

Commons Mail
Java Mail

매뉴얼.

MAIL. Content.

First subject

라이브러리와 관련 사이트

라이브러리에 관련된
jar파일을 다운 받는 방법

01

Second subject

메일 보내기를 위한
NAVER, Google 환경설정

02

Third subject

Commons Mail, Java Mail
코드 설명

메일 보내기 예제 프로그램 시연

03

Commons Mail

Commons 메일 라이브러리

: Apache Commons 프로젝트의 일부로 제공, 자바 메일 라이브러리를 감싸고 있어 더 쉽게 사용할 수 있도록 함.

- JavaMail API보다 더 직관적이고 간결한 API를 제공하여 이메일을 보내는 프로세스를 단순화
- 이메일에 텍스트, HTML, 첨부 파일 등을 추가하는 데 특히 편리
- Commons 라이브러리의 일부로, 자주 사용되는 기능들을 재사용할 수 있어 개발자들이 더 빠르게 이메일 기능을 구현 가능

Java Mail

Java 메일 라이브러리

: Java 플랫폼의 공식 메일 전송 API
이메일을 생성, 송신, 수신, 읽기 및 삭제하는 기능을 제공

- SMTP, IMAP, POP3 등 다양한 프로토콜을 지원하여 이메일을 송수신
- MIME(Multipurpose Internet Mail Extensions)를 통해 텍스트, HTML, 이미지, 첨부 파일 등 다양한 형식의 이메일을 작성
- JavaMail API는 Java EE (Enterprise Edition)와 Java SE (Standard Edition)의 일부로 포함되어 있음.

Maven Repository

Java 개발자들이 필요한 라이브러리와 도구를 쉽게 찾고 다운로드할 수 있도록 제공되는 저장소

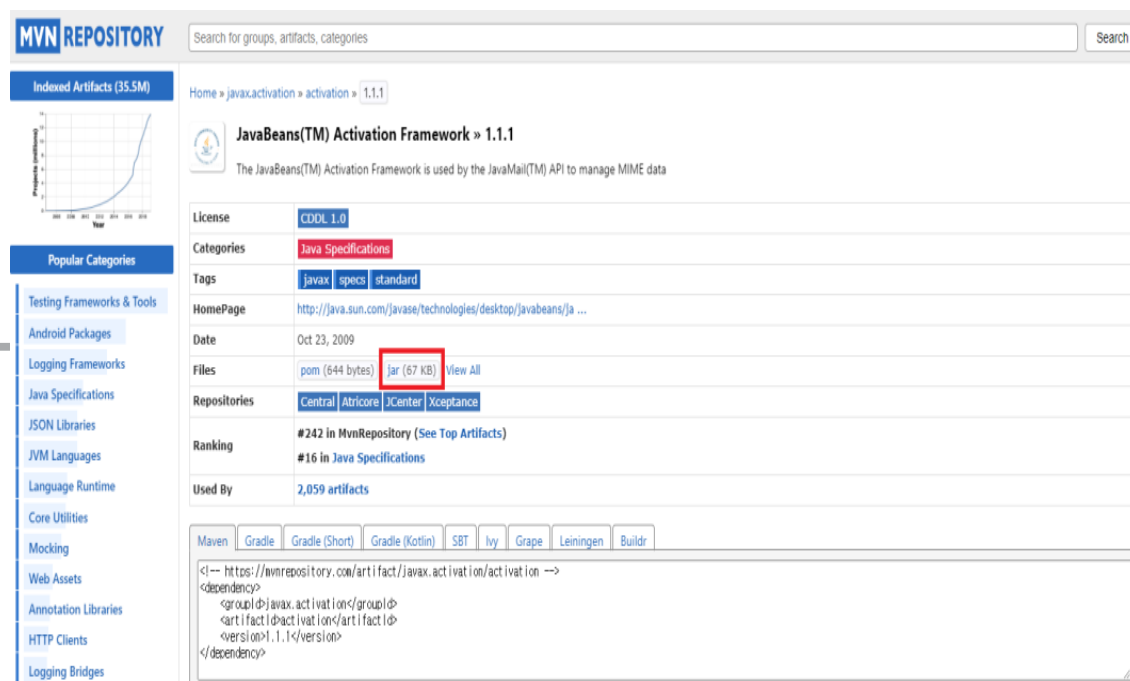
- 편리성, 안정성, 효율성, 다양성, 공동 작업 기능 등 다양한 장점을 제공

The screenshot displays the Maven Repository homepage. At the top, there's a search bar and a navigation menu. The main content area is divided into sections: 'Indexed Artifacts (35.6M)' with a line graph showing growth over time, 'Popular Categories' listing various Java-related tools and frameworks, and 'What's New in Maven' featuring a list of recent updates. The 'What's New' section includes details for several SDKs, such as Tencentcloud SDK Java, AWS Java SDK For Auto Scaling, AWS Java SDK For Elastic Load Balancing, GraphQL DGS, AWS Java SDK For AWS Elastic Beanstalk, and AWS Java SDK For AWS Config. Each entry shows the artifact ID, version, a brief description, the last release date, and the number of usages, along with a link to the Apache license.

