

4. Controller 메서드 추가

MimeMessage 객체를 직접 생성하여 메일을 발송하는 방법과 MimeMessagePreparator 를 사용해서 메일을 전송하는 방법이 있습니다. 두가지 코드 모두 알아보겠습니다.

보내는 이 이메일, 받는 이 이메일, 메일 제목, 메일 내용 4가지의 데이터를 입력해주어야하는데 해당 데이터들을 VO객체에 담아서도 가능하고 Controller 메서드 내에 직접 삽입 하는 것도 가능합니다. 이번 포스팅에선 직접 삽입 방식으로 진행하겠습니다.

메일전송을 위해 필요한 정보를 세팅하였던 "mailSender" Bean을 인젝션(의존성 주입) 하여 사용하겠습니다.

```
@Autowired
private JavaMailSender mailSender;
```

4.1 MimeMessage 객체를 직접 생성하여 메일을 발송하는 방법

이방식을 사용하기 위해 import 되는 클래스들은 아래와 같습니다.

```
import javax.mail.internet.MimeMessage;
import org.springframework.mail.javamail.JavaMailSenderImpl;
import org.springframework.mail.javamail.MimeMessageHelper;
```

*MimeMessage 대신 SimpleMailMessage를 사용할 수도 있습니다. 둘의 차이점은 MimeMessage의 경우 멀티파트 데이터를 처리 할 수 있고 SimpleMailMessage는 단순한 텍스트 데이터만 전송이 가능합니다.

Controller에 메서드는 아래와 같이 작성하면됩니다.

```
@RequestMapping(value = "/sendMail", method = RequestMethod.GET)
public void sendMailTest() throws Exception{

    String subject = "test 메일";
    String content = "메일 테스트 내용";
    String from = "보내는이 아이디@도메인주소";
    String to = "받는이 아이디@도메인주소";

    try {
        MimeMessage mail = mailSender.createMimeMessage();
        MimeMessageHelper mailHelper = new MimeMessageHelper(mail,true,"UTF-8");
        // true는 멀티파트 메세지를 사용하겠다는 의미

        /*
        * 단순한 텍스트 메세지만 사용시엔 아래의 코드도 사용 가능
        * MimeMessageHelper mailHelper = new MimeMessageHelper(mail,"UTF-8");
        */

        mailHelper.setFrom(from);
        // 빈에 아이디 설정한 것은 단순히 smtp 인증을 받기 위해 사용 따라서 보내는이(setFrom())반드시 필요
        // 보내는이와 메일주소를 수신하는이가 볼때 모두 표기 되게 원하신다면 아래의 코드를 사용하시면 됩니다.
        //mailHelper.setFrom("보내는이 이름 <보내는이 아이디@도메인주소>");
        mailHelper.setTo(to);
        mailHelper.setSubject(subject);
    }
}
```

```

        mailHelper.setText(content, true);
        // true는 html을 사용하겠다는 의미입니다.

        /*
        * 단순한 텍스트만 사용하신다면 다음의 코드를 사용하셔도 됩니다. mailHelper.setText(content);
        */

        mailSender.send(mail);

    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

Colored by Color Scripter

- MimeMessageHelper mailHelper = new MimeMessageHelper(mail,true,"UTF-8") 에서 true는 멀티파트 메세지를 사용하겠다는 의미입니다. 단순한 텍스트 메세지만 사용하신다면 아래의 코드를 사용하시면됩니다.

- mailHelper.setText(content, true) 에서 true는 html을 사용한다는 의미입니다. html을 사용하게되면 이미지를 첨부 할 수 있는 태그를 사용 할 수 있습니다.

- 단순한 텍스트만 사용할 시엔 mailHelper.setText(content)로 사용하시면 됩니다.

- 보내는이(from)은 반드시 있어야합니다. mailSender 빈에서 아이디를 기입하였지만 이는 SMTP 사용 권한을 얻어 오는 역할을 수행합니다.

* 메일 주소에 보내는이와 메일주소를 모두 표기를 원한다면 아래와 같이 작성하시면 됩니다.

