

스프링프레임워크 이메일 보내기

스프링프레임워크에서 메일서버(SMTP Server)를 이용하여 메일을 전송하는 방법에 대해서 알아보하고자 합니다. SMTP는 Simple Mail Transfer Protocol로서 전자 메일 전송을 위한 표준 프로토콜입니다. 따라서 SMTP Server란 이메일을 송신하는 서버입니다. (메일서버는 SMTP Server, POP3 Server가 있고 POP3 Server는 이메일을 수신하는 서버입니다.)

메일 전송 기능을 구현하기 위해선 아래의 4가지가 필요합니다.

1. pom.xml 설정
2. mailSender Bean 등록(root-context.xml 또는 servlet-context.xml)
3. Controller 메서드 추가
4. 메일 계정 보안 설정

메일서버(SMTP Server)를 통해 메일을 전송할 때 다음과 같은 두 가지 방법이 있습니다.

첫 번째, 메일 서버가 릴레이(relay)를 허용하는 경우

※릴레이(Relay) : 외부(외부 네트워크)에서 해당 메일서버(smtp 서버)를 경유해서 외부로 메일을 보내는 것을 의미함.

- 이 방식은 계정이 존재하지 않더라도 메일을 발송할 수 있습니다.
- 릴레이(relay)를 허용하는 경우 잘못된 방식(타인에 의한 스팸메일)으로 악용 될 수 있습니다.
- 따라서 특정 IP에서만 릴레이가 되도록 하는게 일반적입니다.
- 해당 방식을 경험해보고 싶으시다면 아래의 포스팅을 참고해주세요.

[서버]Telnet을 이용한 SMTP 메일 발송

두 번째, 메일 서버에 계정 인증 후 메일을 전송하는 방법

위 두 가지 방법 선택은 "2.mailSender Bean 등록"에서 이루어집니다. 아래의 자세한 설명에서 어떻게 설정해야 하는지 설명은 하지만 주 방식은 **계정 인증 후 메일을 전송하는 방법**으로 설명을 진행하고자 합니다.

1. 메일 계정 보안 설정

SMTP Server를 통해서 메일 전송 기능을 구현하기 위해선 사용하고자 하는 메일의 SMTP 보안 단계를 낮추어야 합니다. 네이버의 경우 로그인 시 휴대폰승인을 통한 보안 단계까지 설정이 되어 있을 경우 아래의 설정만으로는 부족합니다.

1.1 구글

1) 구글 메인페이지 오른쪽 상단에 자신의 계정을 클릭하고 아래의 그림과 같이 "Google 계정 관리"를 클릭합니다.



그림 1-1

2) 왼쪽 목록중 [보안]을 클릭합니다.

