

# 자바 메일 보내기(네이버, 구글)



오늘은 우리집 고양이 식빵이 사진으로 시작합니다

~~죽은거아님 보일러지지는거임~~

~~아구보면서 쉴라 그랬는데.. 키자마자 홈런맞길래 하나서 끄고 쓰는 글~~

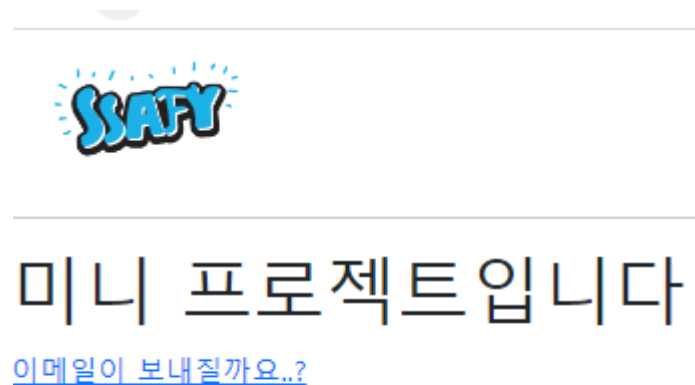
오늘은 메일을 보내보겠습니다! 이메일 인증을 위해서, 혹은 알림 메일을 보내기 위해서 사용하는 기능입니다. 기능을 설명하기 전에 mvn 레포지토리에서 해당 기능을 위해 디펜던시를 우선 추가합니다. (<https://mvnrepository.com/artifact/javax.mail/mail>)

1.4.7	Central	787	Mar 09, 2013
1.4.6	Central	28	Feb 07, 2013
1.4.6-rc1	Central	4	Nov 20, 2012
1.4.5	Central	366	Mar 23, 2012
1.4.5-rc1	Central	5	Feb 08, 2012
1.4.4	Central	324	Jan 27, 2011

그나마? 최근 버전 중 많은 사람이 이용한 1.4.7 버전을 추가했습니다.

검색으로 찾으신다면 **JavaMail API JAR** 와 같은 유사품들이 존재하는데, JAR로 끝나는 이름의 이 API는 오늘 우리가 하는 방식으로 실행이 안됩니다. 이유는 잘 모르겠습니다... 그러니 조심해서 추가해주세요!

오늘의 뷰 화면입니다.



의문문으로 만든 저 -이메일이 보내질까요..?- 버튼이 끝입니다! 그마저도 귀찮아서 a태그로 GET매핑으로 보내줬습니다. 매핑값은 sendEmail 로 해줬습니다. 버튼은 자유롭게 만들면 됩니다! 컨트롤러에 아무튼 도착만 하면 되는겁니다.

그리고 코드를 작성하기 전에

1. 네이버(<https://chasos.tistory.com/37>)
2. 구글(<https://privatenote.tistory.com/172>)

로 각각 SMTP설정을 미리 해주세요.

SMTP가 무엇인지는 ([https://ko.wikipedia.org/wiki/간이\\_우편\\_전송\\_프로토콜](https://ko.wikipedia.org/wiki/간이_우편_전송_프로토콜)) 를 참고부탁드립니다.

간단하게 인터넷에서 이메일을 보내기 위해 사용하는 프로토콜이라고 생각하시면 됩니다.

오늘의 코드입니다.

```
@GetMapping("sendEmail")
public String sendEmail(Model model) {

    // 0. 수신자 설정
    // 랜덤 변수를 만들어서 발생시키고 싶다면 생성하고 반환한다
    String recipient = "haanwhy@gmail.com";
    String code = "abc";
    int randomCode = (int)((Math.random() + 1) * 100000 ); // 랜덤 여섯자리 숫자

    // 1. 발신자의 메일 계정과 비밀번호 설정
    final String user = "onerosen@naver.com";
    // ** 로 되어있지만 발신 계정의 비밀번호를 적어주시면 됩니다
    final String password = "*****";

    // 2. Property에 SMTP 서버 정보 설정
    Properties prop = new Properties();
    prop.put("mail.smtp.host", "smtp.naver.com");
    // 발신자 메일이 naver이기 때문에 smtp.naver.com 작성
    // gmail이라면 smtp.gmail.com
    prop.put("mail.smtp.port", 587); // 발신자 메일이 네이버일때 587 구글 465
    prop.put("mail.smtp.auth", "true");
    // 발신자 메일이 네이버일때는 아래 두 줄을 포함시키면 안된다
    // 발신자 메일이 구글일때는 아래 두 줄을 포함시켜야한다!
    // prop.put("mail.smtp.ssl.enable", "true");
    // prop.put("mail.smtp.ssl.trust", "smtp.gmail.com");

    // 3. SMTP 서버정보와 사용자 정보를 기반으로 Session 클래스의 인스턴스 생성
    // 이 Session은 우리가 사용하던 HttpSession과는 다른 javax.mail 에서 생성하는 session입니다
    // 클래스 이름이 같은 여러 개의 클래스들이 있을 수 있으니
    // 자동 임포트보다는 일일이 하나씩 보면서 javax.mail로 시작하는 클래스를 임포트해주세요
    Session session = Session.getDefaultInstance(prop, new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(user, password);
        }
    });

    // 4. Message 클래스의 객체를 사용하여 수신자와 내용, 제목의 메시지를 작성한다.
    // 5. Transport 클래스를 사용하여 작성한 메시지를 전달한다.

    MimeMessage message = new MimeMessage(session);
    try {
        message.setFrom(new InternetAddress(user));

        // 수신자 메일 주소
        message.addRecipient(Message.RecipientType.TO, new InternetAddress(recipient));

        // Subject(제목)
        message.setSubject("SSAFY 9기 5반입니다..");

        // Text(본문)
```

```

// 아까 만들어준 랜덤 6자리 숫자를 보내봤습니다
message.setText("사실 코드 테스트용 메일이지롱~ : " + randomCode);

Transport.send(message); // send message
model.addAttribute("msg", "이메일 발송 성공!");

} catch (AddressException e) {
    e.printStackTrace();
} catch (MessagingException e) {
    e.printStackTrace();
}

return "/index";

}

