

구글 SMTP 사용(with yml)

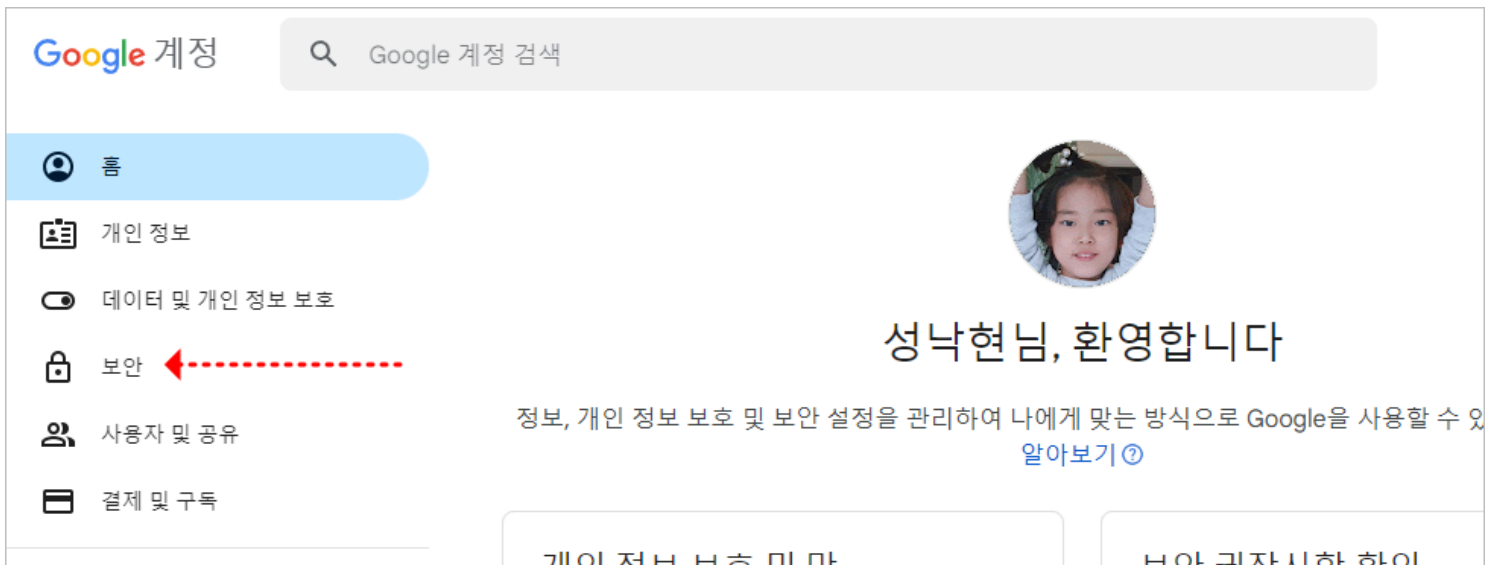
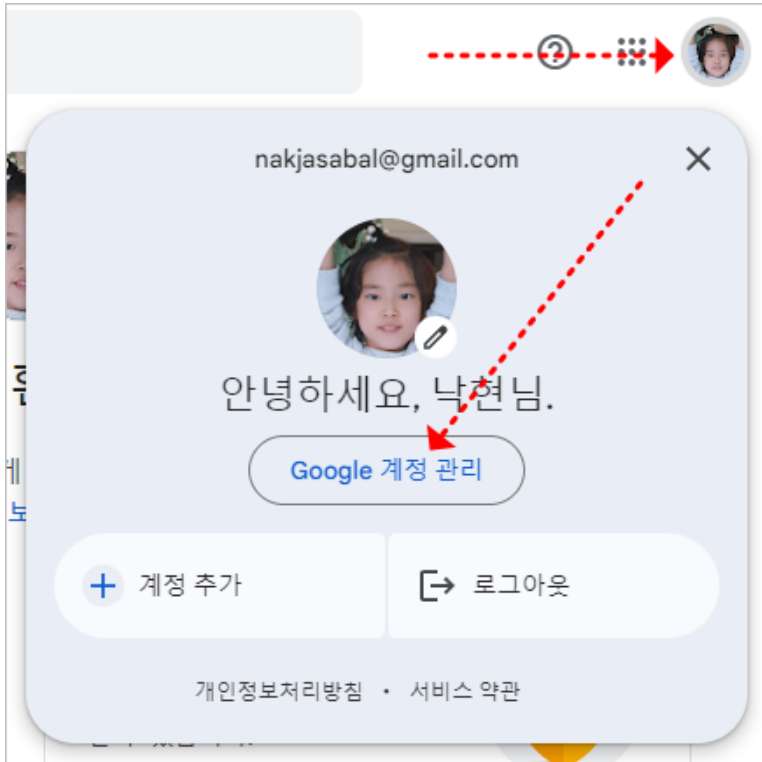
프로젝트명 : B15bSMTP_Google

