

# JSP Email 보내기

1. pom.xml 의존성 라이브러리를 추가한다.

```
<!-- 파일 업로드 -->
<dependency>
    <groupId>com.jfinal</groupId>
    <artifactId>cos</artifactId>
    <version>2017.5</version>
</dependency>
<!-- 메일 보내기 -->
<!-- https://mvnrepository.com/artifact/javax.mail/mail -->
<dependency>
    <groupId>javax.mail</groupId>
    <artifactId>mail</artifactId>
    <version>1.4</version>
</dependency>
```

2. 폼을 만든다. (mailForm1.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <div>
        <form action="sendMail.jsp" method="post">
            <table>
                <tr>
                    <th colspan="2">JSP 메일 보내기</th>
                </tr>
                <tr>
                    <td>from</td>
                    <td><input type="text" name="from" /></td>
                </tr>
                <tr>
                    <td>to</td>
                    <td><input type="text" name="to" /></td>
                </tr>
                <tr>
                    <td>subject</td>
                    <td><input type="text" name="subject" /></td>
                </tr>
                <tr>
                    <td>content</td>
```

```

        <td><textarea name="content"
                    style="width: 170px; height: 200px;"></textarea></td>
    </tr>
    <tr>
        <td colspan="2" style="text-align: right;"><input
            type="submit" value="메일보내기" /></td>
    </tr>
</table>
</form>
</div>
</body>
</html>

```

3. 인증 클래스를 작성한다.(naver메일이나 gmail을 이용하면 된다.)

```

package email.util;

import javax.mail.Authenticator;
import javax.mail.PasswordAuthentication;

public class SMTPAuthentication extends Authenticator{
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        // 자신의 naver나 gmail계정 정보를 넣는다.
        return new PasswordAuthentication("사용자아이디", "비밀번호");
    }
}

```

4. 메일 발송 jsp파일을 만든다.(sendMail.jsp)

```

<%@page import="javax.mail.PasswordAuthentication"%>
<%@page import="email.util.SMTPTAuthentication"%>
<%@page import="javax.mail.Transport"%>
<%@page import="javax.mail.Message"%>
<%@page import="javax.mail.Address"%>
<%@page import="javax.mail.internet.InternetAddress"%>
<%@page import="javax.mail.internet.MimeMessage"%>
<%@page import="javax.mail.Session"%>
<%@page import="javax.mail.Authenticator"%>
<%@page import="java.util.Properties"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
request.setCharacterEncoding("utf-8");

// 입력값 받음
String from = request.getParameter("from");
String to = request.getParameter("to");
String subject = request.getParameter("subject");
String content = request.getParameter("content");

```

```

// SMTP 서버에 접속하기 위한 정보들
Properties p = new Properties(); // 정보를 담을 객체
p.put("mail.smtp.host", "smtp.naver.com"); // 네이버 SMTP 또는 gmail SMTP
p.put("mail.smtp.port", "465");
p.put("mail.smtp.starttls.enable", "true");
p.put("mail.smtp.auth", "true");
p.put("mail.smtp.debug", "true");
p.put("mail.smtp.socketFactory.port", "465");
p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
p.put("mail.smtp.socketFactory.fallback", "false");

try{
    /* 인증 클래스를 익명 클래스로 작성해도 된다.
    Session ses = Session.getInstance(p, new Authenticator(){
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication("사용자아이디", "비번");
        }
    });
    */
    Authenticator auth = new SMTPAuthenticator();
    Session ses = Session.getInstance(p, auth);
    ses.setDebug(true);

    MimeMessage msg = new MimeMessage(ses); // 메일의 내용을 담을 객체
    msg.setSubject(subject); // 제목

    Address fromAddr = new InternetAddress(from);
    msg.setFrom(fromAddr); // 보내는 사람

    Address toAddr = new InternetAddress(to);
    msg.addRecipient(Message.RecipientType.TO, toAddr); // 받는 사람

    msg.setContent(content, "text/html;charset=UTF-8"); // 내용과 인코딩

    Transport.send(msg); // 전송
} catch(Exception e){
    e.printStackTrace();
    out.println("<script>alert('Send Mail Failed..');history.back();</script>");
    // 오류 발생시 뒤로 돌아가도록
    return;
}

out.println("<script>alert('Send Mail Success!!');location.href='mailForm.jsp';</script>");
// 성공 시
%>

```

##### 5. 파일을 첨부해보자(서버에 있는 파일 첨부)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

```

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <div>
        <form action="sendMail2.jsp" method="post">
            <table>
                <tr>
                    <th colspan="2">JSP 메일 보내기</th>
                </tr>
                <tr>
                    <td>from</td>
                    <td><input type="text" name="from" /></td>
                </tr>
                <tr>
                    <td>to</td>
                    <td><input type="text" name="to" /></td>
                </tr>
                <tr>
                    <td>subject</td>
                    <td><input type="text" name="subject" /></td>
                </tr>
                <tr>
                    <td>content</td>
                    <td><textarea name="content"
                        style="width: 170px; height: 200px;"></textarea></td>
                </tr>
                <tr>
                    <td colspan="2" style="text-align: right;"><input
                        type="submit" value="메일보내기" /></td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>

```

## 6. 파일첨부하여 메일 보내기

