



Common Mail

김민교

송인혁

김미진

INDEX.

1

라이브러리
용도

2

지원
사이트

3

방법
&
예제

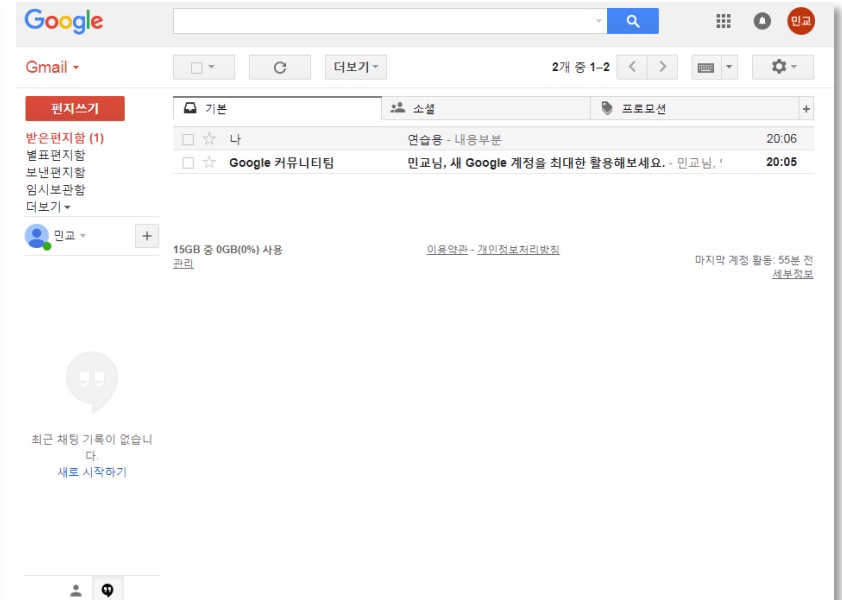
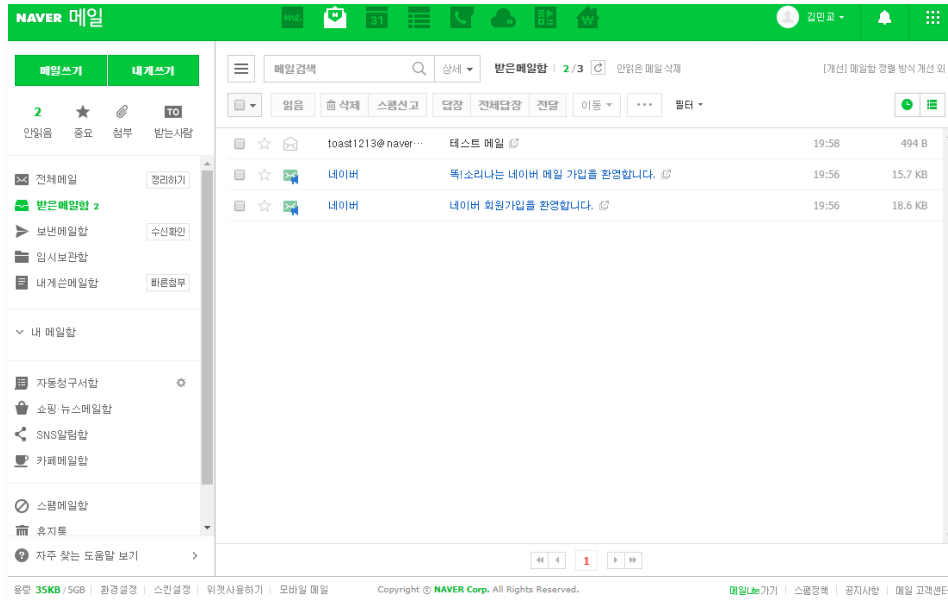
4

팀원
역할

1

라이브러리의 용도

Java를 이용해서 원하는 사용자에게 메일을 보내는 용도로 활용이 가능하고
회원가입 시 인증을 하기 위한 수단으로 사용한다.



2

지원 사이트

MVN Repository 사이트 내 검색기능 활용하여
Common Mail 관련 JAR 파일을 이용한다.

SITE : <https://mvnrepository.com/>

MVNREPOSITORY Search for groups, artifacts, categories Search Categories Popular Contact Us

Home » com.sun.mail » javax.mail » 1.6.0

JavaMail API » 1.6.0
JavaMail API

License CDDL GPL 2.0

Categories Mail Clients

Date (Jul 17, 2017)

Files Download (JAR) (637 KB)

Repositories Central

Used By 365 artifacts

Maven Gradle SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/com.sun.mail/javax.mail -->
<dependency>
  <groupId>com.sun.mail</groupId>
  <artifactId>javax.mail</artifactId>
  <version>1.6.0</version>
</dependency>
```

☒ Include comment with link to declaration

IntelliJ IDEA Level up your code with a Pro Java IDE Get it now

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases
- HTML Parsers
- HTTP Clients

Top Categories

- HTTP Clients
- I/O Utilities
- JDBC Extensions
- JDBC Pools
- JPA Implementation
- JSON Libraries
- JVM Languages
- Logging Framework
- Logging Bridges
- Mail Clients
- Maven Plugins
- Mocking
- Object/Relational M.
- PDF Libraries

Indexed Artifacts (7.96M)

7956k
3978k
0
2004 2017

Indexed

- Code Analyzers
- Collections
- Configuration Librar
- Core Utilities
- Date and Time Utilit
- Dependency Injecti
- Embedded SQL Dat
- HTML Parsers

Popular

- Aspect O
- Actor Fra
- Applicati
- Build Toc
- Bytecode
- Comman
- Cache In
- Cloud Co
- Code An
- Collectio
- Configura
- Core Utili
- Date and
- Depende
- Embedde
- HTML Pai
- HTTP Cli
- I/O Utilit

대부분의 모바일 사이트가
로딩 시간때문에 다수의
방문자를 잃습니다.

사이트를 테스트하세요

Google

10년후내 건강
정답은
유전자 속에!

유전자 분석을 통해,
맞춤 가이드라인을 제안합니다.

framework webapp webserver xml

3

방법 & 예제

Mail의 SMTP 서버를 이용해서
각 메일에 가입되어있는 ID, PW를 이용하여
받는사람 메일로 필요한 내용을 전송한다.