준비사항

- B15aSMTP_Naver 프로젝트를 복사한다.
- 컨텍스트 루트 와 settings.gradle 을 수정한다.
- Refresh Gradle Project 를 눌러 적용한다.

구글 SMTP 설정

23년 11월 현재 구글 SMTP를 사용하려면 “앱 비밀번호”를 생성해야 한다.



2단계 인증을 클릭

Google에 로그인하는 방법

항상 Google 계정에 액세스할 수 있도록 최신 정보가 반영되었는지 확인하세요.

🔒 2단계 인증	✅ 사용 설정일: 2022. 6. 7.	>
👤 패스키	패스키 사용 시작	>
*** 비밀번호	최종 변경일: 2022. 5. 8.	>

아래로 스크롤하여 앱 비밀번호가 나오면 클릭

앱 비밀번호

앱 비밀번호는 권장되지 않으며 대부분의 경우 필요하지 않습니다. 계정을 안전하게 보호하려면 'Google 계정으로 로그인'을 사용하여 앱을 Google 계정에 연결하세요.

앱 비밀번호

없음

이름을 적당히 입력한 후 '만들기'를 클릭

< 앱 비밀번호

앱 비밀번호를 사용하면 최신 보안 표준을 지원하지 않는 오래된 앱 및 서비스에서 Google 계정에 로그인할 수 있습니다.

앱 비밀번호는 최신 보안 표준을 사용하는 최신 앱 및 서비스보다 보안 수준이 낮습니다. 앱 비밀번호를 만들기 전에 앱에 로그인하려면 비밀번호가 필요한지 확인해야 합니다.

[자세히 알아보기](#)

앱 비밀번호가 없습니다.

To create a new app specific password, type a name for it below...

App name

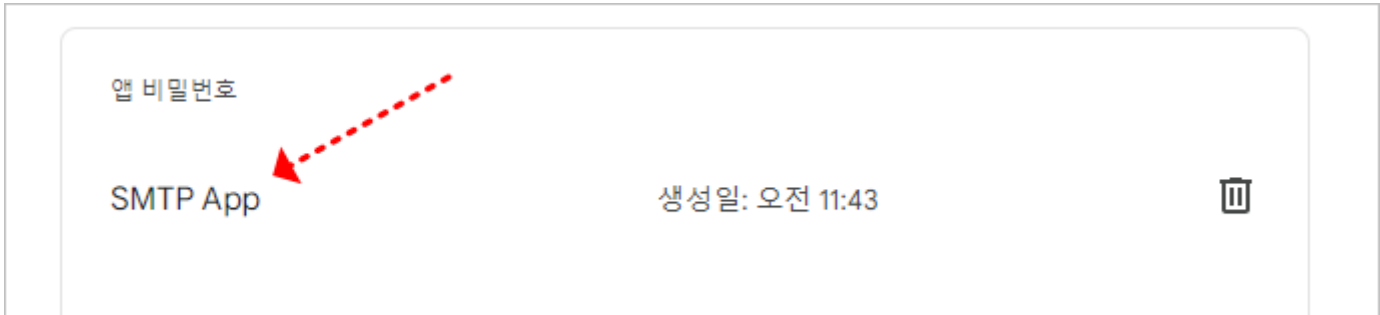
SMTP App

만들기

앱 비밀번호가 이와 같이 생성된다.



