Here is a step by step example for sending email via gmail smtp server, using Spring’s [**org.springframework.mail.javamail.JavaMailSender**](http://static.springsource.org/spring/docs/3.0.x/javadoc-api/org/springframework/mail/javamail/JavaMailSender.html) interface.

**Step 1) Create java project and add runtime dependencies**

Follow [**maven project creation example**](http://howtodoinjava.com/2013/01/02/create-a-simple-java-project-using-maven/) for creating a project. Now update following dependencies.

|  |
| --- |
| <repositories>      <repository>          <id>JBoss repository</id>          <url><http://repository.jboss.org/nexus/content/groups/public/></url>      </repository>  </repositories>  <properties>      <org.springframework.version>3.0.5.RELEASE</org.springframework.version>  </properties>      .............................    <dependency>    <groupId>org.springframework</groupId>    <artifactId>spring-core</artifactId>    <version>${org.springframework.version}</version>  </dependency>  <dependency>    <groupId>org.springframework</groupId>    <artifactId>spring-beans</artifactId>    <version>${org.springframework.version}</version>  </dependency>  <dependency>    <groupId>org.springframework</groupId>    <artifactId>spring-context</artifactId>    <version>${org.springframework.version}</version>  </dependency>  <dependency>      <groupId>javax.mail</groupId>      <artifactId>mail</artifactId>      <version>1.4</version>  </dependency> |

**Step 2) Create Spring application context**

In context file, I will be creating a generic mail sender service which is capable of sending http messages using gmail’s smtp server.

Also, I am making a pre-configured message which can be instantiated on the fly and use for sending the message.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="<http://www.springframework.org/schema/beans>"      xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"      xmlns:context="<http://www.springframework.org/schema/context>"      xsi:schemaLocation="<http://www.springframework.org/schema/beans>  <http://www.springframework.org/schema/beans/spring-beans-3.0.xsd>  <http://www.springframework.org/schema/context>  <http://www.springframework.org/schema/context/spring-context-3.0.xsd>">        <context:component-scan base-package="com.howtodoinjava" />        <!-- SET default mail properties -->      <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">          <property name="host" value="smtp.gmail.com"/>          <property name="port" value="25"/>          <property name="username" value="howtodoinjava@gmail.com"/>          <property name="password" value="password"/>          <property name="javaMailProperties">              <props>                  <prop key="mail.transport.protocol">smtp</prop>                  <prop key="mail.smtp.auth">true</prop>                  <prop key="mail.smtp.starttls.enable">true</prop>                  <prop key="mail.debug">true</prop>              </props>          </property>      </bean>        <!-- You can have some pre-configured messagess also which are ready to send -->      <bean id="preConfiguredMessage" class="org.springframework.mail.SimpleMailMessage">          <property name="to" value="lokeshgupta1981@gmail.com"></property>          <property name="from" value="howtodoinjava@gmail.com"></property>          <property name="subject" value="FATAL - Application crash. Save your job !!"/>      </bean>  </beans> |

**Step 3) Create Mail sender service**

This class uses the beans configured in application-context.xml file and use them to send messages.

|  |
| --- |
| package com.howtodoinjava.mail;    import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.mail.MailSender;  import org.springframework.mail.SimpleMailMessage;  import org.springframework.stereotype.Service;    @Service("mailService")  public class ApplicationMailer  {      @Autowired      private MailSender mailSender;        @Autowired      private SimpleMailMessage preConfiguredMessage;        /\*\*       \* This method will send compose and send the message       \* \*/      public void sendMail(String to, String subject, String body)      {          SimpleMailMessage message = new SimpleMailMessage();          message.setTo(to);          message.setSubject(subject);          message.setText(body);          mailSender.send(message);      }        /\*\*       \* This method will send a pre-configured message       \* \*/      public void sendPreConfiguredMail(String message)      {          SimpleMailMessage mailMessage = new SimpleMailMessage(preConfiguredMessage);          mailMessage.setText(message);          mailSender.send(mailMessage);      }  } |

**Step 4) Test the Mail Sender**

Time to test the code. I am sending two messages from test code. One is instantiated and composed in test class itself, and another one is pre-configured message from application-context.xml file.