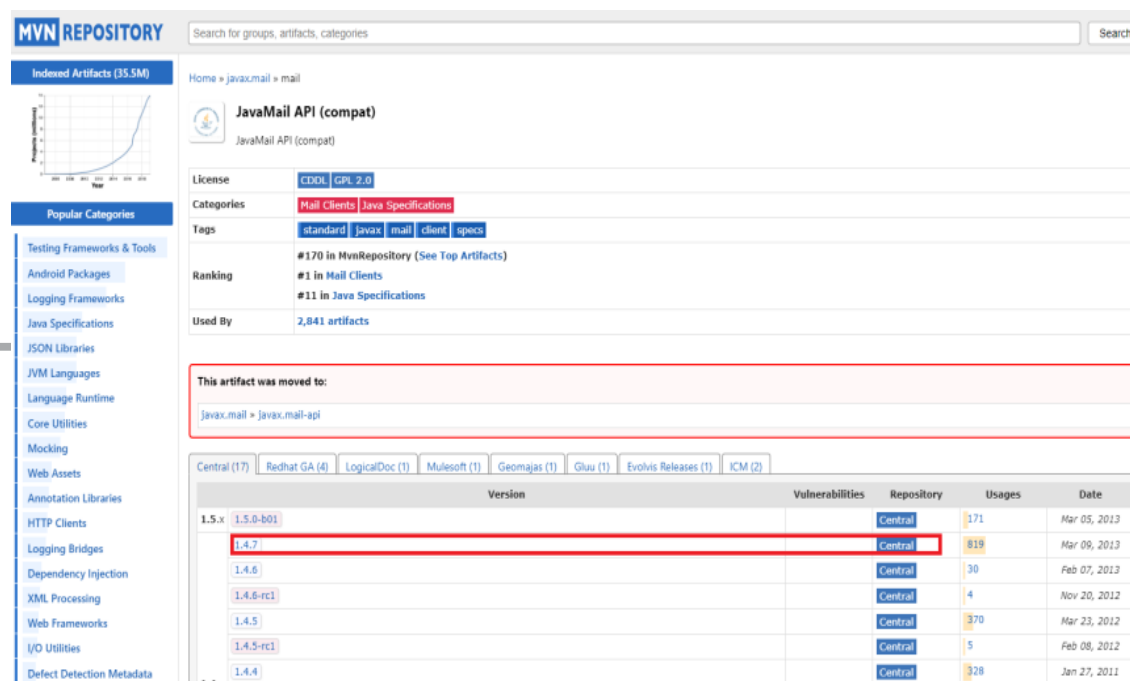
Artifact	Version	Usage	Last Release
Tencentcloud SDK Java	3.1.1016	74 usages	May 10, 2024
AWS Java SDK For Auto Scaling	1.12.720	52 usages	May 10, 2024
AWS Java SDK For Elastic Load Balancing	1.12.720	35 usages	May 10, 2024
GraphQL DGS	8.6.0	29 usages	May 10, 2024
AWS Java SDK For AWS Elastic Beanstalk	1.12.720	25 usages	May 10, 2024
AWS Java SDK For AWS Config	1.12.720	13 usages	May 10, 2024

1. Maven Repository (<https://mvnrepository.com/>) 에서 jar 파일 다운로드

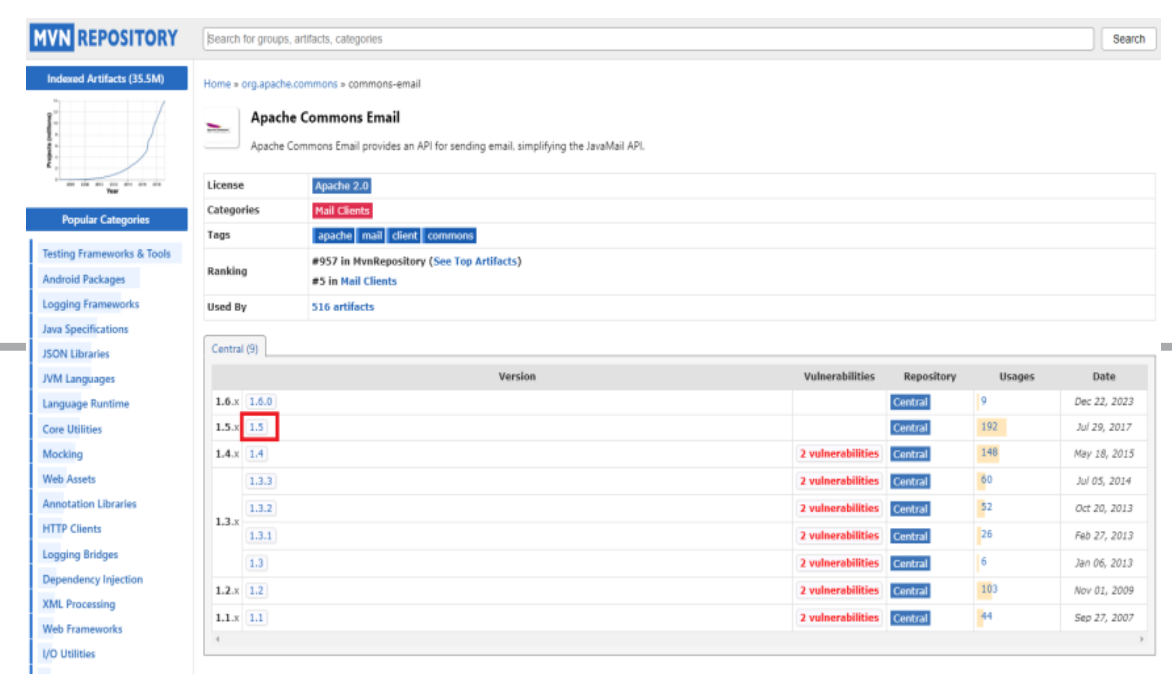
- JavaMail API (compat) 1.4.7
- JavaBeans(TM) Activation Framework 1.1.1
- Apache Commons Email 1.5
- 파일 다운로드 후 이클립스 프로젝트 파일에 jar 파일 import



