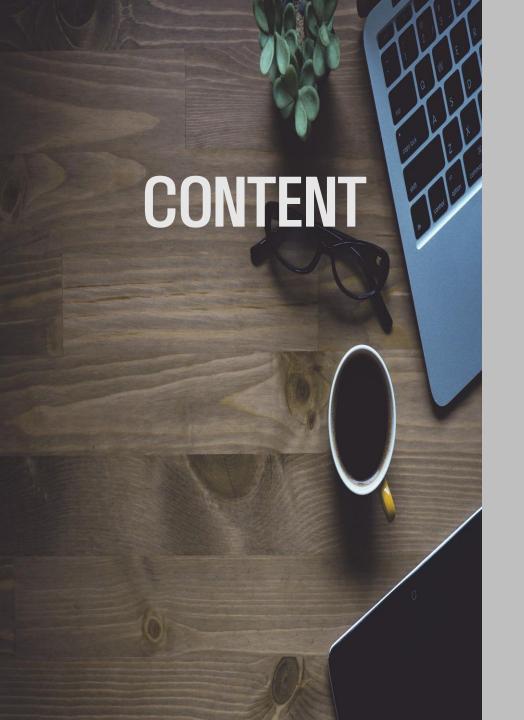
# 10장 스프링에서 지원하는 기능

강사 금영석



log4j란?

2 다중 파일 업로드 하기

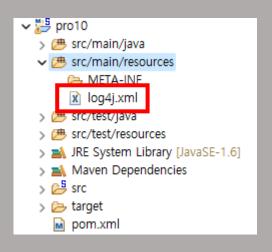
3 썸네일 이미지

4 스프링 이메일

5 스프링 인터셉터

## 1. log4j란?

- ✓ 로그 기능을 제공하는 오픈 소스 라이브러리이다.
- ✓ 애플리케이션에서 웹 사이트에 접속한 사용자 정보나 각 클래스의 메서드호출 시간 등 여러가지 정보를 로그로 출력해서 관리 할 수 있다.
- ✔ 메이븐에서는 프로젝트를 생성할 때 자동으로 log4j 라이브러리가 설치된다.



### ✓ log4j.xml 을 이루는 태그는 아래와 같다.

태그	설명
<appender></appender>	로그의 출력 위치를 결정(파일, 콘솔, DB 등) 한다. log4j API 문서의 XXXAppender로 끝나는 클래스들의 이름을 보면 출력 위치가 어디인지 알 수 있다.
<layout></layout>	Appender가 어디에 출력할 것인지 결정했다면 Layout은 어떤 형식으로 출력할지 출력 레이아 웃을 결정한다.
<logger></logger>	로깅 메시지를 Appender에 전달한다. 개발자가 로그 레벨을 이용해 로그 출력 여부를 조정할수 있다. logge는 로그 레벨을 가지고 있으며, 로그의 출력 여부는 로그문의 레벨과 logger의 레벨을 가지고 결정한다.

### ✓ 여러 가지 Appender 클래스

Appender 클래스	설명
ConsoleAppender	org.apache.log4j.ConsoleAppender 클래스로 콘솔에 로그 메시지를 출력한다.
FileAppender	org.apache.log4j.FileAppender 클래스로 파일에 로그 메시지를 출력한다.
RollingFileAppender	org.apache.log4j.rolling.RollingFileAppender 클래스로 파일 크기가 일정 기준을 넘으면 기존 파일을 백업 파일로 바꾸고 처음부터 다시 기록한다.
DailyRollingAppender	org.apache.log4j.DailyRollingAppender 클래스로 설정한 기간 단위로 새 파일을 만들어 로그 메 시지를 기록한다.

# ✓ PatternLayout 클래스에서 사용하는 여러 가지 출력 속성

속성	설명
%p	debug, info, error, fatal 등 로그 레빌 이름을 출력한다.
%m	로그 메시지를 출력한다.
%d	로깅 이벤트 발생 시각을 출력한다.
%F	로깅이 발생한 프로그램 파일 이름을 출력한다.
%1	로깅이 발생한 caller의 정보를 출력한다.
%L	로깅이 발생한 caller의 라인 수를 출력 한다.
%M	로깅이 발생한 method 이름을 출력한다.
%c	로깅 메시지 앞에 전체 패키지 이름이나 전체 파일 이름을 출력한다.
%n	한 행을 내린다. (next line)

### ✓ log4j의 여러 가지 로그 레벨

레벨	설명
FATAL	시스템 차원에서 심각한 문제가 발생해 애플리케이션 동작이 불가능할 경우에 해당하는 레벨이다. 일반적으로 애플리케이션에서는 사용할 일이 없다.
ERROR	실행 중 문제가 발생한 상태를 나타낸다.
WARN	향후 시스템 오류의 원인이 될 수 있는 경고 메시지를 나타낸다.
INFO	로그인, 상태 변경과 같은 실제 애플리케이션 운영과 관련된 정보 메시지를 나타낸다.
DEBUG	개발을 할 때 디버깅 용도로 사용하는 메시지를 나타낸다.
TRACE	DEBUG 레벨보다 상세한 로깅 정보를 출력하기 위해 도입된 레벨이다.



```
🖈 *log4j.xml 🖂
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "log4j.dtd">
 3@<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
       <!-- Console Appenders -->
 5⊝
       <appender name="console" class="org.apache.log4j.ConsoleAppender">
           <param name="Target" value="System.out" />
           <layout class="org.apache.log4j.PatternLayout">
               <param name="ConversionPattern" value="%-5p: %c - %m%n" />
           </layout>
10
11
       <!-- DailyRollingFile Appenders -->
120
       <appender name="dailyFileAppender" class="org.apache.log4j.DailyRollingFileAppender">
13
           <param name="File" value="I://Spring Source 2//logs//output.log" />
           <param name="Append" value="true" />
14
15⊜
           <layout class="org.apache.log4j.PatternLayout">
16
               <param name="DatePattern" value="'.'yyyy-MM-dd"></param>
17
               <param name="ConversionPattern" value="[%d{HH:mm:ss}][%-5p](%F:%L)-%m%n"></param>
18
           </layout>
19
       </appender>
20
      <!-- Application Loggers -->
21⊕
       <logger name="com.test.pro10">...
24
25
       <!-- 3rdparty Loggers -->
26⊕
       <logger name="org.springframework.core">[]
29
30⊕
       <logger name="org.springframework.beans">[]
33
34⊕
       <logger name="org.springframework.context">...
37
38⊕
       <logger name="org.springframework.web">...
41
42
       <!-- Root Logger -->
43⊝
44
           <priority value="info" />
45
           <appender-ref ref="dailyFileAppender" /</pre>
46
47
48
49 </log4j:configuration>
```



```
■ Console 

Progress Problems 

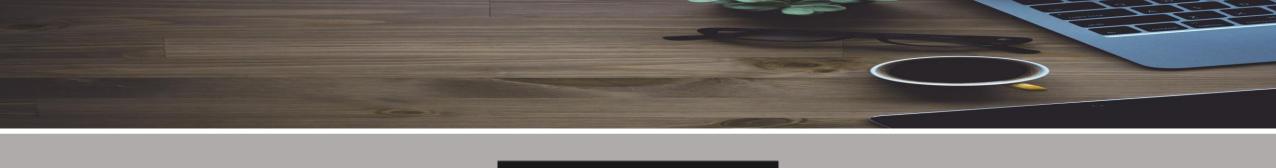
Servers

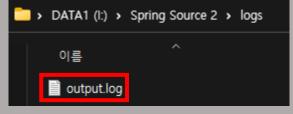
Problems 

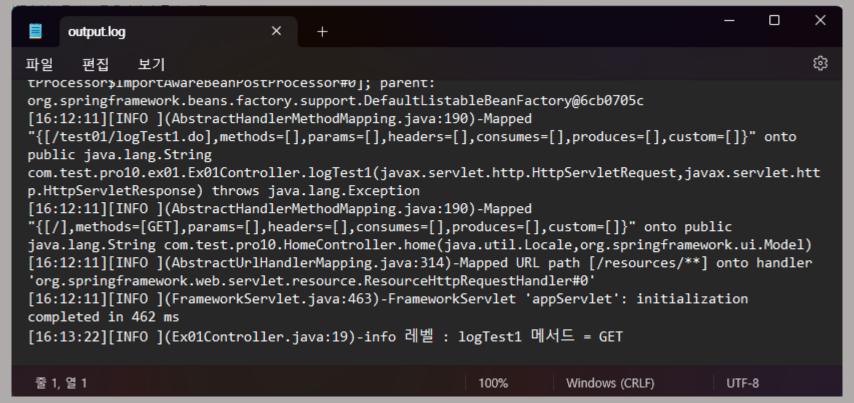
Result

Problems 

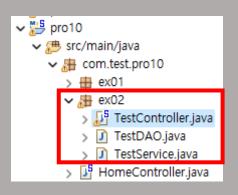
Pro
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre-1.8\bin\javaw.exe (2023. 9. 13. 오후 3:55:35)
 9월 13, 2023 3:55:37 오후 org.apache.catalina.startup.Catalina start
 정보: 서버가 [1437] 밀리초 내에 시작되었습니다.
 INFO: Ex01Controller.class - info 레벨: logTest1 메서드 = GET
                                  x *log4j.xml ⊠
                                    44
                                                              <!-- Root Logger -->
                                    45⊖
                                                              <root>
                                    46
                                                                             ority value="debug" />
                                    47
                                                                           <appender-ref ref="console" />
                                                                             <appender-ref ref="dailyFileAppender" />
                                    48
                                    49
                                                              </root>
                                    50
                                  51 </log4j:configuration>
 □ Console ⋈ □ Progress । Problems ♣ Servers
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre-1.8\bin\javaw.exe (2023. 9. 13. 오후 3:55:35)
 정보: 이름이 [/pro10]인 컨텍스트를 다시 로드하는 것을 완료했습니다.
INFO: Ex01Controller.class - info 레벨: logTest1 메서드 = GET
 DEBUG: Ex01Controller.class - debug 레벨: logTest1 메서드 = GET
```