|  |
| --- |
| package com.howtodoinjava.mail;    import org.springframework.context.ApplicationContext;  import org.springframework.context.support.FileSystemXmlApplicationContext;    public class MailerTest {        public static void main(String[] args)      {          //Create the application context          ApplicationContext context = new FileSystemXmlApplicationContext("application-context.xml");            //Get the mailer instance          ApplicationMailer mailer = (ApplicationMailer) context.getBean("mailService");            //Send a composed mail          mailer.sendMail("lokeshgupta1981@gmail.com", "Test Subject", "Testing body");            //Send a pre-configured mail          mailer.sendPreConfiguredMail("Exception occurred everywhere.. where are you ????");      }  }    Output:    DEBUG: JavaMail version 1.4.7  DEBUG: URL jar:file:/D:/Latest%20Setup/eclipse\_juno\_workspace/SpringAutoWireExample/lib/smtp.jar!/META-INF/javamail.providers  DEBUG: successfully loaded resource: jar:file:/D:/Latest%20Setup/eclipse\_juno\_workspace/SpringAutoWireExample/lib/smtp.jar!/META-INF/javamail.providers  DEBUG: Tables of loaded providers  DEBUG: Providers Listed By Class Name: {com.sun.mail.smtp.SMTPSSLTransport=javax.mail.Provider[TRANSPORT,smtps,com.sun.mail.smtp.SMTPSSLTransport,Oracle], com.sun.mail.smtp.SMTPTransport=javax.mail.Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Oracle]}  DEBUG: Providers Listed By Protocol: {smtps=javax.mail.Provider[TRANSPORT,smtps,com.sun.mail.smtp.SMTPSSLTransport,Oracle], smtp=javax.mail.Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Oracle]}  DEBUG: URL jar:file:/D:/Latest%20Setup/eclipse\_juno\_workspace/SpringAutoWireExample/lib/smtp.jar!/META-INF/javamail.address.map  DEBUG: successfully loaded resource: jar:file:/D:/Latest%20Setup/eclipse\_juno\_workspace/SpringAutoWireExample/lib/smtp.jar!/META-INF/javamail.address.map  DEBUG: getProvider() returning javax.mail.Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Oracle]  DEBUG SMTP: useEhlo true, useAuth true  DEBUG SMTP: trying to connect to host "smtp.gmail.com", port 25, isSSL false  220 mx.google.com ESMTP cc15sm31085096pac.1 - gsmtp  DEBUG SMTP: connected to host "smtp.gmail.com", port: 25    EHLO 192.168.2.6  250-mx.google.com at your service, [14.102.109.172]  250-SIZE 35882577  250-8BITMIME  250-STARTTLS  250-ENHANCEDSTATUSCODES  250 PIPELINING  DEBUG SMTP: Found extension "SIZE", arg "35882577"  DEBUG SMTP: Found extension "8BITMIME", arg ""  DEBUG SMTP: Found extension "STARTTLS", arg ""  DEBUG SMTP: Found extension "ENHANCEDSTATUSCODES", arg ""  DEBUG SMTP: Found extension "PIPELINING", arg ""  STARTTLS  220 2.0.0 Ready to start TLS  EHLO 192.168.2.6  250-mx.google.com at your service, [14.102.109.172]  250-SIZE 35882577  250-8BITMIME  250-AUTH LOGIN PLAIN XOAUTH XOAUTH2  250-ENHANCEDSTATUSCODES  250 PIPELINING  DEBUG SMTP: Found extension "SIZE", arg "35882577"  DEBUG SMTP: Found extension "8BITMIME", arg ""  DEBUG SMTP: Found extension "AUTH", arg "LOGIN PLAIN XOAUTH XOAUTH2"  DEBUG SMTP: Found extension "ENHANCEDSTATUSCODES", arg ""  