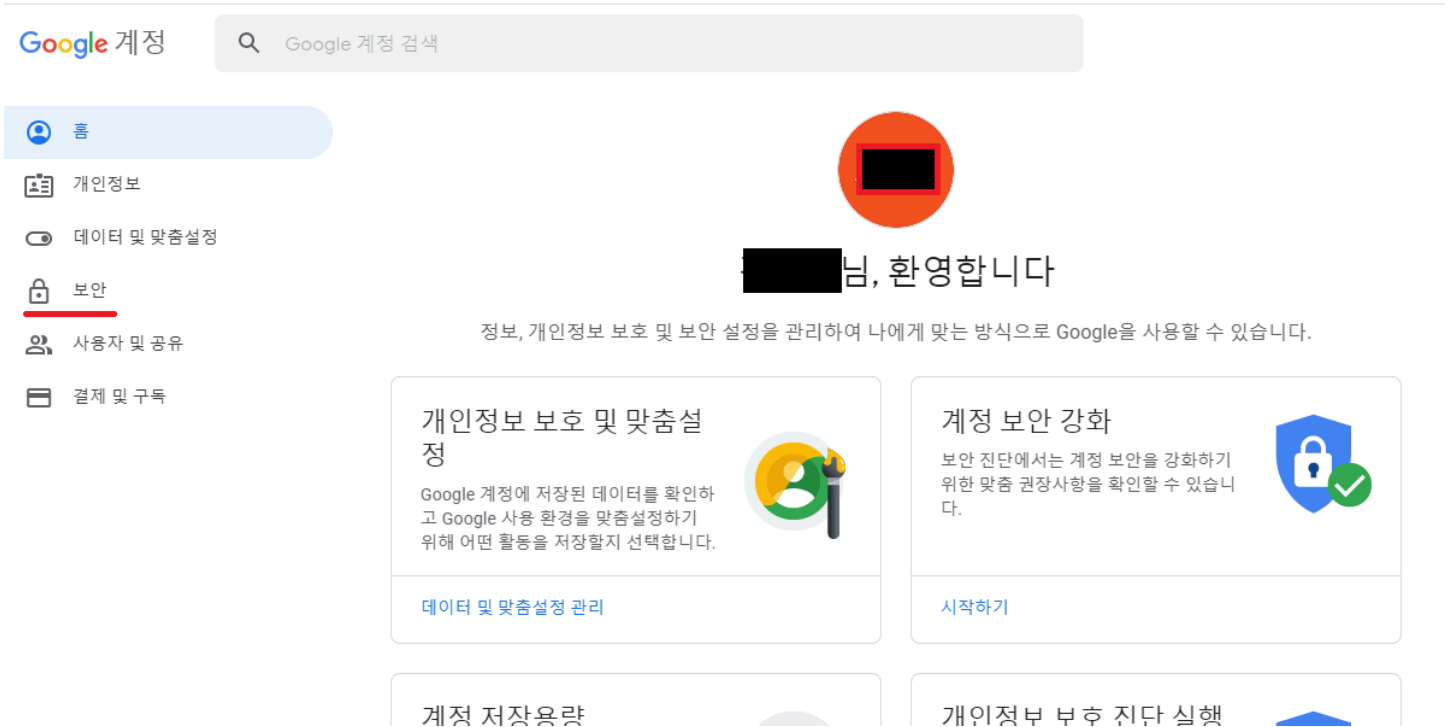


그림 1-3

3) 보안 항목에서 "보안 수준이 낮은 앱의 액세스"를 찾은 후 "액세스 사용 설정(권장하지 않음)"을 클릭합니다.

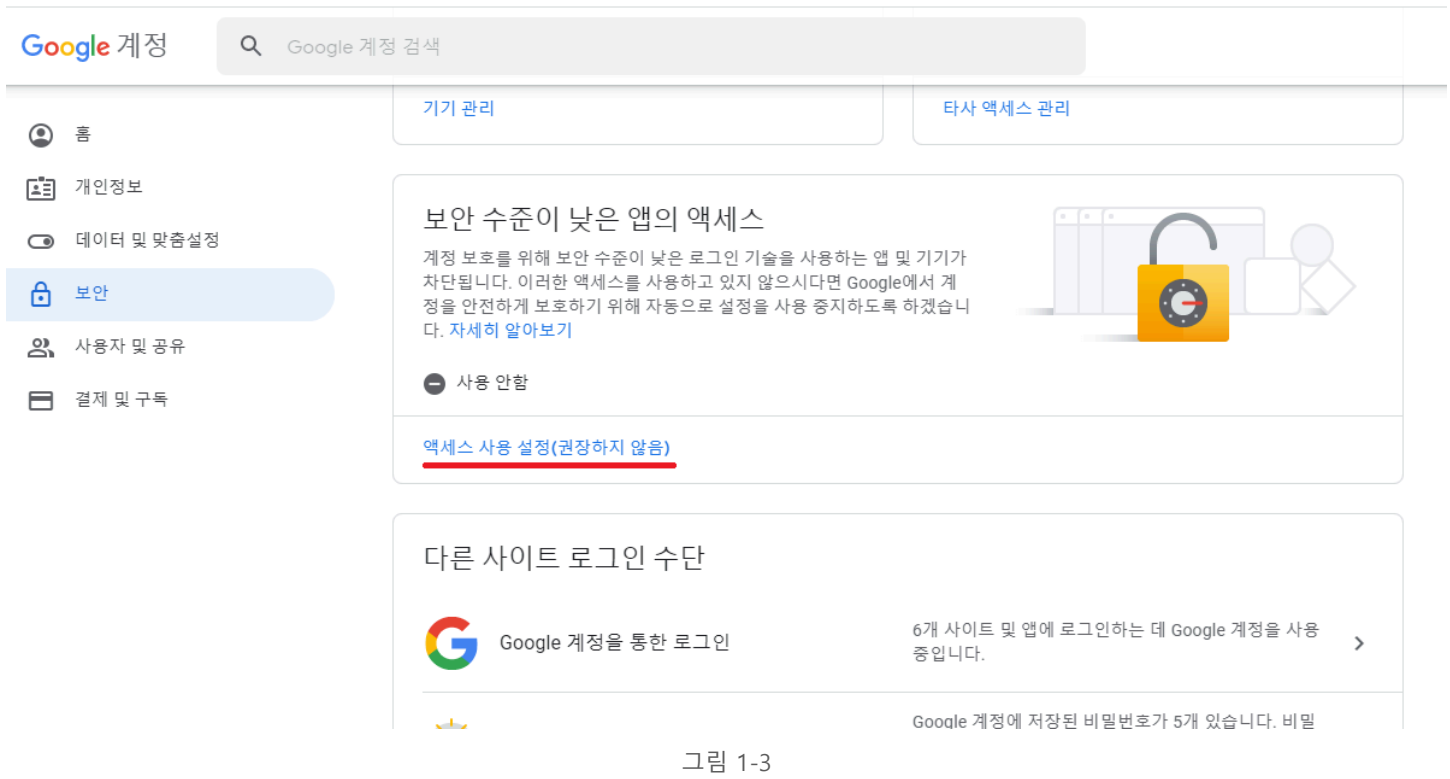


그림 1-3

4) 보안 수준이 낮은 앱 허용을 사용함으로 변경합니다.

