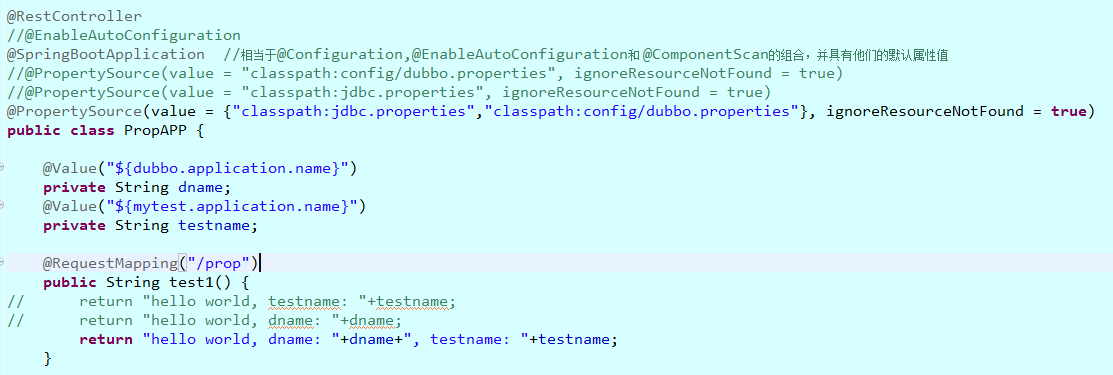
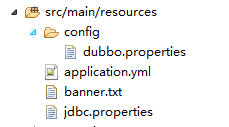
# 读取配置

## 方式一 @Value和@PropertySource

@PropertySource 注解加载文件来配合，注意单个文件和多个文件加载的区别。

@Value 获取配置的值

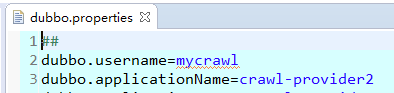


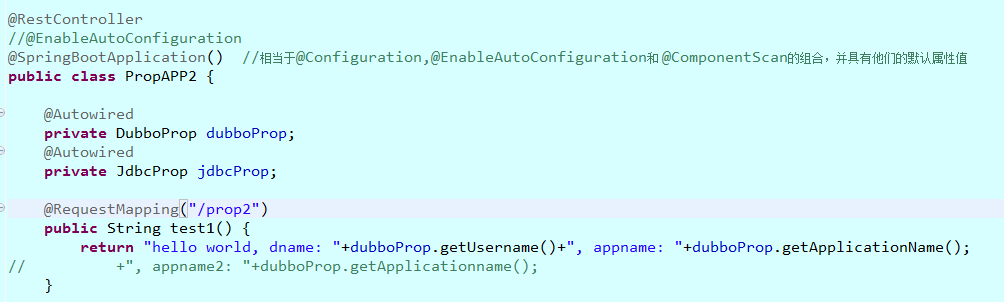


## 方式二 @ConfigurationProperties用类来接收

定义一个类和配置文件对应，标明了前缀，则申明类属性的时候可以将其去掉，剩余部分对应即可，如果前缀是1节，key有3节的，那么需要新建将前缀定义为2节的类。

|  |
| --- |
| @Component  @ConfigurationProperties(prefix = "dubbo")  //指定文件没什么用  //@PropertySource(value = "classpath:config/dubbo.properties", ignoreResourceNotFound = true)  **public** **class** DubboProp {  **private** String username;  **private** String applicationName;  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getApplicationName() {  **return** applicationName;  }  **public** **void** setApplicationName(String applicationName) {  **this**.applicationName = applicationName;  }  } |





此处有个地方需要再次强调，@SpringBootApplication自动扫描所在类的同级包,以及下级包里的所有bean。

也可通过 @SpringBootApplication(scanBasePackages = {"firstdemo","test"}) 来指定扫描包

# 配置文件里面可以生成随机数据

