

# Class[Naver Email]

## Naver Email

이메일 발송 기능은 단순한 메일 용도 뿐만이 아니라,

상품 구매 내역 전달, 예약 확인 정보 전달, 회원 가입 시 인증 처리(ajax) 등 많은 부분에서 다양하게 사용된다.

이메일 전송을 위해서는 **SMTP simple mail transfer protocol** 서버가 별도로 필요하지만 간단한 사용을 위해서 네이버 메일 또는 구글 메일을 사용하면 무료로 편리하게 운영할 수 있다.

하지만 단점은 메일 전송시 전송인의 메일 주소를 변경할 수 없다.

(한 번 발신인의 메일 주소를 설정하면 그 메일 주소로만 발송이 이루어진다.)

위의 단점에도 불구하고 별도의 서버구성 노력이나 비용도 없고 무료로 대량 발송도 가능하기 때문에 많이 사용된다.

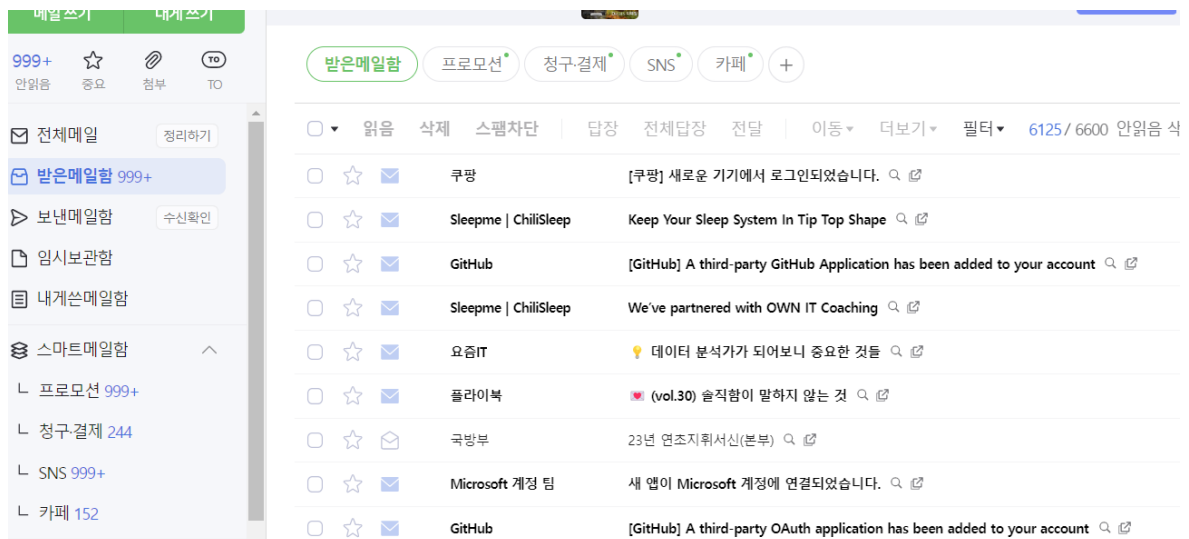
## ▼ 1. 메일 보내기 설정 (네이버 메일함)

### 주의사항

1. 네이버 메일 2차인증 설정이 되어있으면 반드시 해제 해주어야만 한다.

(반대로 구글 메일 이용시에는 2차인증이 설정되어야 한다.)

2. 백신 프로그램이 설치되어 있으면 비활성화해야 메일을 발송할 수 있다.



메일 쓰기

내게 쓰기

999+

안읽음

☆

중요

✎

첨부

10

TO

L SNS 999+

L 카페 152

내 메일함

Unwanted

보낼 편지함

정크

스팸메일함 5

비우기

휴지통

환경설정

고객센터

스마트봇

메일용량

2.19GB / 5GB

공지사항

[안내] 사칭 및 피싱 메일에 주의...

이전 버전으로 가기

환경 설정

기본 설정

메일함관리

메일 자동분류

서명/빠른 답장

부재 중 설정

새 메일 알림

스팸 설정

외부메일 가져오기

POP3/IMAP 설정

단축키

POP3/SMTP 설정

IMAP/SMTP 설정

POP3/SMTP 사용

사용함

사용 안 함

스마트폰, 아웃룩 등에서 네이버 메일을 확인할 수 있도록 POP3/SMTP를 설정합니다.

적용 범위

지금부터 새로 받는 메일만 받음

기존에 받은 메일을 포함하여 받음

읽음 표시

POP3로 읽어진 메일을 읽음 표시

POP3로 읽어진 메일을 읽지 않음으로 표시

원본 저장

네이버 메일에 원본 저장 ?

메일 프로그램 설정에 따라 저장 또는 삭제 ?

외부메일 처리

POP3로 읽어갈 때 외부메일을 포함하지 않음

POP3로 읽어갈 때 외부메일을 포함

기본 설정으로 되돌리기

취소

저장

## 메일 프로그램 환경 설정 안내

스마트폰, 아웃룩 등 외부 메일 프로그램 환경설정에 아래와 같이 등록해 주세요.