```

<%@page import="javax.activation.DataHandler"%>
<%@page import="javax.activation.FileDataSource"%>
<%@page import="java.io.File"%>
<%@page import="javax.mail.internet.MimeMultipart"%>
<%@page import="javax.mail.Multipart"%>
<%@page import="javax.mail.internet.MimeBodyPart"%>
<%@page import="javax.mail.BodyPart"%>
<%@page import="javax.mail.PasswordAuthentication"%>
<%@page import="email.util.SMTPAuthentication"%>

<%@page import="javax.mail.Transport"%>

```

```

<%@page import="javax.mail.Message"%>
<%@page import="javax.mail.Address"%>
<%@page import="javax.mail.internet.InternetAddress"%>
<%@page import="javax.mail.internet.MimeMessage"%>
<%@page import="javax.mail.Session"%>
<%@page import="javax.mail.Authenticator"%>
<%@page import="java.util.Properties"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
request.setCharacterEncoding("utf-8");

// 입력값 받음
String from = request.getParameter("from");
String to = request.getParameter("to");
String subject = request.getParameter("subject");
String content = request.getParameter("content");

// SMTP 서버에 접속하기 위한 정보들
Properties p = new Properties(); // 정보를 담을 객체
p.put("mail.smtp.host", "smtp.naver.com"); // 네이버 SMTP
p.put("mail.smtp.port", "465");
p.put("mail.smtp.starttls.enable", "true");
p.put("mail.smtp.auth", "true");
p.put("mail.smtp.debug", "true");
p.put("mail.smtp.socketFactory.port", "465");
p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
p.put("mail.smtp.socketFactory.fallback", "false");

try{
    /*
    Session ses = Session.getInstance(p, new Authenticator(){
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication("사용자아이디", "비번");
        }
    });
    */
    Authenticator auth = new SMTPAuthenticator();
    Session ses = Session.getInstance(p, auth);
    ses.setDebug(true);

    MimeMessage msg = new MimeMessage(ses); // 메일의 내용을 담을 객체
    msg.setSubject(subject); // 제목

    Address fromAddr = new InternetAddress(from);
    msg.setFrom(fromAddr); // 보내는 사람

    Address toAddr = new InternetAddress(to);
    msg.addRecipient(Message.RecipientType.TO, toAddr); // 받는 사람
    //-----
    // 1. 파일을 첨부할 객체

    Multipart multipart = new MimeMultipart();

```

```

// 2. 메시지를 저장할 객체
BodyPart messageBodyPart = new MimeBodyPart();
// 3. 몸체 내용을 지정
messageBodyPart.setContent(content, "text/html;charset=UTF-8"); // html 전송
// messageBodyPart.setText(content); // 텍스트 전송

// 4. 파일을 첨부할 객체에 추가
multipart.addBodyPart(messageBodyPart);

// 5. 파일을 첨부 한다.
BodyPart filePart = new MimeBodyPart();
File file = new File(application.getRealPath("mailForm.jsp"));
FileDataSource fds = new FileDataSource(file);
filePart.setDataHandler(new DataHandler(fds));
filePart.setFileName(fds.getName());

// 6. 파일을 첨부할 객체에 추가
multipart.addBodyPart(filePart);

// 7. 만들어진 Multipart객체를 보낼 메일에 지정한다
msg.setContent(multipart);
//-----
// 8. 실제 전송
Transport.send(msg); // 전송
} catch(Exception e){
    e.printStackTrace();
    out.println("<script>alert('Send Mail Failed..');history.back();</script>");
    // 오류 발생시 뒤로 돌아가도록
    return;
}

out.println("<script>alert('Send Mail Success!!');location.href='mailForm2.jsp';</script>");
// 성공 시
%>

```

## 7. 서버로 파일을 전송하여 첨부하기

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <div>
        <form action="sendMail3.jsp" method="post" enctype="multipart/form-data">
            <table>
                <tr>
                    <th colspan="2">JSP 메일 보내기</th>
                </tr>
            </table>
        </form>
    </div>

```

```

        <tr>
            <td>from</td>
            <td><input type="text" name="from" /></td>
        </tr>
        <tr>
            <td>to</td>
            <td><input type="text" name="to" /></td>
        </tr>
        <tr>
            <td>subject</td>
            <td><input type="text" name="subject" /></td>
        </tr>
        <tr>
            <td>content</td>
            <td><textarea name="content"
                style="width: 170px; height: 200px;"></textarea></td>
        </tr>
        <tr>
            <td>첨부파일</td>
            <td>
                <input type="file" name="file1"><br>
                <input type="file" name="file2"><br>
                <input type="file" name="file3"><br>
            </td>
        </tr>
        <tr>
            <td colspan="2" style="text-align: right;"><input
                type="submit" value="메일보내기" /></td>
        </tr>
    </table>
</form>
</div>
</body>
</html>

