorPage를 출력한다.
- **public String statAll():** itemDao를 통해 DB에 저장된 모든 판매 기록을 가져와 보여준다. statToday(), statWeek(), statMonth() 모두 동일.
- **public String statQuantity():** itemDao를 통해 모든 판매 기록을 가져와 가장 많이 판매된 5개의 제품 제품 코드를 통해 각각의 리스트에 이름, 코드, 가격, 개수를 저장한다. statEarning()도 동일하다.
- **public String memberList():** memberDao를 통해 저장된 모든 멤버를 출력한다.
- **public String memberRegistFinish():** regist페이지에서 입력받은 rank, name, id, password를 regReq에 저장하고 memRegSvc의 regist함수를 통해 해당 정보를 DB에 등록한다.
- **public String finishDelete():** delete 페이지에서 입력받은 id를 regReq에 저장 후 memRegSvc의 delete 함수를 통해 해당 id를 가진 정보를 DB에서 삭제

## 4. system.pos.item

### 1) Item

- 상품 정보를 저장하기 위한 클래스
- 멤버 변수로 상품 코드(code), 상품 명(name), 상품가격(price), 재고(stock), 입고 날짜(receiveDate)
- 생성자와 각 멤버 변수의 getter를 포함한다.

### 2) ItemDao

- 생성자에서 dataSource를 인자로 받아 JdbcTemplate에 연동
- selectAll(): POS DB의 items테이블에서 모든 목록을 가져와 리턴한다.
- receive(): 매개변수로 받은 item을 items 테이블에 update한다. 재고와 입고 날짜가 변경된다.
- receiveLog(): 입고받은 제품을 기록하는 함수. receiveLog 테이블에 상품 코드와 입고된 제품 개수, 입고 날짜를 기록한다.
- change(): 재고 수정을 위한 함수, 매개변수로 받은 item을 items테이블에서 같은 코드정보를 가지는 제품의 재고(stock)와 입고날짜를 변경한다.
- insert(): 재고 등록을 위한 함수. 매개변수로 받은 item을 items테이블에 기록
- insertSoldItem(): 판매된 제품을 기록하기 위한 함수. 매개변수로 받은 item과 count를 sellLog 테이블에 code, count, 총가격, 판매 날짜를 저장
- selectByCode(): items 테이블에서 매개변수로 받은 code와 일치하는 code를 가져와 해당 item을 반환
- delete(): DB에서 매개변수로 받은 item의 code와 일치하는 물품을 제거한다.
- sell(): 판매 후 재고 수정을 위한 함수. 매개변수로 받은 item의 코드와 일치하는 상품의 stock를 수정한다
- selectTodaySellLog(): 당일 판매된 상품 정보들을 가져오는 함수. sellLog의 테이블에서 판매 날짜를 통해 당일이면 그 정보들을 반환한다. selectWeekSellLog(), selectMonthSellLog()도 각각 1주일, 한달로 기능은 동일하다.

### 3) ItemRegisterRequest

item등록을 위한 정보를 저장하는 클래스  
item과 동일한 멤버 변수를 가지고 setter와 getter를 가진다.

### 4) ItemRegisterService

- item등록을 처리하는 클래스
- regist(): 매개변수로 받은 ItemRegisterRequest를 통해 item을 만들고 itemDao의 insert함수의 매개변수로 넘겨준다.
  - insertSoldItem(): 마찬가지로 ItemRegisterRequest를 통해 item객체를 새로 만들고 그 item객체와 매개변수 count를 itemDao의 insertSoldItem의 매개변수로 넘겨준다.

- delete(): ItemRegisterRequest를 통해 item객체를 새로 만들고 temDao의 delete의 매개변수로 넘겨준다.

#### 5) SellLog

판매 정보를 저장하기 위한 클래스

멤버 변수로 상품코드(code), 상품개수(count), 총가격(totalPrice), 판매날짜(saleDate), 번호(num)을 가진다.  
멤버변수들의 getter를 가진다.

