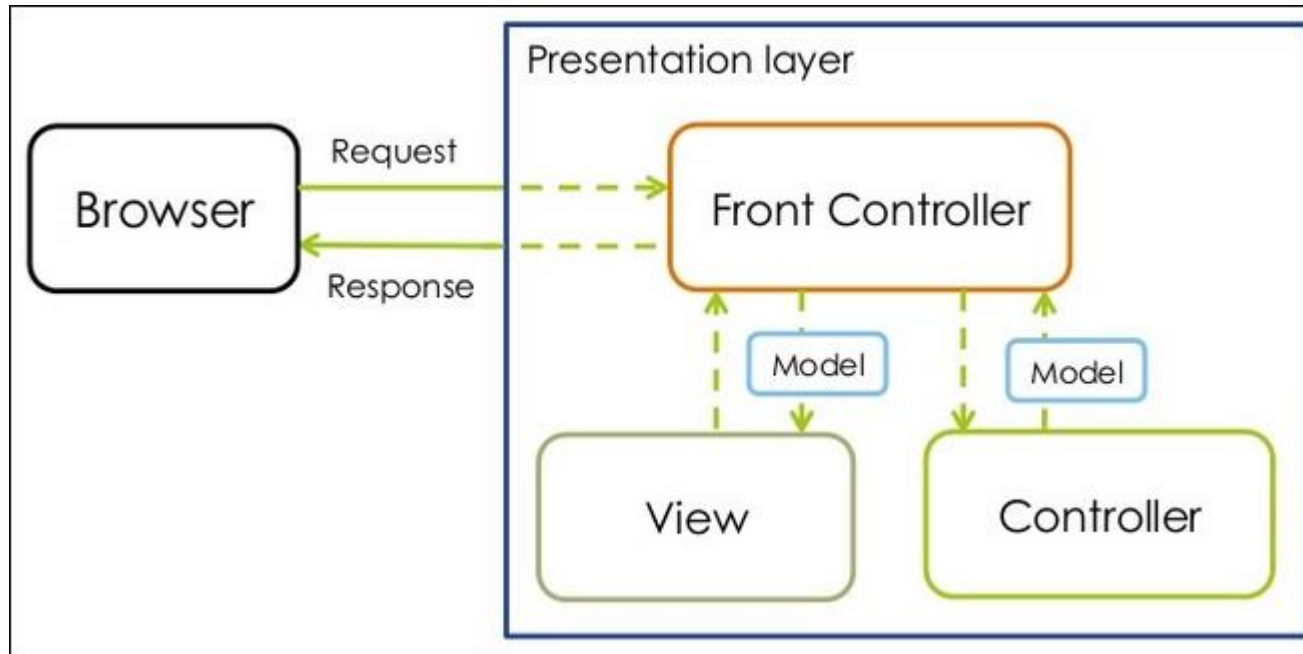


Spring Boot demo
(view page 만들기)

Spring MVC 구조



Project 만들기

New Spring Starter Project



Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

☐ Add project to working sets

Working sets:

New Spring Starter Project Dependencies



Spring Boot Version:

Frequently Used:

<input type="checkbox"/> Java Mail Sender	<input checked="" type="checkbox"/> MyBatis Framework	<input checked="" type="checkbox"/> Oracle Driver
<input type="checkbox"/> Spring Data JDBC	<input type="checkbox"/> Spring Reactive Web	<input type="checkbox"/> Spring Session
<input checked="" type="checkbox"/> Spring Web	<input type="checkbox"/> Thymeleaf	

Available:

- Alibaba
- Amazon Web Services
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Ops
- Pivotal Cloud Foundry

Selected:

- X MyBatis Framework
- X Oracle Driver
- X Spring Web

● Controller 작성

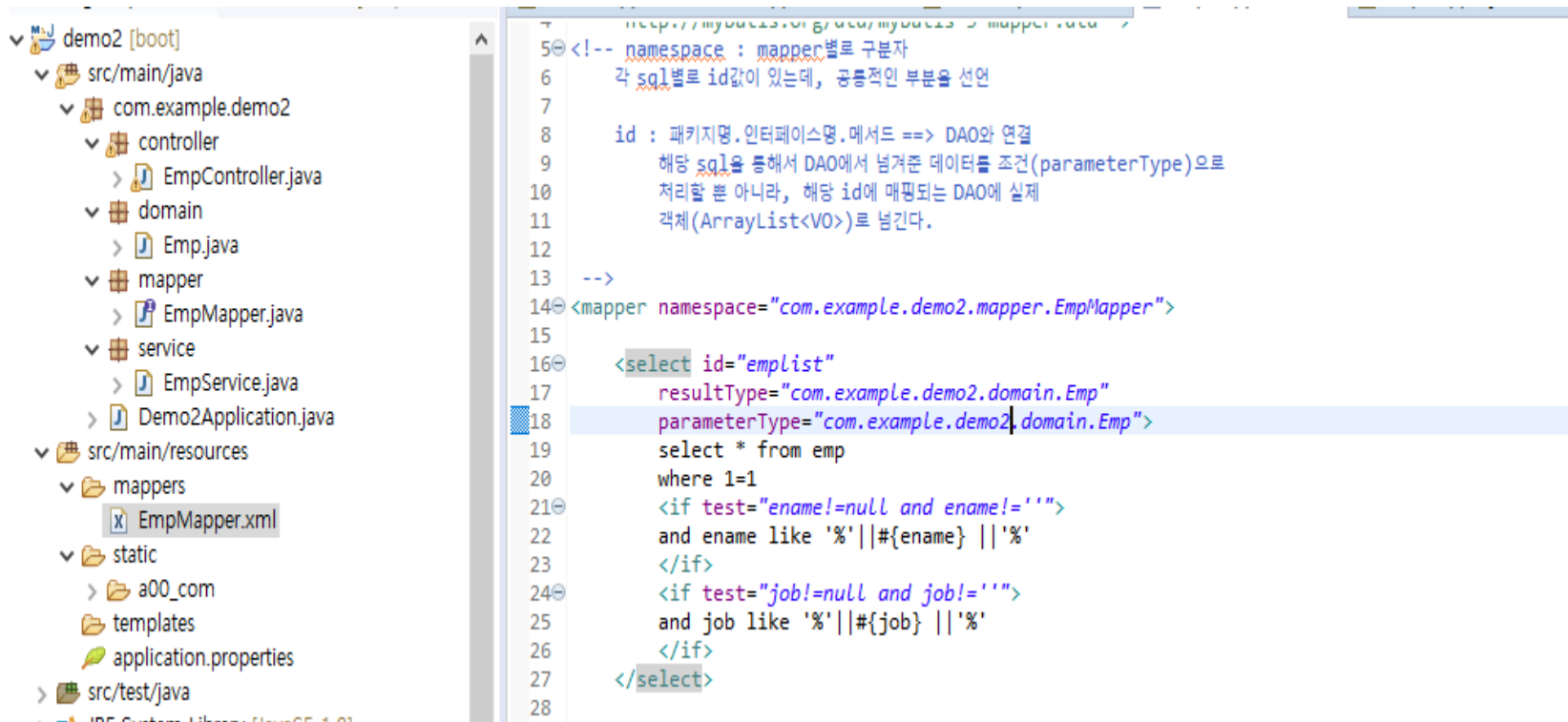
The screenshot displays an IDE environment for developing a Spring Boot application. On the left, the Package Explorer shows the project structure. The 'demo2' project is expanded, revealing the 'src/main/java' directory. Within 'com.example.demo2', the 'controller' package is highlighted, and 'EmpController.java' is selected. The main editor window shows the code for 'EmpController.java'. The code includes the package declaration 'package com.example.demo2.controller;', the import 'import java.util.List;', and the '@Controller' annotation. The 'EmpController' class contains an '@Autowired' field 'EmpService service' and a single endpoint method 'empList' mapped to '/empList'. The method prints the size of the employee list and adds it to the model. The bottom console shows the application's startup logs, including a warning about an unsatisfied dependency.

```
1 package com.example.demo2.controller;
2
3 import java.util.List;
4
5 @Controller
6 public class EmpController {
7
8     @Autowired(required=false)
9     private EmpService service;
10
11     @RequestMapping("/empList")
12     public String empList(Emp sch, Model md) throws Exception{
13         System.out.println("## controller ##");
14         System.out.println(service.empList(sch).size());
15         md.addAttribute("emplist", service.empList(sch) );
16
17         return "empList";
18     }
19 }
20
21
22
23
24
25
26
27
28
29
```

