

23장 이메일 인증번호로 비밀번호 변경

- 참고: <https://smujihoon.tistory.com/96>

23.1 라이브러리 추가

- 이메일 관련 라이브러리(javax.mail.jar)를 다운로드받아 프로젝트에 추가(/WEB-INF/lib)해 놓는다.
 - 다운로드: <https://javaee.github.io/javamail/>

23.2 Naver smtp 설정 하기

- Naver 에 자신의 이메일로 들어가서 [환경설정 > pop3/imap 설정]을 아래 그림과 같이 설정한다.

The screenshot shows the Naver email settings page. The 'POP3/SMTP 설정' tab is selected. Under 'POP3/SMTP 사용', the '사용함' (Use) radio button is selected. The '확인' (Confirm) button is highlighted. At the bottom, the 'SMTP 서버명' is set to 'smtp.naver.com' and the 'SMTP 포트' is set to '465, 보안 연결(SSL) 필요'.

23.3 코드 작성하기

- 아이디와 이메일을 입력받고 아이디와 이메일이 있으면 등록된 이메일로 난수를 보내서 인증하도록 한다. 일단 있는 회원인지 확인부터하고 있다면 서버 정보를 properties 로 넣어주고 인증번호를 만들기 위해 StringBuffer 로 추가한 뒤 toString 해서 데이터 전송시 편하게 바꿨다.
- 끝부분에 인증키를 세션값에 넣는 이유는 request 로는 보관이 용이하지않기 때문에 세션에 넣고 다음 서블릿에서 불러와 사용자가 입력한 인증값과 비교하기위해 사용한다.

23.3.1 searchPwd.do

```
[/WEB-INF/commandHandlerURI.properties]
```

```
01 /changePwd.do=controller.ChangePwdHandler
02 /searchPwd.do=controller.SearchPwdHandler
```

```
[src/controller/SearchPwdHandler.java]
```

```
01 package controller;
02
03 import java.util.Properties;
04 import java.util.Random;
05
06 import javax.mail.Message;
07 import javax.mail.PasswordAuthentication;
08 import javax.mail.Session;
09 import javax.mail.Transport;
10 import javax.mail.internet.InternetAddress;
11 import javax.mail.internet.MimeMessage;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import javax.servlet.http.HttpSession;
15
16 import bean.Member;
17 import mvc.command.CommandHandler;
18 import service.MemberService;
19
20 public class SearchPwdHandler implements CommandHandler {
21
22     private static final String FORM_VIEW = "/WEB-INF/view/login/searchPwd.jsp";
23
24     @Override
25     public String process(HttpServletRequest req, HttpServletResponse res) throws Exception {
26         if (req.getMethod().equalsIgnoreCase("GET")) {
27             return processForm(req, res);
28         } else if (req.getMethod().equalsIgnoreCase("POST")) {
29             return processSubmit(req, res);
30         } else {
31             res.setStatus(HttpServletResponse.SC_METHOD_NOT_ALLOWED);
32             return null;
33         }
34     }
35
36     private String processForm(HttpServletRequest req, HttpServletResponse res) {
37         return FORM_VIEW;
38     }
39
40     private String processSubmit(HttpServletRequest req, HttpServletResponse res) throws Exception
41     {
42         String memberId = req.getParameter("name");
43         String email = req.getParameter("email");
44
45         // 먼저 아이디로 회원정보를 받아오고 가져온 데이터에서 email값을 비교하여 존재하지
46         않으면 인증메일 보내지 못함
47         Member m = new MemberService().memberLogin(memberId);
48         m.setEmail(email);
49
50         if (m == null || !m.getEmail().equals(email)) {
51             req.setAttribute("msg", "아이디나 이메일 정보가 맞지 않습니다");
52             req.setAttribute("loc", "/member/searchPw");
53             req.getRequestDispatcher("/views/common/msg.jsp").forward(req, res);
54             return null;
55         }
56     }
57 }
```

```

55     }
56
57     // mail server 설정
58     String host = "smtp.naver.com";
59     String user = "***@naver.com"; // 자신의 네이버 계정
60     String password = "****"; // 자신의 네이버 비밀번호
61
62     // 메일 받을 주소
63     String to_email = m.getEmail();
64
65     // SMTP 서버 정보를 설정한다.
66     Properties props = new Properties();
67     props.put("mail.smtp.host", host);
68     props.put("mail.smtp.port", 465);
69     props.put("mail.smtp.auth", "true");
70     props.put("mail.smtp.ssl.enable", "true");
71
72     // 인증 번호 생성기
73     StringBuffer temp = new StringBuffer();
74     Random rnd = new Random();
75     for (int i = 0; i < 10; i++) {
76         int rIndex = rnd.nextInt(3);
77         switch (rIndex) {
78             case 0:
79                 // a-z
80                 temp.append((char) ((int) (rnd.nextInt(26)) + 97));
81                 break;
82             case 1:
83                 // A-Z
84                 temp.append((char) ((int) (rnd.nextInt(26)) + 65));
85                 break;
86             case 2:
87                 // 0-9
88                 temp.append((rnd.nextInt(10)));
89                 break;
90         }
91     }
92     String AuthenticationKey = temp.toString();
93     System.out.println(AuthenticationKey);
94
95     Session session = Session.getDefaultInstance(props, new javax.mail.Authenticator() {
96         protected PasswordAuthentication getPasswordAuthentication() {
97             return new PasswordAuthentication(user, password);
98         }
99     });
100
101     // email 전송
102     try {
103         MimeMessage msg = new MimeMessage(session);
104         msg.setFrom(new InternetAddress(user, "bootstrap"));
105         msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to_email));
106
107         // 메일 제목
108         msg.setSubject("안녕하세요? bootstrap 인증메일입니다.");
109         // 메일 내용
110         msg.setText("인증번호:" + temp);
111
112         Transport.send(msg);
113         System.out.println("이메일 전송");
114
115     } catch (Exception e) {
116         e.printStackTrace(); // TODO: handle exception
117     }
118     HttpSession saveKey = req.getSession();
119     saveKey.setAttribute("AuthenticationKey", AuthenticationKey);