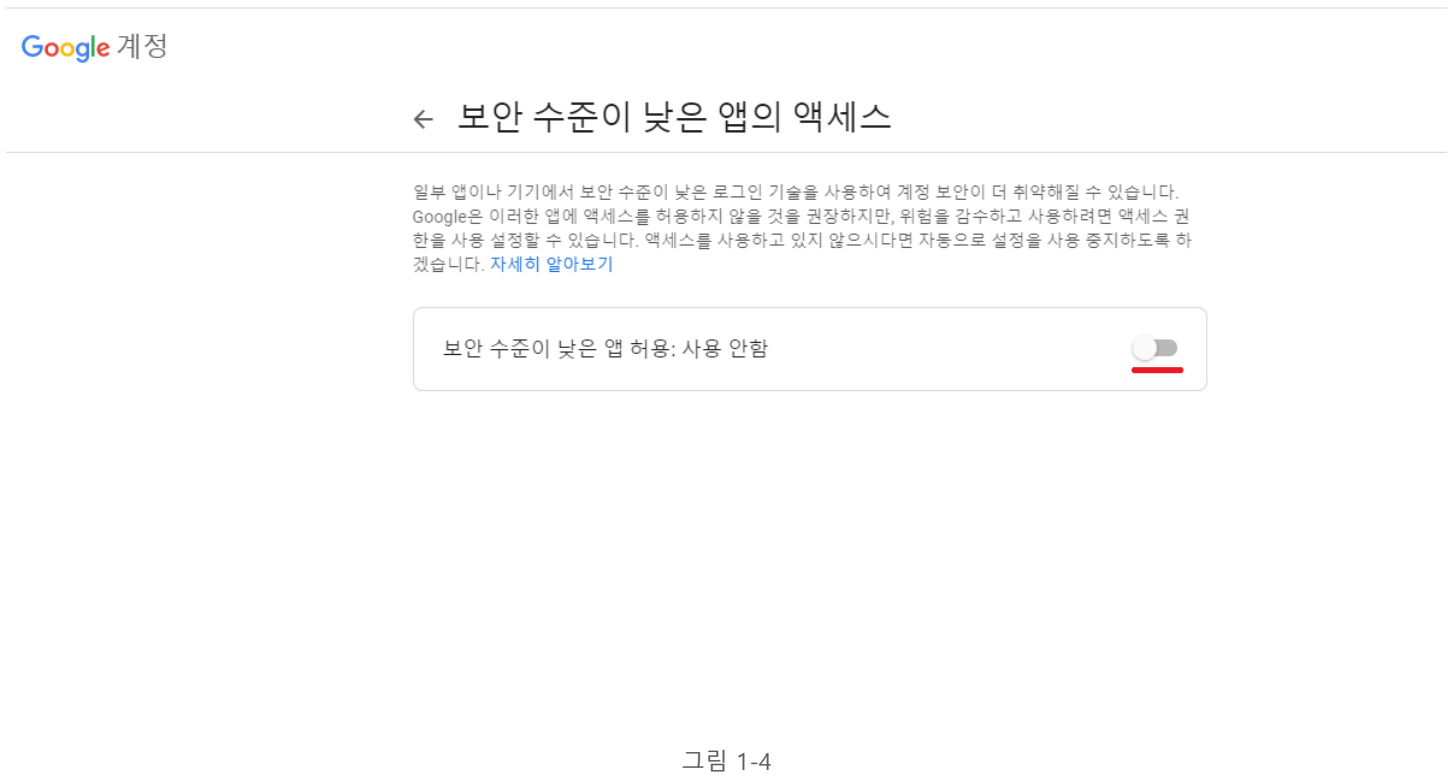


그림 1-4

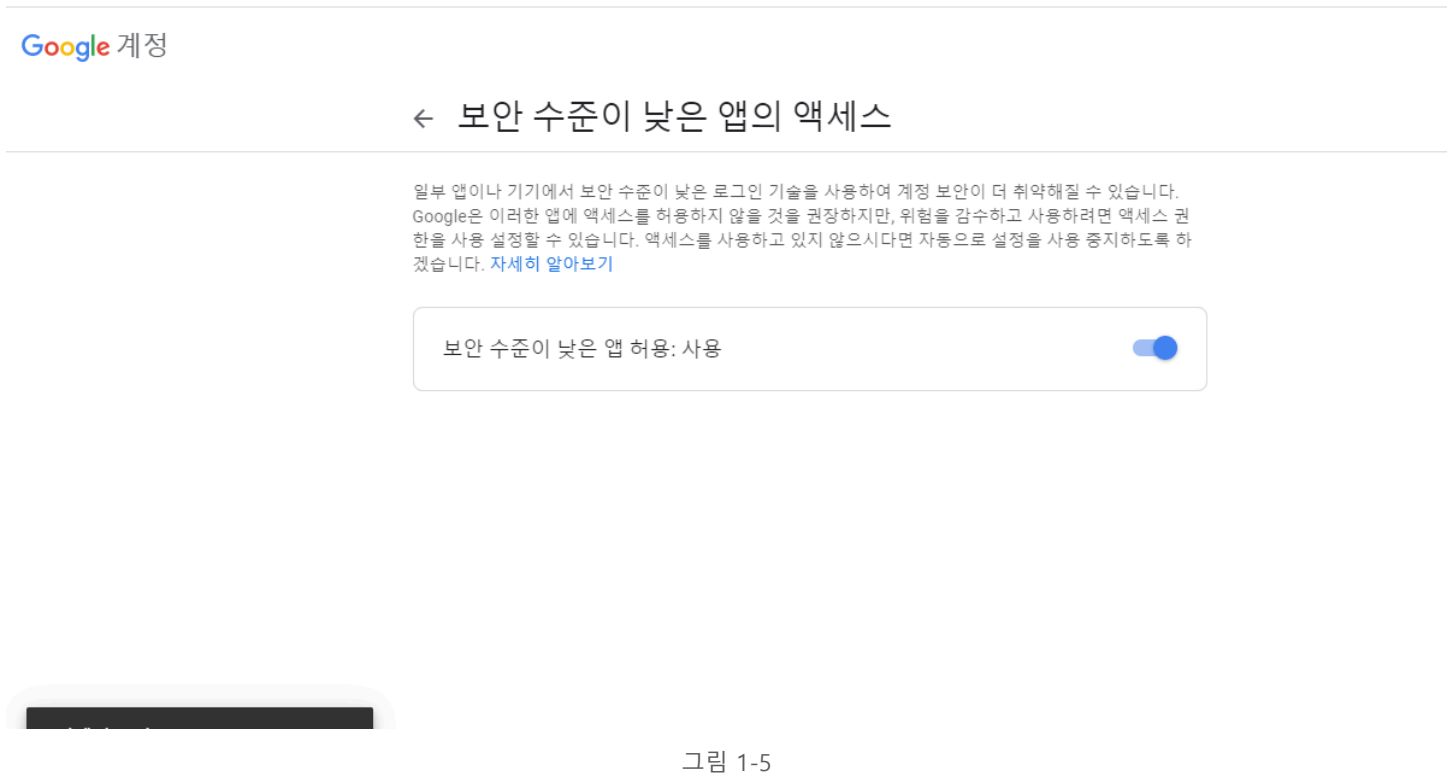


그림 1-5

1.2 네이버

1) 네이버 메일 페이지 들어간 후 왼쪽 최하단에 "환경 설정"을 클릭합니다.



그림 1-6

2) 상단 목록 중 "POP3/IMAP 설정"을 클릭합니다.



그림 1-7

3) "POP3/SMTP 사용"을 '사용함'으로 체크 후 확인버튼을 클릭합니다.