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



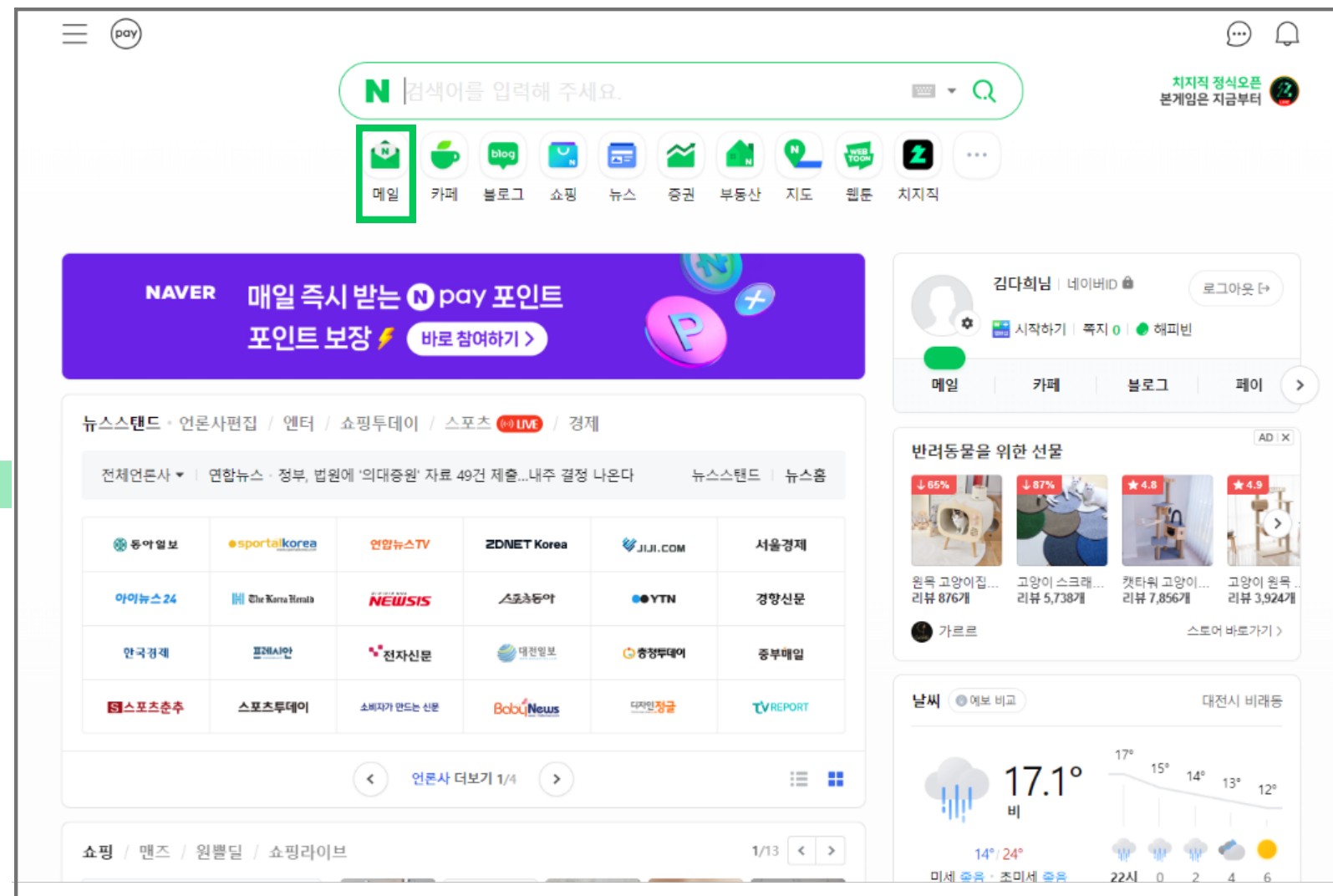
<https://mvnrepository.com/artifact/javax.mail/mail>