### ✔ AOP를 이용한 로그 출력



```
m pro10/pom.xml ⊠
 34
             </dependency>
 35
 36
            <!-- AspectJ -->
 37⊜
             <dependency>
 38
                <groupId>org.aspectj</groupId>
 39
                <artifactId>aspectjrt</artifactId>
                <version>${org.aspectj-version}</version>
 40
 41
            </dependency>
 42
            <!-- spring-aop -->
 43
 44⊖
             <dependency>
 45
                <groupId>org.springframework</groupId>
 46
                <artifactId>spring-aop</artifactId>
                <version>${org.springframework-version}</version>
 47
 48
             </dependency>
 49⊜
             <dependency>
                <groupId>cglib
 50
 51
                <artifactId>cglib</artifactId>
 52
                 <version>2.2</version>
 53
                <type>jar</type>
 54
                <scope>compile</scope>
 55
            </dependency>
 56⊜
             <dependency>
 57
                <groupId>org.aspectj</groupId>
 58
                <artifactId>aspectjweaver</artifactId>
 59
                <version>${org.aspectj-version}</version>
60
             </dependency>
```



```
1 package com.test.pro10.ex02;
3. 3. import javax.servlet.http.HttpServletRequest;
11
12 @Controller
13 @EnableAspectJAutoProxy
14 public class TestController {
15⊜
       @Autowired
16
       TestService service;
17
18⊖
       @RequestMapping(value="/test02/logTest2.do")
       public String logtest2(HttpServletRequest request, HttpServletResponse response)
⇒19
20
           throws Exception{
21
           service.logTest2();
22
           return "test01/result";
23
24 }
```

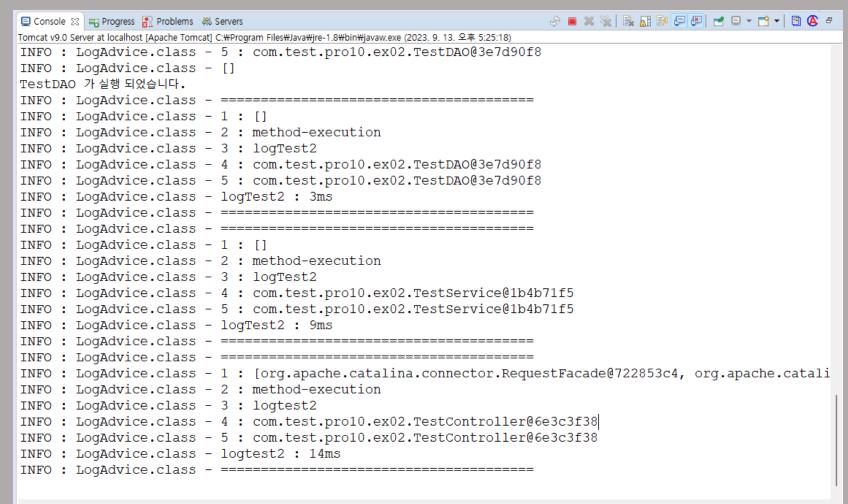
```
package com.test.pro10.ex02;
 39 import org.springframework.beans.factory.annotation.Autowired;
 4 import org.springframework.stereotype.Service;
 5
 6 @Service
 7 public class TestService {
       @Autowired
 9
       TestDAO dao;
⇒10⊝
       public void logTest2() {
211
           // TODO Auto-generated method stub
12
           dao.logTest2();
13
14 }
```



pro10

```
1 package com.test.pro10.common;
 3 import java.util.Arrays; ...
15 @Component
16 @Aspect
17 public class LogAdvice {
       private static final Logger logger
19
           = LoggerFactory.getLogger("LogAdvice.class");
20
21⊖
       @Before("execution(* com.test.pro10.ex02.*.*(..))")
922
       public void startLog(JoinPoint jp) {
23
           logger.info("=======");
24
25
           logger.info("1 : " + Arrays.deepToString(jp.getArgs()));
26
           logger.info("2 : " + jp.getKind());
27
           logger.info("3 : " + jp.getSignature().getName());
28
           logger.info("4 : " + jp.getTarget().toString());
29
           logger.info("5 : " + jp.getThis().toString());
30
31⊝
       @After("execution(* com.test.pro10.ex02.*.*(..))")
₽32
       public void endLog(JoinPoint jp) {
33
           logger.info("=======");
34
35
           logger.info("1 : " + Arrays.deepToString(jp.getArgs()));
36
           logger.info("2 : " + jp.getKind());
37
           logger.info("3 : " + jp.getSignature().getName());
38
           logger.info("4 : " + jp.getTarget().toString());
39
           logger.info("5 : " + jp.getThis().toString());
40
41⊖
       @Around("execution(* com.test.pro10.ex02.*.*(..))")
342
       public Object timeLog(ProceedingJoinPoint pjp) throws Throwable {
           long startTime = System.currentTimeMillis();
43
44
           logger.info(Arrays.toString(pjp.getArgs()));
45
           Object result = pjp.proceed();
46
           long endTime = System.currentTimeMillis();
47
           logger.info(pjp.getSignature().getName() + " : " + (endTime-startTime));
48
           logger.info("===========");
49
           return result;
50
```





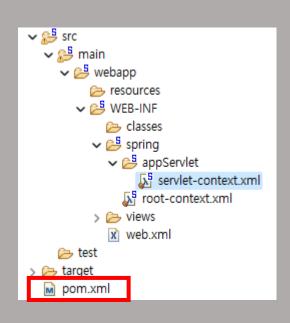
# 2. 다중 파일 업로드 하기

✓ 스프링의 CommonsMultipartResolver 클래스를 이용하면 여러 개의 파일을 한꺼번에 업로드할 수 있다.