계정 정보

아이디 : difbfl4750

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

도움말

[POP3/SMTP 연결이 되지 않아요 >](#)

[메일 프로그램 POP3/SMTP 설정하기 >](#)

Class[Naver Email]

2

POP3/SMTP 설정

IMAP/SMTP 설정

IMAP/SMTP 사용

☒ 사용함
 ☐ 사용 안 함

스마트폰에서 네이버 메일을 더욱 잘 관리하고 쓸 수 있도록 IMAP/SMTP를 설정합니다. (?)

IMAP 동기화 메일 제한

1,000통 ▼

IMAP으로 동기화할 수 있는 최대 메일 통수를 지정할 수 있습니다.  
 너무 많은 메일을 동기화하는 경우 데이터 과금을 주의해 주세요.

기본 설정으로 되돌리기

취소

저장

스마트폰 메일 애플리케이션 계정 설정

스마트폰의 메일 계정 설정에 아래와 같이 등록해 주세요.

계정 정보	아이디 : difbf4750	비밀번호 : 네이버 로그인 비밀번호
서버 명	IMAP 서버명 : imap.naver.com	SMTP 서버명 : smtp.naver.com
포트 정보	IMAP 포트 : 993, 보안연결(SSL) 필요	SMTP 포트 : 587, 보안 연결(TLS) 필요 (TLS가 없는 경우 SSL로 연결)
도움말	<a href="#">IMAP/SMTP 설정 및 해제 방법 &gt;</a>	<a href="#">IMAP이란 무엇인가요? &gt;</a>

POP3 / IMAP

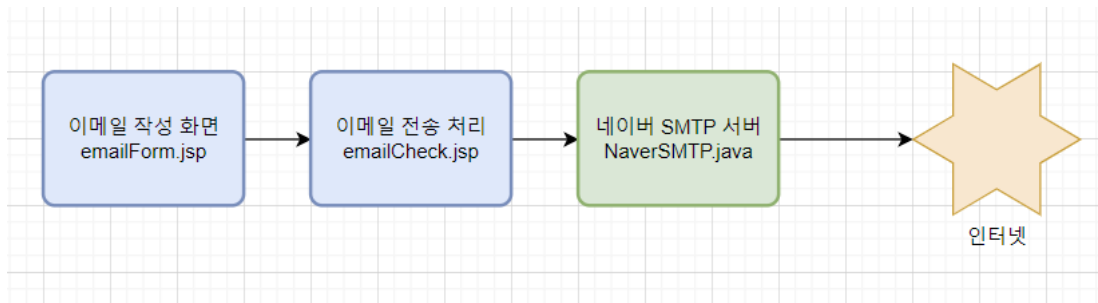
SMTP 는 메일을 보내는 역할을 한다. 하지만 메일을 받을 때는 POP3와 IMAP을 이용한다.

POP3는 클라이언트가 메일 서버에서 메일을 받아오는 프로토콜이다. 즉 메일 서버에 저장된 메일을 사용자의 컴퓨터나 스마트폰으로 가져와서 확인 할 수 있게 해준다. 이 때 주로 outlook과 같은 프로그램을 주로 사용한다.

IMAP은 POP3와 마찬가지로 메일 서버에서 메일을 받는 프로토콜이다. 다른 점은 중앙 서버에서 동기화가 이루어지기 때문에 같은 계정으로 연결된 모든 장치에서 똑같은 내용이 보여지게 된다.

예를 들면 스마트폰에서 메일을 삭제하면 태블릿에서도 함께 지워진다.

메일 전송 과정



## ▼ 2. 메일 보내기 (기본형)

### STEP 1. emailForm.jsp

emailForm.jsp —> emailCheck.jsp —> NaverSMTP.java

간단한 이메일 작성 폼을 만든다.

### 이메일 전송하기

보내는 사람 :

받는 사람 :

제목 :

전송하기

## STEP 2. emailCheck.jsp

emailForm.jsp → emailCheck.jsp → NaverSMTP.java

이메일 처리 페이지를 만든다.

```
<%@page import="java.util.*"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%

    request.setCharacterEncoding("utf-8"); //한글 처리

    Map<String, String> emailInfo = new HashMap<String, String>();

    emailInfo.put("from", request.getParameter("from")); //보내는 사람, 변동 불가, 반드시 본인의 네이버 메일 주소
    emailInfo.put("to", request.getParameter("to")); //받는 사람
    emailInfo.put("subject", request.getParameter("subject")); //메일 제목
    emailInfo.put("content", request.getParameter("content")); //메일 내용

    emailInfo.put("format", "text/plain;charset=UTF-8"); //메일의 형식을 설정. 일단 text로 고정

    try {
        NaverSMTP smtpServer = new NaverSMTP(); //메일 전송 객체 생성
        smtpServer.emailSending(emailInfo);
        out.print("이메일 전송 성공");
    } catch (Exception e) {
        out.print("이메일 전송 실패");
        e.printStackTrace();
    }

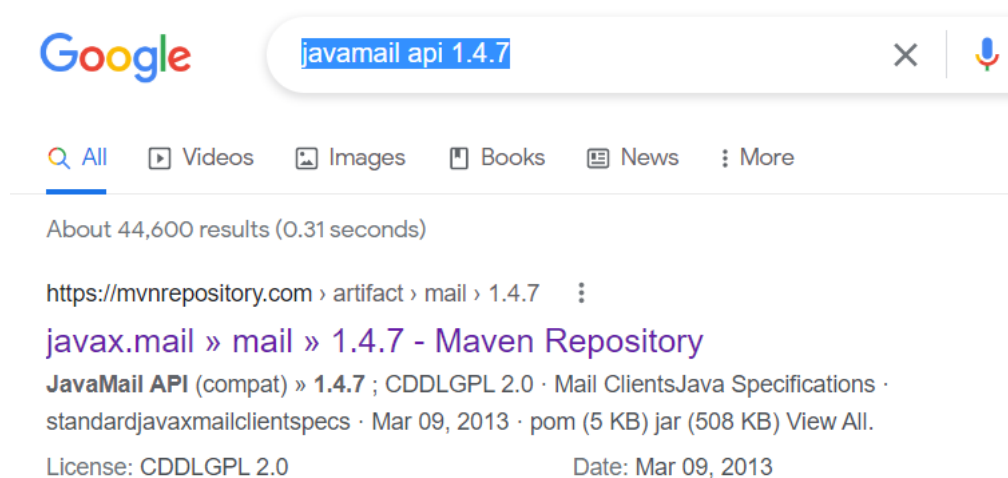
%>
```