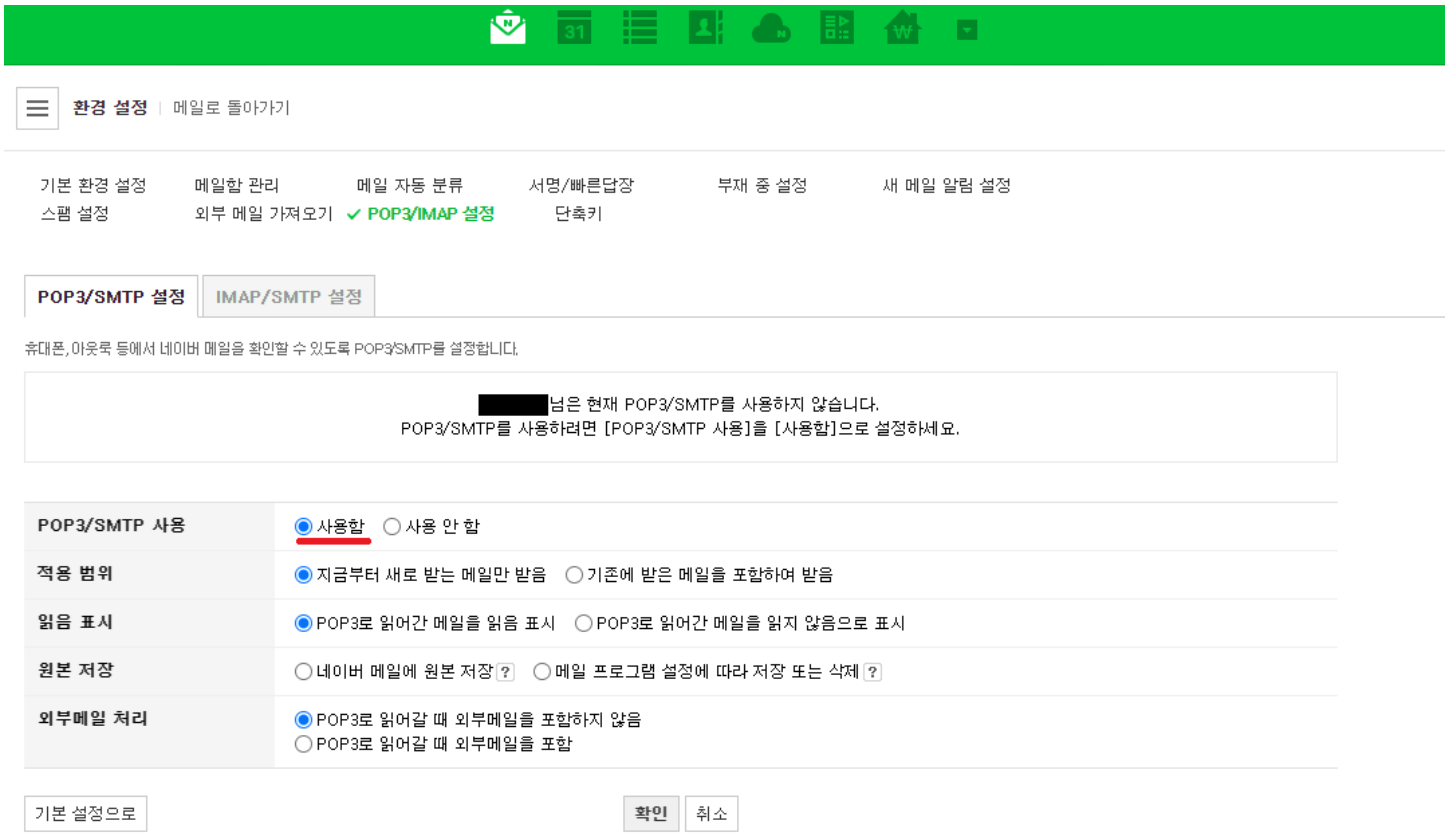


그림 1-8

1.3 다음

1) 다음 메일페이지 들어간 후 왼쪽 목록 중 제일 아래에 "환경 설정"을 클릭합니다.



그림 1-9

2) 상단 목록 중 "IMAP/POP3"을 클릭합니다.

