

Spring Framework

- 스프링을 이용한 메일발송

■ CONTENTS

- MailSender와 JavaMailSender 메일 발송
- MimeMessageHelper 이용 파일 첨부

■ MailSender와 JavaMailSender 메일 발송

- 메일 발송 기능을 위한 MailSender 인터페이스 제공
- SimpleMailMessage(메일 제목, 단순 텍스트 내용)를 전달받아 메일을 발송하는 기능을 정의

```
package org.springframework.mail;  
  
public interface MailSender {  
    void send(SimpleMailMessage simpleMessage) throws MailException;  
    void send(SimpleMailMessage[] simpleMessages) throws MailException;  
}
```

- MailSender 인터페이스를 상속받은 JavaMailSender는 Java Mail API의 MimeMessage를 이용해서 메일을 발송하는 추가적인 기능 정의

```
public interface JavaMailSender extends MailSender {  
    MimeMessage createMimeMessage();  
    MimeMessage createMimeMessage(InputStream contentStream) throws MailException;  
    void send(MimeMessage mimeMessage) throws MailException;  
    void send(MimeMessage[] mimeMessages) throws MailException;  
    void send(MimeMessagePreparator mimeMessagePreparator) throws MailException;  
    void send(MimeMessagePreparator[] mimeMessagePreparators) throws MailException;  
}
```

■ MailSender와 JavaMailSender 메일 발송

- MailException은 RuntimeException이므로 예외 처리가 필요한 경우에 처리
- JavaMailSender 인터페이스의 구현체로 JavaMailSenderImpl 클래스를 제공

■ MailSender와 JavaMailSender 메일 발송

- 라이브러리 의존 설정

```
<!-- 메일 설정 시작 -->
<!-- ===== -->
    <!-- javax.mail/mail -->
    <dependency>
        <groupId>javax.mail</groupId>
        <artifactId>mail</artifactId>
        <version>1.4.7</version>
    </dependency>

    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context-support</artifactId>
        <version>${org.springframework-version}</version>
    </dependency>
<!-- ===== -->
```

■ MailSender와 JavaMailSender 메일 발송

- JavaMailSender 빈 설정

- JavaMailSenderImpl 클래스는 Java Mail API를 이용, SMTP 프로토콜 사용

```
<beans:bean id="mailSender"
  class="org.springframework.mail.javamail.JavaMailSenderImpl">
  <beans:property name="host" value="smtp.gmail.com" />
  <beans:property name="port" value="587" /><!-- 465 or 25-->
  <beans:property name="username" value="[gmail 계정]" />
  <beans:property name="password" value="[비밀번호]" />
  <beans:property name="defaultEncoding" value="utf-8" />
  <beans:property name="javaMailProperties">
    <beans:props>
      <beans:prop key="mail.transport.protocol">smtp</beans:prop>
      <beans:prop key="mail.smtp.auth">true</beans:prop>
      <beans:prop key="mail.smtp.starttls.enable">true</beans:prop>
      <beans:prop key="mail.debug">true</beans:prop>
    </beans:props>
  </beans:property>
</beans:bean>
```

- port 프로퍼티 : 포트 번호, 기본 값 25
- defaultEncoding 프로퍼티 : 기본 인코딩 설정
- username 프로퍼티 : 인증을 위한 사용자명
- password 프로퍼티 : 인증을 위한 사용자 비밀번호

■ MailSender와 JavaMailSender 메일 발송

- SimpleMailmessage 이용 메일 발송
 - 단순 텍스트 구성 메일 메시지 생성할 때 이용

메서드	설명
setFrom(String from)	발신자 설정
setReplyTo(String replyTo)	응답 주소 설정
setTo(String to)	수신자 설정
setTo(String[] to)	수신자 목록 설정
setCc(String cc)	참조자 설정
setCc(String[] cc)	참조자 목록 설정
setBcc(String bcc)	숨은 참조자 설정
setBcc(String[] bcc)	숨은 참조자 목록 설정
setSentDate(Date sentDate)	메일 발송일 설정
setSubject(String subject)	메일 제목 설정
setText(String text)	메일 내용 설정

■ MailSender와 JavaMailSender 메일 발송

```
// Member.java
public class Member {
    private String id;
    private String name;
    private String email;

    public Member() { super(); }
    public Member(String id, String name, String email) {
        super();
        this.id = id;
        this.name = name;
        this.email = email;
    }
    public String getId() { return id; }
    public void setId(String id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }
} // Member
```