## STEP 3. NaverSMTP.java

emailForm.jsp → emailCheck.jsp → NaverSMTP.java

## 네이버 메일 서버 전송 클래스

라이브러리 설치



mvnrepository.com/artifact/javax.mail/mail/1....

# MVN REPOSITORY

Search for groups, artifacts, categories

Categories | Popular | Cor

Indexed Artifacts (31.6M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- Core Utilities
- JVM Languages
- Mocking

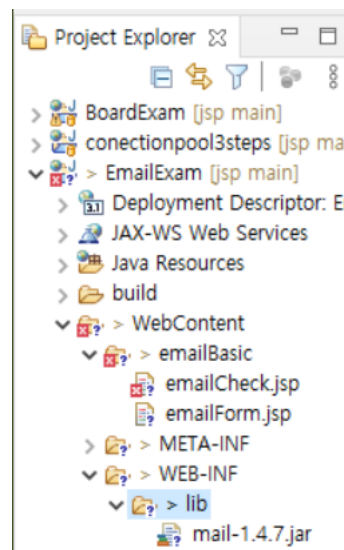
Home » javax.mail » mail » 1.4.7

## JavaMail API (compat) » 1.4.7

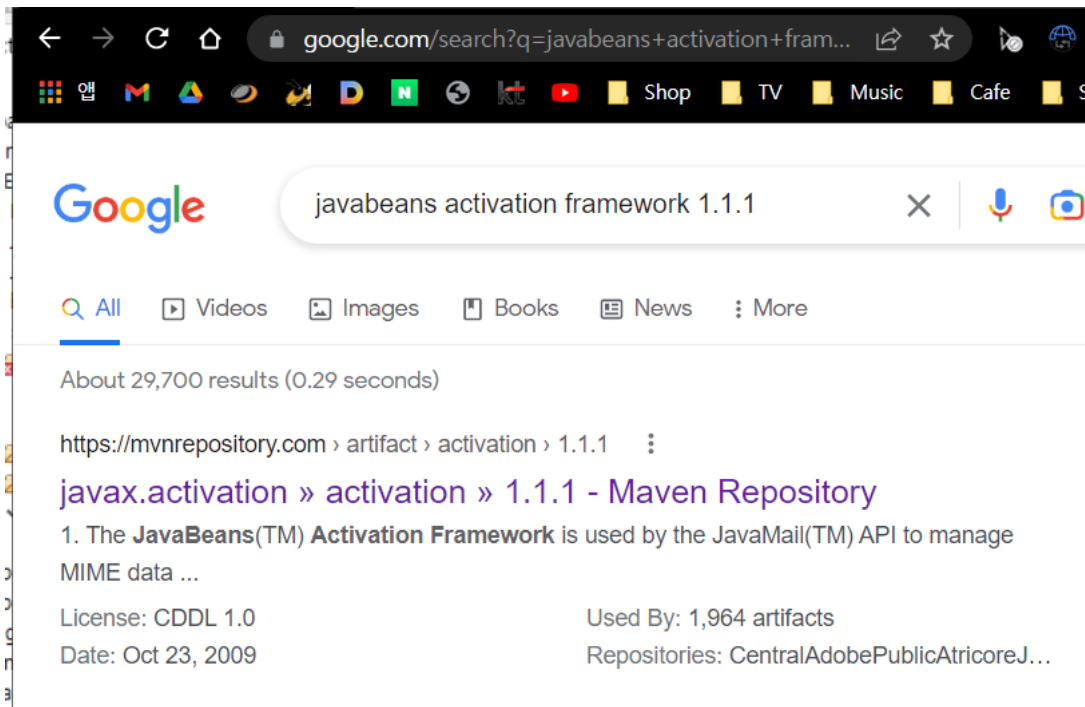
JavaMail API (compat)