✓ ComonsMultiResolver 클래스 속성

속성	설명
maxUploadSize	최대로 업로드가 가능한 파일의 크기를 설정할 수 있다.
maxInMemorySize	디스크에 임시 파일을 생성하기 전 메모리에 보관 할 수 있는 최대 바이트 크기를 설정한 다.
defaultEncoding	전달되는 매개변수의 인코딩을 설정한다.





```
m pro10/pom.xml ⋈
 58
                <artifactId>aspectjweaver</artifactId>
 59
                <version>${org.aspectj-version}</version>
 60
            </dependency>
 61
 62
            <!-- 다중 파일 다운로드 관련 -->
 63⊜
            <dependency>
                <groupId>commons-fileupload</groupId>
 64
 65
                <artifactId>commons-fileupload</artifactId>
                <version>1.2.1
 66
 67
            </dependency>
            <dependency>
 68⊜
 69
                <groupId>commons-io
 70
                <artifactId>commons-io</artifactId>
 71
                <version>1.4</version>
 72
            </dependency>
```



```
Second Second
```

```
x servlet-context.xml
 9
10
        <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure
11
12
        <!-- Enables the Spring MVC @Controller programming model -->
13
        <annotation-driven />
14
15
        <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resour
16
        <resources mapping="/resources/**" location="/resources/" />
17
18
        <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEF
19⊜
        <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
            <beans:property name="prefix" value="/WEB-INF/views/" />
9i20
<sub>0</sub>i21
            <beans:property name="suffix" value=".jsp" />
22
        </beans:bean>
23
        <beans:bean id="multipartResolver"</pre>
24⊖
25
            class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
            <beans:property name="maxUploadSize" value="52428800"></beans:property>
0i26
9<sub>i</sub>27
            <beans:property name="maxInMemorySize" value="1000000">/beans:property>
<mark>9</mark>i28
            <beans:property name="defaultEncoding" value="utf-8"></beans:property>
29
        </beans:bean>
30
31 </beans:beans>
```



```
    → 
    □ pro10

    → 
    □ src/main/java

    → com.test.pro10

    → common

    → ex01

    → ex02

    → ex03

    → FileDownloadController.java

    → I FileUploadController.java

    → I HomeController.java
```

```
🕽 *FileDownloadController.java 🖂
1 package com.test.pro10.ex03;
 3⊕ import java.io.File; ...
13
14 @Controller
15 public class FileDownloadController {
        private static String IMAGE REPO = "I:\\Spring Source 2\\image repo";
16
17⊜
        @RequestMapping(value="/test03/download.do")
18
        public void download (@RequestParam ("imageFileName") String imageFileName,
19
                HttpServletRequest request, HttpServletResponse response) throws Exception {
20
            OutputStream out = response.getOutputStream();
21
            String downFile = IMAGE REPO + "\\" + imageFileName;
22
            File file = new File(downFile);
23
            response.setHeader("Cache-Control", "no-cache");
24
            response.addHeader("Content-disposition", "attachment; fileName=" + imageFileName);
25
            FileInputStream in = new FileInputStream(file);
26
            byte[] buffer = new byte[1024*8];
27
            while(true) {
28
                int cnt = in.read(buffer);
29
                if (cnt == -1) {
30
                    break;
31
32
                out.write(buffer, 0, cnt);
33
            in.close();
34
35
            out.close();
36
37 }
```



```
🔝 FileUploadController.java 🖂
 1 package com.test.pro10.ex03;
 3⊕ import java.io.File;
18
19 @Controller
20 public class FileUploadController {
        private static final String IMAGE REPO = "I:\\Spring Source 2\\image repo";
21
22⊖
        @RequestMapping(value="/test03/form.do")
23
        public String form() {
24
            return "test02/uploadForm";
25
2.6⊖
        @RequestMapping(value="/test03/upload.do")
27
        public ModelAndView upload(MultipartHttpServletRequest mRequest,
28
            HttpServletResponse response) throws Exception {
29
            mRequest.setCharacterEncoding("utf-8");
№30
            Map map = new HashMap();
231
            Enumeration<String> enu = mRequest.getParameterNames();
32
            while(enu.hasMoreElements()) {
33
                String name = (String)enu.nextElement();
34
                String value = mRequest.getParameter(name);
№35
                map.put(name, value);
36
37
            List<String> fileList = fileProcess(mRequest);
№38
            map.put("fileList", fileList);
39
            ModelAndView mav = new ModelAndView();
40
            mav.addObject("map", map);
            mav.setViewName("test02/result");
41
42
            return mav;
43
```



```
🔎 *FileUploadController.java 🖂
45⊝
        private List<String> fileProcess(MultipartHttpServletRequest mRequest)
46
            throws Exception {
47
            List<String> fileList = new ArrayList<String>();
48
            Iterator<String> fileNames = mRequest.getFileNames();
49
            while(fileNames.hasNext()) {
50
                String fileName = fileNames.next();
51
                MultipartFile mFile = mRequest.getFile(fileName);
52
                String originalFileName = mFile.getOriginalFilename();
53
                fileList.add(originalFileName);
54
                File file = new File(IMAGE REPO + "\\" + fileName);
                if(mFile.getSize() != 0) {
56
                    if(! file.exists()) {
                        if(file.getParentFile().mkdirs()) {
58
                             file.createNewFile();
59
60
61
                    mFile.transferTo(new File(IMAGE REPO+"\\"+originalFileName));
62
63
64
            return fileList;
65
66
```



```
✓ № src

  🗸 🔑 main
    🗸 🐸 webapp
        resources

✓ № WEB-INF

           classes
         > 🐸 spring
         views
           > / test01

✓ 

    test02

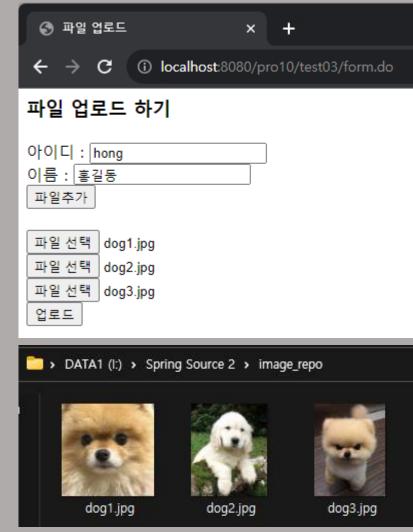
               result.jsp
               uploadForm.jsp
             home.jsp
           x web.xml
    test
> 📂 target
  m pom.xml
```

```
1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
 2 <%0 page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"%>
 4⊖<html>
 5⊖<head>
       <meta charset="UTF-8">
       <title>파일 업로드</title>
       <script src="http://code.jquery.com/jquery-latest.js"></script>
 9⊝
       <script>
           var cnt = 1;
10
11
           function fn addFile(){
12
               $("#d file").append("<br><input type='file' name='file"+cnt+"' />");
13
               cnt++;
14
15
       </script>
16 </head>
17⊖<body>
18
       <h3>파일 업로드 하기</h3>
19⊝
       <form method="post" action="/pro10/test03/upload.do" enctype="multipart/form-data">
20
           <label>아이디: </label><input type="text" name="id"><br>
21
           <label>이름: </label><input type="text" name="name"><br>
22
           <input type="button" value="파일추가" onclick="fn addFile()"><br>
23⊜
           <div id="d file">
24
           </div>
25
           <input type="submit" value="@로드">
26
       </form>
27 </body>
28 </html>
```



```
1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
 2 < @ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"%>
 4⊖<html>
 50 < head>
       <meta charset="UTF-8">
       <title>결과창</title>
 8 </head>
 9⊖ <body>
10 <h3>업로드가 완료되었습니다.</h3>
       <h3>파일 업로드 하기</h3>
11
12
      <label>아이디: </label><input type="text" value="${map.id }" readonly="readonly"><br>
       <label>이름: </label><input type="text" value="${map.name }" readonly="readonly"><br>
13
      <div>
14⊖
15⊜
           <c:forEach var="imageFileName" items="${map.fileList }">
16
               <imq src="/pro10/test03/download.do?imageFileName=${imageFileName }"><br><br>
17
           </c:forEach>
18
       </div>
19
       <a href="/pro10/test03/form.do">다시 업로드 하기</a>
20 </body>
21 </html>
```







# 3. 썸네일 이미지

✓ 다운 받은 이미지를 작은 사이즈의 이미지로 처리 할 수 있다.