<https://mvnrepository.com/artifact/org.apache.commons/commons-email>

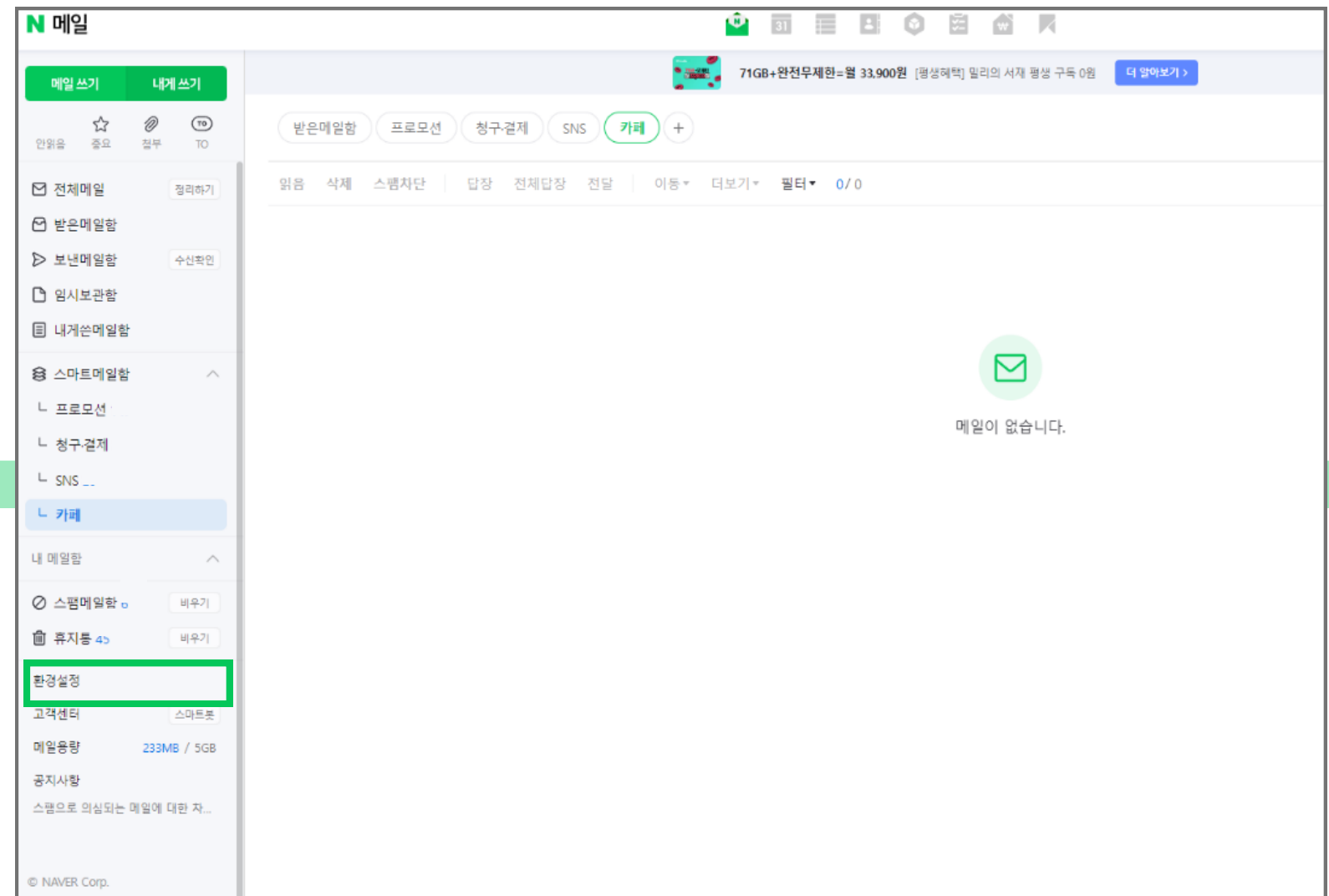
NAVER - 환경설정

02



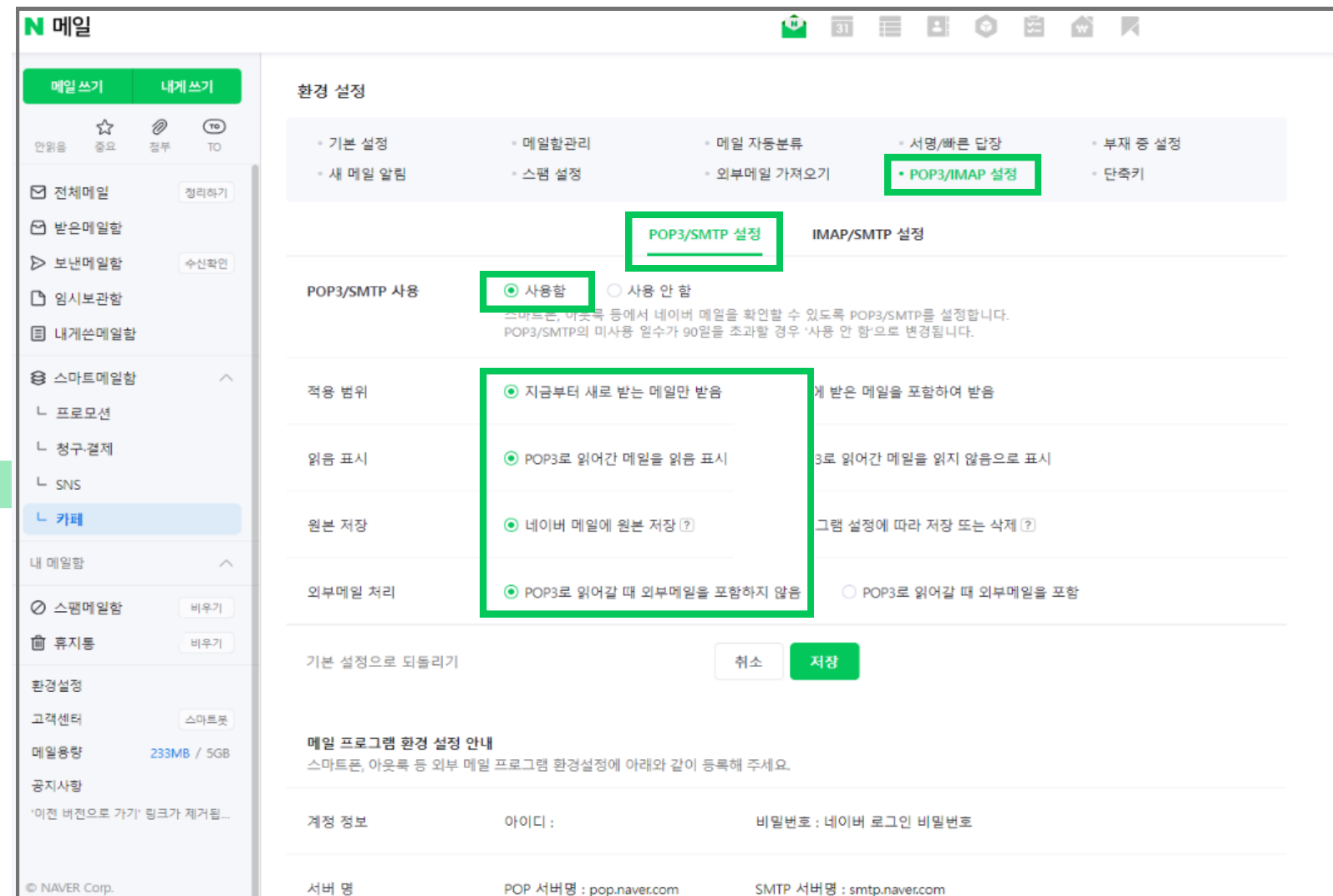
1