License	CDDL   GPL 2.0
Categories	Mail Clients   Java Specifications
Tags	standard   javax   mail   client   specs
Date	Mar 09, 2013
Files	pom (5 KB)   jar (508 KB)   View All
Repositories	Central   Java.net   Mulesoft   Redhat GA   Velocity
Ranking	#156 in MvnRepository (See Top Artifacts) #1 in Mail Clients #8 in Java Specifications
Used By	2,753 artifacts

**Note:** There is a new version for this artifact



lib 아래로 복사



**MVN REPOSITORY**

Search for groups, artifacts, categories

Home » javax.activation » activation » 1.1.1

**JavaBeans(TM) Activation Framework » 1.1.1**

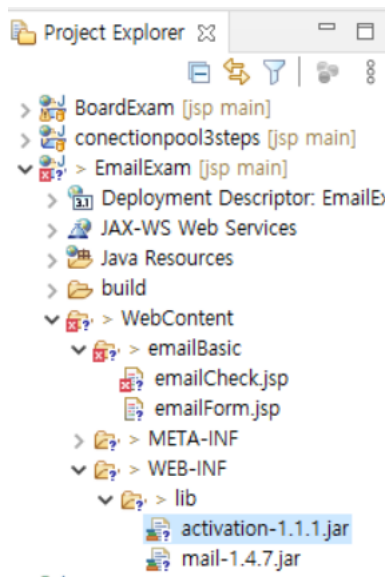
The JavaBeans(TM) Activation Framework is used by the JavaMail(TM) API to manage MIME data

License	CDDL 1.0
Categories	Java Specifications
Tags	javax specs standard
HomePage	http://java.sun.com/javase/technologies/desktop/javabeans/ja ...
Date	Oct 23, 2009
Files	pom (644 bytes) jar (67 KB) View All
Repositories	Central AdobePublic Atricare JCenter Redhat GA Xceptance
Ranking	#228 in MvnRepository (See Top Artifacts) #15 in Java Specifications
Used By	1,965 artifacts

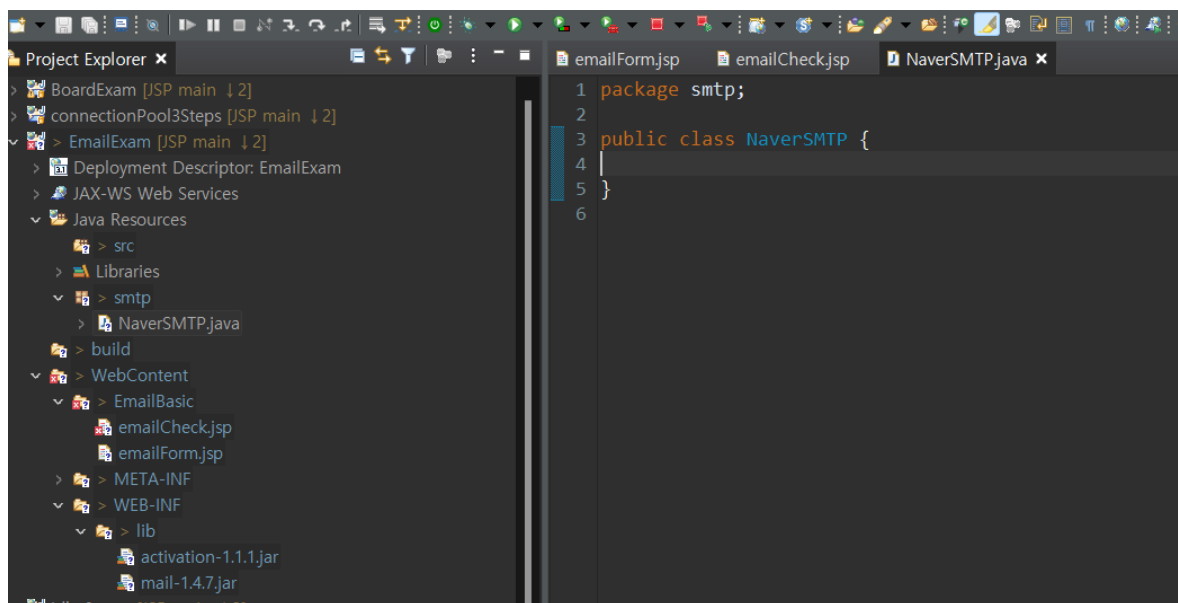
Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

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

jar 다운로드



lib 아래로 복사



메일 전송은

1. 세션 생성 (네이버/다음/구글 메일 서버에 사용자 인증)
2. 메세지 작성 (emailCheck.jsp 에서 만든 객체)
3. 전송

메일 전송

```
package smtp;

import java.util.*;

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