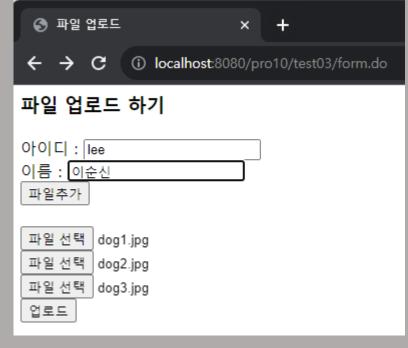
```
m pro10/pom.xml ⊠
                 <artifactId>commons-io</artifactId>
 70
                 <version>1.4</version>
 71
 72
             </dependency>
 73
 74
             <!-- 썸네일 관련 -->
 75⊜
             <dependency>
                  <groupId>net.coobird</groupId>
 76
                 <artifactId>thumbnailator</artifactId>
 77
 78
                 <version>0.4.8</version>
 79
             </dependency>
```



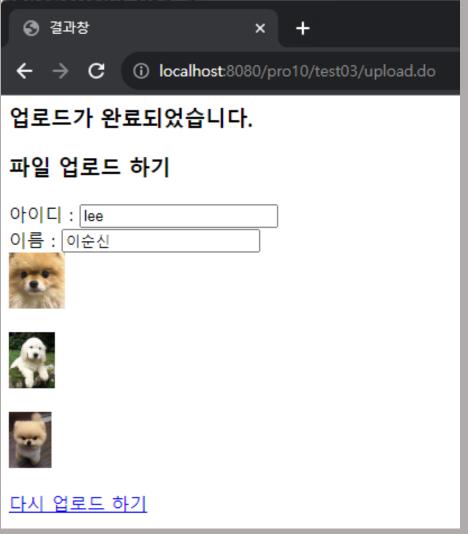
```
☑ FileDownloadController2.java ⋈
 1 package com.test.pro10.ex04;
 3⊕ import java.io.File; ...
15
16 @Controller
17 public class FileDownloadController2 {
       private static String IMAGE REPO = "I:\\Spring Source 2\\image repo";
19⊜
       @RequestMapping(value="/test04/download.do")
20
       public void download(@RequestParam("imageFileName") String imageFileName,
21
                HttpServletRequest request, HttpServletResponse response) throws Exception {
           OutputStream out = response.getOutputStream();
           String filePath = IMAGE REPO + "\\" + imageFileName;
            File image = new File(filePath);
           int lastIndex = imageFileName.lastIndexOf(".");
           String fileName = imageFileName.substring(0,lastIndex);
           File thumbnail = new File(IMAGE REPO+"\\"+"thumbnail"+"\\"+fileName+".png");
28
           if (image.exists()) {
29
                thumbnail.getParentFile().mkdirs();
30
                Thumbnails.of(image).size(50,50).outputFormat("png").toFile(thumbnail);
31
32
33
            FileInputStream in = new FileInputStream(thumbnail);
34
           byte[] buffer = new byte[1024 * 8];
35
           while (true) {
36
                int count = in.read(buffer);
37
                if (count == -1) {
38
                    break;
39
40
                out.write(buffer, 0, count);
41
42
           in.close();
43
           out.close();
44
45 }
```

```
☐ result.jsp ⋈
 1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
 2 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"%>
 4⊖<html>
 5⊖<head>
       <meta charset="UTF-8">
       <title>결과창</title>
 8 </head>
 9⊖ <body>
10 <h3>업로드가 완료되었습니다.</h3>
11
       <h3>파일 업로드 하기</h3>
12
       <label>아이디: </label><input type="text" value="${map.id }" readonly="readonly"><br>
       <label>이름: </label><input type="text" value="${map.name }" readonly="readonly"><br>
13
       <div>
14⊖
15⊜
            <c:forEach var="imageFileName" items="${map.fileList }">
16
               <%-- <img src="/pro10/test03/download.do?imageFileName=${imageFileName }"><br> --%>
                <imq src="/pro10/test04/download.do?imageFileName=${imageFileName }"><br><br>
17
18
            </c:forEach>
19
       </div>
        <a href="/pro10/test03/form.do">다시 업로드 하기</a>
20
21 </body>
22 </html>
```









### ✓ 받은 이미지 작은 이미지 만들어서 바로 출력하기

```
      ✓ № pro10

      ✓ # src/main/java

      ✓ # com.test.pro10

      > ➡ common

      > ➡ ex01

      > ➡ ex02

      > ➡ ex03

      > ➡ ex04

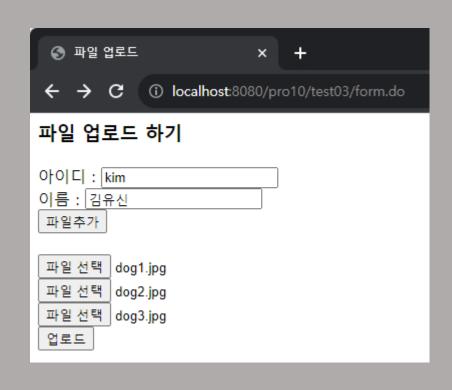
      ✓ ➡ ex05

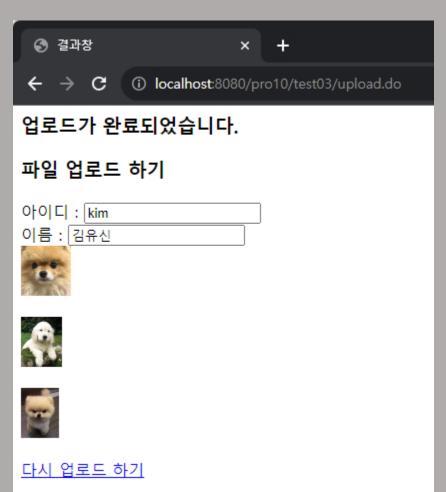
      > ➡ FileDownloadController3.java
```

```
1 package com.test.pro10.ex05;
 3⊕ import java.io.File; ...
16 @Controller
17 public class FileDownloadController3 {
       private static String IMAGE_REPO = "I:\\Spring Source 2\\image_repo";
19⊜
       @RequestMapping(value="/test05/download.do")
20
       public void download(@RequestParam("imageFileName") String imageFileName,
21
               HttpServletRequest request, HttpServletResponse response) throws Exception {
           OutputStream out = response.getOutputStream();
22
           String filePath = IMAGE REPO + "\\" + imageFileName;
23
24
            File image = new File(filePath);
25
           int lastIndex = imageFileName.lastIndexOf(".");
26
           String fileName = imageFileName.substring(0,lastIndex);
27
            File thumbnail = new File(IMAGE REPO+"\\"+"thumbnail"+"\\"+fileName+".png");
28
           if (image.exists()) {
29
30
               Thumbnails.of(image).size(50,50).outputFormat("png").toOutputStream(out);
31
32
33
            FileInputStream in = new FileInputStream(thumbnail);
34
           byte[] buffer = new byte[1024 * 8];
35
           while (true) {
36
               int count = in.read(buffer);
37
               if (count == -1) {
38
                   break;
39
40
               out.write(buffer, 0, count);
41
42
           in.close();
43
           out.close();
44
45 }
```

```
☐ result.jsp 
☐ 
 1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
 2 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"%>
 4⊖<html>
 50 < head>
       <meta charset="UTF-8">
       <title>결과창</title>
 8 </head>
 9⊖ <body>
10 <h3>업로드가 완료되었습니다.</h3>
       <h3>파일 업로드 하기</h3>
11
12
       <label>아이디: </label><input type="text" value="${map.id }" readonly="readonly"><br>
       <label>이름: </label><input type="text" value="${map.name }" readonly="readonly"><br>
13
       <div>
14⊖
15⊜
           <c:forEach var="imageFileName" items="${map.fileList }">
16
               <%-- <imq src="/pro10/test03/download.do?imageFileName=${imageFileName }"><br> --%>
               <%-- <img src="/pro10/test04/download.do?imageFileName=${imageFileName }"><br> --%>
17
               <img src="/pro10/test05/download.do?imageFileName=${imageFileName }"><br><br>
18
19
           </c:forEach>
20
       </div>
21
       <a href="/pro10/test03/form.do">다시 업로드 하기</a>
22 </body>
23 </html>
```







