

# Spring ve E-Posta Gönderme



# Spring ve E-Posta Desteği

- Spring, e-posta gönderme işlemini sistemin **Java Mail altyapısından soyutlanmasını** sağlayan yardımcı sınıflar sunar
- Alt düzey sistem hatalarını daha anlamlı **üst düzey e-posta hatalarına** dönüştürür
- Alt düzeyde **resource'ların ele alınmasını** da kolaylaştırır

# Spring ve E-Posta Desteği

- Spring e-posta desteğinin temel yapı taşı **MailSender** arayüzüdür
- **JavaMailSender** arayüzü de MIME mesaj gönderimi gibi ilave kabiliyetler sunar
- Her ikisinde de alt tarafta **Java Mail API**'si kullanılmaktadır

# Mail Mesajları

- **MailMessage** bütün e-posta mesajları için ortak arayüzdür
- Farklı gerçekleştirmeleri mevcuttur
- **SimpleMailMessage** sınıfı ile from, to, cc, subject, text body gibi bölümleri içeren basit e-posta mesajları oluşturulabilir
- **MimeMailMessage** sınıfı ile de MIME mesajları oluşturulabilir
- **MimeMessagePreparator** sınıfı ile mime mesajlarının hazırlanması gerçekleştirilir

# MailSender Konfigürasyonu

```
<bean id="mailSender"
class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.mycompany.com"/>
    <property name="port" value="25"/>
    <property name="username" value="user1"/>
    <property name="password" value="secret"/>
    <property name="javaMailProperties">
        <props>
            <prop key="mail.smtp.auth">true</prop>
        </props>
    </property>
</bean>
```

# Text İçerikli Mesaj Gönderme

```
SimpleMailMessage msg = new SimpleMailMessage();
msg.setFrom("me@example.com");
msg.setTo("you@example.com");
msg.setText("Hello world!");

mailSender.send(msg);
```

# Mime Mesaj Gönderme: 1.yol

```
MimeMessagePreparator preparator = new MimeMessagePreparator() {  
  
    public void prepare(MimeMessage mimeMessage) throws Exception {  
        mimeMessage.setRecipient(Message.RecipientType.TO,  
            new InternetAddress("you@example.com"));  
        mimeMessage.setFrom(new InternetAddress("mail@mycompany.com"));  
        mimeMessage.setText("Hello world!");  
    }  
};  
  
javaMailSender.send(preparator);
```

# Mime Mesaj Gönderme: 2.yol

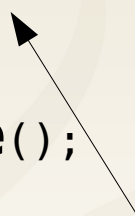
```
MimeMessage message = javaMailSender.createMimeMessage();  
  
MimeMessageHelper helper = new MimeMessageHelper(message);  
  
helper.setTo("test@host.com");  
helper.setText("<html><body>Thank you!</body></html>", true);  
  
javaMailSender.send(message);
```

↓  
İçeriğin html olduğu  
belirtilebilir



# Attachment Gönderme

Multi part mesaj olduğu  
boolean flag ile belirtilir



```
MimeMessage message = javaMailSender.createMimeMessage();

MimeMessageHelper helper = new MimeMessageHelper(message, true);
helper.setTo("test@host.com");

helper.setText("Check out this image!");

FileSystemResource file = new FileSystemResource(
    new File("c:/Sample.jpg"));
helper.addAttachment("CoolImage.jpg", file);

javaMailSender.send(message);
```

# Inline Resource Gönderme

```
MimeMessage message = javaMailSender.createMimeMessage();

MimeMessageHelper helper = new MimeMessageHelper(message, true);
helper.setTo("test@host.com");

helper.setText("<html><body><img src=cid:id1234></body></html>", true);

FileSystemResource res = new FileSystemResource(
    new File("c:/Sample.jpg"));
helper.addInline("id1234", res);

javaMailSender.send(message);
```



Önce text mesajın ardından da inline resource'ların eklenmesi gerekir. Önce inline resource'lar eklenirse mesaj düzgün biçimde gönderilmeyecektir

# Mesaj Şablonları

- Kurumsal uygulamaların çoğunda mesajların içeriği, yapısı ve nasıl görüldüğü **şablonlar** vasıtası ile uygulama kodu dışında yönetilir
- Spring, FreeMarker, Velocity gibi “**template kütüphaneleri**”ni destekler
- Önce **şablon üzerinden mesaj içeriği** elde edilir ve ardından mesaj gönderilir

# Mesaj Şablonları Örnek Kullanım

```
<beans...>
```

```
<bean id="templateEngine"  
class="org.springframework.ui.velocity.VelocityEngineFactoryBean">  
  <property name="velocityProperties">  
    <value>  
      resource.loader=class  
      class.resource.loader.class=  
      org.apache.velocity.runtime.resource.loader.ClasspathResourceLoader  
    </value>  
  </property>  
</bean>  
</beans>
```

Spring Container içerisinde **template engine bean** olarak tanımlanmış olmalıdır

# Mesaj Şablonları Örnek Kullanım

```
<html>
  <body>
    <h3>Hi ${user.userName}, welcome to our system!</h3>
    <div>
      Your email address is <a href="mailto:${user.emailAddress}">${
user.emailAddress}</a>.
    </div>
  </body>
</html>
```

Şablon mesaj içeriği hazırlanır, classpath'de, file system'de veya başka bir yerde saklanır

# Mesaj Şablonları Örnek Kullanım

```
MimeMessagePreparator preparator = new MimeMessagePreparator() {  
    public void prepare(MimeMessage mimeMessage) throws Exception {  
        MimeMessageHelper message = new MimeMessageHelper(mimeMessage);  
        message.setTo(user.getEmailAddress());  
        message.setFrom("me@example.com");  
        Map modelMap = new HashMap();  
        modelMap.put("user", user);  
  
        String text = VelocityEngineUtils.mergeTemplateIntoString(  
            templateEngine, "com/example/template.vm", modelMap);  
  
        message.setText(text, true);  
    }  
};
```

```
javaMailSender.send(preparator);
```

templateEngine bean'i, şablon dosyası ve şablon input'unu içeren model map kullanılarak mesaj içeriği elde edilir

# Dumbster ile E-Mail Gönderme Testi

```
private SimpleSmtpServer smtpServer;
```

```
@Before
```

```
public void setUp() {  
    smtpServer = SimpleSmtpServer.start();  
}
```

Dumbster isimli bir kütüphane ile Standalone ortamda test amaçlı bir SMTP sunucu çalıştırılabilir

```
@Test
```

```
public void testEmail() {  
    // mail gönderme işlemleri...  
    int size = smtpServer.getReceivedEmailSize();  
    if (size > 0) {  
        Iterator receivedEmails = smtpServer.getReceivedEmail();  
        while (receivedEmails.hasNext()) {  
            Object message = receivedEmails.next();  
            // mesaj üzerinde işlemler...  
        }  
    }  
}
```

```
@After
```

```
public void tearDown() {  
    smtpServer.stop();  
}
```

Test sonunda SMTP sunucunun kapatılması gerekir

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