```

import javax.mail.internet.MimeMessage;

public class NaverSMTP {

    private final Properties serverInfo; //서버 정보
    private final Authenticator auth;    //인증 정보

    public NaverSMTP() {
        //생성자의 2가지 특징
        //1. 클래스명과 동일 2.리턴 타입이 없다.
        serverInfo = new Properties();
        serverInfo.put("mail.smtp.host", "smtp.naver.com"); //메일 서버 지정
        serverInfo.put("mail.smtp.port", "465"); // 포트 번호 거의 대부분 동일
        serverInfo.put("mail.smtp.starttls.enable", "true");
        serverInfo.put("mail.smtp.auth", "true");
        serverInfo.put("mail.smtp.debug", "true");
        serverInfo.put("mail.smtp.socketFactory.port", "465");
        serverInfo.put("mail.smtp.socketFactory.class",
            "javax.net.ssl.SSLSocketFactory");
        serverInfo.put("mail.smtp.socketFactory.fallback", "false");

        serverInfo.put("mail.smtp.ssl.protocols", "TLSv1.2"); //맥에서 반드시

        // 사용자 인증 정보
        auth = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("본인네이버 아이디", "본인의 네이버비번");
            }
        };
    }

    public void emailSending(Map<String, String> mailInfo) throws MessagingException {

        //1 세션 생성 ( 네이버/다음/구글 메일 서버에 사용자 인증)

        Session session = Session.getInstance(serverInfo, auth);
        session.setDebug(true);

        //2 메시지 작성 (emailCheck.jsp 에서 만든 객체)

        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress(mailInfo.get("from"))); //보내는 사람
        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(mailInfo.get("to"))); //받는사람
        msg.setSubject(mailInfo.get("subject")); //메일 제목
        msg.setContent(mailInfo.get("content"), mailInfo.get("format"));

        //3 전송

        Transport.send(msg);

    }
}

```

### ▼ 3. 메일 보내기 (디자인)

2번의 경우는 이메일을 단순히 글자로만 보낼 수 있다.

이제는 HTML 을 사용하여 다양하게 꾸며서 메일을 전송해보자.

먼저 메일 작성시 그냥 텍스트 또는 HTML 형식을 사용할 수 있도록 선택하는 옵션을 추가하자.

## 이메일 전송하기

보내는 사람 :

받는 사람 :

제목 :

형식 : ☐ TEXT ☐ HTML

전송하기

### HTML 꾸미기

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>이메일 템플릿</title>
</head>
<body>
<h2> 메일 템플릿 </h2>
<br><br><br>
내용 : __CONTENT__
<br><br><br>
<font color="red">디자인 메일 발송 연습입니다. </font>
</body>
</html>
```

### emailCheck.jsp

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

<%@page import="java.io.*"%>
<%@page import="smtp.*"%>
<%@page import="java.util.*"%>

<%

    request.setCharacterEncoding("utf-8"); //한글 처리

    Map<String, String> emailInfo = new HashMap<String, String>();

    emailInfo.put("from", request.getParameter("from")); //보내는 사람 , 변동 불가, 반드시 본인의 네이버 메일주소
    emailInfo.put("to", request.getParameter("to")); //받는 사람
    emailInfo.put("subject", request.getParameter("subject")); // 메일 제목

    //emailInfo.put("content", request.getParameter("content")); // 메일 내용
    //emailInfo.put("format", "text/plain;charset=UTF-8"); // 메일 형식
```

```

//HTML 또는 TEXT 로 메일 전송 방법 설정

String format = request.getParameter("format");
String content = request.getParameter("content");

if (format.equals("text")) {
    emailInfo.put("content", request.getParameter("content")); // 메일 내용
    emailInfo.put("format", "text/plain;charset=UTF-8"); // 메일 형식
} else if (format.equals("html")) {
    //html 형식으로 변환하여 저장
    content = content.replace("\r\n", "<br/>"); //줄바꿈을 html 형식으로 변환

    String htmlContent = ""; // 전체 내용을 html 형식으로 바꾸어 저장할 변수

    //html 메일 템플릿 파일 읽어오기. 경로가 있을때는 경로명까지 주의
    String templatePath = application.getRealPath("/emailDesign/emailDesign.html");
    BufferedReader br = new BufferedReader(new FileReader(templatePath));

    String oneLine;
    while ((oneLine = br.readLine()) != null) {
        htmlContent += oneLine + "\n";
    }

    br.close();

    // 읽어온 템플릿의 자리표시자인 __CONTENT__ 부분을 메일 내용으로 대체
    htmlContent = htmlContent.replace("__CONTENT__", content);

    // 변경 내용 저장
    emailInfo.put("content", htmlContent); // 메일 내용
    emailInfo.put("format", "text/html;charset=UTF-8"); // 메일 형식
}

try {
    NaverSMTP smtpServer = new NaverSMTP(); //메일 전송 객체 생성
    smtpServer.emailSending(emailInfo);
    out.print("이메일 전송 성공");
} catch (Exception e) {
    out.print("이메일 전송 실패");
    e.printStackTrace();
}

%>

```

## ▼ Modal 창 띄우기 응용

### modal

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh"
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxg0cBQBXU=" crossorigin="

<!--
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
    Launch demo modal
</button> -->

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
    <div class="modal-content">

```

```

<div class="modal-header">
  <h1 class="modal-title fs-5" id="exampleModalLabel">Modal title</h1>
  <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
  메일 발송 성공 !!!
</div>
<div class="modal-footer">
  <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
  <button type="button" class="btn btn-primary">Save changes</button>
</div>
</div>
</div>
</div>

<script>
  $(function() {
    $("#exampleModal").modal("show");
  });
</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBDXo:
</body>
</html>

