

전자정부 표준프레임워크

종합실습 easycompany

eGovFrame



Contents



실행환경 종합실습(Easycompany) 목차

1. _로그인 처리 (LAB4-1)
2. _사원목록 조회, ajax(autocomplete) (LAB4-2)
3. _사원정보 등록화면 이동 (LAB4-3)
4. _사원정보 등록화면, 부서 멀티콤보 처리 (LAB4-4)
5. _사원정보 등록, 상세조회 (LAB4-5)
6. _사원정보 수정 (LAB4-6)

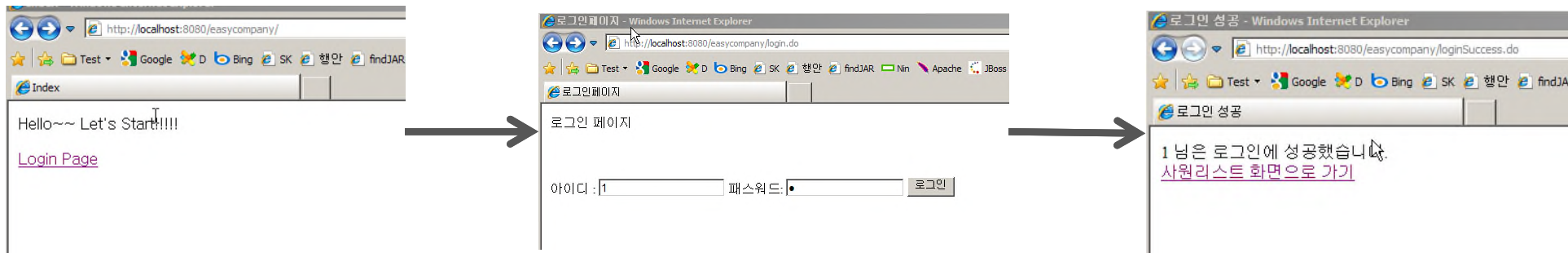
LAB 4-1 로그인 처리

Step 4-1-01. LoginController.java 에서 LoginService 를 이용하여 로그인 여부 체크 Account 객체를 리턴 받는다.

Step 4-1-02. LoginServiceImpl.java 에서 구현되어져 있는 LoginMapper의 authenticate 메소드를 이용하여 검증하여 결과 (Account)객체를 리턴한다.

Step 4-1-03. LoginController.java 에서 가져온 account 객체를 session에 "UserAccount"라는 이름으로 저장한다.

Step 4-1-04. LoginController.java 에서 loginSuccess.do 로 호출된 처리를 위한 메소드를 만든다.(GET 방식만 지원)



LAB 4-2 사원목록 조회, ajax(autoComplete)

Step 4-2-01. EmployeeController.java 에서 employeeList.do 경로로 요청과 메소드를 매핑해준다.

@RequestMapping 을 이용한다.

Step 4-2-02. EmployeeController.java 에서 화면에서 넘어오는 pageNo 파라미터 값을 반드시 넘어오지 않아도 됨을 명시적 표현해 본다. @RequestParam 의 required 를 이용한다.

Step 4-2-03. employeeelist.jsp에서 ajax사용을 위한 jquery autocomplete 코드를 생성한다.

The image shows two screenshots of a web application. The left screenshot shows a login success message: "로그인 성공" (Login Success) and a message "1 님은 로그인에 성공했습니다." (User 1 has successfully logged in). Below this is a link "사원리스트 화면으로 가기" (Go to Employee List Screen). An arrow points from this link to the right screenshot. The right screenshot shows the "사원 정보 리스트 페이지" (Employee Information List Page). It contains a search form with fields for "사원번호" (Employee Number), "부서번호" (Department Number), and "이름" (Name), and a "검색" (Search) button. Below the search form is a table with the following data:

사원번호	부서번호	이름	나이	이메일
1	1200	김길동	28	kkd@easycompany.com
2	1100	김길수	39	kks@easycompany.com
3	1200	강감찬	17	kko@easycompany.com

Below the table is a link "사원추가" (Add Employee) and a pagination bar showing "1 2 3 4 5 6 7 8".

LAB 4-3

사원정보 등록화면 이동

Step 4-3-01. EmployeeController.java 에서 요청되는 insertEmployee.do 와 메소드를 매핑한다.

단 GET 방식에 대해서만 처리하도록 한다.

Step 4-3-02. EmployeeController.java 에서 employee 객체(이름:employee)를 ModelAttribute 를 이용하여 세팅하여 보자.

Step 4-3-03. EmployeeController.java 에서 deptInfoOneDepthCategory 객체(이름동일)를 ModelAttribute 를 이용하여 세팅하여보자.

호출되는 서비스는 departmentService 서비스의 getDepartmentIdNameList 메소드(디폴트로 "1" 을 스트링 형태로 넘긴다.)이며
get반환값은 Map<String,String> 타입이다.