## 4. 스프링 이메일

- ✓ 스프링에서는 이메일 라이브러리를 이용해서 쉽게 이메일 기능을 구현할 수 있다.
- ✓ 구글 SMTP 서버를 이용해 이메일 기능을 구현 할 수 있다.

```
∨ 🔑 src

✓ № main

✓ № webapp

                                                                                               resources

✓ № WEB-INF

                                                                                                                      classes

✓ № spring

                                                                                                                        > 🔑 appServlet
                                                                                                                                               mail-context.xml
                                                                                                                                              root-context.xml

✓ 

    test01

                                                                                                                                                                         result.jsp

✓ 

begin{picture}
    best (1)
    constant (2)
    constant (3)
    constant (3)

                                                                                                                                                                       result.jsp
                                                                                                                                                                         uploadForm.jsp
                                                                                                                                            home.jsp
                                                                                                                        x web.xml
                                            test
                                lmx.mod in
```

```
m pro10/pom.xml ⊠
        properties>
 10⊝
            <java-version>1.6</java-version>
 11
            <org.springframework-version>4.1.1.RELEASE
 12
            <org.aspectj-version>1.6.10</org.aspectj-version>
 13
 14
            <org.slf4j-version>1.6.6</org.slf4j-version>
 15
        </properties>
 16⊖
        <dependencies>
 17
            <!-- Spring -->
 18⊜
            <dependency>
 19
                <groupId>org.springframework</groupId>
 20
                <artifactId>spring-context</artifactId>
 21
                <version>${org.springframework-version}</version>
```



```
m pro10/pom.xml ⋈
 78
                <version>0.4.8
 79
            </dependency>
 80
            <!-- 스프링 이메일 -->
 81
 82⊜
            <dependency>
 83
                <groupId>org.springframework</groupId>
                <artifactId>spring-context-support</artifactId>
 84
 85
                <version>${org.springframework-version}</version>
 86
            </dependency>
 87⊜
            <dependency>
                <groupId>javax.mail</groupId>
 88
                <artifactId>javax.mail-api</artifactId>
 89
 90
                <version>1.5.4
 91
            </dependency>
 92⊜
            <dependency>
 93
                <groupId>com.sun.mail
                <artifactId>javax.mail</artifactId>
 94
 95
                <version>1.5.3
 96
            </dependency>
 97
 98
            <!-- Logging -->
 99⊜
            <dependency>
100
                <groupId>org.slf4j</groupId>
101
                <artifactId>slf4j-api</artifactId>
102
                <version>${org.slf4j-version}</version>
103
            </dependency>
```



```
✓ №5 pro10

  > # src/main/java
  > # src/main/resources
  > 🌁 src/test/java
  > # src/test/resources
  > A JRE System Library [JavaSE-1.6]
  > Maven Dependencies
 🗸 🔑 src
    🗸 🔑 main

✓ № webapp

           resources

✓ № WEB-INF

              classes
            > 🔑 spring
             views
             x web.xml
      test
  > 📂 target

    pom.xml
```

```
x web.xml ⋈
 1 <?xml version="1.0" encoding="UTF-8"?>
 20 <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 https://java.sun.com/xml/ns/javaee/web-app 2 5.xsd">
       <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
 7⊝
       <context-param>
 8
           <param-name>contextConfigLocation</param-name>
 9⊜
               /WEB-INF/spring/*.xml
10
11
           </paralle-value>
12
       </context-param>
13
       <!-- Creates the Spring Container shared by all Servlets and Filters -->
14
15⊜
       <listener>
           tener-class>org.springframework.web.context.ContextLoaderListener-
16
17
        </listener>
```



```
✓ №5 pro10

  > # src/main/java
  > # src/main/resources
  > # src/test/java
  > # src/test/resources
  > M JRE System Library [JavaSE-1.6]
  > Maven Dependencies

✓ № src

✓ № main

✓ № webapp

           resources

✓ № WEB-INF

              classes

✓ № spring

               > 65 annServlet
                ואן root-context.xmi
            > 🗁 views
              x web.xml
       test
  > 📂 target
    M pom.xml
```

```
x *mail-context.xml ⊠
 1 <?xml version="1.0" encoding="UTF-8"?>
 20 < 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">
 6
 7⊝
      <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
9<sub>i</sub> 8
          0<sub>i</sub> 9
          property name="port" value="465">
0<sub>i</sub>10
          cproperty name="username" value="송신자 이메일 주소">
9<sub>i</sub>11
          cproperty name="password" value="송신자 이메일 비밀 키"></property>
0<sub>i</sub>12⊖
          property name="javaMailProperties">
13⊜
             ops>
                14
15
                prop key="mail.smtp.auth">true
                16
17
                18
                prop key="mail.debug">true
19
             </props>
20
          </property>
21
      </bean>
229
      <bean id="preConfigureMessage" class="org.springframework.mail.SimpleMailMessage">
%i23
          cproperty name="to" value="수신자 이메일 주소"></property>
<mark>9</mark>i24
          roperty name="from" value="송신자 이메일 주소">
% 25
          cproperty name="subject" value="테스트 메일 입니다.">
26
      </bean>
27 </beans>
```



```
1 package com.test.pro10.ex06;
 3 import java.io.PrintWriter;
 5 import javax.servlet.http.HttpServletRequest;
 6 import javax.servlet.http.HttpServletResponse;
 8 import org.springframework.beans.factory.annotation.Autowired;
 9 import org.springframework.stereotype.Controller;
10 import org.springframework.web.bind.annotation.RequestMapping;
11
12 @Controller
13 public class MailController {
       @Autowired
14⊖
15
       private MailService mailService;
16
17⊝
       @RequestMapping(value="/test06/sendMail.do")
18
       public void sendSimpleMail(HttpServletRequest request, HttpServletResponse response) throws Exception {
19
           request.setCharacterEncoding("utf-8");
           response.setContentType("text/html;charset=utf-8");
20
21
22
23
24
25
           PrintWriter out = response.getWriter();
           mailService.sendMail("수신자 이메일", "테스트 메일", "안녕하세요. 테스트 메일 입니다.");
           mailService.sendPreConfigureMail("테스트 메일 입니다.");
           out.print("메일을 보냈습니다.");
26 }
```