네이버 로그인 후 메일 화면으로 이동



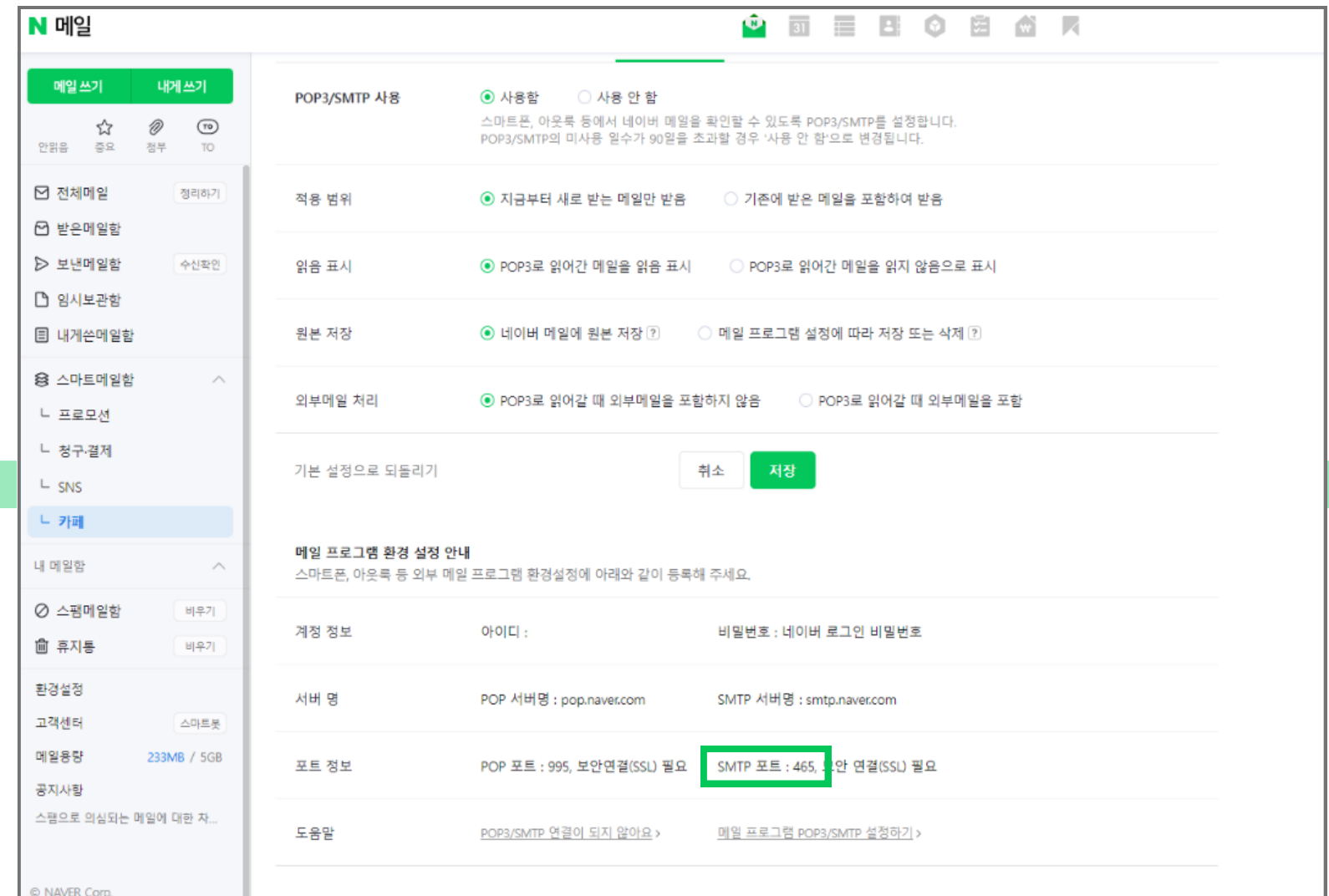
2

네이버 메일 화면 좌측에 환경 설정 선택



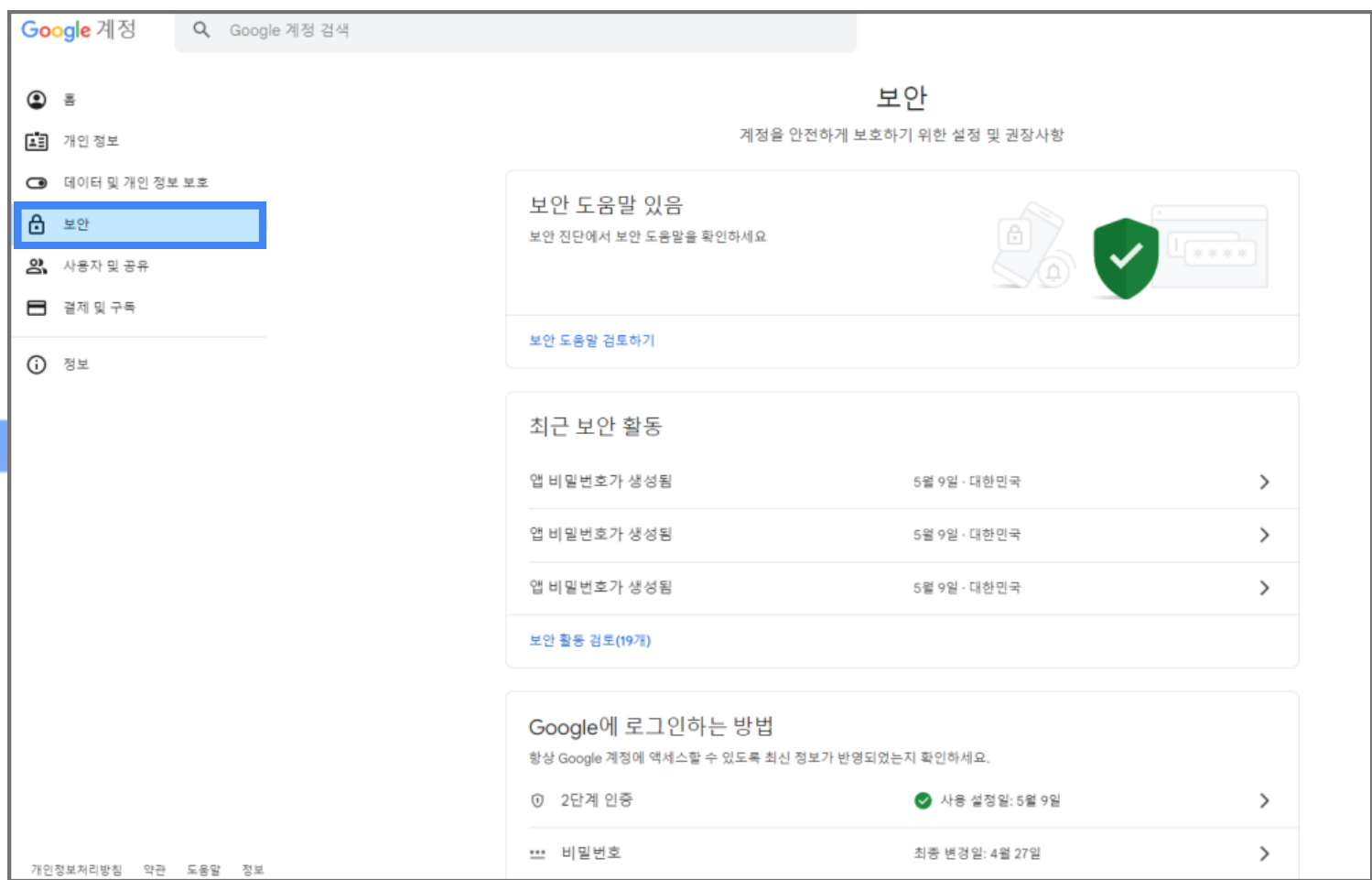
3

POP/IMAP 설정 > POP3/SMTP 설정



