

**\*\* 정의 \*\***

인터페이스 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하고 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력이다.

**\*\* 수준 5 \*\***

**\* 학습시간 : 60시간 (2019.07.12 금 5교시 ~ 2019.07.23 화 8교시) \***

작성자 : 김 문 선

## [ 2001020212\_16v4 인터페이스 구현 ]

### [2001020212\_16v4.1 인터페이스 설계서 확인하기]

#### \*\*\* 수행준거

- 1.1 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다.
- 1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다.
- 1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.

### [2001020212\_16v4.2 인터페이스 기능 구현하기]

#### \*\*\* 수행준거

- 2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다.
- 2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다.
- 2.3 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의할 수 있다.
- 2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.

### [2001020212\_16v4.3 인터페이스 구현 검증하기]

#### \*\*\* 수행준거

- 3.1 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다.
- 3.2 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다.
- 3.3 인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> 프로그램 제작 전, 필요한 테이블을 생성한다.

```
SQL> create table employees (  
2   id varchar2(10) primary key ,  
3   password varchar2(10) not null,  
4   name varchar2(24),  
5   lev char(1),  
6   enter varchar2(10) ,  
7   gender char(1),  
8   phone varchar2(30) ) ;
```

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> 프로그램 제작 전, 필요한 테이블에 내용을 입력한다.

```
선택 SQL 명령을 실행
SQL> insert into employees (id,password,name,lev,enter,gender,phone)
2 values ('javasam','12345','김유신','A','1990-01-01','1','010-1010-1234');
1 개의 행이 만들어졌습니다.
SQL>
SQL>
SQL> insert into employees values('javasam','12345','김유신','A','1990-01-01','1','010-1010-1234');
insert into employees values('javasam','12345','김유신','A','1990-01-01','1','010-1010-1234')
*
1행에 오류:
ORA-00001: 무결성 제약 조건 (SYSTEM.SYS_C008996)에 위배됩니다

SQL> insert into employees values('hongsam','12345','홍길동','A','1991-02-02','2','010-1010-2234');
1 개의 행이 만들어졌습니다.

SQL> insert into employees values('leesam','12345','이순신','B','1992-03-03','2','010-1010-3234');
1 개의 행이 만들어졌습니다.

SQL> insert into employees values('banana','12345','바나나','B','1993-04-04','1','010-1010-4234');
1 개의 행이 만들어졌습니다.

SQL> insert into employees values('coffee','12345','카푸치노','A','1994-05-05','1','010-1010-5234');
1 개의 행이 만들어졌습니다.

SQL> insert into employees values('apple','12345','스티브','B','1995-06-06','2','010-1010-6234');
1 개의 행이 만들어졌습니다.

SQL> commit;
커밋이 완료되었습니다.
```

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> business.EService.java

```
package business;

import java.util.ArrayList;

public interface EService {

    ArrayList<EmpVO> selectList();

    EmpVO selectOne(EmpVO vo);

    int insert(EmpVO vo);

    int update(EmpVO vo);

    int delete(EmpVO vo);

}
```

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> business.EServiceImple.java

```
package business;

import java.util.ArrayList;

@Service("employee")
public class EServiceImpl implements EService {

    @Autowired
    private EmpDAO dao;

    @Override
    public ArrayList<EmpVO> selectList() {
        return dao EmpList();
    }

    @Override
    public EmpVO selectOne(EmpVO vo) {
        return dao eDetail(vo);
    }

    @Override
    public int insert(EmpVO vo) {
        return dao eInsert(vo);
    }

    @Override
    public int update(EmpVO vo) {
        return dao eUpdate(vo);
    }

    @Override
    public int delete(EmpVO vo) {
        return dao eDelete(vo);
    }
}
```

# [ 2001020212\_16v4 인터페이스 구현 ]

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
  </context-param>

  <!-- Creates the Spring Container shared by all Servlets and Filters -->
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>

  <!-- Processes application requests -->
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

  <!-- This is Filter 한글제출을 방지하기 위함 -->
  <filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

## [ 2001020212\_16v4 인터페이스 구현 ]

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

## &gt; servlet-context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd
        http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd">

    <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resources directory -->
    <resources mapping="/resources/**" location="/resources/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <context:component-scan base-package="com.ncs.green.business.util,vo" />

