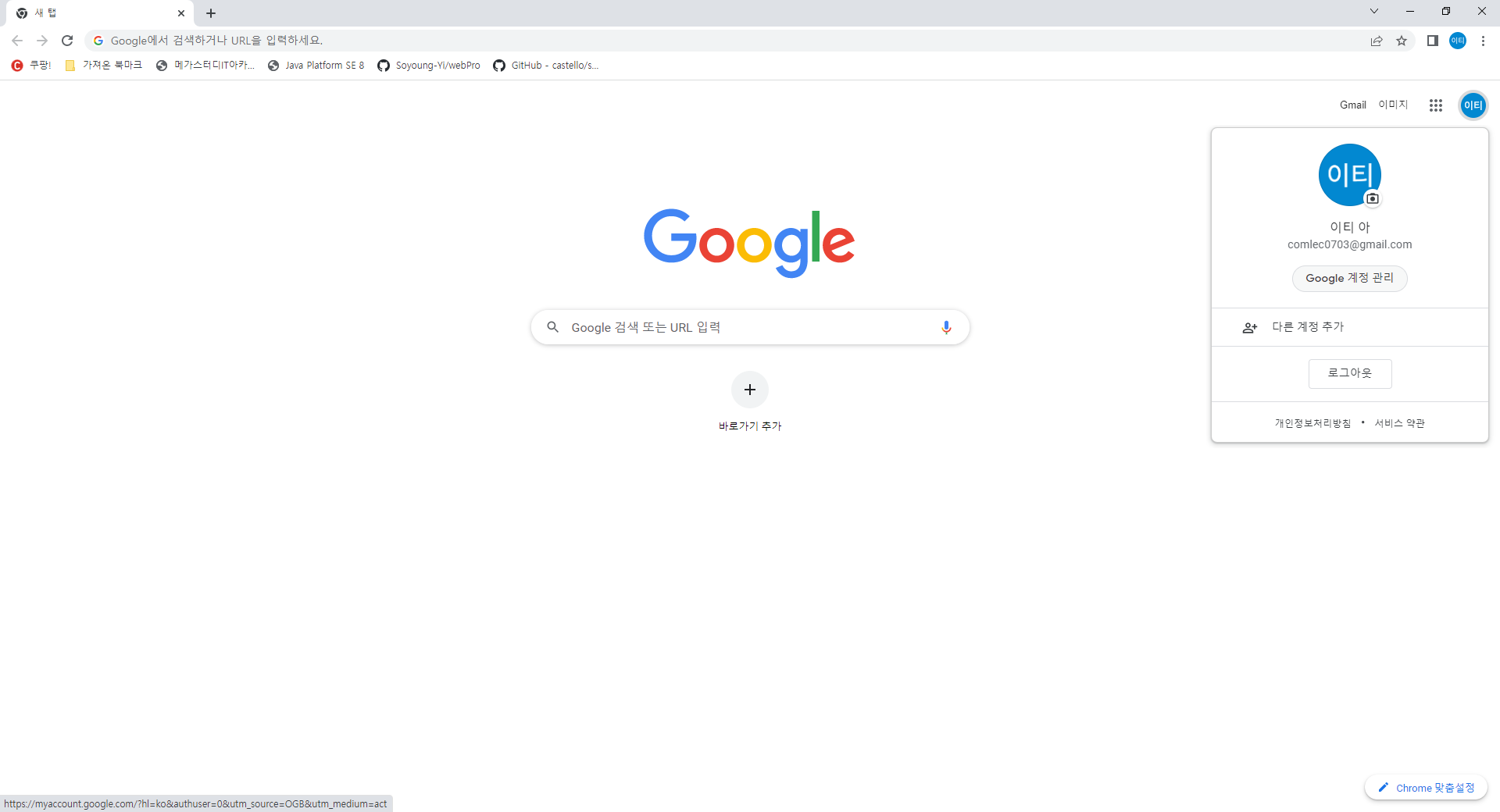
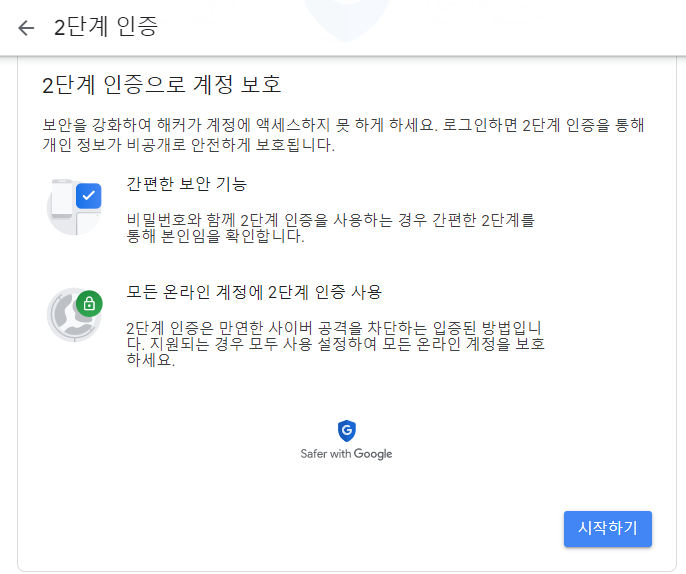
[ 18 ] 메일발송