#### 6) StockChangeRequest

재고 변경을 위해 정보를 저장하기 위한 클래스

멤버 변수로 상품코드(code), 재고(stock), 입고날짜(receiveDate)를 가진다.  
각 변수의 getter/setter를 가진다.

#### 7) StockChangeService

- change(): 재고 변경을 위한 함수. itemDao에서 매개변수로 받은 정보를 통해 같은 코드를 가지는 상품 정보를 가져오고 재고를 변경하여 itemDao의 change함수 매개변수로 준다.
- receive(): 입고를 위한 함수. itemDao에서 매개변수로 받은 정보를 통해 같은 코드를 가지는 상품 정보를 가져오고 입고이므로 현재 재고에 더하여 itemDao의 receive와 receiveLog함수의 매개변수에 각각 준다.
- sell(): 판매를 위한 함수. itemDao에서 매개변수로 받은 정보를 통해 같은 코드를 가지는 상품 정보를 가져오고 현재 재고에서 count만큼 차감 시킨후 itemDao의 sell함수에 매개변수로 item객체를 준다.

## 5. system.pos.member

#### 1) Member

멤버의 정보를 저장하는 클래스

멤버변수로 직급(rank), 이름(name), id, password, 등록날짜(regDate)를 가진다.  
멤버변수들의 getter를 가진다.

#### 2) MemberDao

생성자에서 jdbcTemplate를 통해 DB와 연동한다

- selectAll(): member 테이블에서 모든 정보를 가져와 반환한다.
- insert(): 매개변수로 받은 member객체를 member테이블에 저장한다.
- selectById(): DB에서 매개변수로 받은 id와 동일한 id를 가지는 멤버 정보를 가져와 반환한다.
- delete(): 매개변수로 받은 member의 id와 동일한 id를 가지는 정보를 DB에서 삭제한다.

#### 3) memberRegistRequest

멤버 등록을 위해 정보를 저장하는 클래스

멤버 변수는 Member와 동일하다  
멤버 변수의 getter/setter를 가진다.

#### 4) memberRegistService

멤버 등록을 하기 위한 클래스

- regist(): 매개변수로 받은 memberRegistRequest를 통해 새로운 member객체를 만들고 date를 추가하여 memberDao의 insert함수의 매개변수로 넘겨준다.
- delete(): 매개변수로 받은 memberRegistRequest를 통해 같은 id를 가지는 member를 MemberDao를 통해 불러오고 그 멤버를 memberDao의 delete함수 매개변수로 넘겨준다.

## 6. login

### 1) login

session을 초기화하고

로그인을 위한 id와 password를 입력받을 blank를 보여준다.

로그인을 누르면 해당 정보는 loginCheck로 넘어간다.

### 2) logincheck

login에서 넘겨받은 id가 null이면 잘못 입력되었다고 보여주며 이전 login 창으로 돌아가는 버튼이 존재한다.

## 7. member

### 1) delete

멤버를 삭제하기 위한 페이지

삭제할 멤버 아이디를 입력할 칸이 존재하고, 삭제 버튼을 누르면 해당 정보가 finishDelete로 넘어간다

### 2) finishDelete

id정보를 가져와 성공적으로 삭제되면 해당 페이지가 출력된다.

멤버메뉴로 돌아가거나 메인화면으로 돌아가는 버튼이 존재한다.

### 3) errorPage

delete에서 없는 정보를 입력하면 해당 페이지가 출력된다.

멤버 메뉴로 돌아가는 버튼이 존재한다.

### 4) regist

멤버 등록을 위한 페이지

멤버 등록을 위해 직급, 이름, id, password를 입력할 빈 칸이 존재하며, 등록 버튼을 누르면 해당 정보가 finishRegist로 넘어간다

### 5) finishRegist

멤버 등록이 성공적으로 완료되면 해당 페이지가 출력된다.

멤버메뉴로 돌아가거나 메인화면으로 돌아가는 버튼이 존재한다.

### 6) view

등록된 모든 정보를 테이블에 직급, 이름, 아이디, 비밀번호, 등록날짜 순으로 출력하여 보여준다.

## 8. menu

### 1) stockMenu

재고 관리 메뉴이다.