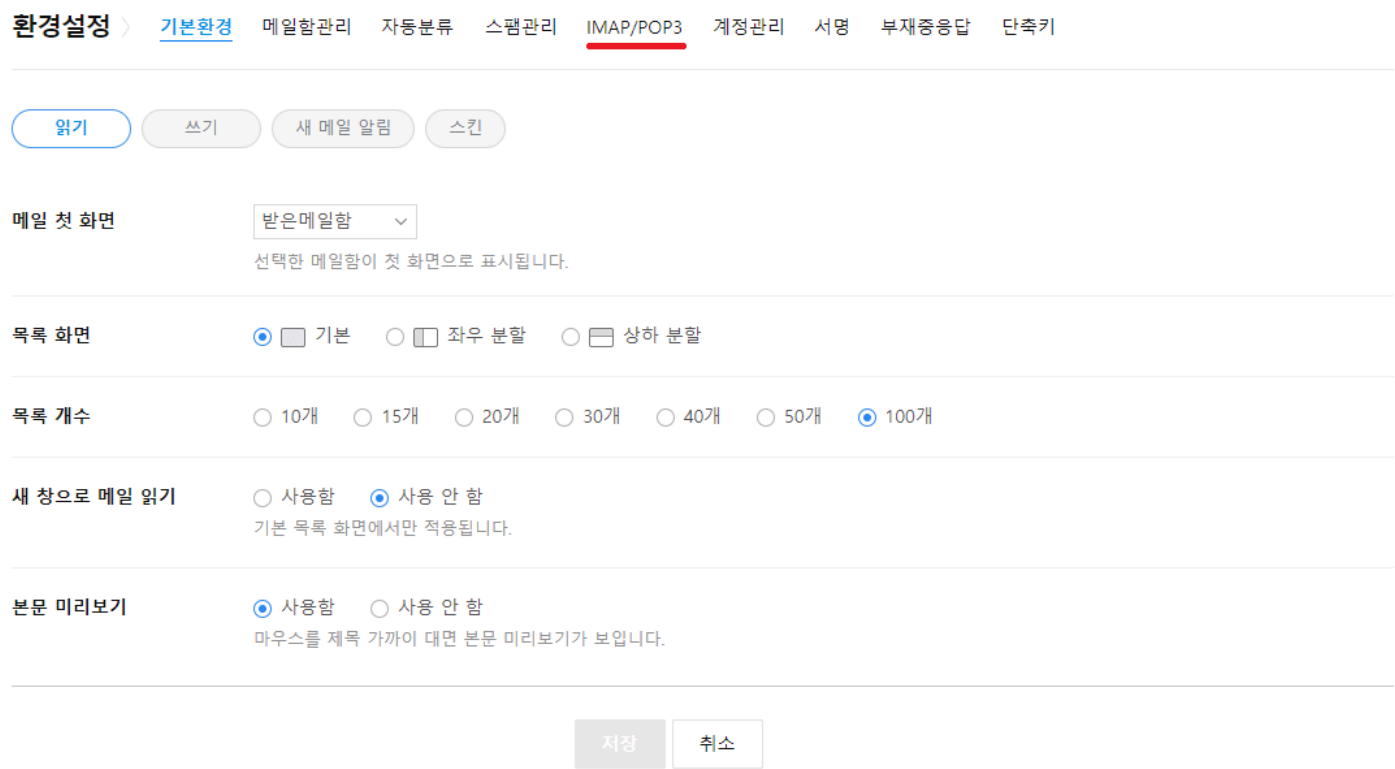


그림 1-10

3) "IMAP / SMTP 사용"을 '사용함'으로 체크 후 저장을 클릭합니다.

IMAP POP3 SKT 휴대폰 메일

IMAP / SMTP 사용 ☒ 사용함 ☐ 사용 안 함

메일 프로그램에서 메일을 읽거나 작성하는 즉시 웹 Daum 메일에도 반영돼서 스마트폰에서 메일을 더욱 잘 관리할 수 있습니다.

저장 취소

IMAP / SMTP 설정값 안내

스마트폰의 메일 계정 설정에 아래와 같이 등록해주세요.

IMAP 서버명	imap.daum.net	SMTP 서버명	smtp.daum.net
IMAP 포트번호	993, 보안연결(SSL) 필요	SMTP 포트번호	465, 보안연결(SSL) 필요
아이디		비밀번호	Daum 메일 접속 시 비밀번호
설정 방법	iOS 안드로이드 윈도우폰		

그림 1-11

2. pom.xml 설정

2.1 mail 라이브러리를 추가해줍니다.

- 버전은 자신에게 맞는 버전을 선택하시면 됩니다.(글쓰는 현재 가장 최신 버전은 1.5.0-b01이지만 아직 안정적이지 않은 거 같아서 가장 많이 사용한 1.4.7을 선택하였습니다.)

```

1 <!-- https://mvnrepository.com/artifact/javax.mail/mail -->
2 <dependency>
3   <groupId>javax.mail</groupId>
4   <artifactId>mail</artifactId>
5   <version>1.4.7</version>
6 </dependency>
7

```

Colored by Color Scripter

```

VamPa/pom.xml MemberController.java
148 <dependency>
149   <groupId>org.springframework</groupId>
150   <artifactId>spring-jdbc</artifactId>
151   <version>${org.springframework-version}</version>
152 </dependency>
153 <!-- log4j2 -->
154 <!-- https://mvnrepository.com/artifact/org.bgee.log4jdbc-log4j2/log4jdbc-log4j2-jdbc4 -->
155 <dependency>
156   <groupId>org.bgee.log4jdbc-log4j2</groupId>
157   <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
158   <version>1.16</version>
159 </dependency>
160 <!-- smtp 메일 전송 관련 라이브러리 -->
161 <!-- https://mvnrepository.com/artifact/javax.mail/mail -->
162 <dependency>
163   <groupId>javax.mail</groupId>
164   <artifactId>mail</artifactId>
165   <version>1.4.7</version>
166 </dependency>
167
168
169 <!-- @Inject -->
170 <dependency>
171   <groupId>javax.inject</groupId>
172   <artifactId>javax.inject</artifactId>
173   <version>1</version>
174 </dependency>
175
176 <!-- Servlet -->
177 <!-- <dependency>

```

그림 2-1

- java에서 메일을 보내기 위해 필요한 라이브러리입니다.

- 실제 전송되는 메일 객체를 가집니다.

- 주요 클래스로 Session, Message, Address, Authenticator, Transport 등이 있습니다.

- 해당 라이브러리만으로도 메일 전송 기능이 구현 가능합니다. 하지만 개발자가 모든 세팅을 직접 해주어야 하기 때문에 번거롭습니다.

- 이번 포스팅에선 spring-context-support 라이브러리를 사용할 것이기 때문에, mail 라이브러리에선 javax.mail.internet.MimeMessage만 사용됩니다.

2.2 spring-context-support 라이브러리를 추가합니다.

```
1 <!-- https://mvnrepository.com/artifact/org.springframework/spring-context-support -->
2 <dependency>
3   <groupId>org.springframework</groupId>
4   <artifactId>spring-context-support</artifactId>
5   <version>${org.springframework-version}</version>
6 </dependency>
```