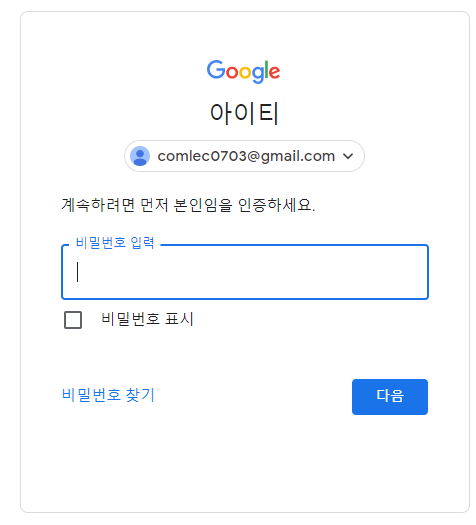
1. 우선 gmail 계정을 이용한다. 보안수준을 2단계로 설정하고 앱 비밀번호를 생성한다



계정관리 -> 보안 -> 2단계 인증 -> 앱 비밀번호 생성



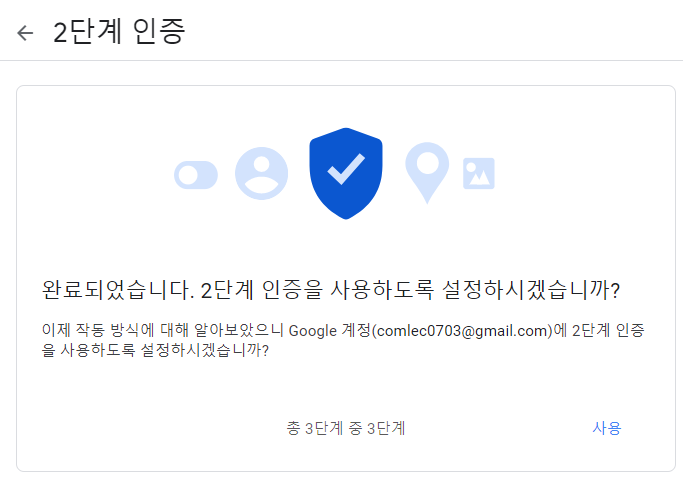


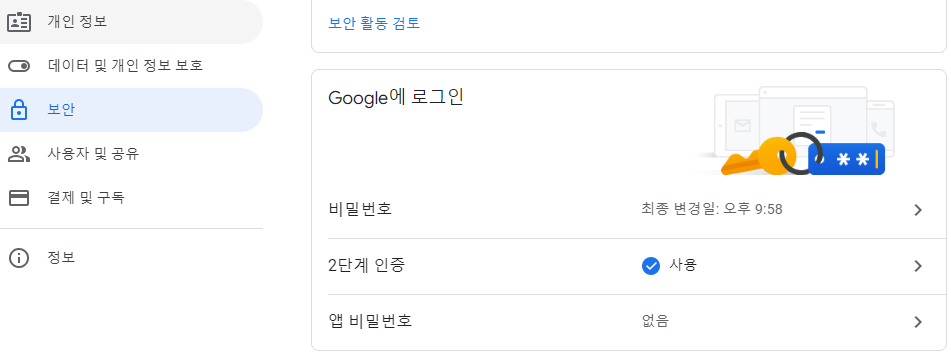