사원 정보 리스트 페이지

사원정보 부서정보

사원번호: 부서번호: 이름:

사원번호	부서번호	이름	나이	이메일
1	1200	김길동	28	kkd@easycompany.com
2	1100	김길수	39	kks@easycompany.com
3	1200	강감찬	17	kkc@easycompany.com

[사원 추가](#)

1 2 3 4 5 6 7 8 9 10

사원 정보 입력 페이지

사원정보 부서정보

사원번호	<input type="text"/>
부서번호	<input type="text" value="상위부서를 선택하세요."/> <input type="text" value="근무부서를 선택하세요."/>
이름	<input type="text"/>
비밀번호	<input type="text"/>
주민번호	<input type="text"/> - <input type="text"/>
나이	<input type="text" value="0"/>
이메일	<input type="text"/>

LAB 4-4 사원정보 등록화면, 부서 멀티콤보 처리

Step 4-4-01. EmployeeController.java 에서 요청되는 insertEmployee.do 와 메소드를 매핑한다.

단 POST 방식에 대해서만 처리하도록 한다. 리턴타입은 String 이다.

Step 4-4-02. EmployeeController.java 에서 DefaultBeanValidator 의 validate 메소드를 이용하여 employee 객체 값을 체크한다.

Step 4-4-03. EmployeeController.java 에서 DefaultBeanValidator 를 실행한 결과 에러가 있을 경우
addemployee.jsp 원래 페이지를 다시 보여주고 에러 메시지를 뿌려준다.

```
if (bindingResult.hasErrors()) {  
    return "addemployee";  
}
```

Step 4-4-04. addemployee.jsp 에서 jquery 이용하여 ajax 방식으로 통신하여 근무부서를 조회한다.

LAB 4-5

사원정보 등록, 상세조회

Step 4-5-01. EmployeeServiceImpl.java 에서 insertEmployee 메소드는 Employee 를 파라미터로 받아 empMapper 를 실행한다.

(결과는 저장된 개수)

Step 4-5-02. EmpMapper.java 에서 insertEmployee 메소드는 EgovAbstractMapper에서 제공되는 insert() 메소드를 이용하여 com.easycompany.mapper.EmpMapper.insertEmployee ID를 가진 sql문을 실행한다.

Step 4-5-03. Employee.xml 파일에 insertEmployee 에 해당하는 sql문을 만들어준다. (DBIO 를 이용하여 생성하도록 한다.)

Insert 문의 id : `insertEmployee` 이고 parameterClass 는 `com.easycompany.service.Employee` 이고

SQL 문은 `insert into employee (employeeid, name, age, departmentid, email, password)`

`values (#{employeeid}, #{name}, #{age}, #{departmentid}, #{email}, #{password})`

The diagram illustrates the data flow from a web form to a summary table. On the left, a form titled '사원정보 부서정보' (Employee Information Department Information) contains the following fields: 사원번호 (Employee ID) with value 123456789, 부서번호 (Department ID) with a dropdown menu showing '금융사업부' (Financial Business Dept) and '증권대리점' (Securities Branch), 이름 (Name) with value '강한사람' (Strong Person), 비밀번호 (Password) with masked characters '****', 주민번호 (Resident ID) with a split input field, 나이 (Age) with value 1, and 이메일 (Email) with an empty field. Below the form are 'SAVE' and 'LIST' buttons. A large arrow points from the form to a table on the right. The table, titled '사원정보' (Employee Information), displays the entered data in a structured format. At the bottom of the table are '확인' (Check) and '재수정' (Re-edit) buttons.

사원번호	부서번호	이름	비밀번호	나이	이메일
123456789	5200	강한사람	1234	1	

LAB 4-6 사원정보 수정

Step 4-6-01. EmployeeController.java 에서 updateEmployee.do 요청 시 @RequestMapping 에 value 와 method (GET)를 만들어준다.

Step 4-6-02. EmployeeController.java 에서 getEmployeeInfo 메소드에 @RequestMapping을 이용하여 updateEmployee.do 요청 시, 파라미터로 “employeeid” key 가 넘어온다.

이것으로 EmployeeService 의 getEmployeeInfoById를 이용하여 Employee를 구해 리턴한다.

(참고: referenceDataOneDepthDept())

사원정보 부서정보

사원번호 : 부서번호 : 이름 :

	사원번호	부서번호	이름	나이	이메일
1	1200	김길동	28	kkd@easycom	
2	1100	김길수	39	kks@easycom	
3	1200	강감찬	17	kkc@easycom	

사원정보 부서정보

사원번호	1
부서번호	경영기획실 ▼점심메뉴기획팀 ▼
이름	김길동
비밀번호	• <input type="password"/>
주민번호	<input type="text"/> - <input type="text"/>
나이	28
이메일	kkd@easycompany.com

쿼리 전송 저장 리스트페이지

수고 하셨습니다. ^^*