Colored by Color Scripter

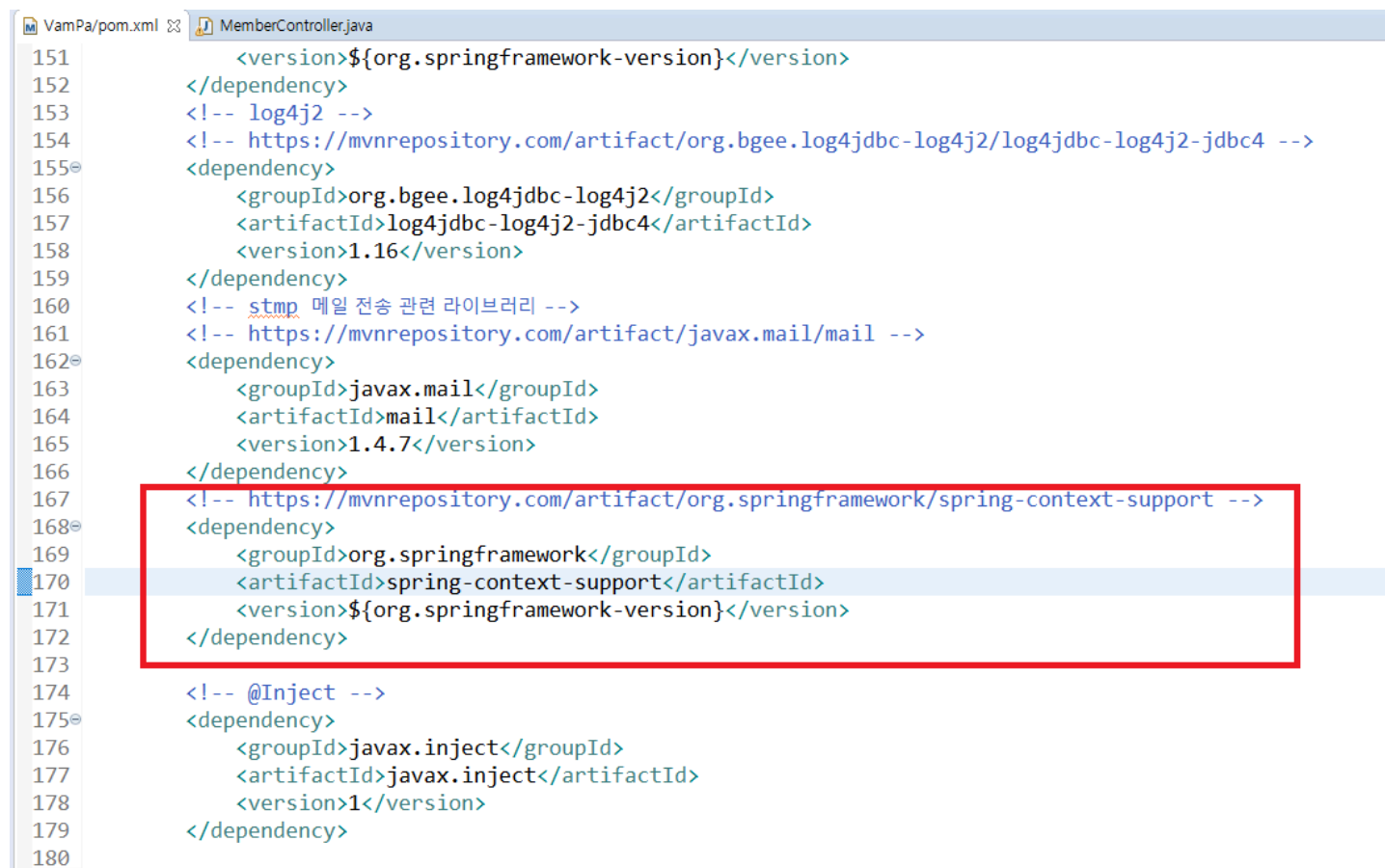


그림 2-2

- JavaMailSender 인터페이스(JavaMailSenderImpl)를 사용하기 위해 필요로 합니다. mail 라이브러리를 일일이 세팅을 할 필요 없이 몇 가지의 설정(MailSender Bean설정)만으로도 사용될 수 있도록 도와줍니다.

- JavaMailSender 인터페이스는 MIME 형식의 메일을 보낼 수 있도록 해줍니다.(javax.mail.MimeMessage 사용 가능)

- 스프링 프레임웍에서는 기본적으로 JavaMailSender 인터페이스와 비슷한 역할을 하는 메인 인터페이스 MainSender가 있습니다. 하지만 MainSender는 SimpleMailMessage만 사용 가능합니다. (javax.mail.MimeMessage 사용 불가능)

* SimpleMailMessage는 단순한 텍스트 메시지만 사용할 수 있고, MimeMessage는 텍스트와 더불어 이미지와 같은 첨부파일을 같이 메시지에 포함시킬 수 있습니다.

3. mailSender Bean 등록

Bean 등록은 root-context.xml에 설정해주시면 됩니다. '0. 개요'에서 말했듯이 릴레이를 허용하는 서버를 사용한 경우의 빈 설정과 아이디/비밀번호로 인증 후 발송하는 빈 설정은 구글 / 네이버 / 다음 순으로 알아보겠습니다.