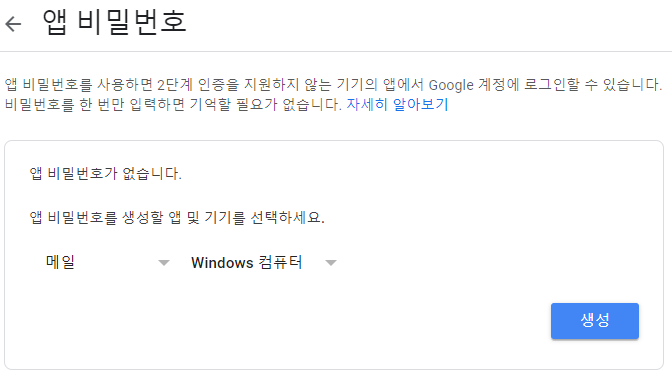
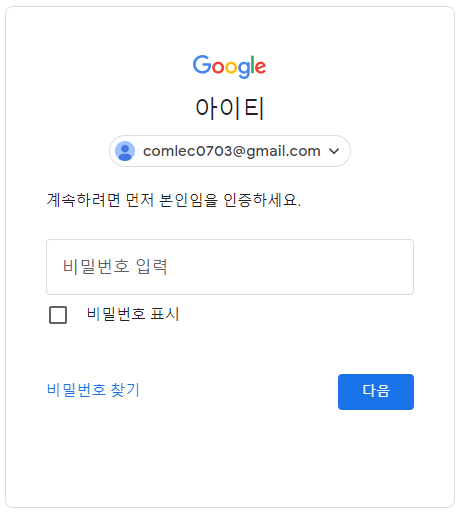
본인의 핸드폰으로 코드를 받는다

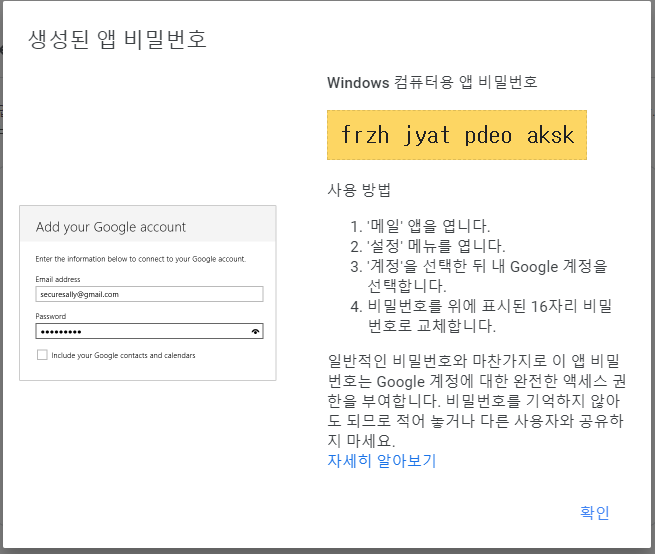


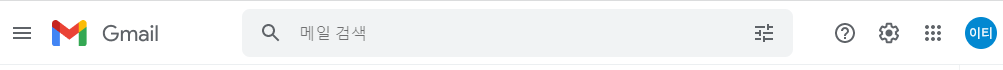
본인의 핸드폰으로 받은 코드를 입력하여 2다계 인증을 완료하도록 설정한다