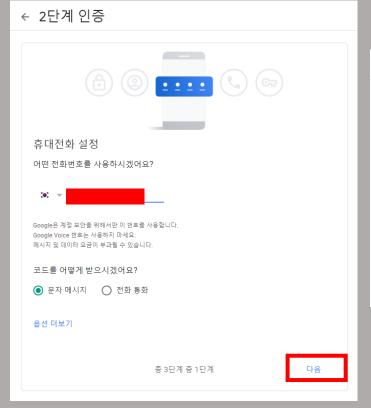


```
1 package com.test.pro10.ex06;
 3⊕ import javax.mail.internet.MimeMessage;
11
12 @Service
13 public class MailService {
        @Autowired
       private JavaMailSender mailSender;
15
16⊖
       @Autowired
       private SimpleMailMessage preConfigureMessage;
17
18
19⊜
        @Async
20
       public void sendMail(String to, String subject, String body) {
21
           MimeMessage message = mailSender.createMimeMessage();
22
           try {
23
               MimeMessageHelper messageHelper =
24
                       new MimeMessageHelper(message, true, "UTF-8");
25
               messageHelper.setCc("참조 이메일 주소");
26
               messageHelper.setFrom("송신자 주소", "송신자 이름");
               messageHelper.setSubject(subject);
27
28
               messageHelper.setTo(to);
29
               messageHelper.setText(body);
30
               mailSender.send(message);
           } catch(Exception e) {
31
32
               e.printStackTrace();
33
34
35⊜
        @Async
36
       public void sendPreConfigureMail(String message) {
           SimpleMailMessage mailMessage = new SimpleMailMessage(preConfigureMessage);
37
           mailMessage.setText(message);
38
           mailSender.send(mailMessage);
39
40
41 }
```

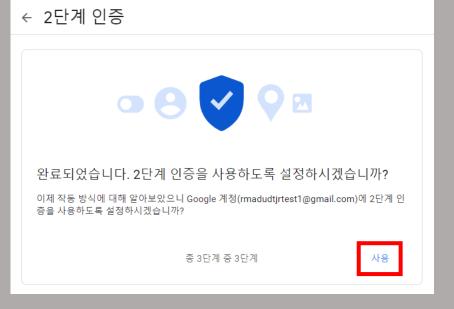


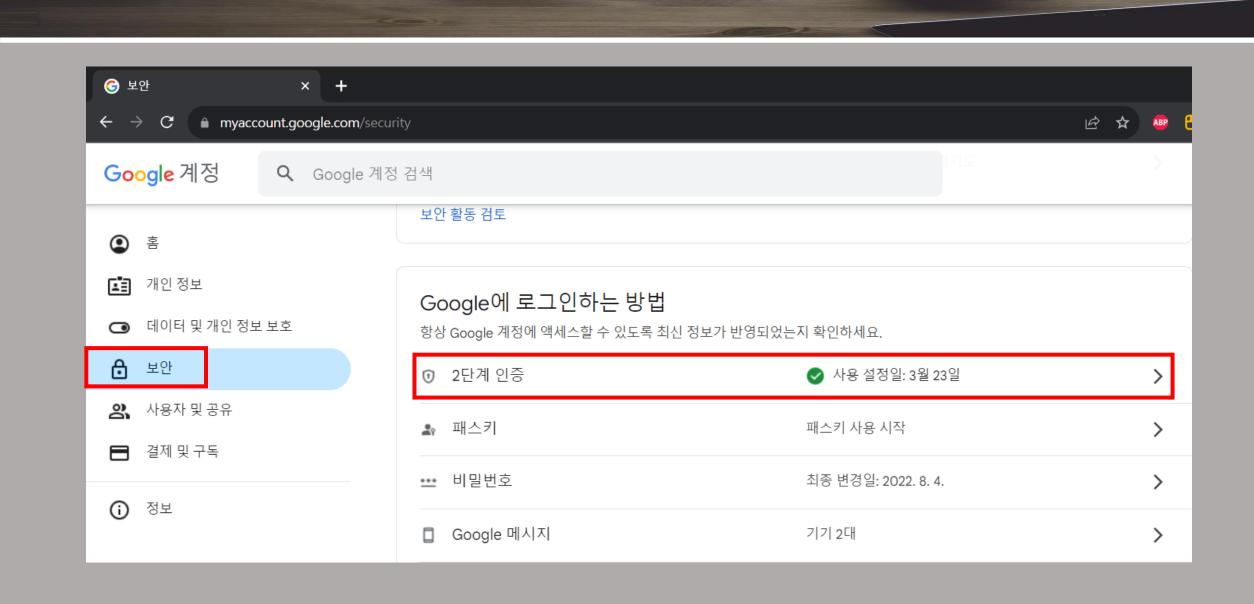
### ✓ 지메일 앱 비밀번호 만들기

URL: <a href="https://myaccount.google.com/">https://myaccount.google.com/</a>











# ← 2단계 인증 인증 2단계가 필요 없는 기기 개인 컴퓨터와 같이 신뢰할 수 있는 기기에서 두 번째 단계를 건너될 수 있습니다. ▲ 신뢰하는 기기 2단계 인증을 건너워도록 설정한 기기에서 신뢰할 수 있음 상태를 취소합니다. 모두 취소 앱 비밀번호 앱 비밀번호는 권장되지 않으며 대부분의 경우 필요하지 않습니다. 계정을 안전하게 보호하려면 'Google 계정으로 로그인을 사용하여 업을 Google 계정에 연결하세요. 앱 비밀번호 없음

# ← 앱비밀번호를 사용하면 최신 보안 표준을 지원하지 않는 오래된 앱 및 서비스에서 Google 계정에 로그인할 수 있습니다. 앱 비밀번호는 최신 보안 표준을 사용하는 최신 앱 및 서비스보다 보안 수준이 낮습니다. 앱 비밀번호를 만들기 전에 앱에 로그인하려면 비밀번호가 필요한지확인해야 합니다. 자세히 알아보기 앱 비밀번호가 없습니다. To create a new app specific password, type a name for it below... App name EmailTest

생성된 앱 비밀번호 키를 복사해서 저장할 것

사용 방법
설정하려는 애플리케이션 또는 기기의 Google 계정 설정으로 이동합니다. 비밀번호를 위에 표시된 16자리 비밀번호로 교체합니다.
일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되므로 적어 놓거나 다른 사용자와 공유하지 마세요.

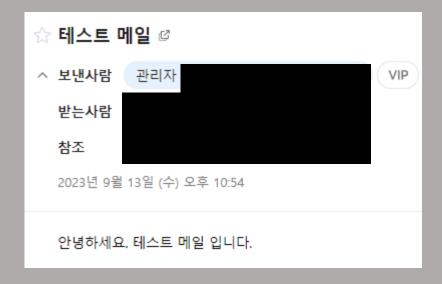


```
x *mail-context.xml 🖂
 1 <?xml version="1.0" encoding="UTF-8"?>
 29 < beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 3
 4
      xsi:schemaLocation="http://www.springframework.org/schema/beans
 5
      http://www.springframework.org/schema/beans/spring-beans.xsd">
 6
 7⊝
       <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
9<sub>i</sub> 8
          property name="host" value="smtp.qmai.com">
0i 9
          roperty name="port" value="465">
9<sub>i</sub>10
          property name="username" value=".
                                                   "></property>
<sub>0</sub>11
          property name="password" value=".
                                                     "></property>
          property name="javaMailProperties">
% 12⊖
                                          password 에 저장한 키를 입력한다.
13⊜
             props>
14
                 15
                 prop key="mail.smtp.auth">true
16
                 17
                 18
                 prop key="mail.debug">true
19
             </props>
20
          </property>
21
      </bean>
22⊖
       <bean id="preConfigureMessage" class="org.springframework.mail.SimpleMailMessage">
%i23
          property name="to" value=":
                                                   "></property>
<sub>0</sub>i24
          property name="from" value=".
                                                        "></property>
%i 25
          roperty name="subject" value="테스트 메일 입니다.">
26
      </bean>
27 </beans>
```