Problems @ Javadoc Declaration Console

```
<terminated> demo - DemoApplication [Spring Boot App] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2020-03-21 15:51:22.178 INFO 19900 --- [main] org.apache.jasper.servlet.TldScanner
2020-03-21 15:51:22.184 INFO 19900 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/]
2020-03-21 15:51:22.184 INFO 19900 --- [main] o.s.web.context.ContextLoader
2020-03-21 15:51:22.261 WARN 19900 --- [main] ConfigServletWebServerApplicationContext
Related cause: org.springframework.beans.factory.UnsatisfiedDependencyException: Error creat
Related cause: org.springframework.beans.factory.UnsatisfiedDependencyException: Error creat
2020-03-21 15:51:22.265 INFO 19900 --- [main] o.apache.catalina.core.StandardSer
```

Mapper 작성



The image shows an IDE with a project structure on the left and an XML file editor on the right.

Project Structure (Left):

- demo2 [boot]
 - src/main/java
 - com.example.demo2
 - controller
 - EmpController.java
 - domain
 - Emp.java
 - mapper
 - EmpMapper.java
 - service
 - EmpService.java
 - Demo2Application.java
 - src/main/resources
 - mappers
 - EmpMapper.xml
 - static
 - a00_com
 - templates
 - application.properties
 - src/test/java

XML File Content (Right):

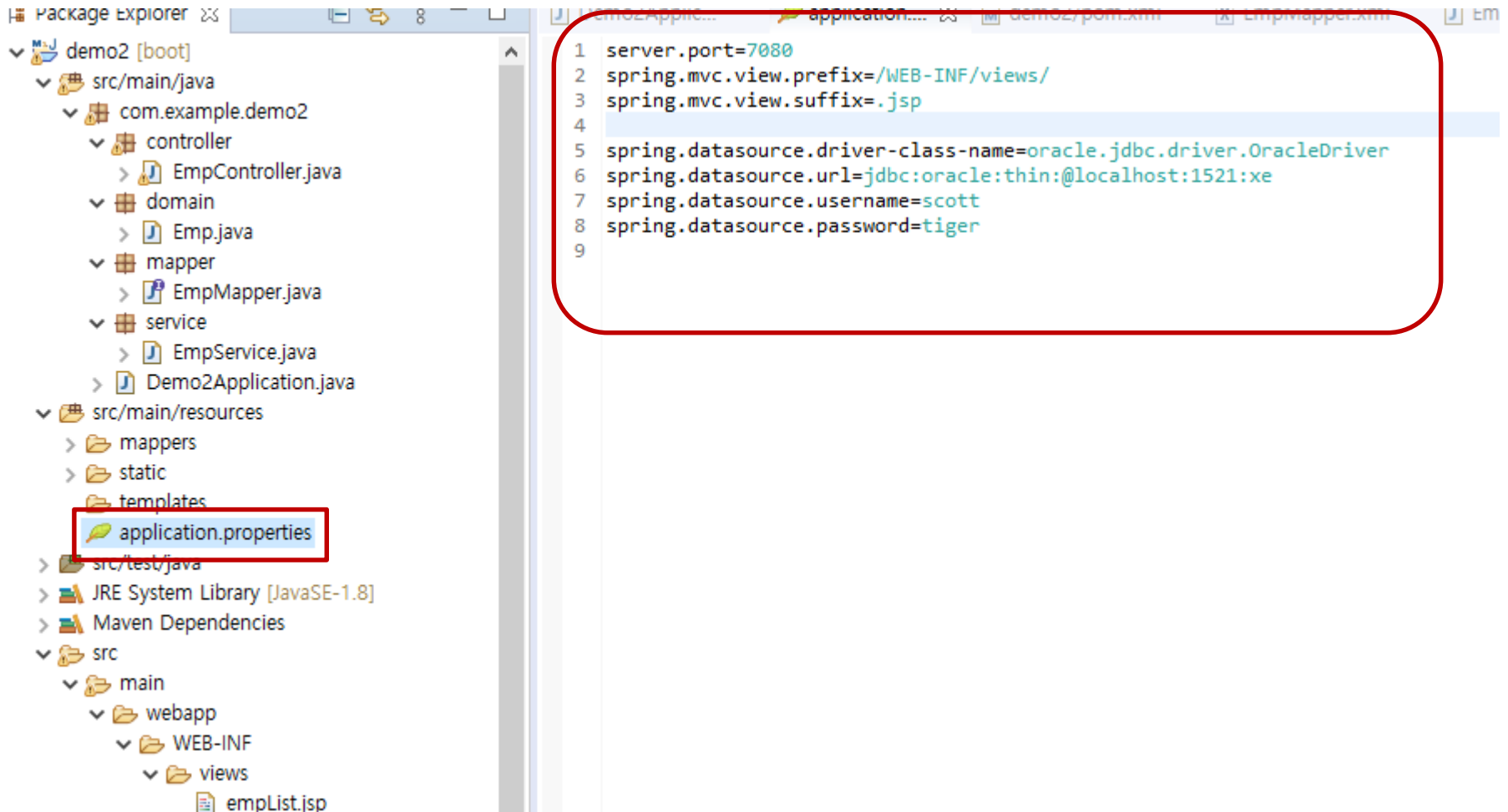
```
5<!-- namespace : mapper별 구분자
6   각 sql별 id값이 있는데, 공통적인 부분을 선언
7
8   id : 패키지명.인터페이스명.메서드 ==> DAO와 연결
9   해당 sql을 통해서 DAO에서 넘겨준 데이터를 조건(parameterType)으로
10  처리할 뿐 아니라, 해당 id에 매핑되는 DAO에 실제
11  객체(ArrayList<VO>)를 넘긴다.
12
13 -->
14<mapper namespace="com.example.demo2.mapper.EmpMapper">
15
16  <select id="emplist"
17    resultType="com.example.demo2.domain.Emp"
18    parameterType="com.example.demo2.domain.Emp">
19    select * from emp
20    where 1=1
21    <if test="ename!=null and ename!=''">
22      and ename like '%|#{ename} ||'%'
23    </if>
24    <if test="job!=null and job!=''">
25      and job like '%|#{job} ||'%'
26    </if>
27  </select>
28
```