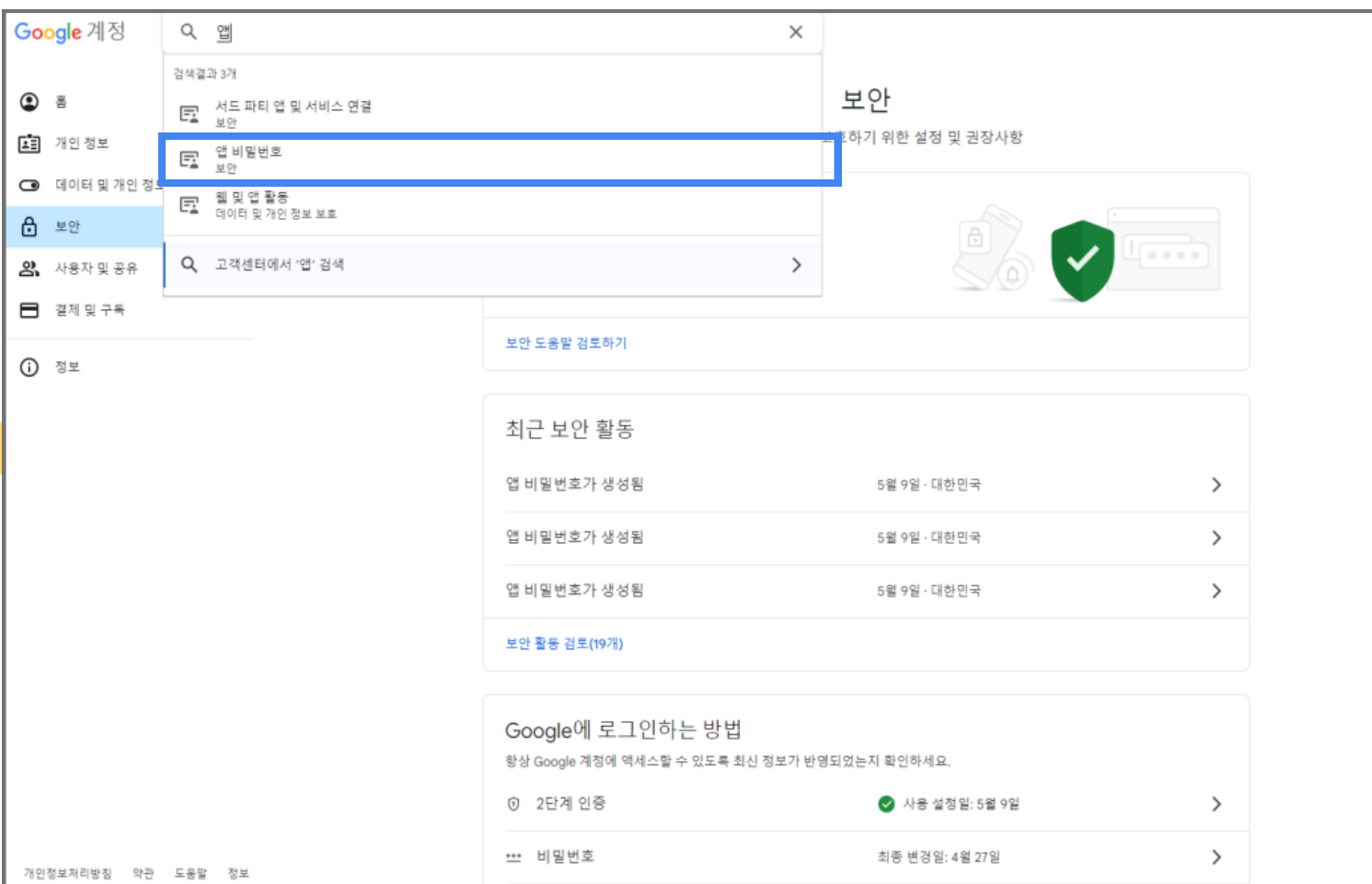
4

SMTP 포트 번호 확인



1

구글 로그인 후 계정 설정으로 이동
-> 좌측 보안 선택



2

상단 검색창에 앱 검색 후 앱 비밀번호 클릭

Google

계정

< 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.