```

## emailCheck & modal

```

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

<%@page import="java.io.*"%>
<%@page import="smtp.*"%>
<%@page import="java.util.*"%>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh'
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMX0G+800xUf+/Im1MZjXxxg0CBQBXU=" crossorigin="
<%

    request.setCharacterEncoding("utf-8"); //한글 처리

    Map<String, String> emailInfo = new HashMap<String, String>();

    emailInfo.put("from", request.getParameter("from")); //보내는 사람 , 변동 불가, 반드시 본인의 네이버 메일주소
    emailInfo.put("to", request.getParameter("to")); //받는 사람
    emailInfo.put("subject", request.getParameter("subject")); // 메일 제목

    //emailInfo.put("content", request.getParameter("content")); // 메일 내용
    //emailInfo.put("format", "text/plain;charset=UTF-8"); // 메일 형식

    //HTML 또는 TEXT 로 메일 전송 방법 설정

    String format = request.getParameter("format");
    String content = request.getParameter("content");

    if (format.equals("text")) {
        emailInfo.put("content", request.getParameter("content")); // 메일 내용
        emailInfo.put("format", "text/plain;charset=UTF-8"); // 메일 형식
    } else if (format.equals("html")) {
        //html 형식으로 변환하여 저장
        content = content.replace("\r\n", "<br/>"); //줄바꿈을 html 형식으로 변환

        String htmlContent = ""; // 전체 내용을 html 형식으로 바꾸어 저장할 변수

        //html 메일 템플릿 파일 읽어오기 경로가 있을때는 경로명까지 주의
        String templatePath = application.getRealPath("/emailDesign/emailDesign.html");
        BufferedReader br = new BufferedReader(new FileReader(templatePath));

        String oneLine;
        while ((oneLine = br.readLine()) != null) {
            htmlContent += oneLine + "\n";
        }

        br.close();
    }

```

```

// 읽어온 템플릿의 자리표시자인 __CONTENT__ 부분을 메일 내용으로 대체
htmlContent = htmlContent.replace("__CONTENT__", content);

// 변경 내용 저장
emailInfo.put("content", htmlContent); // 메일 내용
emailInfo.put("format", "text/html;charset=UTF-8"); // 메일 형식

}

try {
    NaverSMTP smtpServer = new NaverSMTP(); // 메일 전송 객체 생성
    smtpServer.emailSending(emailInfo);
} /* out.print("이메일 전송 성공"); */

%>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">Modal title</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        메일 발송 성공 !!!
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>

<script>
  $(function() {
    $("#exampleModal").modal("show");
  });
</script>

<%
  } catch (Exception e) {
%>

  <!-- Modal -->
  <div class="modal fade" id="exampleModal2" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h1 class="modal-title fs-5" id="exampleModalLabel">Modal title</h1>
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
        <div class="modal-body">
          메일 발송 실패 !!!
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
          <button type="button" class="btn btn-primary">Save changes</button>
        </div>
      </div>
    </div>
  </div>

  <script>
    $(function() {
      $("#exampleModal2").modal("show");
    });
  </script>

  <%
  }
%>

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBDXo

## ▼ 4. 응용

### 회원 가입 완료 이메일 발송 추가하기

#### DATABASE, TABLE 생성

MySQL#market#user# - HeidiSQL 12.3.0.6589

파일 편집 검색 쿼리 도구 이동 도움말

Host: 127.0.0.1 Database: market Table: user

CREATE TABLE `user` (  
 `to` VARCHAR(50) NOT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',  
 `pass` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',  
 `name` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',  
 `content` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4\_0900\_ai\_ci',  
 PRIMARY KEY (`to`) USING BTREE  
)  
COLLATE='utf8mb4\_0900\_ai\_ci'  
ENGINE=InnoDB  
;

| # | 이름      | 데이터 유형  | 길이/설정 | 부호 없...                  | NULL 허...                           | 0으로...                   | 기본값    |
|---|---------|---------|-------|--------------------------|-------------------------------------|--------------------------|--------|
| 1 | to      | VARCHAR | 50    | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | 기본값 없음 |
| 2 | pass    | VARCHAR | 50    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL   |
| 3 | name    | VARCHAR | 50    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL   |
| 4 | content | VARCHAR | 50    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL   |

```
CREATE TABLE `user` (  
  `to` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `pass` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `name` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `content` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  PRIMARY KEY (`to`) USING BTREE  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
;
```

#### UserDTO.java

```
package jdbc;  
  
public class UserDTO {  
  
    private String to;  
    private String pass;  
    private String name;  
    private String content;  
  
    public String getTo() {  
        return to;  
    }  
    public void setTo(String to) {  
        this.to = to;  
    }  
    public String getPass() {  
        return pass;  
    }
```

```

    }
    public void setPass(String pass) {
        this.pass = pass;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getContent() {
        return content;
    }
    public void setContent(String content) {
        this.content = content;
    }
}

public UserDTO(String to, String pass, String name, String content) {
    super();
    this.to = to;
    this.pass = pass;
    this.name = name;
    this.content = content;
}
}

```

## UserDAO.java

```

package jdbc;

import java.sql.*;

import javax.naming.NamingException;

import util.*;

public class UserDAO {

    // Connection Pool 셋팅
    // 1.context.xml
    // 2.ConnectionPool.java
    // 3.connector lib

    public static int join(String to, String pass, String name, String content)
        throws NamingException, SQLException{