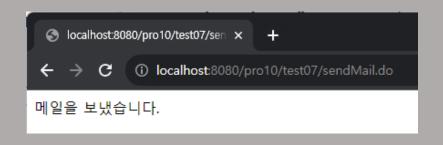
### ✔ HTML 형식으로 이메일 보내기

```
1 package com.test.pro10.ex07;
 3⊕ import java.io.PrintWriter;
12 @Controller
13 public class MailController2 {
       @Autowired
15
       private MailService2 mailService2;
16
17⊝
       @RequestMapping(value="/test07/sendMail.do")
18
       public void sendSimpleMail(HttpServletRequest request, HttpServletResponse response) throws Exception {
19
           request.setCharacterEncoding("utf-8");
20
           response.setContentType("text/html;charset=utf-8");
21
           PrintWriter out = response.getWriter();
22
           String msg = new String();
23
           msq = "<html><body>";
24
           msg +="<h1>내 강아지 보여 줄께요</h1>";
25
26
27
           msq +="<imq src='https://www.dailysecu.com/news/photo/202104/123449 145665 1147.png'>";
           msq +="더 많은 사진은 <a href='https://www.zooseyo.com/main/main.php'>여기를</a> 클릭해";
           msq +="</body></html>";
28
29
           mailService2.sendMail("
                                                      ", "분양 받을래?", msg);
30
           out.print("메일을 보냈습니다.");
31
32 ]
```



```
🔝 MailService2.java 🖂
1 package com.test.pro10.ex07;
 3 import javax.mail.internet.MimeMessage;
 11
 12 @Service
13 public class MailService2 {
14⊖
        @Autowired
15
        private JavaMailSender mailSender;
16⊖
        @Autowired
№17
        private SimpleMailMessage preConfigureMessage;
18
19⊝
        @Async
20
        public void sendMail(String to, String subject, String body) {
21
            MimeMessage message = mailSender.createMimeMessage();
22
            try -
23
                MimeMessageHelper messageHelper =
24
                        new MimeMessageHelper(message, true, "UTF-8");
25
                messageHelper.setCc("
26
                messageHelper.setFrom("
                                                                   "관리자");
27
                messageHelper.setSubject(subject);
28
                messageHelper.setTo(to);
29
                messageHelper.setText(body, true);
30
                mailSender.send(message);
31
            } catch(Exception e) {
32
                e.printStackTrace();
33
34
35 }
```







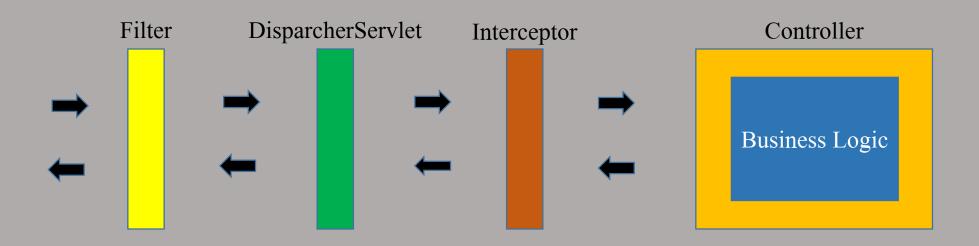
# 5. 스프링 인터셉터

### ✓ 인터셉터(Interceptor)

- 브라우저에서 요청 했을 때 요청 메서드 호출 전후에 개발자가 원하는 기능을 수행하 게 할 수 있다.
- 필터와 기능이 유사하지만 필터보다는 좀 더 자유롭게 위치를 변경해서 기능을 수행하게 할 수 있다는 장점이 있다.
- 쿠키(Cookie) 제어, 파일 업로드 등의 작업을 수행할 수 있다.

## ✓ 스프링의 HandlerInterceptor 클래스의 메서드

메서드	기능
preHandle()	컨트롤러 실행 전 호출 된다.
postHandle()	컨트롤러 실행 후 DispatcherServlet이 뷰로 보내기 전에 호출된다.
afterCompletion()	View 까지 수행하고 나서 호출된다.





### ✓ 인터셉터를 사용한 다국어 기능

```
✓ №5 pro10

                > # src/main/java
                > # src/main/resources
                > # src/test/java
                 > # src/test/resources
                > M JRE System Library [JavaSE-1.6]
                > Maven Dependencies

✓ 

Since

Since
                                🗸 😂 main

✓ № webapp

                                                                                   resources

✓ № WEB-INF

                                                                                                   classes
                                                                                    spring
                                                                                                        SannServlet
                                                                                                                                       servlet-context.xml
                                                                                                                                       mail-context.xml
                                                                                                                        root-context.xml
                                                                                   > 🗁 views
                                                                                                     x web.xml
                                                test
                > 📂 target
                                  m pom.xml
```

```
x *servlet-context.xml 🖂
25⊜
        <beans:bean id="multipartResolver"</pre>
26
             class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
<sub>0</sub>i27
             <beans:property name="maxUploadSize" value="52428800"></beans:property>
0i28
             <beans:property name="maxInMemorySize" value="1000000">/beans:property>
<sub>0</sub>i29
             <beans:property name="defaultEncoding" value="utf-8"></beans:property>
        </beans:bean>
30
        <interceptors>
31⊜
32⊜
             <interceptor>
33
                 <mapping path="/test08/*.do"/>
34
                 <mapping path="/*/*.do"/>
                 <beans:bean class="com.test.pro10.ex08.LocaleInterceptor"></beans:bean>
@35
36
             </interceptor>
37
        </interceptors>
    </beans:beans>
```



```
> # src/main/java
  > # src/main/resources
  > # src/test/java
  > # src/test/resources
  > A JRE System Library [JavaSE-1.6]
  > Maven Dependencies

✓ 

Src

→ 

→ 

main

       resources

✓ 

WEB-INF

              classes

✓ 

Spring

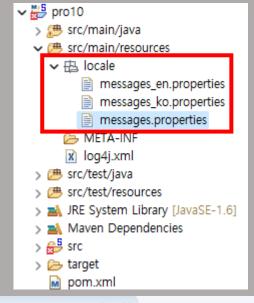
              > 🚑 appServlet
                55 mail-context yml
                   message-context.xn
                 AT 1001-CONTEXT.XIII
            > 📂 views
              x web.xml
      test
  > 📂 target

    pom.xml
```

```
🗷 *message-context.xml 🖂
 1 <?xml version="1.0" encoding="UTF-8"?>
 20 < beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  3
        xsi:schemaLocation="http://www.springframework.org/schema/beans
 5
        http://www.springframework.org/schema/beans/spring-beans.xsd">
  6
 7⊝
        <bean id="localeResolver"</pre>
 8
            class="org.springframework.web.servlet.i18n.SessionLocaleResolver"></bean>
        <bean id="messageSource"</pre>
 9⊝
10
            class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
9<sub>i</sub> 11⊖
            property name="basenames">
12⊖
                st>
13
                    <value>classpath:locale/messages</value>
                </list>
14
15
            </property>
9i16
            cproperty name="defaultEncoding" value="UTF-8">
            cacheSeconds" value="60">
<sub>9</sub>i17
18
        </bean>
19 </beans>
```



회원 정보 이름 직업 홍길동 학생 보내기 취소 종료



messages\_ko.properties 
1 site.title=\uD68C\uC6D0\uC815\uBCF4
2 site.name=\uC774\uB984
3 site.job=\uC9C1\uC5C5
4 name=\uD64D\uAE38\uB3D9
5 job=\uC9C1\uC5C5
6 btn.send=\uBCF4\uB0B4\uAE30
7 btn.cancel=\uCDE8\uC18C
8 btn.finish=\uC885\uB8CC

messages\_en.properties \( \text{S} \)

