Spring属性文件properties的读取与使用

实际项目中，通常将一些可配置的定制信息放到属性文件中（如数据库连接信息，邮件发送配置信息等），便于统一配置管理。例中将需配置的属性信息放在属性文件resource目录下。配置文件如下：

|  |
| --- |
| #properties文件  mail.host=172.16.130.170  mail.user=zxx  mail.password=zxx  uploadPath=upload  backupPath=backup |
| #spring配置  <bean id="propertyConfigurer"  class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  <property name="locations">  <list>  <value>classpath:env/env.properties</value>  <value>classpath:app-config.properties</value>  </list>  </property>  </bean> |

## spring配置文件中注入bean(**引用${})**

|  |
| --- |
| #spring装配bean  <bean id="javaMailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">  <property name="host" value="${mail.host}" />  <property name="username" value="${mail.user}" />  <property name="password" value="${mail.password}" />  <property name="javaMailProperties">  <props>  <prop key="mail.smtp.auth">true</prop>  <prop key="mail.smtp.port">25</prop>  </props>  </property>  </bean> |
| #注入bean  @Autowired  private javaMailSender JavaMailSender; |

## 代码中使用@Value("${}")注入值

|  |
| --- |
| #装配bean，注入值。  @Component  **public** **class** AppConfig {  @Value("${uploadPath}")  **private** String uploadPath;    @Value("${backupPath}")  **private** String backupPath;  **public** String getUploadPath() {  **return** uploadPath;  }  **public** **void** setUploadPath(String uploadPath) {  **this**.uploadPath = uploadPath;  }  **public** String getBackupPath() {  **return** backupPath;  }  **public** **void** setBackupPath(String backupPath) {  **this**.backupPath = backupPath;  }  }  #注入  @Autowired  **private** AppConfig appConfig; |