        String sql = "INSERT INTO user VALUES(?,?,?,?)";

        Connection conn = ConnectionPool.get();

        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, to);
        pstmt.setString(2, pass);
        pstmt.setString(3, name);
        pstmt.setString(4, content);

        int result = pstmt.executeUpdate();

        return result;
    }
}

```

## UserCheck.jsp

```

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

<%@page import="jdbc.UserDAO"%>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh"
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxg0cBQBXU=" crossorigin="true">

```

```

<%

request.setCharacterEncoding("utf-8"); //한글 처리

String to = request.getParameter("to");
String pass = request.getParameter("pass");
String name = request.getParameter("name");
String content = request.getParameter("content");

int result = UserDao.join(to, pass, name, content);

//회원 가입 성공시 모달 창으로 "회원 가입 완료", "축하합니다."

//회원 가입 실패시 모달 창으로 "회원 가입 실패", "다시 시도하세요"

if(result == 1){
%>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">회원 가입 성공</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        회원 가입을 축하합니다.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>

<script>
  $(function() {
    $("#exampleModal").modal("show");
  });
</script>

<%
} else {
%>

<!-- Modal -->
<div class="modal fade" id="exampleModal2" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">회원 가입 실패</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        회원 가입에 실패하였습니다. 다시 시도하세요
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>

<script>
  $(function() {
    $("#exampleModal2").modal("show");
  });
</script>

<%

```



```

    }
    %>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBO"

```

## UserForm.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GlH

```

```

<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navb
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            회원관리
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">로그인</a></li>
            <li><a class="dropdown-item" href="#">로그아웃</a></li>
            <li><a class="dropdown-item" href="UserForm.jsp">회원가입</a></li>
            <li><a class="dropdown-item" href="#">회원탈퇴</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>

<div class="container">
<form action="UserCheck.jsp" method="post" class="row g-3">

  <div class="col-md-6">
    <label for="inputEmail4" class="form-label">Email</label>
    <input name="to" type="email" class="form-control" id="inputEmail4">
  </div>
  <div class="col-md-6">
    <label for="inputPassword4" class="form-label">Password</label>
    <input name="pass" type="password" class="form-control" id="inputPassword4">
  </div>
  <div class="col-12">
    <label for="inputAddress" class="form-label">Name</label>
    <input name="name" type="text" class="form-control" id="inputAddress">
  </div>
  <div class="col-12">
    <label for="inputAddress2" class="form-label">Content</label>
    <input name="content" type="text" class="form-control" id="inputAddress2">
  </div>

  <div class="col-12">
    <button type="submit" class="btn btn-primary">회원 가입</button>
  </div>
</form>
</div>

```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdKMBDXo"
</body>
</html>
```

## Welcome.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>가입 환영</title>
</head>
<body>
<h2> 가입 환영 </h2>
<br><br><br>
성명 : __NAME__
이메일 : __TO__
암호 : __PASS__
<br><br><br>
<font color="red">__NAME__님 회원 가입을 축하합니다. </font>
</body>
</html>
```

## ▼ Connection Pool 설정

### ConnectionPool.java

```
package util;

import java.sql.*;
import javax.naming.*;
import javax.sql.DataSource;

public class ConnectionPool {
    private static DataSource _ds = null;

    public static Connection get() throws NamingException, SQLException {
        if (_ds == null ) {
            _ds = (DataSource) (new InitialContext()).lookup("java:comp/env/jdbc/market");
        }
        return _ds.getConnection();
    }
}
```