</beans:beans>
```



## [2001020212\_16v4.2 인터페이스 기능 구현하기]

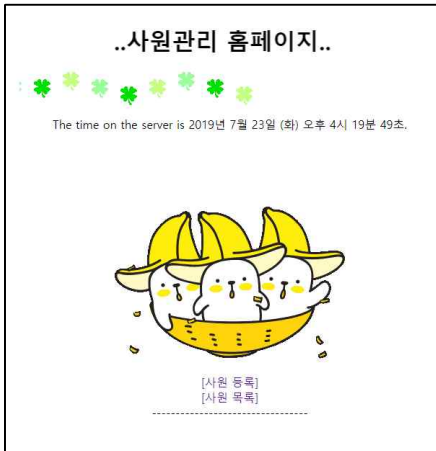
> home.jsp 소스코드

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page session="true" %>
<html>
<head>
    <title>** 인터페이스 구현 프로젝트 ** </title>
<style type = "text/css">
    a {
        text-decoration : none;
    }
    body {
        text-align:center;
    }
</style>

</head>
<body>
<marquee width="500px">
<h1>
    ..사원관리 홈페이지..
</h1>
</marquee><br>
<img src = "resources/image/bar.gif"><br>
<P> The time on the server is ${serverTime}. </P>
<br>
<img src = "resources/image/banana2.gif"><br>
<a href="joinf"> [사원 등록]</a><br>
<a href="eList">[사원 목록]</a><br>
-----<br>
</body>
</html>
```

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp 실행화면



[2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원 목록] elist 소스코드

```
@RequestMapping(value="/elist")
public ModelAndView elist(ModelAndView mv) {
    ArrayList<EmpVO> elist = service.selectList();
    mv.addObject("apple", elist);

    mv.setViewName("employee/emplist");
    return mv;
}
```

[ 2001020212\_16v4 인터페이스 구현 ]

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원 목록] 실행화면

**\*\* Employee List \*\***



I D	Password	Name	Lev	EnterDay	Gender	Phone
<a href="#">javasam</a>	12345	김유신	A VIP	1990-01-01	1	010-1010-1234
<a href="#">hongsam</a>	12345	홍길동	A VIP	1991-02-02	2	010-1010-2234
<a href="#">leesam</a>	12345	이순신	B 나무	1992-03-03	2	010-1010-3234
<a href="#">banana</a>	12345	바나나	B 나무	1993-04-04	1	010-1010-4234
<a href="#">coffee</a>	12345	카푸치노	A VIP	1994-05-05	1	010-1010-5234

[\[HOME\]](#)

[\[이전 화면\]](#)

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원 등록] 실행화면

**\*\* Employee Join \*\***

ID	<input type="text"/>
Password	<input type="password"/>
Name	<input type="text"/>
Level	<input type="text" value="관리자"/>
Enter Day	<input type="text" value="연도-월-일"/>
Gender	<input type="text"/>
Phone	<input type="text"/>

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원 등록] 실행화면

**\*\* Employee Join \*\***


ID	<input type="text"/>
Password	<input type="password"/>
Name	<input type="text"/>
Level	<input type="text" value="관리자"/>
Enter Day	<input type="text" value="연도-월-일"/>
Gender	<input type="text"/>
Phone	<input type="text"/>

[ 2001020212\_16v4 인터페이스 구현 ]

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원등록] 실행 성공화면

**\*\* Employee List \*\***



I D	Password	Name	Lev	EnterDay	Gender	Phone
<a href="#">jvasam</a>	12345	김유신	A VIP	1990-01-01	1	010-1010-1234
<a href="#">hongsam</a>	12345	홍길동	A VIP	1991-02-02	2	010-1010-2234
<a href="#">leesam</a>	12345	이순신	B 나무	1992-03-03	2	010-1010-3234
<a href="#">banana</a>	12345	바나나	B 나무	1993-04-04	1	010-1010-4234
<a href="#">coffee</a>	12345	카푸치노	A VIP	1994-05-05	1	010-1010-5234
<a href="#">moonsun</a>	12345	김문선	A VIP	1994-11-11	2	010-1212-3333

[\[HOME\]](#)  
[\[이전 화면\]](#)

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원등록] 실행 실패화면

**\*\* Employees DoFinish \*\***

사원 등록 실패~~

다시 하시겠습니까 ?