■ MailSender와 JavaMailSender 메일 발송

```
// SendMailService.java
```

```
public interface SendMailService {  
    public void sendEmail(Member member);  
} // SendMailService
```

■ MailSender와 JavaMailSender 메일 발송

```
// SimpleMailMessageTest.java
public class SimpleMailMessageTest implements SendMailService {
    private MailSender mailSender;
    public void setMailSender(MailSender mailSender) {
        this.mailSender = mailSender;
    }
    @Override
    public void sendEmail(Member member) {
        SimpleMailMessage message = new SimpleMailMessage();
        message.setSubject("[공지] 회원 가입 안내");
        message.setText("회원 가입을 축하합니다.");
        message.setFrom("kyung_ho@naver.com");
        message.setTo(member.getEmail());
        try {
            mailSender.send(message);
        } catch (MailException e) {
            e.printStackTrace();
        }
    } // sendEmail
} // SimpleMailMessageTest
```

■ MailSender와 JavaMailSender 메일 발송

- **SimpleMailmessage 재사용**

- SimpleMailMessage 클래스는 다른 SimpleMailMessage 클래스로부터 내용을 복사해서 사용하는 기능을 제공
- 생성자를 통해 객체를 전달하면 설정 정보를 모두 복사
- simpleMailMessage를 스프링 설정파일에서 설정하고 필요로 하는 빈에서 재사용 가능

■ MailSender와 JavaMailSender 메일 발송

```
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:p="http://www.springframework.org/schema/p" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemalocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```
    <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl"
        p:host="smtp.gmail.com"
        p:port="25"
        p:defaultencoding="UTF-8"> </bean>
```

```
    <bean id="templateMailMessage" class="org.springframework.mail.SimpleMailMessage"
        p:from="jin.ryoo@gmail.com"
        p:subject="회원 가입을 환영합니다.">
</bean>
```

```
    <bean id="simpleMailMessageTest2" class="SimpleMailMessageTest2"
        p:mailsender-ref="mailSender"
        p:templatemailmessage-ref="templateMailMessage">
</bean>
</beans>
```

■ MailSender와 JavaMailSender 메일 발송

```
// SimpleMailMessageTest2.java
```

```
import org.springframework.mail.MailException;  
import org.springframework.mail.MailSender;  
import org.springframework.mail.SimpleMailMessage;
```

```
public class SimpleMailMessageTest2 implements SendMailService {
```

```
    private MailSender mailSender;
```

```
    private SimpleMailMessage templateMailMessage;
```

```
    public void setMailSender(MailSender mailSender) {
```

```
        this.mailSender = mailSender;
```

```
    }
```

```
    public void setTemplateMailMessage(SimpleMailMessage templateMailMessage) {
```

```
        this.templateMailMessage = templateMailMessage;
```

```
    }
```

■ MailSender와 JavaMailSender 메일 발송

```
@Override
public void sendEmail(Member member) {
    SimpleMailMessage message = new SimpleMailMessage(templateMailMessage);
    message.setText("회원 가입을 축하합니다.");
    message.setTo(member.getEmail());
    try {
        mailSender.send(message);
    } catch (MailException e) {
        e.printStackTrace();
    }
} // sendEmail

} // SimpleMailMessageTest2
```

■ MailSender와 JavaMailSender 메일 발송

- **Java Mail API - MimeMessage 이용 메시지 생성**
 - 파일 첨부 또는 HTML로 구성되어 있는 경우 MimeMessage를 이용
 - JavaMailSender 인터페이스는 MimeMessage 객체를 생성해주는 createMimeMessage() 메서드를 제공
 - 이 메서드가 리턴한 MimeMessage 객체를 이용해서 메시지를 구성한 뒤 메일 발송

■ MailSender와 JavaMailSender 메일 발송

```
// MimeMessageTest.java
```

```
import javax.mail.MessagingException;  
import javax.mail.Message.RecipientType;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;
```

```
import org.springframework.mail.MailException;  
import org.springframework.mail.javamail.JavaMailSender;
```

```
public class MimeMessageTest implements SendMailService {  
    private JavaMailSender mailSender;  
  
    public void setMailSender(JavaMailSender mailSender) {  
        this.mailSender = mailSender;  
    }  
}
```


