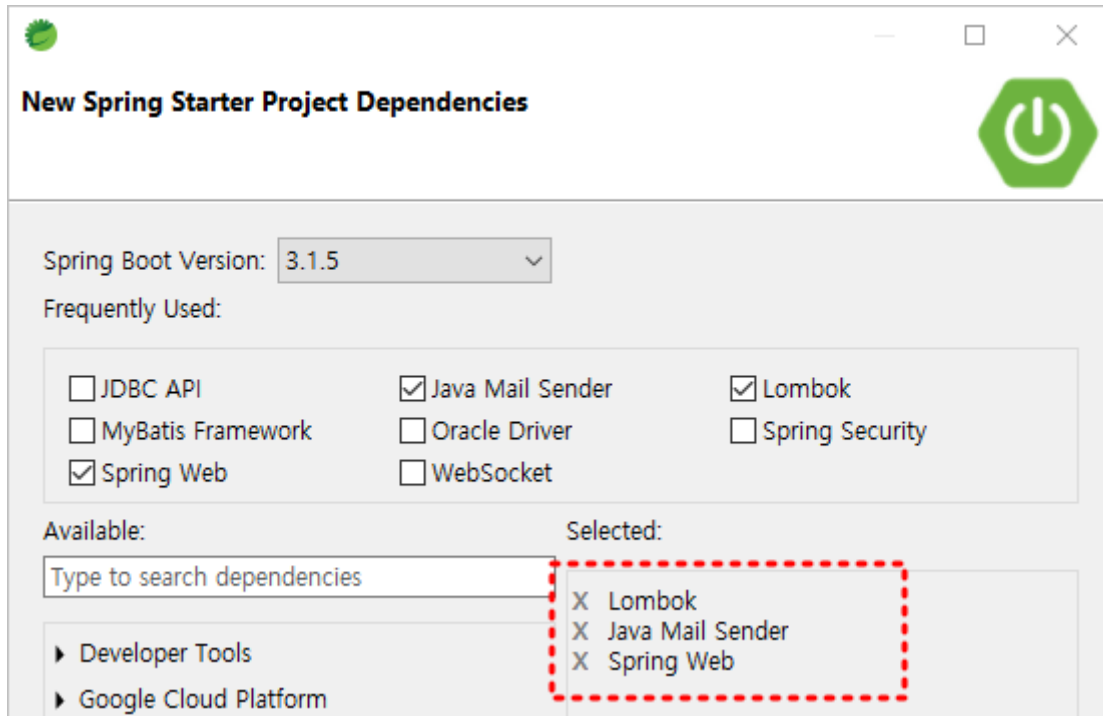


네이버 SMTP 사용(with properties)

프로젝트명 : B15aSMTP_Naver

준비사항

- 의존설정 : Spring Web, Lombok, Java Mail Sender
- JSP 사용을 위한 설정을 한다.
- Refresh Gradle Project 를 눌러 적용한다.



SMTP란..??

- Simple Mail Transfer Protocol의 약자
- 인터넷을 통해 이메일을 보내고 받는데 사용되는 통신 프로토콜
- 메일 서버 및 기타 메시지 전송 에이전트(MTA)는 SMTP를 사용하여 메일 송수신한다.

Naver 설정



| | | |
|---|----------------------------|------------------------------|
| 메일 프로그램 환경 설정 안내 스마트폰, 아웃룩 등 외부 메일 프로그램 환경설정에 아래와 같이 등록해 주세요. | | |
| 계정 정보 | 아이디 : nakjasabal | 비밀번호 : 네이버 로그인 비밀번호 |
| 서버 명 | POP 서버명 : pop.naver.com | SMTP 서버명 : smtp.naver.com |
| 포트 정보 | POP 포트 : 995, 보안연결(SSL) 필요 | SMTP 포트 : 465, 보안 연결(SSL) 필요 |

뷰 : webapp/WEB-INF/views/home.jsp

```

9<body>
10    <h2>Naver SMTP 서버를 활용한 이메일 전송</h2>
11    <ul>
12        <li><a href="/">루트</a></li>
13        <li><a href="/emailSendMain.do">이메일보내기</a></li>
14    </ul>
15 </body>
16 </html>

```

컨트롤러 : com.edu.springboot.MainController.java

```

27    @GetMapping("/emailSendMain.do")
28    public String emailSendMain() {
29        return "emailSendMain";
30    }
31
32    @Autowired
33    EmailSending email;
34
35    @PostMapping("/emailSendProcess.do")
36    public String emailSendProcess(InfoDTO infoDTO) {
37        email.myEmailSender(infoDTO);
38        return "home";
39    }
40 }