```

```

120         // 패스워드 바꿀때 뭘 바꿀지 조건에 들어가는 id
121         req.setAttribute("id", memberId);
122         req.getRequestDispatcher("/WEB-INF/view/login/changePwd.jsp").forward(req, res);
123         return null;
124     }
125
126     private String trim(String str) {
127         return str == null ? null : str.trim();
128     }
129 }

```

[/WEB-INF/view/login/searchPwd.jsp]

```

01 <%@ page language="java" contentType="text/html; charset=UTF-8"
02     pageEncoding="UTF-8"%>
03 <!DOCTYPE html>
04 <html>
05 <head>
06 <meta charset="UTF-8">
07 <title>Insert title here</title>
08 </head>
09 <body>
10
11     <form action="searchPwd.do" method="post">
12         <input type="text" name="name" class="form-control" placeholder="이름">
13         <input type="email" name="email" class="form-control"
14             placeholder="email">
15         <button type="submit" class="btn btn-primary btn-block btn-flat">전송</button>
16     </form>
17
18 </body>
19 </html>

```

23.3.2 changePwd.do

[src/controller/ChangePwdHandler.java]

```

01 package controller;
02
03 import java.util.Properties;
04 import java.util.Random;
05
06 import javax.mail.Message;
07 import javax.mail.PasswordAuthentication;
08 import javax.mail.Session;
09 import javax.mail.Transport;
10 import javax.mail.internet.InternetAddress;
11 import javax.mail.internet.MimeMessage;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import javax.servlet.http.HttpSession;
15
16 import bean.Member;
17 import mvc.command.CommandHandler;
18 import service.MemberService;
19
20 public class ChangePwdHandler implements CommandHandler {
21
22     private static final String FORM_VIEW = "/WEB-INF/view/login/changePwd.jsp";
23
24     @Override

```

```

25     public String process(HttpServletRequest req, HttpServletResponse res) throws Exception {
26         if (req.getMethod().equalsIgnoreCase("GET")) {
27             return processForm(req, res);
28         } else if (req.getMethod().equalsIgnoreCase("POST")) {
29             return processSubmit(req, res);
30         } else {
31             res.setStatus(HttpServletResponse.SC_METHOD_NOT_ALLOWED);
32             return null;
33         }
34     }
35
36     private String processForm(HttpServletRequest req, HttpServletResponse res) {
37         return FORM_VIEW;
38     }
39
40     private String processSubmit(HttpServletRequest req, HttpServletResponse res) throws Exception
41     {
42         String password = req.getParameter("password");
43         String email = req.getParameter("email");
44
45         // 먼저 아이디로 회원정보를 받아오고 가져온 데이터에서 email값을 비교하여 존재하지
46         // 않으면 인증메일 보내지 못함
47         Member m = new MemberService().memberLogin(email);
48
49         String AuthenticationKey = (String)
50 req.getSession().getAttribute("AuthenticationKey");
51         String AuthenticationUser = req.getParameter("AuthenticationUser");
52         if (!AuthenticationKey.equals(AuthenticationUser)) {
53             System.out.println("인증번호 일치하지 않음");
54             req.setAttribute("msg", "인증번호가 일치하지 않습니다");
55             req.setAttribute("loc", "/member/searchPw");
56             req.getRequestDispatcher("/WEB-INF/view/login/changePwdFail.jsp").forward(req, res);
57             return null;
58         }
59
60         // 아래 부분에 새 비밀번호로 변경하는 로직이 필요하다.
61
62         return "/WEB-INF/view/login/login.jsp";
63     }
64
65     private String trim(String str) {
66         return str == null ? null : str.trim();
67     }
68 }
69

```

[/WEB-INF/view/login/changePwd.jsp]

```

01 <%@ page language="java" contentType="text/html; charset=UTF-8"
02     pageEncoding="UTF-8"%>
03 <!DOCTYPE html>
04 <html>
05 <head>
06 <meta charset="UTF-8">
07 <title>Insert title here</title>
08 </head>
09 <body>
10
11     <h1>비밀번호 변경</h1>
12
13     <form action="changePwd.do" method="post">
14         <input type="hidden" name="email" value="kshelp@naver.com">
15         <label>인증번호: <input type="text" name="AuthenticationUser" placeholder="이메일 확
16 인"></label><br>

```

```
17         <label>새 비밀번호: <input type="password" name="pwd1" placeholder="6글자 이상
18     "></label><br>
19         <label>새 비밀번호 확인: <input type="password" name="pwd2" placeholder="영문, 숫자,
20     특수문자 조합"></label><br>
21         <button type="submit" class="btn btn-primary btn-block btn-flat">변경</button>
22     </form>
23
24 </body>
25 </html>
```