```
mailHelper.setFrom("보내는이 이름 <보내는이 아이디@도메인주소>");
```

4.2 MimeMessagePreparator를 사용해서 메일을 전송하는 방법

해당 방식은 MimeMessage를 직접 객체화 하지 않고 MimeMessagePreparator 클래스를 사용하는 방식입니다. 이방식을 사용하기위해 import 되는 클래스는 다음과 같습니다.

- 4.1과 다른점은 import org.springframework.mail.javamail.MimeMessagePreparator 가 추가적으로 추가 된다는 점입니다.

```

import javax.mail.internet.MimeMessage;
import org.springframework.mail.javamail.JavaMailSenderImpl;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.mail.javamail.MimeMessagePreparator;

```

- Controller에 메서드는 다음과 같이 작성 하시면 됩니다.

- new MimeMessageHelper(mail,true,"UTF-8") / mailHelper.setText(content, true) 는 4.1 설명과 동일합니다.

```
@RequestMapping(value = "/sendMail", method = RequestMethod.GET)
public void sendMailTest2() throws Exception{

    String subject = "test 메일";
    String content = "메일 테스트 내용.";
    String from = "보내는이 아이디@도메인주소";
    String to = "받는이 아이디@도메인주소";

    final MimeMessagePreparator preparator = new MimeMessagePreparator() {

        public void prepare(MimeMessage mimeMessage) throws Exception{
            final MimeMessageHelper mailHelper = new MimeMessageHelper(mimeMessage, true, "UTF-8");

            mailHelper.setFrom(from);
            mailHelper.setTo(to);
            mailHelper.setSubject(subject);
            mailHelper.setText(content, true);

        }

    };

    try {
        mailSender.send(preparator);
    } catch(Exception e) {
        e.printStackTrace();
    }

}
```

Colored by Color Scripter CS

5. 텍스트 / 이미지 / 업로드 전송

설명하기 앞서 이미지 / 업로드 전송을 위해선 **멀티파트 메세지**를 허용해야하며 **html 허용**을 설정해주어야합니다. 해당방법은 4.1설명을 참고 부탁드립니다. 해당 설명은 4.1 의 Controller 메서드 코드를 기반으로 진행하겠습니다.

```
// 멀티파트 메세지 허용
MimeMessageHelper mailHelper = new MimeMessageHelper(mail,true,"UTF-8");

// html
mailHelper.setText(content, true);
```

Colored by Color Scripter

5.1 단순 텍스트

4.1 코드 예시가 메세지 내용을 단순한 텍스트로 전송하는 것입니다.

5.2 이미지

이미지를 첨부하는 방법은 html 허용을 설정했기때문에 이미지 태그()를 사용합니다. 예를 들면 다음과 같습니다.

```
String content = "메일 테스트 내용" + "<img src=\"이미지 경로\">";

@RequestMapping(value = "/sendMail", method = RequestMethod.GET)
public void sendMailTest() throws Exception{

    String subject = "test 메일";
```

```
String content = "메일 테스트 내용" + "<img src=\"https://t1.daumcdn.net/cfile/tistory/214DCD42594CC40625\">";
String from = "보내는이 아이디@도메인주소";
String to = "받는이 아이디@도메인주소";

try {
    MimeMessage mail = mailSender.createMimeMessage();
    MimeMessageHelper mailHelper = new MimeMessageHelper(mail,true,"UTF-8");

    mailHelper.setFrom(from);
    mailHelper.setTo(to);
    mailHelper.setSubject(subject);
    mailHelper.setText(content, true);

    mailSender.send(mail);

} catch (Exception e) {
    e.printStackTrace();
}
```

Colored by Color Scripter

5.3 업로드

업로드 된 파일을 전송하기 위해선 추가적인 클래스 import가 필요합니다.

```
import org.springframework.core.io.FileSystemResource;
import java.io.File;
```

Colored by Color Scripter

추가해주어야할 코드는 다음과 같습니다.