[사원 등록] [HOME](#)

[이전 화면](#)



## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> home.jsp -> [사원 등록] ejoin 소스코드

```
@RequestMapping(value = "/ejoin")
public ModelAndView ejoin(ModelAndView mv, HttpServletRequest request, EmpVO vo) {
    int cnt = service.insert(vo);

    if (cnt > 0) {
        mv.setViewName("redirect:/elist");
    } else {
        mv.addObject("fCode", "JF");
        mv.setViewName("employee/doFinish");
    } // else

    return mv;
} // ejoin
```

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원목록] 수정 eupdate 소스코드

```
@RequestMapping(value = "/eupdate")
public ModelAndView eupdate(ModelAndView mv, EmpVO vo) {
    int cnt = service.update(vo);

    if (cnt > 0) {
        mv.setViewName("redirect:/elist");
    } else { // 수정실패
        mv.addObject("fCode", "U");
        mv.setViewName("employee/finish");
    } // else
    return mv;
} // eupdate
```

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원목록] 수정 eupdate 실행화면

**\*\* Employee Detail \*\***

ID	<input type="text" value="beongsook"/>
Password	<input type="password" value="....."/>
Name	<input type="text" value="임병숙"/>
Level	<input type="text" value="관리자 ▼"/>
Enter Day	<input type="text" value="1980-03-11"/>
Gender	<input type="text" value="2"/>
Phone	<input type="text" value="010-4848-9898"/>

=====> [\[사원 삭제\]](#)

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원목록] 수정 eupdate 성공 화면

**\*\* Employee List \*\***



I D	Password	Name	Lev	EnterDay	Gender	Phone
<a href="#">jasasam</a>	12345	김유신	A VIP	1990-01-01	1	010-1010-1234
<a href="#">hongsam</a>	12345	홍길동	A VIP	1991-02-02	2	010-1010-2234
<a href="#">leesam</a>	12345	이순신	B 나무	1992-03-03	2	010-1010-3234
<a href="#">banana</a>	12345	바나나	B 나무	1993-04-04	1	010-1010-4234
<a href="#">coffee</a>	12345	카푸치노	A VIP	1994-05-05	1	010-1010-5234
<a href="#">youmee</a>	12345	김유미	A VIP	1999-03-03	2	010-1214-4433
<a href="#">moonsun</a>	12345	김문선	A VIP	1994-11-11	2	010-1212-3333
<a href="#">moonae</a>	12345	김문애	A VIP	1999-02-02	2	010-2222-3333
<a href="#">hongki</a>	12345	김홍기	A VIP	1933-03-02	1	010-2345-2121
<a href="#">beongsook</a>	12345	속속이	A VIP	1980-03-11	2	010-4848-9898

[\[HOME\]](#)

[\[이전 화면\]](#)

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원목록] 수정 eupdate 실패 화면

**\*\* Employees DoFinish \*\***

**정보 수정에 실패했습니다~~~**

[empList HOME](#)

[\[이전 화면\]](#)

## [2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원 목록] 삭제(탈퇴) edelete 소스코드

```
@RequestMapping(value = "/edelete")
public ModelAndView edelete(ModelAndView mv, EmpVO vo, EmpDAO dao) {

    int cnt = service.delete(vo);
    // 3) View 출력
    if (cnt > 0) {
        mv.setViewName("redirect:/elist");

    } else { // 탈퇴실패
        mv.addObject("fCode", "U");
        mv.setViewName("employee/doFinish");
    } // else
    return mv;
} // edelete
```

[ 2001020212\_16v4 인터페이스 구현 ]

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원목록] 삭제(탈퇴) edelete 성공화면

\*\* Employee List \*\*



I D	Password	Name	Lev	EnterDay	Gender	Phone
<a href="#">javasam</a>	12345	김유신	A VIP	1990-01-01	1	010-1010-1234
<a href="#">hongsam</a>	12345	홍길동	A VIP	1991-02-02	2	010-1010-2234
<a href="#">leesam</a>	12345	이순신	B 나무	1992-03-03	2	010-1010-3234
<a href="#">banana</a>	12345	바나나	B 나무	1993-04-04	1	010-1010-4234
<a href="#">coffee</a>	12345	카푸치노	A VIP	1994-05-05	1	010-1010-5234
<a href="#">youmee</a>	12345	김유미	A VIP	1999-03-03	2	010-1214-4433
<a href="#">moonsun</a>	12345	김문선	A VIP	1994-11-11	2	010-1212-3333
<a href="#">moonae</a>	12345	김문애	A VIP	1999-02-02	2	010-2222-3333
<a href="#">hongki</a>	12345	김홍기	A VIP	1933-03-02	1	010-2345-2121

[HOME]

[이전 화면]

-> id [beongsook]을 가진 사원을 탈퇴(삭제) 성공함.

[2001020212\_16v4.2 인터페이스 기능 구현하기]

> [사원 목록] 삭제(탈퇴) edelete 실패화면

**\*\* Employees DoFinish \*\***

**회원 탈퇴 실패~~**

HOME

[이전 화면]