■ MailSender와 JavaMailSender 메일 발송

@Override

```
public void sendEmail(Member member) {  
    MimeMessage message = mailSender.createMimeMessage();  
    try {  
        message.setSubject("[공지] 회원 가입 안내", "UTF-8");  
        String htmlContent = "<strong>안녕하세요</strong>, 반갑습니다.";  
        message.setText(htmlContent, "UTF-8", "html");  
        message.setFrom(new InternetAddress( " jin90188548@gmail.com"));  
        message.addRecipient(RecipientType.TO, new InternetAddress(member.getEmail()));  
        mailSender.send(message);  
    } catch (MessagingException e) {  
        e.printStackTrace();  
        return;  
    } catch (MailException e) {  
        e.printStackTrace();  
        return;  
    } // try - catch  
}  
} // MimeMessageTest
```

■ MimeMessageHelper 이용 파일 첨부

- MimeMessage를 이용하여 파일 첨부할 수 있지만 복잡하므로, 스프링이 제공하는 MimeMessageHelper 클래스를 사용

■ MimeMessageHelper 이용 파일 첨부

- **MimeMessageHelper 이용 첨부 파일 추가**

- 생성자의 첫 번째 파라미터는 메일 메시지를 구성할 때 사용 될 MimeMessage 객체를 전달
- 두 번째 파라미터는 MultiPart 여부를 설정 (true 이면 첨부 파일 추가)
- 파일 정보를 제공할 javax.activation.DataSource를 생성
- FileDataSource 클래스를 사용, MimeMessageHelper.addAttachment() 메서드로 파일을 메일에 첨부
- MimeUtilty 클래스는 Java Mail API에서 제공하는 클래스, ASCII 이외의 문자로 구성된 문장을 알맞게 인코딩

■ MimeMessageHelper 이용 파일 첨부

```
// MimeMessageHelperFileTest.java
```

```
import javax.activation.DataSource;  
import javax.activation.FileDataSource;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;  
import javax.mail.internet.MimeUtility;
```

```
import org.springframework.mail.MailException;  
import org.springframework.mail.javamail.JavaMailSender;  
import org.springframework.mail.javamail.MimeMessageHelper;
```

```
public class MimeMessageHelperFileTest implements SendMailService {
```

```
    private JavaMailSender mailSender;
```

```
    public void setMailSender(JavaMailSender mailSender) {  
        this.mailSender = mailSender;  
    }
```

■ MimeMessageHelper 이용 파일 첨부

```
@Override
public void sendEmail(Member member) {
    MimeMessage message = mailSender.createMimeMessage();
    try {
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true,
"UTF-8");
        messageHelper.setSubject("[공지] 회원 가입 안내");
        String htmlContent = "<strong>안녕하세요</strong>, 반갑습니다.";
        messageHelper.setText(htmlContent, true);
        messageHelper.setFrom("ji90188548@gmail.com", "JIN");
        messageHelper.setTo(new InternetAddress(member.getEmail(), member.getName(),
"UTF-8"));
        DataSource dataSource = new FileDataSource("c:\\ww책목록.xlsx");
        messageHelper.addAttachment(MimeUtility.encodeText("책목록.xlsx", "UTF-8", "B"),
dataSource);
        mailSender.send(message);
    } catch (MailException e) {
        e.printStackTrace();
        return;
    } catch (Throwable e) {
```

■ MimeMessageHelper 이용 파일 첨부

```
e.printStackTrace();  
    return;  
}  
}
```

```
} // MimeMessageHelperFileTest
```

■ **MimeMessageHelper** 이용 파일 첨부

- **MimeMessageHelper** 이용 인라인 자원 추가
 - `MimeMessageHelper.addInline()` 메서드를 사용, 메일 내용에 이미지나 파일 등을 삽입
 - `setText()` 메서드를 호출한 이후, `addInline()` 메서드를 호출

■ MimeMessageHelper 이용 파일 첨부

```
// MimeMessageInlineTest.java
```

```
import javax.activation.DataSource;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;
```

```
import org.springframework.mail.MailException;  
import org.springframework.mail.javamail.JavaMailSender;  
import org.springframework.mail.javamail.MimeMessageHelper;
```

```
public class MimeMessageInlineTest implements SendMailService {  
    private JavaMailSender mailSender;  
  
    public void setMailSender(JavaMailSender mailSender) {  
        this.mailSender = mailSender;  
    }  
}
```