```
FileSystemResource file = new FileSystemResource(new File("경로\업로드할파일.형식"));
helper.addAttachment("업로드파일.형식", file);
```

Colored by Color Scripter

```
@RequestMapping(value = "/sendMail", method = RequestMethod.GET)
public void sendMailTest() throws Exception{

    String subject = "test 메일";
    String content = "메일 테스트 내용" + "<img src=\"https://t1.daumcdn.net/cfile/tistory/214DCD42594CC40625\">";
    String from = "보내는이 아이디@도메인주소";
    String to = "받는이 아이디@도메인주소";

    try {
        MimeMessage mail = mailSender.createMimeMessage();
        MimeMessageHelper mailHelper = new MimeMessageHelper(mail,true,"UTF-8");

        mailHelper.setFrom(from);
        mailHelper.setTo(to);
        mailHelper.setSubject(subject);
        mailHelper.setText(content, true);

        FileSystemResource file = new FileSystemResource(new File("D:\\test.txt"));
        mailHelper.addAttachment("업로드파일.형식", file);

        mailSender.send(mail);

    } catch (Exception e) {
        e.printStackTrace();
    }

}
```

Colored by Color Scripter

6. 메일 전송 테스트

테스트 클래스를 통해 Controller 메서드에 추가했던 코드들을 테스트 해보겠습니다. 테스트 클래스를 생성 후 아래의 코드를 작성하였습니다. mailSendTest(), mailSendTest2()가 있는데 둘 중 하나를 사용하시면 됩니다.

```
import java.io.File;

import javax.mail.internet.MimeMessage;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.io.FileSystemResource;
import org.springframework.mail.javamail.JavaMailSenderImpl;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.mail.javamail.MimeMessagePreparator;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
public class MailTestController {

    @Autowired
    JavaMailSenderImpl mailSender;

    // MimeMessage 객체를 직접 생성하여 메일을 발송하는 방법
    @Test
    public void mailSendTest() throws Exception{

        String subject = "test 메일";
        String content = "메일 테스트 내용" + "<img src=\"https://t1.daumcdn.net/cfile/tistory/214DCD42594CC40625\">";
        String from = "보내는이 아이디@도메인주소";
        String to = "받는이 아이디@도메인주소";

        try {
            MimeMessage mail = mailSender.createMimeMessage();
            MimeMessageHelper mailHelper = new MimeMessageHelper(mail, "UTF-8");

            mailHelper.setFrom(from);
            mailHelper.setTo(to);
            mailHelper.setSubject(subject);
            mailHelper.setText(content, true);

            FileSystemResource file = new FileSystemResource(new File("D:\\test.txt"));
            mailHelper.addAttachment("test.txt", file);

            mailSender.send(mail);

        } catch (Exception e) {
            e.printStackTrace();
        }

    }

    // MimeMessagePreparator를 사용해서 메일을 전송하는 방법
    @Test
    public void mailSendTest2() throws Exception{

        String subject = "test 메일";
        String content = "메일 테스트 내용" + "<img src=\"https://t1.daumcdn.net/cfile/tistory/214DCD42594CC40625\">";
        String from = "보내는이 아이디@도메인주소";
        String to = "받는이 아이디@도메인주소";

        try {
            final MimeMessagePreparator preparator = new MimeMessagePreparator() {

                public void prepare(MimeMessage mimeMessage) throws Exception{
                    final MimeMessageHelper mailHelper = new MimeMessageHelper(mimeMessage, true, "UTF-8");

                    mailHelper.setFrom(from);
                    mailHelper.setTo(to);
                    mailHelper.setSubject(subject);
                    mailHelper.setText(content, true);

                    FileSystemResource file = new FileSystemResource(new File("D:\\test.txt"));
                    mailHelper.addAttachment("test.txt", file);

                }

            };

            mailSender.send(preparator);

