01. 프로젝트

- src/main/java
- src/main/resource
- 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-chap15폴더를 생성한다.
- 8장의 pom.xml과 src폴더를 그대로 sp4-chap15폴더에 복사한다.
- sp4-chap15폴더의 pom.xml에서 <artifactId>태그의 값을 sp4-chap14로 변경한다.
- 이클립스에서 sp4-chap15 메이븐 프로젝트를 임포트한다.

http://www.oracle.com/technetwork/java/javamail-1-4-1-141959.html
http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-java-plat-419418.html#jaf-1.1-fr-oth-JPR
각각 파일을 다운로드 받아 압축을 푼 후 톰켓의 bin디렉터리에 복사한다.

예제 1) sp4-chap15 폴더에 pom.xml 파일을 작성한다.

```
<?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-chap15</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>war</packaging>
        <properties>
            <java-version>1.6</java-version>

<org.springframework-version>3.1.0.RELEASE</org.springframework-version>
            <org.aspectj-version>1.6.9</org.aspectj-version>
            <org.slf4j-version>1.6.1</org.slf4j-version>
        </properties>

        <dependencies>
                <dependency>
                        <groupId>javax.servlet.jsp</groupId>
                        <artifactId>jsp-api</artifactId>
```

```xml
        <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>

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

```
            <dependency>
                <groupId>javax.mail</groupId>
                <artifactId>mail</artifactId>
                <version>1.4.3</version>
            </dependency>

            <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-context-support</artifactId>
                <version>3.0.5.RELEASE</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"
        xmlns:context="http://www.springframework.org/schema/context"
        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-3.1.xsd
                http://www.springframework.org/schema/tx
                http://www.springframework.org/schema/tx/spring-tx.xsd
                http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.1.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="mailSender"                          class
="org.springframework.mail.javamail.JavaMailSenderImpl" >
            <property name="host" value="smtp.gmail.com" /> // 74.125.129.109
            <property name="port" value="587" />
            <property name="defaultEncoding" value="utf-8"/>
            <property name="username" value="" />
            <property name="password" value="" />
            <property name="javaMailProperties">
                        <prop key="mail.smtp.starttls.enable" >true</prop>
                        <prop key="mail.smtp.auth" >true</prop>
                        <prop key="mail.smtps.ssl.checkserveridentity">true</prop>
                        <prop key="mail.smtps.debug" >true</prop>
                        <prop key="mail.smtps.ssl.trust">*</prop>
            </property>
        </bean>

        <          b          e          a          n
class="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostPr
```

```
ocessor" />

    <bean id="email" class="spring.Email">
    </bean>
    <bean id="emailSender" class="spring.EmailSender">
    </bean>
     <bean id="emailcontroller" class="Controller.EmailController">
    </bean>
</beans>
```

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>

</beans>
```

예제 4) web.xml

```
<?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-controller.xml
                        classpath:spring-mvc.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>
        <servlet-mapping>
    <servlet-name>dispatcher</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

```
</web-app>
```

예제 5)메일정보를 저장하기 위한 Email클래스를 생성한다.

```
package spring;

public class Email{

    private String subject;
    private String content;
    private String regdate;
    private String reciver;
    private String fromName;

    public String getReciver() {
        return reciver;
    }
    public void setReciver(String reciver) {
        this.reciver = reciver;
    }

    public String getSubject() {
        return subject;
    }
    public void setSubject(String subject) {
        this.subject = subject;
    }
    public String getContent() {
        return content;
    }
    public void setContent(String content) {
        this.content = content;
    }
    public String getRegdate() {
        return regdate;
    }
    public void setRegdate(String regdate) {
        this.regdate = regdate;
    }
    public String getFromName() {
        return fromName;
    }
    public void setFromName(String fromName) {
        this.fromName = fromName;
    }

}
```

예제 6) EmailSender 클래스 생성

```
package spring;

import javax.mail.MessagingException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.MailException;
import org.springframework.mail.javamail.JavaMailSender;


public class EmailSender  {

        @Autowired
    protected JavaMailSender  mailSender;

    public void SendEmail(Email email) throws Exception {
        MimeMessage msg = mailSender.createMimeMessage();

        //System.out.println(email.getReciver());
        //System.out.println(email.getSubject());
        //System.out.println(email.getContent());
        try {

                msg.setSubject(email.getSubject());
                msg.setText(email.getContent());
                msg.setFrom(new InternetAddress(email.getFromName()));
                msg.setRecipient(MimeMessage.RecipientType.TO          ,          new
InternetAddress(email.getReciver()));
                msg.setText(text,"UTF-8", "html");
        }catch(MessagingException e) {
                System.out.println("MessagingException");
                e.printStackTrace();
        }
            try {
                mailSender.send(msg);
            }catch(MailException e) {
                System.out.println("MailException발생");
                e.printStackTrace();
            }
    }
}
```

예제 7) EmailController 를 생성한다.

```
package Controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import spring.Email;
import spring.EmailSender;

@Controller
@RequestMapping("/email")
public class EmailController {
    @Autowired
    private EmailSender emailSender;
    @Autowired
    private Email email;
    @RequestMapping("/send")
    public ModelAndView sendEmailAction () throws Exception {

        String reciver = "highland0@nate.com"; //받을사람의 이메일입니다.
        String subject = "이메일 제목";
        String content = "이메일 내용입니다.";
        String fromName ="test@aaaa.com";

        email.setReciver(reciver);
        email.setSubject(subject);
        email.setContent(content);
        email.setFromName(fromName);
```

```
            emailSender.SendEmail(email);
            return new ModelAndView("redirect:/success");
        }
}
```

setFrom(String from) : 발신자 설정
setReplyTo(String replyTo) : 응답 주소 설정
setTo(String to) : 수신자 설정
setTo(String[] to) : 수신자 목록 설정
setCc(String cc) : 참조자 설정
setCc(String[] cc) : 참조자 목록 설정
setBcc(String bcc) : 숨은 참조자 설정
setBcc(String[] bcc) : 숨은 참조자 목록 설정
setSentDate(Date sentDate) : 메일 발송일 설정
setSubject(String subject) : 메일 제목 설정
setText(String text) : 메일 내용 설정