● css/bootstrap설정

- src/main/resources
 - mappers
 - EmpMapper.xml
 - static
 - a00_com
 - css
 - images
 - packages
 - bootstrap.min.css
 - bootstrap.min.js
 - jquery-3.4.1.js
 - jquery-ui.css
 - jquery-ui.js
 - jquery.min.js
 - popper.min.js
 - w2ui-1.2.css
 - w2ui-1.2.js
 - templates
 - application.properties

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

● 뷰 리졸버(View Resolver)/DB 연동 설정 설정



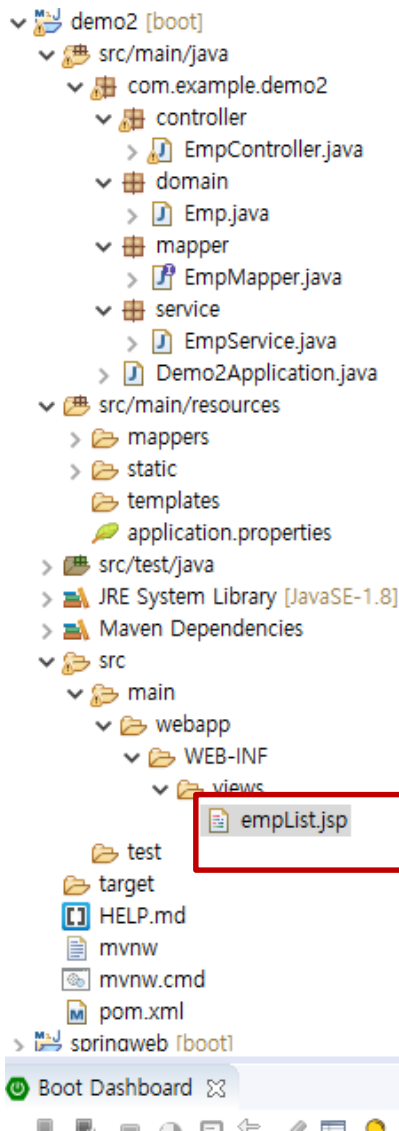
The screenshot shows an IDE interface with the Package Explorer on the left and the application.properties file open in the editor on the right. The Package Explorer shows a project named 'demo2 [boot]' with a structure including 'src/main/java' (containing 'com.example.demo2' with sub-packages 'controller', 'domain', 'mapper', 'service' and files 'EmpController.java', 'Emp.java', 'EmpMapper.java', 'EmpService.java', 'Demo2Application.java') and 'src/main/resources' (containing 'mappers', 'static', 'templates', and 'application.properties', which is highlighted with a red box). The editor shows the content of 'application.properties' with a red rounded rectangle highlighting the configuration for the View Resolver and Database Connection.

```
1 server.port=7080
2 spring.mvc.view.prefix=/WEB-INF/views/
3 spring.mvc.view.suffix=.jsp
4
5 spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
6 spring.datasource.url=jdbc:oracle:thin:@localhost:1521:xe
7 spring.datasource.username=scott
8 spring.datasource.password=tiger
9
```