DEBUG SMTP: Found extension "PIPELINING", arg ""  DEBUG SMTP: Attempt to authenticate using mechanisms: LOGIN PLAIN DIGEST-MD5 NTLM  DEBUG SMTP: AUTH LOGIN command trace suppressed  DEBUG SMTP: AUTH LOGIN succeeded  DEBUG SMTP: use8bit false  MAIL FROM:<user@user-PC>  250 2.1.0 OK cc15sm31085096pac.1 - gsmtp  RCPT TO:<lokeshgupta1981@gmail.com>  250 2.1.5 OK cc15sm31085096pac.1 - gsmtp  DEBUG SMTP: Verified Addresses  DEBUG SMTP:   lokeshgupta1981@gmail.com  DATA  354  Go ahead cc15sm31085096pac.1 - gsmtp  Date: Mon, 27 May 2013 21:06:28 +0530 (IST)  To: lokeshgupta1981@gmail.com  Message-ID: <7224872.0.1369668988268.JavaMail.user@user-PC>  Subject: Test Subject  MIME-Version: 1.0  Content-Type: text/plain; charset=us-ascii  Content-Transfer-Encoding: 7bit    Testing body  .  250 2.0.0 OK 1369668992 cc15sm31085096pac.1 - gsmtp  QUIT  221 2.0.0 closing connection cc15sm31085096pac.1 - gsmtp  DEBUG: getProvider() returning javax.mail.Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Oracle]  DEBUG SMTP: useEhlo true, useAuth true  DEBUG SMTP: trying to connect to host "smtp.gmail.com", port 25, isSSL false  220 mx.google.com ESMTP qp4sm28954590pbc.41 - gsmtp  DEBUG SMTP: connected to host "smtp.gmail.com", port: 25    EHLO 192.168.2.6  250-mx.google.com at your service, [14.102.109.172]  250-SIZE 35882577  250-8BITMIME  250-STARTTLS  250-ENHANCEDSTATUSCODES  250 PIPELINING  DEBUG SMTP: Found extension "SIZE", arg "35882577"  DEBUG SMTP: Found extension "8BITMIME", arg ""  DEBUG SMTP: Found extension "STARTTLS", arg ""  DEBUG SMTP: Found extension "ENHANCEDSTATUSCODES", arg ""  DEBUG SMTP: Found extension "PIPELINING", arg ""  STARTTLS  220 2.0.0 Ready to start TLS  EHLO 192.168.2.6  250-mx.google.com at your service, [14.102.109.172]  250-SIZE 35882577  250-8BITMIME  250-AUTH LOGIN PLAIN XOAUTH XOAUTH2  250-ENHANCEDSTATUSCODES  250 PIPELINING  DEBUG SMTP: Found extension "SIZE", arg "35882577"  DEBUG SMTP: Found extension "8BITMIME", arg ""  DEBUG SMTP: Found extension "AUTH", arg "LOGIN PLAIN XOAUTH XOAUTH2"  DEBUG SMTP: Found extension "ENHANCEDSTATUSCODES", arg ""  DEBUG SMTP: Found extension "PIPELINING", arg ""  DEBUG SMTP: Attempt to authenticate using mechanisms: LOGIN PLAIN DIGEST-MD5 NTLM  DEBUG SMTP: AUTH LOGIN command trace suppressed  DEBUG SMTP: AUTH LOGIN succeeded  DEBUG SMTP: use8bit false  MAIL FROM:<howtodoinjava@gmail.com>  250 2.1.0 OK qp4sm28954590pbc.41 - gsmtp  RCPT TO:<howtodoinjava@gmail.com>  250 2.1.5 OK qp4sm28954590pbc.41 - gsmtp  DEBUG SMTP: Verified Addresses  DEBUG SMTP:   howtodoinjava@gmail.com  DATA  354  Go ahead qp4sm28954590pbc.41 - gsmtp  Date: Mon, 27 May 2013 21:06:41 +0530 (IST)  From: howtodoinjava@gmail.com  To: howtodoinjava@gmail.com  Message-ID: <32482448.1.1369669001408.JavaMail.user@user-PC>  Subject: FATAL - Application crash. Save your job !!  MIME-Version: 1.0  Content-Type: text/plain; charset=us-ascii  Content-Transfer-Encoding: 7bit    Exception occurred everywhere.. where are you ????  .  250 2.0.0 OK 1369669005 qp4sm28954590pbc.41 - gsmtp  QUIT  221 2.0.0 closing connection qp4sm28954590pbc.41 - gsmtp |