BUG总结

Bug名称：Spring发送邮件

日期：2017.3.8

遇到的BUG : 在使用spring和JavaMail发送邮件的时候出现了530的错误，具体BUG如下：

Exception in thread "main" org.springframework.mail.MailAuthenticationException: Authentication failed; nested exception is javax.mail.AuthenticationFailedException

Caused by: javax.mail.AuthenticationFailedException

造成原因:

1. 邮箱没有开通 POP3/SMTP服务IMAP/SMTP服务
2. 密码用的是自己账号密码
3. 端口使用错误的默认端口

解决方案

1. 开通邮箱POP3/SMTP服务IMAP/SMTP服务
2. 密码使用授权码
3. 3.端口使用465

Maven中jar包的依赖信息

dependency>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

<version>1.4.7</version>

</dependency>

Mail.properties 属性的配置：

mail.host=smtp.qq.com

mail.username=XXX@qq.com

mail.password=XXXXX

mail.smtp.auth=true

mail.smtp.timeout=25000

mail.from=1843984481@qq.com

mail.smtp.port=465

Spring-context.xml 中的配置信息:

<!-- 邮件发送器 -->

<bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">

<property name="host" value="${mail.host}"/>

<property name="username" value="${mail.username}"/>

<property name="password" value="${mail.password}"/>

<property name="defaultEncoding" value="UTF-8"/>

<property name="port" value="${mail.smtp.port}"/>

<property name="javaMailProperties" >

<props>

<prop key="mail.smtp.auth">${mail.smtp.auth}</prop>

<prop key="mail.smtp.timeout">${mail.smtp.timeout}</prop>

<prop key="mail.smtp.socketFactory.port">${mail.smtp.port}</prop>

<prop key="mail.smtp.socketFactory.class">javax.net.ssl.SSLSocketFactory</prop>

<prop key="mail.smtp.socketFactory.fallback">false</prop>

</props>

</property>

</bean>

<!-- 邮件发送人 -->

<bean id="simpleMailMessage" class="org.springframework.mail.SimpleMailMessage">

<property name="from" value="${mail.from}"></property>

</bean>

邮件发送实体类的编写：

public class SendMailServiceImpl implements SendMailService {

@Autowired

private MailSender mailSender;

@Autowired

private SimpleMailMessage simpleMailMessage;

@Override

public void send(String recipient, String subject, String content) {

simpleMailMessage.setTo(recipient);

simpleMailMessage.setSubject(subject);

simpleMailMessage.setText(content);

mailSender.send(simpleMailMessage);

}

@Override

public void send(List<String> recipients, String subject, String content) {

simpleMailMessage.setTo(recipients.toArray(new String[recipients.size()]));

simpleMailMessage.setSubject(subject);

simpleMailMessage.setText(content);

mailSender.send(simpleMailMessage);

}

}

BUG2 ajax请求服务器返回乱码的问题

原因：原因是这可以说是spring mvc的一个bug，spring MVC有一系列HttpMessageConverter去处理用@ResponseBody注解的返回值，如返回list或其它则使用 MappingJacksonHttpMessageConverter，返回string，则使用 StringHttpMessageConverter，而这个convert使用的是字符集是iso-8859-1,而且是final的。所以在当返回json中有中文时会出现乱码。

1. public static final Charset DEFAULT\_CHARSET = Charset.forName("ISO-8859-1");

解决方案

1.

<mvc:annotation-driven >

<!-- 消息转换器 -->

<mvc:message-converters register-defaults="true">

<bean class="org.springframework.http.converter.StringHttpMessageConverter">

<property name="supportedMediaTypes" value="text/html;charset=UTF-8"/>

</bean>

</mvc:message-converters>

</mvc:annotation-driven>

解决方案2：<mvc:annotation-driven>

<mvc:message-converters>

<bean class="org.springframework.http.converter.StringHttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<span style="white-space:pre"> </span> <value>text/html;charset=UTF-8</value>

<value>application/json;charset=UTF-8</value>

<value>\*/\*;charset=UTF-8</value>

</list>

</property>

</bean>

</mvc:message-converters>

</mvc:annotation-driven>

解决方案3

@RequestMapping(value = "\*\*\*",produces="text/html;charset=UTF-8;")

都解决不了：去死吧

# Mybatis 用注解实现批量插入

用例:

@Insert("<script>"+

"insert into tag\_main(name, create\_time, update\_time) "

+ "values "

+ "<foreach collection =\"list\" item=\"tag\" index= \"index\" separator =\",\"> "

+ "(#{tag.name},#{tag.createTime},#{tag.updateTime}) "

+ "</foreach > "

+ "</script>")

Integer saveTags(List<Tag> list);

用上述方法便可实现批量插入