● Pom.xml 의존성 라이브러리 추가

```
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
</dependency>
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.3.0</version>
</dependency>
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>1.2.3</version>
</dependency>
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0.3</version>
</dependency>
</dependencies>
```


● JSP 페이지 작성

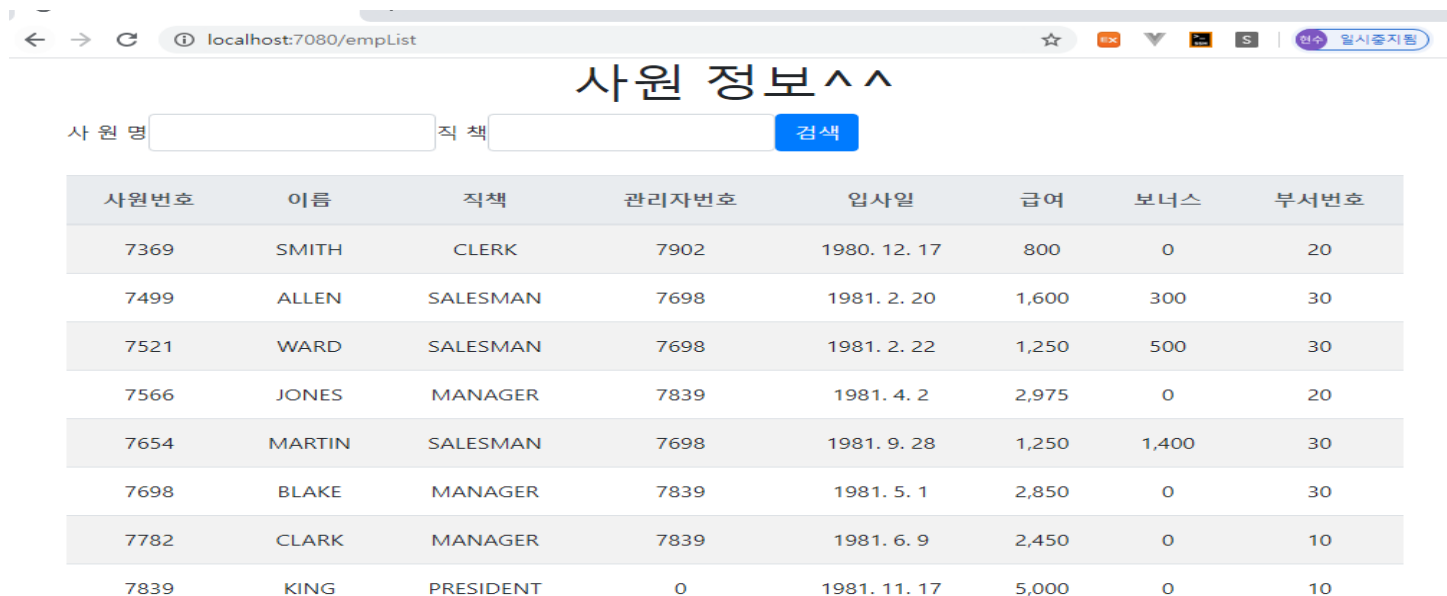


```
20 <script src="{path}/a00_com/popper.min.js" >/script>
21 <script src="{path}/a00_com/bootstrap.min.js">/script>
22 <script src="{path}/a00_com/jquery-ui.js">/script>
23 <script type="text/javascript">
24     $(document).ready(function(){
25         $("h1,table").attr("align","center");
26         $("h1").html("사원 정보^^");
27     });
28 </script>
29 </head>
30 <body>
31 <div class="container">
32     <h1></h1>
33     <!-- 사원명과 직책 화면 조회 처리 -->
34 <form class="form-inline" method="post">
35     <label for="ename"> 사 원 명 </label>
36     <input class="form-control" id="ename" name="ename" value="{param.ename}"/>
37     <label for="job"> 직 책 </label>
38     <input class="form-control" id="job" name="job" value="{param.job}"/>
39     <input type="submit" class="btn btn-primary" value="검색"/>
40 </form>
41 <br>
42 <table class="table table-hover table-striped">
43 <thead class="thead-light">
44 <tr><th>사원번호</th><th>이름</th><th>직책</th><th>관리자번호</th>
45 <th>입사일</th><th>급여</th><th>보너스</th><th>부서번호</th>
46 </tr>
47 </thead>
48 <c:forEach var="emp" items="{emplist }">
49 <tr>
50 <td>{emp.empno}</td>
51 <td>{emp.ename}</td>
52 <td>{emp.job}</td>
53 <td>{emp.mgr}</td>
54 <td><fmt:formatDate value="{emp.hiredate}"/></td>
55 <td><fmt:formatNumber value="{emp.sal}"/> </td>
56 <td><fmt:formatNumber value="{emp.comm}"/></td>
57 <td>{emp.deptno}</td>
58 </tr>
59 </c:forEach>
60 </table>
61 </div>
62
```