```

DTO : com.study.springboot.InfoDTO.java

```
1 package com.edu.springboot.smtp;
2
3 import lombok.Data;
4
5 @Data
6 public class InfoDTO {
7     private String mailServer;
8     private String from;
9     private String to;
10    private String subject;
11    private String format;
12    private String content;
13 }
```

뷰 : webapp/WEB-INF/views/emailSendMain.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>SMTP 이메일 전송</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(function(){
    let naMail = "네이버아이디@naver.com";//보내는 사람
    let goMail = "구글아이디@gmail.com";//받는 사람

    $('input:text[name="from"]').val(naMail);
    $('input:text[name="to"]').val(goMail);
});
</script>
</head>
<body>
<h2>이메일 전송하기</h2>
<form method="post" action="emailSendProcess.do">
<table border=1>
    <tr>
        <td>
```

```
보내는 사람 : <input type="text" name="from" />
</td>
</tr>
<tr>
<td>
받는 사람 : <input type="text" name="to" />
</td>
</tr>
<tr>
<td>
제목 : <input type="text" name="subject" size="50"
value="제목은 제목일뿐" />
</td>
</tr>
<tr>
<td>
형식 :
<input type="radio" name="format"
value="text" checked />Text
<input type="radio" name="format"
value="html" />HTML
</td>
</tr>
<tr>
<td>
<textarea name="content" cols="60"
rows="10">내용은 내용일뿐</textarea>
</td>
</tr>
<tr>
<td>
<button type="submit">전송하기</button>
</td>
</tr>
</table>
</form>
</body>
</html>
```

여기까지 작성하세요.

프로퍼티 : resources/application.properties

```
8 # SMTP 설정
9 spring.mail.host=smtp.naver.com
10 spring.mail.port=465
11 spring.mail.username=[redacted]@naver.com 네이버 아이디
12 spring.mail.password=[redacted] 네이버 비밀번호
13
14 spring.mail.properties.mail.debug=true
15 spring.mail.properties.mail.smtp.auth=true
16 spring.mail.properties.mail.smtp.ssl.enable=true
17 spring.mail.properties.mail.smtp.ssl.trust=smtp.naver.com
18 |
```

이메일전송 : com.study.springboot.EmailSending.java

```
1 package com.edu.springboot.smtp;
2
3 import java.io.BufferedReader;
14
15 @Component
16 @RequiredArgsConstructor
17 public class EmailSending {
18
19     private final JavaMailSender mailSender;
20
21     @Value("${spring.mail.username}")
22     private String from;
23
24     public void myEmailSender(InfoDTO infoDTO) {
25
26         try {
27             MimeMessage m = mailSender.createMimeMessage();
28             MimeMessageHelper h = new MimeMessageHelper(m, "UTF-8");
29
30             h.setFrom(from);
31             h.setTo(infoDTO.getTo());
32             h.setSubject(infoDTO.getSubject());
33             if (infoDTO.getFormat().equals("text")) {
34                 h.setText(infoDTO.getContent());
35             }
36         }
37     }
38 }
```

```

36         else {
37             String emailTpl = readHTMLFile();
38             String contents = infoDTO.getContent()
39                 .replace("\r\n", "<br/>");
40             emailTpl = emailTpl.replace("__CONTENT__", contents);
41             h.setText(emailTpl, true);
42         }
43
44         mailSender.send(m);
45         System.out.println("메일 전송 완료..!!");
46     }
47     catch (Exception e) {
48         e.printStackTrace();
49     }
50 }
51

```

```

52 public String readHTMLFile() {
53     StringBuffer htmlContents = new StringBuffer();
54     try {
55         String uploadDir = ResourceUtils
56             .getFile("classpath:static/").toPath().toString();
57         String templatePath = uploadDir+"/MailTemplate.html";
58         BufferedReader br = new BufferedReader(
59             new FileReader(templatePath));
60         String oneLine;
61         while ((oneLine = br.readLine()) != null) {
62             htmlContents.append(oneLine + "\n");
63         }
64         br.close();
65     }
66     catch (Exception e) {
67         e.printStackTrace();
68     }
69     return htmlContents.toString();
70 }
71 }

```

메일템플릿 : resources/static/MailTemplate.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>MustHave 메일 템플릿</title>
6 </head>
7 <body>
8     <h2>MustHave 메일 템플릿</h2>
9     <table border=1 width="100%">
10         <tr>
11             <td width="50">내용</td>
12             <td>__CONTENT__</td>
13         </tr>
14         <tr>
15             <td>이미지</td>
16             <td><img src="https://github.com/goldenrabbit2020/must.
17         </tr>
18     </table>
19 </body>
20 </html>
```

이미지URL :

https://github.com/goldenrabbit2020/musthave_jsp/blob/main/GOLDEN-RABBIT_LOGO_150.png?raw=true