

01. 프로젝트

- src/main/java
 - src/main/resources
 - src/main/webapp : 웹 어플리케이션을 구현할 때 필요한 코드가 위치하는 곳
 - src/main/webapp/WEB-INF : web.xml이 위치하는 곳
 - src/main/webapp/WEB-INF/view
-
- sp4-chap14폴더를 생성한다.
 - 8장의 pom.xml과 src폴더를 그대로 sp4-chap14폴더에 복사한다.
 - sp4-chap14폴더의 pom.xml에서 <artifactId>태그의 값을 sp4-chap14로 변경한다.
 - 이클립스에서 sp4-chap14 메이븐 프로젝트를 임포트한다.

```
CREATE TABLE TB_FILE
```

```
(  
  IDX NUMBER,  
  BOARD_IDX NUMBER NOT NULL,  
  ORIGINAL_FILE_NAME VARCHAR2(260 BYTE) NOT NULL,  
  STORED_FILE_NAME VARCHAR2(36 BYTE) NOT NULL,  
  FILE_SIZE NUMBER,  
  CREA_DTM DATE DEFAULT SYSDATE NOT NULL,  
  CREA_ID VARCHAR2(30 BYTE) NOT NULL,  
  DEL_GB VARCHAR2(1 BYTE) DEFAULT 'N' NOT NULL,  
  PRIMARY KEY (IDX)  
);
```

예제 1) sp4-chap14 폴더에 pom.xml 파일을 작성한다.
MultipartHttpServletRequest 추가한다.

```
<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
    http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  <modelVersion>4.0.0</modelVersion>  
  <groupId>sp4</groupId>  
  <artifactId>sp4-chap14</artifactId>  
  <version>0.0.1-SNAPSHOT</version>  
  <packaging>war</packaging>
```

```
<dependencies>
  <dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.2</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>4.1.0.RELEASE</version>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>4.1.0.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>com.mchange</groupId>
    <artifactId>c3p0</artifactId>
    <version>0.9.2.1</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.30</version>
  </dependency>
</dependencies>
```

```

    <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>1.2.17</version>
    </dependency>

    <dependency>
        <groupId>commons-io</groupId>
        <artifactId>commons-io</artifactId>
        <version>2.0.1</version>
    </dependency>

    <dependency>
        <groupId>commons-fileupload</groupId>
        <artifactId>commons-fileupload</artifactId>
        <version>1.2.2</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.1</version>
            <configuration>
                <source>1.7</source>
                <target>1.7</target>
                <encoding>utf-8</encoding>
            </configuration>
        </plugin>
    </plugins>
</build>

</project>

```

예제 2) 서비스 클래스를 위한 스프링 설정 클래스를 작성한다.

src\main\resources\spring-controller.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
```

```

        public String submitReport1(
            @RequestParam("studentNumber") String studentNumber,
            @RequestParam("report") MultipartFile report) throws
IOException {
            printInfo(studentNumber, report);
            return "report/submissionComplete";
        }

        private void printInfo(String studentNumber, MultipartFile report) throws
IOException {
            originalFile = report.getOriginalFilename();
            originalFileExtension =
originalFile.substring(originalFile.lastIndexOf("."));
            originalFileName = originalFile + originalFileExtension;
            storedFileName = UUID.randomUUID().toString().replaceAll("-", "") +
originalFileExtension;

            System.out.println(studentNumber + "가 업로드 한 파일명: "
                + originalFile);
            System.out.println(studentNumber + "가 업로드 한 저장된 파일명: "
                + storedFileName);
            System.out.println(studentNumber + "가 업로드 한 파일 사이즈: "
                + report.getSize());

            file = new File(filePath + storedFileName);
            report.transferTo(file);
        }

        @RequestMapping(value = "/report/submitReport2.do", method =
RequestMethod.POST)
        public String submitReport2(MultipartHttpServletRequest request) throws
IOException {
            String studentNumber = request.getParameter("studentNumber");
            MultipartFile report = request.getFile("report");
            printInfo(studentNumber, report);
            return "report/submissionComplete";
        }

        @RequestMapping(value = "/report/submitReport3.do", method =
RequestMethod.POST)
        public String submitReport3(ReportCommand reportCommand) throws

```



```

IOException {
    printInfo(reportCommand.getStudentNumber(),
reportCommand.getReport());
    return "report/submissionComplete";
}
}

```

예제 7) 컨트롤러를 스프링 빈에 등록한다. src\main\resources\spring-controller.xml
 <?xml version="1.0" encoding="UTF-8"?>

```

<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">

```

... 생략

```

    <bean class="controller.ReportSubmissionController" />

```

```

</beans>

```

예제 8) JSP구현

컨트롤러가 생성한 결과를 뷰 코드를 만든다.

뷰 코드는 JSP를 이용해서 구현한다.

src\main\webapp\WEB-INF\view\report\submissionForm.jsp

```

<%@ page contentType="text/html; charset=EUC-KR"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<% request.setCharacterEncoding("UTF-8"); %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>리포트 제출</title>
</head>
<body>
<h3>@RequestParam 사용</h3>
<form action="submitReport1.do" method="post" enctype="multipart/form-data">
    학번: <input type="text" name="studentNumber" />

```