● Application 실행(sqlSessionFactory설정) & Brower 결과 확인



```
1 package com.example.demo2;
2
3 import javax.sql.DataSource;
4
5
6
7
8
9
10
11
12
13
14
15 @SpringBootApplication
16 @MapperScan(value = {"com.example.demo2.mapper"})
17 public class Demo2Application {
18
19     public static void main(String[] args) {
20         SpringApplication.run(Demo2Application.class, args);
21     }
22 }
23
24 /*
25  * SqlSessionFactory 설정
26  */
27 @Bean
28 public SqlSessionFactory sqlSessionFactory(DataSource dataSource) throws Exception{
29
30     SqlSessionFactoryBean sessionFactory = new SqlSessionFactoryBean();
31
32     Resource[] res = new PathMatchingResourcePatternResolver().getResources("classpath:mappers/*Mapper.xml");
33     sessionFactory.setMapperLocations(res);
34
35     sessionFactory.setDataSource(dataSource);
36     return sessionFactory.getObject();
37 }
38
39
40
41
42
```



← → ↻ 📍 localhost:7080/empList ☆ 📄 📁 📄 📄 | 🔍 임시중지됨

사원 정보^^

사원명 직책

사원번호	이름	직책	관리자번호	입사일	급여	보너스	부서번호
7369	SMITH	CLERK	7902	1980. 12. 17	800	0	20
7499	ALLEN	SALESMAN	7698	1981. 2. 20	1,600	300	30
7521	WARD	SALESMAN	7698	1981. 2. 22	1,250	500	30
7566	JONES	MANAGER	7839	1981. 4. 2	2,975	0	20
7654	MARTIN	SALESMAN	7698	1981. 9. 28	1,250	1,400	30
7698	BLAKE	MANAGER	7839	1981. 5. 1	2,850	0	30
7782	CLARK	MANAGER	7839	1981. 6. 9	2,450	0	10
7839	KING	PRESIDENT	0	1981. 11. 17	5,000	0	10

스프링 ajax 설정

마지막 ajax 소스 호출 설정추가부분

```
// http://localhost:7080/springweb/emp.do?method=ajaxList
@RequestMapping(params="method=ajaxList")
public String ajaxList(Emp sch, Model d) {
    d.addAttribute("empList", service.empList(sch));
    return "pageJsonReport";
}
```



- springweb
 - Deployment Descriptor: spring
 - JAX-WS Web Services
 - Java Resources
 - JavaScript Resources
 - build
 - WebContent
 - a00_com
 - a01_jquery
 - a02_jqAjax
 - h01_fullcalen
 - META-INF
 - WEB-INF
 - lib
 - view
 - dispatcher-servlet.xml
 - web.xml
 - index.html
 - index.jsp
 - ora
 - a00_first.css
- VueExp
- vueexp2

```
82 <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
83     <property name="order" value="1"/>
84 </bean>
85 <bean id="btnViewResolver"
86     class="org.springframework.web.servlet.view.BeanNameViewResolver">
87     <property name="order" value="0"/>
88 </bean>
89 <!-- java의 Object를 json문자열로 자동변환 처리하는 view
90 controller에서 마지막 return되는 문자열이
91     return "pageJsonReport"이면 view json형식으로
92     처리 한다.
93 -->
94 <bean id="pageJsonReport" class="org.springframework.web.servlet.view.json.MappingJacksonJsonView">
95     <property name="contentType" value="text/html;charset=utf-8" />
96 </bean>
```