```

## 8. 파일 업로드하여 메일 전송하기

```

<%@page import="javax.mail.internet.MimeUtility"%>
<%@page import="java.net.URLEncoder"%>
<%@page import="java.util.Enumeration"%>
<%@page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%@page import="com.oreilly.servlet.MultipartRequest"%>
<%@page import="javax.activation.DataHandler"%>
<%@page import="javax.activation.FileDataSource"%>
<%@page import="java.io.File"%>
<%@page import="javax.mail.internet.MimeMultipart"%>
<%@page import="javax.mail.Multipart"%>
<%@page import="javax.mail.internet.MimeBodyPart"%>
<%@page import="javax.mail.BodyPart"%>
<%@page import="javax.mail.PasswordAuthentication"%>
<%@page import="email.util.SMTPAuthentication"%>

<%@page import="javax.mail.Transport"%>

```

```

<%@page import="javax.mail.Message"%>
<%@page import="javax.mail.Address"%>
<%@page import="javax.mail.internet.InternetAddress"%>
<%@page import="javax.mail.internet.MimeMessage"%>
<%@page import="javax.mail.Session"%>
<%@page import="javax.mail.Authenticator"%>
<%@page import="java.util.Properties"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
request.setCharacterEncoding("utf-8");

// cos-fileupload 라이브러리를 사용하여 파일을 업로드 한다.
String upload_path = application.getRealPath("upload"); // 업로드할 경로
int size = 10 * 1024 * 1024; // 업로드 크기 제한
int filesize = 0;
// SMTP 서버에 접속하기 위한 정보들
Properties p = new Properties(); // 정보를 담을 객체
p.put("mail.smtp.host", "smtp.naver.com"); // 네이버 SMTP
p.put("mail.smtp.port", "465");
p.put("mail.smtp.starttls.enable", "true");
p.put("mail.smtp.auth", "true");
p.put("mail.smtp.debug", "true");
p.put("mail.smtp.socketFactory.port", "465");
p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
p.put("mail.smtp.socketFactory.fallback", "false");

Authenticator auth = new SMTPAuthenticator();
Session ses = Session.getInstance(p,auth);
ses.setDebug(true);

try {
    // 이순간 모드 업로드가 이루어진다.
    MultipartRequest multi =
        new MultipartRequest(request, upload_path, size, "utf-8", new
DefaultFileRenamePolicy());

    // 폼필드의 내용을 읽자.
    String from = multi.getParameter("from");
    String to = multi.getParameter("to");
    String subject = multi.getParameter("subject");
    String content = multi.getParameter("content");

    MimeMessage msg = new MimeMessage(ses); // 메일의 내용을 담을 객체
    msg.setSubject(subject); // 제목

    Address fromAddr = new InternetAddress(from);
    msg.setFrom(fromAddr); // 보내는 사람

    Address toAddr = new InternetAddress(to);
    msg.addRecipient(Message.RecipientType.TO, toAddr); // 받는 사람

    // 1. 파일을 첨부할 객체

```



```

Multipart multipart = new MimeMultipart();
//=====
// 본문
// 2. 메시지를 저장할 객체
BodyPart messageBodyPart = new MimeBodyPart();
// 3. 몸체 내용을 지정
messageBodyPart.setContent(content, "text/html;charset=UTF-8"); // html 전송
// messageBodyPart.setText(content); // 텍스트 전송

// 4. 파일을 첨부할 객체에 추가
multipart.addBodyPart(messageBodyPart);
//=====
// 파일 첨부
// 파일 처리를 하자
Enumeration<String> files = multi.getFileNames(); // 필드명 리스트
while(files.hasMoreElements()){
    String fieldName = files.nextElement();
    File file = multi.getFile(fieldName);
    if(file!=null){
        // 5. 파일을 첨부 한다.
        BodyPart filePart = new MimeBodyPart();
        FileDataSource fds = new FileDataSource(file);
        filePart.setDataHandler(new DataHandler(fds));
        // 한글 파일이름 깨지게 하지 않기
        filePart.setFileName(MimeUtility.encodeText(multi.getOriginalFileName(fieldName)));
        // 6. 파일을 첨부할 객체에 추가
        multipart.addBodyPart(filePart);
    }
}
// 7. 만들어진 Multipart객체를 보낼 메일에 지정한다
msg.setContent(multipart);
// 8. 실제 전송
Transport.send(msg); // 전송
} catch (Exception e) {
    e.printStackTrace();
    out.println("<script>alert('Send Mail Failed..');history.back();</script>");
    // 오류 발생시 뒤로 돌아가도록
    return;
}
//성공 시
out.println("<script>alert('Send Mail Success!!');location.href='mailForm3.jsp';</script>");
%>

```

- maven을 사용하지 않을 경우에는 두개의 라이브러리를 WEB-INF/lib 폴더로 복사해야 한다.

mail-1.4.jar , activation-1.1.jar