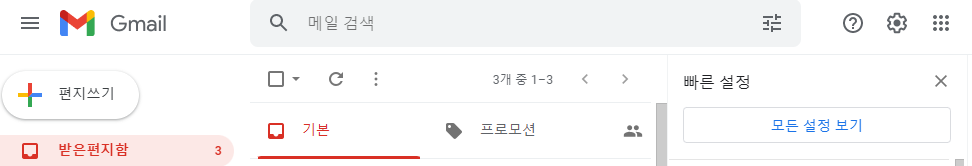


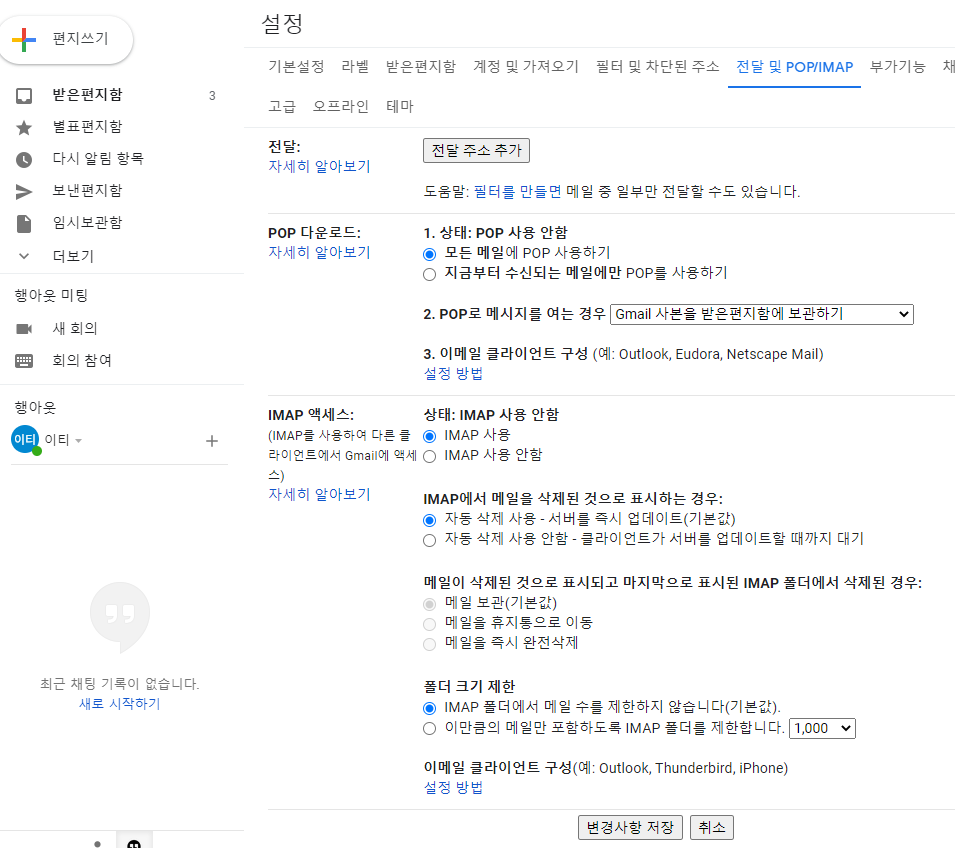












출처 : https://kitty-geno.tistory.com/43

1. 의존추가(pom.xml)

<!-- https://mvnrepository.com/artifact/javax.mail/mail -->

<dependency>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

<version>1.4.7</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>3.0.5.RELEASE</version>

</dependency>

1. web.xml 한글 filter 추가

<!-- 한글처리 -->

<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>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

1. 메일 정보(mail.properties)

mail.host=smtp.gmail.com

mail.port=587

mail.username=comlec0703

mail.passwoard= wuwwnsdxceqltqah

1. 빈생성 (servlet-context.xml)

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans:beans xmlns=*"http://www.springframework.org/schema/mvc"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:beans=*"http://www.springframework.org/schema/beans"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd"*>

<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

<!-- Enables the Spring MVC @Controller programming model -->

<annotation-driven />

<!-- Handles HTTP GET requests for /resources/\*\* by efficiently serving up static resources in the ${webappRoot}/resources directory -->

<resources mapping=*"/resources/\*\*"* location=*"/resources/"* />

<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->

<beans:bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<beans:property name=*"prefix"* value=*"/WEB-INF/views/"* />

<beans:property name=*"suffix"* value=*".jsp"* />

</beans:bean>

<context:component-scan base-package=*"com.lec.ch18"* />

<context:property-placeholder location=*"classpath:META-INF/property/mail.properties"*/>

<beans:bean id=*"mailSender"*

class=*"org.springframework.mail.javamail.JavaMailSenderImpl"*>

<beans:property name=*"host"* value=*"${mail.host}"*/>

<beans:property name=*"port"* value=*"${mail.port}"*/>

<beans:property name=*"username"* value=*"${mail.username}"*/>

<beans:property name=*"password"* value=*"${mail.passwoard}"*/>

<beans:property name=*"javaMailProperties"*>

<beans:props>

<beans:prop key=*"mail.smtp.auth"*>true</beans:prop>

<beans:prop key=*"mail.smtp.starttls.enable"*>true</beans:prop>

<beans:prop key=*"mail.debug"*>true</beans:prop>

<beans:prop key=*"mail.transport.protocol"*>smtp</beans:prop>

<beans:prop key=*"mail.smtp.ssl.trust"*>smtp.gmail.com</beans:prop>

<beans:prop key=*"mail.smtp.ssl.protocols"*>TLSv1.2</beans:prop>

</beans:props>

</beans:property>

</beans:bean>

1. 소스 추가(컨트롤러)