앱 전용 비밀번호를 새로 만들려면 아래에 앱 이름을 입력하세요...

앱 이름

메일

만들기

개인정보처리방침

약관

도움말

정보

3

앱 이름 입력

< 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 만들려면 최신 보안 표준을 지원하는 앱에 로그인해야 합니다.
[자세히 알아보기](#)

생성된 앱 비밀번호

기기용 앱 비밀번호

kbwh jbdn uwgu yecn

사용 방법

설정하려는 애플리케이션 또는 기기의 Google 계정 설정으로 이동합니다. 비밀번호를 위해 표시된 16자리 비밀번호로 교체합니다.
일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되도록 적어 놓거나 다른 사용자와 공유하지 마세요.

확인

도움말

정보

4

앱 비밀번호 생성

Commons Mail

전체 코드

03

```
package kr.or.ddit.basic;

import org.apache.commons.mail.DefaultAuthenticator;
import org.apache.commons.mail.Email;
import org.apache.commons.mail.EmailException;
import org.apache.commons.mail.SimpleEmail;

public class CommonMail {
    public static void main(String[] args) {
        // 이메일 객체 생성
        Email email = new SimpleEmail();

        // SMTP 서버 설정
        email.setHostName("smtp.gmail.com");
        email.setSmtpPort(587); // Gmail SMTP 포트
        email.setAuthenticator(new DefaultAuthenticator("sender@gmail.com", "uozf lalr lzan qjep")); // 발신자 Gmail 계정 정보
        email.setStartTLSRequired(true); // TLS 사용

        // 발신자 이메일 설정
        try {
            email.setFrom("sender@gmail.com");
        } catch (EmailException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }

        try {
            // 이메일 내용 설정
            email.setSubject("제목");
            email.setMsg("내용");

            // 수신자 이메일 설정
            email.addTo("receiver@naver.com");

            // 이메일 전송
            email.send();

            System.out.println("이메일 전송 성공!");
        } catch (EmailException e) {
            e.printStackTrace();
        }
    }
}
```



Apache Commons Email

이메일 전송을 위한 API 제공

```
import org.apache.commons.mail.DefaultAuthenticator;  
import org.apache.commons.mail.Email;  
import org.apache.commons.mail.EmailException;  
import org.apache.commons.mail.SimpleEmail;
```



DefaultAuthenticator : Username,password을 입력 받아 SMTP 서버 인증에 사용

Email : 이메일의 송/수신자, 제목, 본문 설정 및 메일 송신에 사용

EmailException : 이메일작업 중 발생하는 예외 처리에 사용

SimpleEmail : 텍스트 기반 이메일 전송에 사용

// 이메일 객체 생성

Email **email** = **new** SimpleEmail();



Simple Email () - 생성자 호출하여 객체 생성
Email 인터페이스로 참조

```
// SMTP 서버 설정 - SMTP (Simple Mail Transfer Protocol): 이메일을 보내는 데 사용되는 표준 프로토콜
email.setHostName("smtp.gmail.com");
email.setSmtpPort(587); // Gmail SMTP 포트
email.setAuthenticator(new DefaultAuthenticator("sender@gmail.com", "uozf lalr lzan qjep")); // 발신자 Gmail 계정 정보
```

setHostName() : SMTP 호스트 이름 설정

- Google : smtp.gmail.com
- naver : smtp.naver.com

setSmtpPort() : SMTP 포트 설정

- 송신에 사용됨
 - TLS 사용 시 : 587
 - SSL 사용 시 : 465
- (구글은 TLS를 사용하며, 네이버는 SSL/TLS를 사용)

setAuthenticator() : 사용자의 인증 정보 설정



DefaultAuthenticator(name, pw) : 사용자의 이름과 비밀번호 전달

- SMTP 서버에 로그인하기 위함
 - Google : 앱 비밀번호 입력
 - naver : 기존 비밀번호 입력
- (2단계 인증 사용할 경우에는 앱 비밀번호 입력)

```
email.setStartTLSRequired(true); // TLS 사용
```



TLS 활성화하기 위해 사용

```
// 발신자 이메일 설정
try {
    email.setFrom("sender@gmail.com");
} catch (EmailException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
```

setFrom() : 송신자 이메일 주소 입력

```
try {  
    // 이메일 내용 설정  
    email.setSubject("제목");  
    email.setMsg("내용");  
  
    // 수신자 이메일 설정  
    email.addTo("receiver@naver.com");  
  
    // 이메일 전송  
    email.send();  
  
    System.out.println("이메일 전송 성공!");  
} catch (EmailException e) {  
    e.printStackTrace();  
}
```



setSubject() : 이메일 제목 입력
setMsg() : 이메일 내용 입력
addTo() : 수신자 이메일 주소 입력
send() : 메일 발송