```

xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx.xsd">

<tx:annotation-driven transaction-manager="transactionManager"/>

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
    destroy-method="close">
    <property name="driverClass" value="oracle.jdbc.driver.OracleDriver"
/>
    <property name="jdbcUrl" value="jdbc:oracle:thin:@localhost:1521:XE"
/>
    <property name="user" value="smrit" />
    <property name="password" value="oracle" />
</bean>

<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource" />
</bean>

<bean id="multipartResolver"
class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <property name="maxUploadSize" value="100000000" />
    <property name="maxInMemorySize" value="100000000" />
</bean>
</beans>

```

COMmonsMultipartResolver는 스프링에서 파일업로드 기능을 구현해놓은 클래스다. value="100000000"이라고 되어있는데, 이는 서버에 업로드할 수 있는 첨부파일의 최대 크기를 의미한다.

단위는 byte로 10,000,000byte이기 때문에 10MB로 설정이 되어있다.

해당 크기 이상의 파일이 전송되면 에러가 발생한다

예제 3) 스프링 MVC를 위한 기본 설정 파일 작성

src\main\resources\spring-mvc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/mvc
                           http://www.springframework.org/schema/mvc/spring-mvc.xsd">
```

```
    <mvc:annotation-driven />
```

```
    <mvc:default-servlet-handler />
```

```
    <mvc:view-resolvers>
```

```
        <mvc:jsp prefix="/WEB-INF/view/" />
```

```
    </mvc:view-resolvers>
```

```
    <mvc:resources mapping="/**" location="/WEB-INF/view/"
                  cache-period="31556926"/>
```

```
<mvc:interceptors>
```

```
    <mvc:interceptor>
```

```
        <mvc:mapping path="/edit/**/*"/>
```

```
        <bean class="org.springframework.web.servlet.mvc.WebContentInterceptor">
```

```
            <property name="cacheSeconds" value="0"/>
```

```
            <property name="cacheMappings">
```

```
                <props>
```

```
                    <prop key="/api/photo/**">86400</prop>
```

```
                </props>
```

```
            </property>
```

```
        </bean>
```

```
    </mvc:interceptor>
```

```
</mvc:interceptors>
```

```
</beans>
```

```
<mvc:mapping path="/api/**/*"/>
```

현디렉터리 이하는 웹 케시에 저장 하지 말것

```
<prop key="/api/photo/**">86400</prop>
```

현 디렉터리는 하루 동안 예외적으로 처리

캐시에 저장하는 예

```
<mvc:resources mapping="/js/**" location="/js/" cache-period="86400"/>
<mvc:resources mapping="/style/**" location="/style/" cache-period="86400"/>
<mvc:resources mapping="/image/**" location="/image/"
cache-period="31556926"/>
```

예제 4) web.xml 파일을 작성한다. src\main\webapp\WEB-INF\web.xml
contextConfiguration 초기화 파라미터와 DispatcherServlet 설정한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>
        classpath:spring-mvc.xml
        classpath:spring-controller.xml
      </param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

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

  <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>

```

13~20행 : contextConfiguration 초기화 파라미터.

예제 5) Command 를 구현한다.

src\main\java\controller\ReportCommand.java

```

package controller;
import org.springframework.web.multipart.MultipartFile;

public class ReportCommand {

    private String studentNumber;
    private MultipartFile report;

    public String getStudentNumber() {
        return studentNumber;
    }

    public void setStudentNumber(String studentNumber) {
        this.studentNumber = studentNumber;
    }

    public MultipartFile getReport() {
        return report;
    }
}

```

```

        public void setReport(MultipartFile report) {
            this.report = report;
        }
    }
}

```

예제 6) 컨트롤러를 구현한다.

src\main\java\controller\ReportSubmissionController.java

```
package controller;
```

```
import java.io.File;
import java.io.IOException;
import java.util.UUID;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.multipart.MultipartHttpServletRequest;
```

```
@Controller
```

```
public class ReportSubmissionController {
```

```

        private static final String filePath =
"E:\\spring4\\sp4-chap14\\src\\main\\webapp\\WEB-INF\\view\\report\\file\\";
        File file = new File(filePath);
        String originalFile = null;
        String originalFileName = null;
        String originalFileExtension = null;
        String storedFileName = null;

```

```

        @RequestMapping(value = "/report/submission.do", method =
RequestMethod.GET)
        public String form() {
            return "report/submissionForm";
        }

```

```

        @RequestMapping(value = "/report/submitReport1.do", method =
RequestMethod.POST)

```

```
<br/>
리포트파일: <input type="file" name="report" />
<br/>
<input type="submit" />
</form>
```

<h3>MultipartHttpServletRequest 사용</h3>

```
<form action="submitReport2.do" method="post" enctype="multipart/form-data">
    학번: <input type="text" name="studentNumber" />
    <br/>
    리포트파일: <input type="file" name="report" />
    <br/>
    <input type="submit" />
</form>
```

<h3>커맨드 객체 사용</h3>

```
<form action="submitReport3.do" method="post" enctype="multipart/form-data">
    학번: <input type="text" name="studentNumber" />
    <br/>
    리포트파일: <input type="file" name="report" />
    <br/>
    <input type="submit" />
</form>
```

</body>

</html>

예제 9) src\main\webapp\WEB-INF\view\report\submissionComplete.jsp

```
<%@ page contentType="text/html; charset=EUC-KR"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>리포트 제출 완료</title>
</head>
<body>
리포트 제출 완료
</body>
</html>
```