3.1 메일 서버가 릴레이(relay)를 허용하는 경우

- root-context.xml에 아래의 코드를 추가해줍니다.

```
1
2 <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
3   <property name="host" value="smtp 도메인 주소"/>
4 </bean>
```

Colored by Color Scriptor CS

- host의 value값은 아래의 링크에서 '2. 사용 가능한 메일서버 찾기(nslookup)'을 참고하셔서 해당 주소를 입력하시면 됩니다.

[서버]Telnet을 이용한 SMTP 메일 발송

3.2 메일 서버에 계정 인증 후 메일을 전송하는 방법

host, port, username, password <property> 태그는 각 메일서버에 맞게 설정해주시면 됩니다. 각 메일서버에서 가장 크게 다른 점은 name이 javaMailProperties인 <property> 태그입니다. 해당 태그는 각 메일서버에서 사용 중인 보안 프로토콜을 인증하기 위해 사용되는 코드가 삽입됩니다.

1) gmail

- root-context.xml에 아래의 코드를 추가해줍니다.

```
1 <!-- gmail설정 -->
```

CS


```
2 <bean id="mailSender2" class="org.springframework.mail.javamail.JavaMailSenderImpl">
3   <property name="host" value="smtp.gmail.com" />
4   <property name="port" value="587" />
5   <property name="username" value="your_email"/> <!-- 자신의 이메일 아이디 -->
6   <property name="password" value="your_password!"/> <!-- 자신의 비밀번호 -->
7   <!-- 보안연결 TLS과 관련된 설정 -->
8   <property name="javaMailProperties">
9     <props>
10      <prop key="mail.smtp.auth">true</prop>
11      <prop key="mail.smtp.starttls.enable">true</prop>
12    </props>
13  </property>
14 </bean>
```

Colored by Color Scripter

- host : "smtp.gmail.com" 입력

- port : "587" 입력

- username : 자신의 이메일 아이디 입력

- password : 자신의 이메일 비밀번호 입력

- javaMailProperties인 <property> 태그에 삽입된 코드는 TLS 인증 코드입니다.

* SSL이란 웹사이트와 브라우저 사이에서 전송되는 데이터를 암호화하여 인터넷 연결을 보안을 유지하는 표준 기술입니다. TLS는 더 강력한 버전의 SSL입니다. TLS 또한 SSL이라 불리기도 합니다.

2) naver

- root-context.xml에 아래의 코드를 추가해줍니다.

```
1 <!-- navermail설정 -->
2 <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
3   <property name="host" value="smtp.naver.com"/> <!-- 메이서버 호스트 -->
4   <property name="port" value="465"/> <!-- 메이서버 포트번호 -->
5   <property name="username" value="your_email"/> <!-- 자신의 이메일 아이디 -->
6   <property name="password" value="your_password"/> <!-- 자신의 비밀번호 -->
7   <!-- 보안연결 SSL과 관련된 설정 -->
8   <property name="javaMailProperties">
9     <props>
10      <prop key="mail.smtp.auth">true</prop>
11      <prop key="mail.smtp.starttls.enable">true</prop>
12      <prop key="mail.smtps.checkserveridentity">true</prop>
13      <prop key="mail.smtps.ssl.trust">*</prop>
14      <prop key="mail.debug">true</prop>
15      <prop key="mail.smtp.socketFactory.class">javax.net.ssl.SSLSocketFactory</prop>
16    </props>
17  </property>
18 </bean>
```

Colored by Color Scripter

- host : "smtp.naver.com" 입력

- port : "465" 입력

- username : 자신의 이메일 아이디 입력

- password : 자신의 이메일 비밀번호 입력

- javaMailProperties인 <property> 태그에 삽입된 코드는 SSL 인증 코드입니다.

3) daum

- root-context.xml에 아래의 코드를 추가해줍니다.

```
1 <!-- navermail설정 -->
2 <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
3   <property name="host" value="smtp.naver.com"/> <!-- 메이서버 호스트 -->
4   <property name="port" value="465"/> <!-- 메이서버 포트번호 -->
5   <property name="username" value="your_email"/> <!-- 자신의 이메일 아이디 -->
6   <property name="password" value="your_password"/> <!-- 자신의 비밀번호 -->
7   <!-- 보안연결 SSL과 관련된 설정 -->
8   <property name="javaMailProperties">
9     <props>
10    <prop key="mail.smtp.auth">true</prop>
11    <prop key="mail.smtp.starttls.enable">true</prop>
12    <prop key="mail.smtps.checkserveridentity">true</prop>
13    <prop key="mail.smtps.ssl.trust">*</prop>
14    <prop key="mail.debug">true</prop>
15    <prop key="mail.smtp.socketFactory.class">javax.net.ssl.SSLSocketFactory</prop>
16    </props>
17  </property>
18 </bean>
```

Colored by Color Scripter

- host : "smtp.daum.net" 입력

- port : "465" 입력

- username : 자신의 이메일 아이디 입력

- password : 자신의 이메일 비밀번호 입력

- javaMailProperties인 <property>태그에 삽입된 코드는 네이버와 동일하게 SSL 인증 코드입니다.