**import** javax.mail.Message;

**import** javax.mail.internet.InternetAddress;

**import** javax.mail.internet.MimeMessage;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.mail.SimpleMailMessage;

**import** org.springframework.mail.javamail.JavaMailSender;

**import** org.springframework.mail.javamail.MimeMessagePreparator;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

@Controller

**public** **class** HomeController {

@Autowired

**private** JavaMailSender mailSender;

**private** **static** **final** Logger ***logger*** = LoggerFactory.*getLogger*(HomeController.**class**);

@RequestMapping(value = "/", method = RequestMethod.***GET***)

**public** String home() {

***logger***.info("회원가입 페이지");

**return** "join";

}

@RequestMapping(value = "textMail", method = RequestMethod.***GET***)

**public** String textMail(String name, String email, Model model) {

SimpleMailMessage message = **new** SimpleMailMessage();

//message.setFrom("comlec0703@gmail.com");// 관리자 메일 주소(보내는 사람)

message.setTo(email); // 받을 메일 주소

message.setSubject("[TEXT 가입인사]"+name + "님 회원가입에 감사드립니다"); // 받을 메일 제목

String content = name+"님 회원가입 감사합니다.\n <h1>태그 안 먹음</h1>";

message.setText(content);

mailSender.send(message); // 메일 보냄

model.addAttribute("mailSendResult", "TEXT 메일이 발송되었습니다");

**return** "sendResult";

}

@RequestMapping(value = "htmlMail", method = RequestMethod.***GET***)

**public** String htmlMail(**final** String name, **final** String email, Model model) {

MimeMessagePreparator message = **new** MimeMessagePreparator() {

String content =

"<div style=\"width:500px; margin: 0 auto;\">\n" +

" <h1>홍길동님의 회원가입 감사합니다.</h1>\n" +

" <div>\n" +

" <p>아무개 사이트에서만 쓰실 수 있는 감사쿠폰을 드립니다</p>\n" +

" <img src=\"https://t1.daumcdn.net/daumtop\_chanel/op/20170315064553027.png\" />\n" +

" <hr color=\"red\">\n" +

" </div>\n" +

" <div>\n" +

" <p style=\"color:red;\">빨간 글씨 부분</p>\n" +

" <p style=\"color:blue\">파란 글씨 부분</p>\n" +

" <img src=\"http://localhost:8090/ch18/resources/100.jpg\">\n" +

" </div>\n" +

" <p>서울시 어떤구 몰라17길 51 어떤빌딩 4층</p>\n" +

"</div>";

@Override

**public** **void** prepare(MimeMessage mimeMessage) **throws** Exception {

mimeMessage.setRecipient(Message.RecipientType.***TO***, **new** InternetAddress(email));

mimeMessage.setFrom(**new** InternetAddress("comlec0703@gmail.com"));

mimeMessage.setSubject("[HTML 가입인사]"+name + "님 회원가입에 감사드립니다");

mimeMessage.setText(content, "utf-8", "html");

}

};

mailSender.send(message); // 메일 보냄

model.addAttribute("mailSendResult", "HTML 메일이 발송되었습니다");

**return** "sendResult";

}

}

1. 뷰단(join.jsp)

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<body>

<form action=*"textMail"*>

<p>이름 <input type=*"text"* name=*"name"* required=*"required"*></p>

<p>메일 <input type=*"text"* name=*"email"* required=*"required"*></p>

<input type=*"submit"* value=*"회원가입(TEXT단순 메일감)"*>

</form>

<hr color=*"green"*>

<form action=*"htmlMail"*>

<p>이름 <input type=*"text"* name=*"name"* required=*"required"*></p>

<p>메일 <input type=*"text"* name=*"email"* required=*"required"*></p>

<input type=*"submit"* value=*"회원가입(HTML메일감)"*>

</form>

</body>

</html>

1. sendResult.jsp

<body>

<h2>${mailSendResult }</h2>

</body>