확인을 누르면 다음과 같이 생성된것을 볼수 있다.



여기까지 진행하세요.

프로퍼티 파일 : /B15bSMTP_Google/src/main/resources/application.properties.tmp

```
8# SMTP 설정
9spring.mail.host=smtp.gmail.com
10spring.mail.port=587
11spring.mail.username=[redacted]@gmail.com 구글 아이디
12spring.mail.password=[redacted] 발급받은 비밀번호
13
14spring.mail.properties.mail.debug=true
15spring.mail.properties.mail.smtp.connectiontimeout=5000
16spring.mail.properties.mail.smtp.starttls.enable=true
17spring.mail.properties.mail.smtp.starttls.required=true
18spring.mail.properties.mail.smtp.auth=false
19|
```

내용은 이와 같다. 이번 프로젝트에서는 설정 파일로 yaml 파일을 사용해본다.

비활성화를 위해 파일명 끝에 tmp를 붙여준다.

yaml 파일생성 : resources/application.yaml

들여쓰기는 반드시 2칸씩 해야한다.

```
1 #포트설정
2 server:
3   port: '8586'
4 #JSP설정
5 spring:
6   mvc:
7     view:
8       prefix: /WEB-INF/views/
9       suffix: .jsp
10 #SMTP설정
11 mail:
12   host: smtp.gmail.com
13   port: '587'
14   username: [redacted]@gmail.com 구글 아이디
15   password: [redacted] 발급받은 비밀번호
16   properties:
17     mail:
18       debug: 'true'
19     smtp:
20       connectiontimeout: '5000'
21       auth: 'false'
22     starttls:
23       enable: 'true'
24       required: 'true'
25
```

여기까지 작성하세요.

참고 : 프로퍼티 ⇒ yaml로 변경하기

<https://mageddo.com/tools/yaml-converter>

Yaml이란..??

- 시스템 간에 데이터를 주고 받을 필요가 있을때 XML, JSON을 주로 사용
- 이와 유사한 형태로 2001년에 Clark Evans에 의해 제안된 새로운 포매팅 방식이 YAML / YML

Yaml의 어원

- Yaml Ain't Markup Language
- 'YAML은 마크업 언어가 아니다'라는 재귀적 말장난 같은 이름
- 마크업이 핵심이 아니라 데이터가 중심이라는 것을 강조한 의미

.yaml 과 .yml 차이점

- yaml과 yml파일 확장자는 모두 해석 및 구문이 동일
- 옛날 Windows에서 파일확장자가 3자로 제한되는 특성때문에 분리됨
- 이런 이유로 .html 대신 .htm 으로 사용된 적이 있었음
- 따라서 현 시점에서는 .yaml 로 사용하면 됨

XML	JSON	YAML
<pre>1 <Servers> 2 <Server> 3 <name>Server1</name> 4 <owner>Prajwal</owner> 5 <status>active</status> 6 </Server> 7 <Server> 8 <name>Server2</name> 9 <owner>John</owner> 10 <status>inactive</status> 11 </Server> 12 </Servers></pre>	<pre>1 { 2 "Servers": [3 { 4 "name": "Server1", 5 "owner": "Prajwal", 6 "status": "active" 7 }, 8 { 9 "name": "Server2", 10 "owner": "John", 11 "status": "inactive" 12 } 13] 14 }</pre>	<pre>1 Servers: 2 - name: Server1 3 owner: Prajwal 4 status: active 5 - name: Server2 6 owner: John 7 status: inactive</pre>

뷰 : webapp/WEB-INF/views/home.jsp

```
9=<body>
10  <h2>Google 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>
```

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

```
6 <meta charset="UTF-8">
7 <title>SMTP 이메일 전송</title>
8 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jq
9=<script>
10 $(function(){
11   let naMail = "nakjasabal@naver.com";
12   let goMail = "nakjasabal@gmail.com";
13
14   $('input:text[name="from"]').val(goMail);
15   $('input:text[name="to"]').val(naMail);
16 });
17 </script>
18 </head>
```