1 site.title=Member Info
2 site.name=Kil-Dong HOng
3 site.job=Student
4 name=NAME
5 job=JOB
6 btn.send=SEND
7 btn.cancel=CANCEL
8 btn.finish=FINISH

\*messages.properties \( \)
1 site.title=\uD68C\uC6D0\uC815\uBCF4
2 site.name=\uC774\uB984
3 site.job=\uC9C1\uC5C5
4 name=\uD64D\uAE38\uB3D9
5 job=\uC9C1\uC5C5
6 btn.send=\uBCF4\uB0B4\uAE30
7 btn.cancel=\uCDE8\uC18C
8 btn.finish=\uC885\uB8CC



```
1 package com.test.pro10.ex08;
 3@import javax.servlet.http.HttpServletRequest;
 4 import javax.servlet.http.HttpServletResponse;
   import org.springframework.stereotype.Controller;
   import org.springframework.web.bind.annotation.RequestMapping;
   @Controller
10 public class LocaleController {
11⊖
       @RequestMapping("/test08/locale.do")
12
       public String locale (HttpServletRequest request,
13
               HttpServletResponse response) throws Exception {
14
           System.out.println("localeController 입니다.");
15
           return "test08/locale";
16
17 }
```



```
1 package com.test.pro10.ex08;
  3⊕ import java.util.Locale;
10
11 public class LocaleInterceptor extends HandlerInterceptorAdapter {
12
13⊝
        @Override
14
15
16
17
18
19
20
21
       public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) {
            HttpSession session=request.getSession();
            String locale=request.getParameter("locale");
            if(locale==null)
                locale="ko";
            session.setAttribute("org.springframework.web.servlet.i18n.SessionLocaleResolver.LOCALE", new Locale(locale));
            return true;
22 }
```



```
✓ № src

✓ № main

✓ № webapp

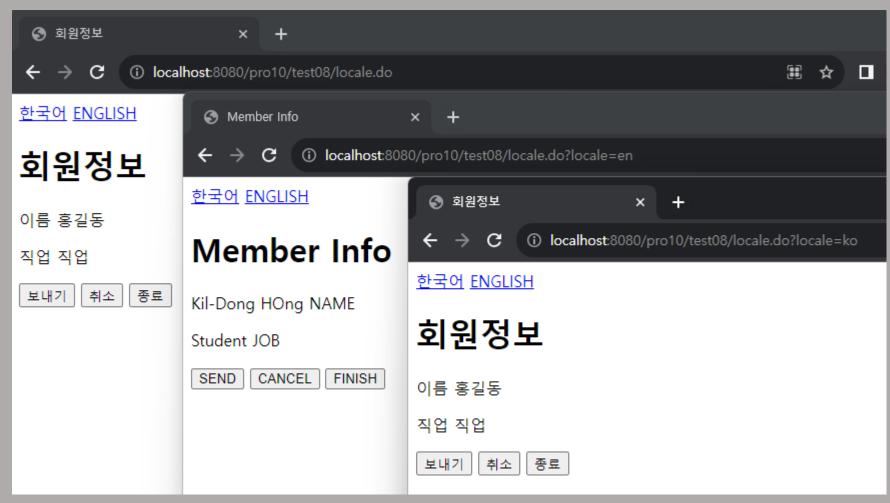
        resources

✓ № WEB-INF

          classes
        > 😂 spring
        views
          > / test01
          > 🗁 test02
          locale.jsp
            home.jsp
          x web.xml
   test
> 📂 target
  M pom.xml
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8" isELIgnored="false"%>
  3 < 0 taglib prefix="spring" uri="http://www.springframework.org/tags" %>
 4 <!DOCTYPE html>
 5⊖<html>
 6⊖ <head>
       <meta charset="UTF-8">
       <title><spring:message code="site.title" text="Member Info" /></title>
 9 </head>
10⊖<body>
11
       <a href="/pro10/test08/locale.do?locale=ko">한국어</a>
12
       <a href="/pro10/test08/locale.do?locale=en">ENGLISH</a>
13
       <h1><spring:message code="site.title" text="Member Info"></spring:message></h1>
14⊖
       <spring:message code="site.name" text="no name" />
15
           <spring:message code="name" text="no name" />
16⊖
       <spring:message code="site.job" text="no job" />
17
           <spring:message code="job" text="no job" />
18
       <input type="button" value="<spring:message code='btn.send'/>">
       <input type="button" value="<spring:message code='btn.cancel'/>">
19
       <input type="button" value="<spring:message code='btn.finish'/>">
21 </body>
22 </html>
```







### ✔ 인터셉터를 사용해 요청명에서 View 이름 가져오기

```
✓ № pro10

                 > # src/main/java
                 > # src/main/resources
                 > # src/test/java
                 > # src/test/resources
                 > M JRE System Library [JavaSE-1.6]
                 > Maven Dependencies

✓ 

Since

Since
                               🗸 🔑 main
                                                  🗸 🐸 webapp
                                                                                 resources

✓ № WEB-INF

                                                                                                 classes

✓ № spring

                                                                                                   Servlet

    servlet-context.xml

                                                                                                                    mail-context.xml אָּגְ
                                                                                                                                      message-context.xml
                                                                                                                  √5 root-context.xml
                                                                                  > 🗁 views
                                                                                                    x web.xml
                                                test
                 > 📂 target
                                  M pom.xml
```

```
x *servlet-context.xml ⋈
        <interceptors>
31⊖
32⊝
            <interceptor>
33
                <mapping path="/test08/*.do"/>
34
                <mapping path="/*/*.do"/>
35
                <beans:bean class="com.test.pro10.ex08.LocaleInterceptor"></beans:bean>
36
            </interceptor>
37
        </interceptors>
38⊜
        <interceptors>
39⊜
            <interceptor>
                <mapping path="/*/*.do"/>
40
                <beans:bean class="com.test.pro10.common.ViewNameInterceptor"></beans:bean>
Q41
            </interceptor>
42
43
        </interceptors>
    </beans:beans>
```



```
Strict

Str
```

```
1 package com.test.pro10.common;
 3⊕ import javax.servlet.http.HttpServletRequest;
 8 public class ViewNameInterceptor extends HandlerInterceptorAdapter{
 9
10⊝
        @Override
△11
       public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
12
               throws Exception {
213
           // TODO Auto-generated method stub
14
           try {
15
               String viewName = getViewName(request);
16
               request.setAttribute("viewName", viewName);
17
           } catch (Exception e) {
18
               e.printStackTrace();
19
20
           return true;
21
229
       private String getViewName(HttpServletRequest request) {
23
           String contextPath = request.getContextPath();
24
           String uri = (String) request.getAttribute("javax.servlet.include.request uri");
25
26
27
           if (uri == null || uri.trim().equals("")) {
               uri = request.getRequestURI();
```

```
uri = request.getRequestURI();
           int begin = 0;
30
           if (!((contextPath == null) || ("".equals(contextPath)))) {
               begin = contextPath.length();
           int end;
           if (uri.indexOf(";") != -1) {
36
               end = uri.indexOf(";");
           } else if (uri.indexOf("?") != -1) {
               end = uri.indexOf("?");
           } else {
40
               end = uri.length();
41
42
43
           String fileName = uri.substring(begin, end);
44
           if (fileName.indexOf(".") != -1) {
               fileName = fileName.substring(0, fileName.lastIndexOf("."));
46
47
           if (fileName.lastIndexOf("/") != -1) {
48
               fileName = fileName.substring(fileName.lastIndexOf("/", 1), fileName.length());
49
50
           return fileName;
52 }
```