```

처음보는 클래스들이 가득하지만 각각이 무슨 역할인지 열심히 외울 필요는 없습니다.(저도 잘모름)

1. 어떤 과정, 흐름을 거쳐서 진행되는가
2. 어떤 설정을 해주어야 하는가
3. 나(작성자)는 어떤 코드, 변수를 수정할 수 있을까

를 중점적으로 생각하시면 됩니다.

서치를 통해 우선적으로 찾은 설정들이 구글과 네이버 관련 설정들이라 다른 메일의 경우 어떻게 설정해야할지는 자세히 모르겠습니다. 이걸 추후 더 찾아서 수정하도록 할게요. 궁금하시다면 여러분도 잠깐 찾아보시기~

## 생각해봅시다!

위의 코드에서는 메일 수신자를 직접 작성한 상태로 진행했습니다. 하지만 우리는 최근 사용자 등록, 회원가입을 할 때 꼭 이메일을 작성하고 있습니다! input의 type을 email로 하면 @를 안넣으면 이메일 형식을 지켜달라고 난리를 치는 모습을 많이 봤습니다. 그렇다면 회원 정보에는 회원의 이메일 개인 정보가 포함되어 있고, 내가 관리자로서 이메일을 보내고 싶은 회원을 지정하면 해당 회원의 이메일을 메일 수신자로 변경하도록 코드를 짤 수 있습니다.

또, 코드에서는 제목을 "-SSAFY 9기 5반입니다.."- , 본문을 "-사실 코드 테스트용 메일이지롱~ : "- 라고 설정했지만 이것도 내가 작성하는 값(input 태그라던지)로 변경할 수 있습니다.

그리고 메일 발신 후에 메인 화면으로 보내기 위해 반환형을 String으로 지정했지만, 내가 6자리 랜덤 변수를 만들어서 반환하고 싶다면 해당 변수에 맞게 반환형을 수정할 수 있습니다. 그렇다면 회원가입이나 본인인증같은 기능에서 메일 보내기를 통해서 인증을 할 수 있습니다!! 와!!!

당장 생각나는 변경점들은 이정도지만 만드려는 기능에 따라서 더 추가될 부분이 많다고 생각합니다. 여러분도 다양하게 생각해보시면 좋을 것 같아요.

아무튼 그렇게 메일을 보내면~  
이렇게 잘 수신되는 모습과



발신되는 모습을 확인할 수 있었습니다.



랜덤 6자리 숫자를 추가하고 발신한 모습입니다. 우리가 흔히 이메일 인증에서 볼 수 있는 모습이네요!



onerosen@naver.com

나에게 ▼

사실 코드 테스트용 메일이지롱~ : 113663

이렇게 오늘은 야매 이메일 보내기 기능을 구현해봤습니다. 여러분은 오늘도 메일 보내기는  
요로코롬 진행되는거구나... 하는 사실을 깨달았습니다! 당장 구현하라고 하면 물론 어려움  
이 있겠지만, 천천히 해보신다면 도움이 될 거라 생각합니다. 어쩌다보니 급커브를 틀어서  
이메일 인증을 구현했는데, 다음에도 예고없이 생각나는대로 진행하겠습니다. ㅎ

오늘도 역시 봐주셔서 감사드리고 언제든지 질문은 환영~~~~~

상세코드는 깃허브에서~~~