        } catch (Exception e) {
            e.printStackTrace();
        }

    }

}
```


}

}

Colored by Color Scriptor

테스트 결과입니다.

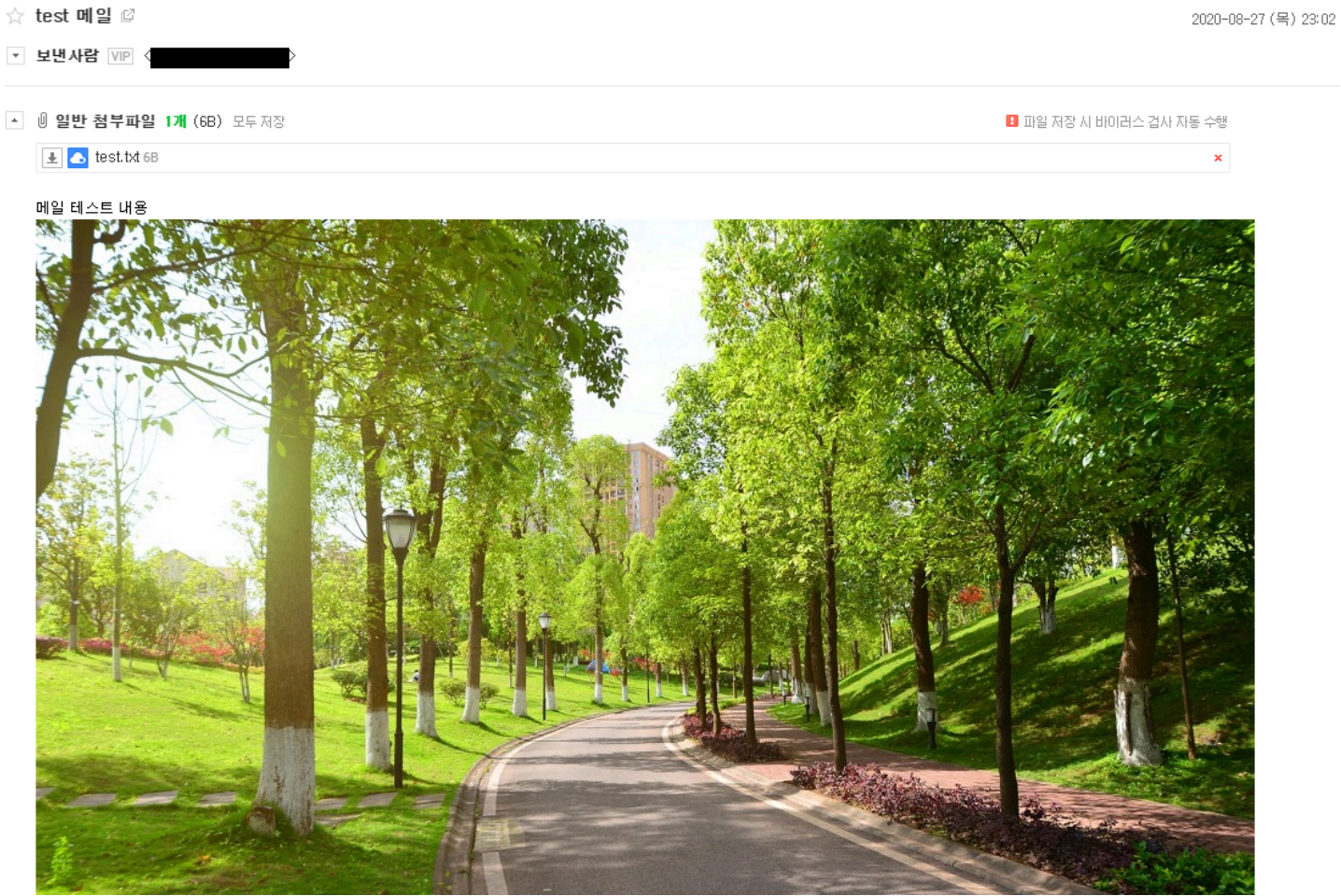


그림 6-1