입고, 제품등록, 제품 삭제, 재고확인, 재고수정, 재고 검색을 위한 페이지고 넘어가는 버튼이 존재한다.

### 2) statMenu

통계 메뉴이다.

전체 매출, 당일 매출, 일주일 매출, 한달 매출, 판매개수 top5, 판매액 top5 페이지로 넘어가는 버튼이 있다.

### 3) memberMenu

멤버 관리 메뉴이다.

매니저와 점장만 접근 가능하도록 했으며, 직원은 경고창으로 접근 불가능하다고 보여준다.

계정확인, 계정등록, 계정 삭제를 위한 페이지로 넘어가는 버튼이 존재한다.

### 4) sell

제품 판매를 위한 페이지.

전체 물품의 금액합을 위해 totalSum변수를 만들고 session에 저장한다.

제품 판매를 위해 code와 제품 개수 count를 입력받을 빈칸이 존재하며 추가 버튼은 누르면 해당 제품이 추가된다.

### 5) addSell

sell에서 받은 정보를 출력해서 보여주며, 이어서 물품을 더 추가할 수 있고 구입 완료를 할 수 있다.

### 6) finishSell



앞에서 추가된 모든 제품들이 정상적으로 DB에 처리가 되면 해당 페이지가 나타난다.

#### 7) errorPage

sell/addSell 페이지에서 없는 제품 코드를 입력받으면 해당 페이지가 나타나며 잘못된 상품 정보라고 알려주고 이전으로 돌아가는 버튼이 있다.

## 9. stat

#### 1)all

테이블의 형태로 sellList에 저장된 모든 상품 정보를 가져와 보여준다.  
today, week, month 모두 동일하다.

#### 2) earning5

각 상품 정보를 request로 받아 저장하고  
각각 테이블 형태로 정보들을 표시한다.  
quantity5도 동일하다.

## 10. stock

#### 1) change/ finishChange

직원은 접근 불가능한 페이지이다.  
재고 수정을 위해 코드와 제품 개수를 입력받을 빈칸이 존재하고 재고 수정 버튼을 누르면 finishChage 페이지로 넘어간다.  
정상적으로 완료되면 finishChange 페이지가 출력된다.

#### 2) delete/finishDelete

마찬가지로 직원은 접근 불가능한 페이지  
제품 삭제를 위해 코드를 입력받을 빈칸이 있으며 삭제 버튼을 누르면 해당 정보는 finishDelete로 넘어간다.  
정상적으로 삭제되면 finishDelete페이지가 나타난다.

#### 3) receive/finishReceive

입고를 위한 페이지  
입고를 위해 제품 코드와 제품 개수를 입력받는 빈칸이 존재.  
입고 완료 버튼을 클릭하면 해당 정보는 finishReceive로 넘어간다.  
정상적으로 입고가 완료되면 finishReceive페이지가 나타나게 된다.

#### 4) regist/finishRegist

제품 등록을 위한 페이지, 직원은 접근 불가능하다.  
제품 등록을 위해 코드, 제품명, 가격을 입력할 빈 칸이 존재하면 제품 등록 버튼을 누르면  
해당 정보들은 finishRegist페이지로 넘어간다.  
정상적으로 제품 등록이 완료되면 finishRegist페이지가 나타난다.

#### 5) search/finishSearch

제품 검색을 위한 페이지이며 검색을 위한 제품 코드를 입력할 빈칸이 존재한다.  
검색 버튼을 클릭하면 해당 정보는 finishSearch로 넘어가게 되며 정상적으로 검색이 완료되면  
finishSearch페이지가 나타난다.

#### 6) view

전체 제품을 테이블 형태로 보여준다

#### 7) errorpage

change, delete, receive, search에서 잘못된 상품 정보를 입력하게 될 경우 해당 페이지가 나타나며  
이전으로 돌아갈 수 있다.

## 11. pos\_main

메인화면으로 판매, 재고관리, 통계, 회원관리 버튼이 존재한다.

접근 권한을 설정하고, 누가 로그인 했는지 보여주기위해 session에 해당 member정보를 저장한다.