```
package main;

import common.Util;

public class TestMain {

    public static void main(String[] args) {
        Util ut = new Util();

        ut.email("toast1213@naver.com");
        ut.email("toast12T@gmail.com");
    }

}
```

3

방법 & 예제

Mail의 SMTP 서버를 이용해서
각 메일에 가입되어있는 ID, PW를 이용하여
받는사람 메일로 필요한 내용을 전송한다.

```
public class Util {
    public void email(String email){
        try {
            if (email.contains("naver.com")) {
                // 네이버 메일보내기
                // 메일 관련 정보
                String host = "smtp.naver.com";
                final String username = "toast1213"; // 아이디만
                final String password = "#toast1213";
                int port = 465;

                // 메일 내용
                System.out.println("메일 내용");
                String recipient = email; // 받는사람 메일식
                String subject = "테스트 메일";
                String body = "내용";

                Properties props = System.getProperties();

                props.put("mail.smtp.host", host);
                props.put("mail.smtp.port", port);
                props.put("mail.smtp.auth", "true");
                props.put("mail.smtp.ssl.enable", "true");
                props.put("mail.smtp.ssl.trust", host);
```

3

방법 & 예제

Mail의 SMTP 서버를 이용해서
각 메일에 가입되어있는 ID, PW를 이용하여
받는사람 메일로 필요한 내용을 전송한다.

```
System.out.println("세션");
Session session = Session.getDefaultInstance(props,
    new javax.mail.Authenticator() {
        String un = username;
        String pw = password;

        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(un, pw);
        }
    });
//session.setDebug(true); // for debug

System.out.println("메세지");
Message mimeTypeMessage = new MimeMessage(session);
mimeTypeMessage.setFrom(new InternetAddress("toast1213@naver.com")); // 보낸사람표시
mimeTypeMessage.setRecipient(Message.RecipientType.TO, new InternetAddress(recipient));
mimeTypeMessage.setSubject(subject);
mimeTypeMessage.setText(body);

Transport.send(mimeTypeMessage);
```

3

방법 & 예제

Mail의 SMTP 서버를 이용해서
각 메일에 가입되어있는 ID, PW를 이용하여
받는사람 메일로 필요한 내용을 전송한다.

```
else if (email.contains("gmail.com")) {  
    String host = "smtp.gmail.com";  
    String username = "toast12T";  
    String password = "#toast1213";
```

```
// 메일 내용
```

```
String recipient = email;  
String subject = "연습용";  
String body = "내용부분";
```

```
// properties 설정
```

```
Properties props = new Properties();  
props.put("mail.smtps.auth", "true");
```

```
// 메일 세션
```

```
Session session = Session.getDefaultInstance(props);
```

```
MimeMessage msg = new MimeMessage(session);
```

```
// 메일 관련
```

```
msg.setSubject(subject);  
msg.setText(body);  
msg.setFrom(new InternetAddress(username));  
msg.addRecipient(Message.RecipientType.TO, new InternetAddress(  
    recipient));
```

```
// 발송 처리
```

```
Transport transport = session.getTransport("smtps");  
transport.connect(host, username, password);  
transport.sendMessage(msg, msg.getAllRecipients());  
transport.close();
```


3

방법 & 예제

Mail의 SMTP 서버를 이용해서
각 메일에 가입되어있는 ID, PW을 이용하여
받는사람 메일로 필요한 내용을 전송한다.

☆ 테스트 메일 

17-10-16 (월) 19:58

▲ 보낸 사람 ☆ <toast1213@naver.com>

받는 사람 <toast1213@naver.com>

내용

맨위로 ▲

 toast12t@gmail.com

20:06 (1시간 전) ☆



나에게 ▾

내용부분

4

팀원의 역할

프로젝트에서 팀원이 수행하는 역할을 알아보자.

PL (project leader)

- 주인의식(Ownership)이 있어야 함.
- 하고 있는 일에 전적으로 몰입하고 사명감이 있어야 함.
- 성공을 위한 강한 에너지와 열정(Need for Achievement)이 있어야 함.
- 부문별 목표보다는 조직전체를 생각하는 팀 정신 소유자(Team Player)
- 자유로운 의사소통(Communicator)을 할 수 있어야 함.
- 거시적 안목에서 일을 처리할 수 있는 자신감(Confidence)이 있어야 함.
- 팀원들을 격려하고 도와주며 활력을 줄 수 있어야(Empowerment) 함.
- 팀원을 관리하기 보다는 계발, 육성(Develop)할 수 있어야 함.

4

팀원의 역할

프로젝트에서 팀원이 수행하는 역할을 알아보자.

TA (Technical Architect)

프로젝트 전체팀에 대한 하드웨어 및 네트워크 아키텍처를 설계한다.

DA(Data Architect)

프로젝트 전체팀에 대해서 데이터 아키텍처 설계를 담당한다.

AA (Application Architect)

애플리케이션에 대한 표준 가이드 및 아키텍처 구조를 담당한다.

팀 규모에 따라 대규모 팀인 경우 각 개발팀마다 AA를 배치하고,

소규모팀인 경우에는 프로젝트 전체팀에 대해서 애플리케이션

아키텍처 설계를 담당한다.

QnA



감사합니다.