Java Mail

전체 코드

```
package kr.or.ddit.basic;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class JavaMail {
    public static void main(String[] args) {
        // Gmail 계정 정보
        final String username = "보낼 이메일주소";
        final String password = "앱 비밀번호";

        // SMTP 서버 설정
        Properties props = new Properties();
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.port", "587");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");

        // 인증 객체 생성
        Authenticator auth = new Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(username, password);
            }
        };

        // 세션 생성
        Session session = Session.getInstance (props, auth);

        try {
            // 메시지 생성
            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress("보낼 이메일 주소"));
            message.setRecipients(Message.RecipientType.TO, InternetAddress.parse ("받을 이메일 주소"));
            message.setSubject("제목 입력");
            message.setText("내용 입력");

            // 이메일 전송
            Transport.send (message);

            System.out.println("메일을 성공적으로 전송하였습니다.!");
        } catch (MessagingException e) {
            e.printStackTrace();
        }
    }
}
```

사용한 라이브러리

```
import javax.mail.Authenticator;  
import javax.mail.Message;  
import javax.mail.MessagingException;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Session;  
import javax.mail.Transport;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;
```

```
// Gmail 계정 정보  
final String username = "보낼 이메일주소";  
final String password = "앱 비밀번호";
```

Username : 보낼 이메일 주소

password : 비밀번호 또는 앱 비밀번호를 할당

final :: 이메일 계정 정보는 보안에 민감한 정보이며, 타인에 의해 의도치 않게 수정되어서는 안 됨

// SMTP 서버 설정

```
Properties props = new Properties();  
props.put("mail.smtp.host", "smtp.gmail.com");  
props.put("mail.smtp.port", "587");  
props.put("mail.smtp.auth", "true");  
props.put("mail.smtp.starttls.enable", "true");
```

Properties는 key, value들을 관리하는 데 사용 (SMTP 서버 설정에 대한 정보를 담는다)

- 1) 서버의 호스트 (**host**) 주소 설정! - 이메일을 보낼 때 사용할 SMTP 서버의 주소를 지정 (네이버 메일이면 smtp.naver.com)
- 2) 서버의 포트 (**port**) 번호 지정! - SMTP 서버와 통신하기 위한 포트 번호 지정 (587은 TLS를 사용하는 SMTP 서버에 대한 포트 번호, SSL을 사용하면 465)
- 3) 사용자 인증 (**auth**) 을 활성화할지 여부 설정! - false로 설정하면 인증 비활성화 (false로 설정해도(사용자 인증 없이) 이메일 전송이 가능함)
-> 안전한 이메일 전송을 보장하지 않을 뿐임. 누구나 해당 SMTP 서버를 사용하여 스팸 방지를 위해 이메일을 보낼 수 있게 되기 때문에 비권장.
- 4) SMTP 연결에 대해 TLS (**starttls.enable**) 를 사용할지 여부 설정! - false로 바뀌도 전송 가능 (google은 tls를 활성화 해야만 전송 가능)
-> 이메일 통신의 보안 수준이 낮아져 보안에 취약해질 수 있기 때문에 비권장.

// 인증 객체 생성

```
Authenticator auth = new Authenticator() {  
    protected PasswordAuthentication getPasswordAuthentication() {  
        return new PasswordAuthentication(username, password);  
    }  
};
```

Authenticator인증 객체 생성

- 1) getPasswordAuthentication()는 추상 메서드를 오버라이딩
- 2) 앞서 선언한 username과 password를 정보를 저장

(Session을 생성할 때 사용되어 SMTP서버에 사용자를 인증하기 위해 사용됨.)

// 세션 생성

```
Session session = Session.getInstance (props, auth);
```



SMTP 서버와의 통신을 하기 위한 세션을 생성

- 1) Message객체를 생성할 때 Session객체가 필요
- 2) props와 auth는 앞서 생성한 SMTP 서버 설정과 인증 정보를 담고있는 Properties 객체, auth는 인증 객체

```
try {  
    // 메시지 생성  
    Message message = new MimeMessage(session);  
    message.setFrom(new InternetAddress("보낼 이메일 주소"));  
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse("받을 이메일 주소"));  
    message.setSubject("제목 입력");  
    message.setText("내용 입력");  
  
    // 이메일 전송  
    Transport.send(message);  
  
    System.out.println("메일을 성공적으로 전송하였습니다.!!");  
} catch (MessagingException e) {  
    System.out.println("메일 전송 실패!...");  
}
```

1. MimeMessage(session)

1) MimeMessage 클래스의 생성자

이메일의 MIME(Multipurpose Internet Mail Extensions) 형식 메시지를 생성

2) 앞서 설명한 인증과 SMTP설정 정보를 갖고 있는 Session을 매개변수로 생성

```
try {  
    // 메시지 생성  
    Message message = new MimeMessage(session);  
    message.setFrom(new InternetAddress("보낼 이메일 주소"));  
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse("받을 이메일 주소"));  
    message.setSubject("제목 입력");  
    message.setText("내용 입력");  
  
    // 이메일 전송  
    Transport.send(message);  
  
    System.out.println("메일을 성공적으로 전송하였습니다.!!");  
} catch (MessagingException e) {  
    System.out.println("메일 전송 실패!...");  
}
```

2.message.setFrom(address)

1) 이메일의 발신자 주소를 설정

address는 InternetAddress 객체로 발신자 이메일 주소를 나타냄


```
try {  
    // 메시지 생성  
    Message message = new MimeMessage(session);  
    message.setFrom(new InternetAddress("보낼 이메일 주소"));  
    message.setRecipients(Message.RecipientType.TO, InternetAddress.parse("받을 이메일 주소"));  
    message.setSubject("제목 입력");  
    message.setText("내용 입력");  
  
    // 이메일 전송  
    Transport.send(message);  
  
    System.out.println("메일을 성공적으로 전송하였습니다.!!");  
} catch (MessagingException e) {  
    System.out.println("메일 전송 실패!...");  
}
```

Transport.send(message);

- 이메일을 실제로 전송하는 메서드

Message 객체를 전달하여 해당 이메일을 SMTP 서버를 통해 전송

CODE

 **시연**