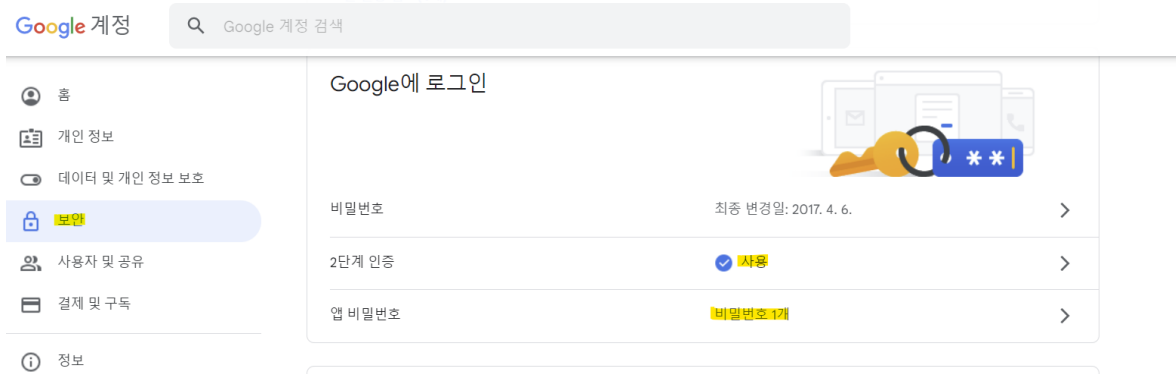
### context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
  <Resource name="jdbc/market"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/market?serverTimezone=UTC"
    username="root"
    password="0000"
    maxTotal="16"
    maxIdle="4"
    maxWaitMillis="-1" />
</Context>
```

## ▼ 5. Gmail 이용

GmailSMTP도 2차인증 설정과 비밀번호 설정 이외에는 NaverSMTP와 사용법이 비슷하다.

1. Google 계정에서 2단계 인증 설정.
2. 앱 비밀번호 받기.



### UserCheck.jsp

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh"
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPujgMXy0G+800xUf+/Im1MZjXxxg0cBQBXU=" crossorigin="
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBDXo:

<%@page import="jdbc.*"%>
<%@page import="java.io.*"%>
<%@page import="java.util.*"%>
<%@page import="smtp.*"%>

<%

    request.setCharacterEncoding("utf-8"); //한글 처리

    String to = request.getParameter("to");
    String pass = request.getParameter("pass");
    String name = request.getParameter("name");
    String content = request.getParameter("content");

    Map<String, String> emailInfo = new HashMap<String, String>();

    emailInfo.put("from", "difbfl4750@gmail.com"); //보내는 사람, 변동 불가, 반드시 본인의 네이버 메일 주소
    emailInfo.put("to", request.getParameter("to")); //받는 사람
    emailInfo.put("subject", "[구글] 회원가입을 축하합니다."); //메일 제목

    content = content.replace("\r\n", "<br/>"); //줄바꿈을 html 형식으로 변환

    String htmlContent = ""; //전체 내용을 html 형식으로 바꾸어 저장할 변수

    //html 메일 템플릿 파일 읽어오기
    String templatePath = application.getRealPath("/Welcome.html");
    BufferedReader br = new BufferedReader(new FileReader(templatePath));

    String oneLine;
    while ((oneLine = br.readLine()) != null) {
        htmlContent += oneLine + "\n";
    }

    br.close();

    // 읽어온 템플릿의 자리표시자인 __CONTENT__ 부분을 메일 내용으로 대체
    htmlContent = htmlContent.replace("__NAME__", name);
    htmlContent = htmlContent.replace("__TO__", to);
    htmlContent = htmlContent.replace("__PASS__", pass);
```

```

// 변경 내용 저장

emailInfo.put("content", htmlContent); //메일 내용
emailInfo.put("format", "text/html;charset=UTF-8"); //메일 형식


GoogleSMTP smtpServer = new GoogleSMTP(); //메일 전송 객체 생성
smtpServer.emailSending(emailInfo);


//회원가입 성공시 모달 창으로 "회원가입 완료", "축하합니다."
//회원가입 실패시 모달 창으로 "회원가입 실패", "다시 시도하세요"

int result = UserDao.join(to, pass, name, content);


if(result == 1) {

// response.sendRedirect("modalTest.jsp?check=success");
%> <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">이메일 전송 체크</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        회원가입 완료. 축하합니다!
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">확인</button>
      </div>
    </div>
  </div>
</div>

<script>
  $(function() {
    $("#exampleModal").modal("show");
  });
</script>

<% } else {
// response.sendRedirect("modalTest.jsp?check=fail");
%> <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">이메일 전송 체크</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        회원가입 실패. 다시 시도하세요.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">확인</button>
      </div>
    </div>
  </div>
</div>

<script>
  $(function() {
    $("#exampleModal").modal("show");
  });
</script>
<%
}
%>

```

## GoogleSMTP.java

```

package smtp;

import java.util.*;

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

```

```

public class GoogleSMTP {

    private final Properties serverInfo; //서버 정보
    private final Authenticator auth; //인증 정보

    public GoogleSMTP() {
        //생성자의 2가지 특징
        //1. 클래스명과 동일 2. 리턴 타입이 없다.
        serverInfo = new Properties();
        serverInfo.put("mail.smtp.host", "smtp.gmail.com"); //메일 서버 지정
        serverInfo.put("mail.smtp.port", "465"); //포트 번호(거의 대부분 동일)
        serverInfo.put("mail.smtp.starttls.enable", "true");
        serverInfo.put("mail.smtp.auth", "true");
        serverInfo.put("mail.smtp.debug", "true");
        serverInfo.put("mail.smtp.socketFactory.port", "465");
        serverInfo.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        serverInfo.put("mail.smtp.socketFactory.fallback", "false");

        serverInfo.put("mail.smtp.ssl.protocols", "TLSv1.2"); //맥에서 사용하기 위한 코드

        // 사용자 인증 정보
        auth = new Authenticator() {

            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("difbfl4750@gmail.com", "앱 비밀번호");
            }

        };
    }

    public void emailSending(Map<String, String> mailInfo) throws AddressException, MessagingException {

        //1. 세션 생성 ( 네이버/다음/구글 메일 서버에 사용자 인증)

        Session session = Session.getInstance(serverInfo, auth);
        session.setDebug(true);

        //2. 메시지 작성 (emailCheck.jsp 에서 만든 객체)

        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress(mailInfo.get("from"))); //보내는 사람
        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(mailInfo.get("to"))); //받는 사람
        msg.setSubject(mailInfo.get("subject")); //메일 제목
        msg.setContent(mailInfo.get("content"), mailInfo.get("format"));
        //3. 전송

        Transport.send(msg);
    }

}

```