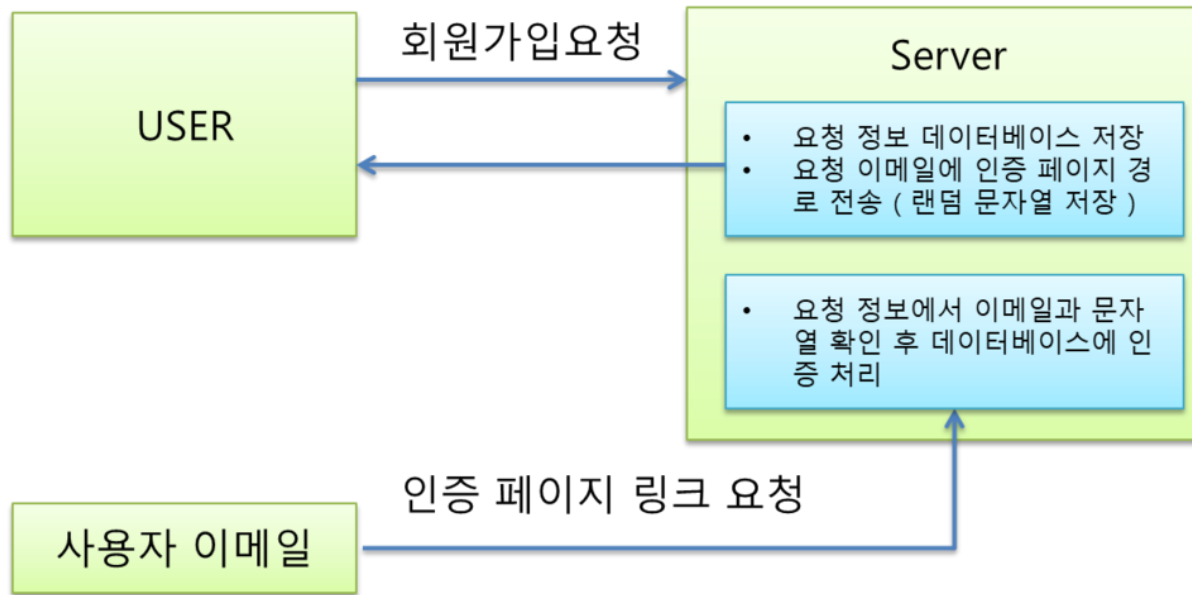

■ MimeMessageHelper 이용 파일 첨부

```
@Override
public void sendEmail(Member member) {
    MimeMessage message = mailSender.createMimeMessage();
    try {
        MimeMessageHelper messageHelper = new MimeMessageHelper(message, true,
"UTF-8");
        messageHelper.setSubject("[공지] 회원 가입 안내");
        String htmlContent = "<strong>안녕하세요</strong>, 반갑습니다." + "<img
src='http://i1.daumcdn.net/cfs.tistory/static/images/xBoxReplace_250.png' width='250'
height='250'>";
        messageHelper.setText(htmlContent, true);
        messageHelper.setFrom("jin90188548@gmail.com", "JIN");
        messageHelper.setTo(new InternetAddress(member.getEmail(), member.getName(),
"UTF-8"));
        messageHelper.addInline("abc", new FileDataSource("abc.jpg"));
        mailSender.send(message);
    } catch (MailException e) {
        e.printStackTrace();
        return;
    } catch (Throwable e) {
```

■ MimeMessageHelper 이용 파일 첨부

```
        e.printStackTrace();  
        return;  
    }  
}  
  
} // MimeMessageInlineTest
```

■ 회원 가입 시에 입력한 이메일의 인증 , 비밀번호 변경



1. 회원 가입 폼
2. 가입처리 (회원 테이블에 인증 확인 컬럼이 필요 함)
 - DB 저장
 - 저장 시 임의의 문자열 저장
 - 가입 이메일로 인증 페이지 경로 전송
3. 인증 처리 페이지는 이메일과 문자열 비교 후 인증 처리 (컬럼 값 변경)
4. 이메일 에서 링크 클릭 → 인증페이지에서 인증 처리