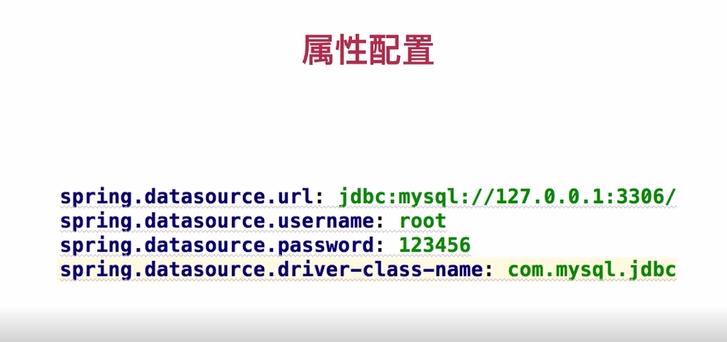
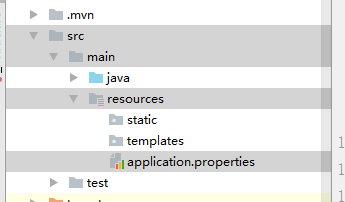
## 1.属性配置介绍



# 2.打开配置文件



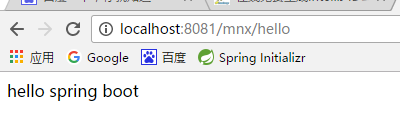
# 3.编辑配置文件application.properties信息

## 1.编辑浏览器访问端口 和文件路径

**server.port**=**8081  
server.context-path**=**/mnx**

## 2.测试 打开浏览器地址

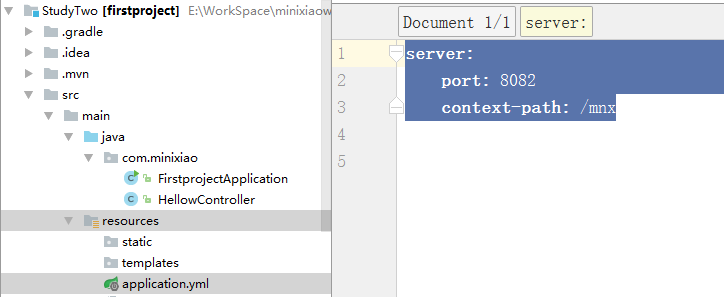
<http://localhost:8081/mnx/hello>



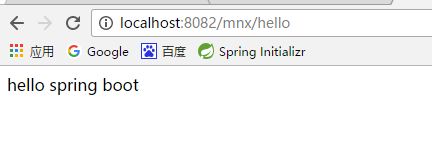
# 4.换一种配置文件的格式 application.yml

## 1.修改文件信息

**server:  
 port:** 8082  
 **context-path:** /mnx



## 2.测试成功 8082 端口



# 5.利用配置文件进行注入@Value("${age}") 注解进行注入

## 1.配置文件写入 相关属性

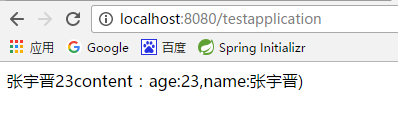
**server:****port:** 8080  
**name:** 张宇晋  
**age:** 23  
**content: "age:${age},name:${name}"**

## 2.Java controler文件中

@RestController**public class** HellowController {

@Value(**"${age}"**)   
**private int age**;  
@Value(**"${name}"**)  
**private** String **name**;  
@Value(**"${content}"**)  
**private** String **content**;  
 @RequestMapping(value = **"/testapplication"** ,method = RequestMethod.***GET***)  
 **public** String TestApplication(){  
 **return name**+**age**+**"content："**+**content**;  
 }

## 3.测试成功 <http://localhost:8080/testapplication>



# 6、添加前缀

## 1.配置文件中

*# 3***server:  
 port:** 8080  
person: **age:** 23  
 **name:** 张宇晋

## 2、新建一个javabean进行获取值

*//把普通pojo实例化到spring容器中，相当于配置文件中的<bean id="" class="">*@Component  
*//识别配置文件的前缀prefix，可以通过一个注解ConfigurationProperties然后根据属性进行注入*@ConfigurationProperties(prefix = **"person"**)  
**public class** TestTwoPersonBean {  
 **private** String **name**;  
 **private int age**;  
 **public** String getName(){  
 **return this**.**name**;  
 }  
 **public void** setName(String name){  
 **this**.**name**= name;  
 }  
  
 **public int** getAge(){  
 **return this**.**age**;  
 }  
 **public void** setAge(**int** age){  
 **this**.**age**= age;  
 }  
}

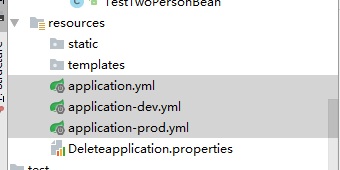
## 3.controler进行测试

@Autowired *// 省略 get/setter 直接进行使用***private** TestTwoPersonBean **person**;  
 @RequestMapping(value = **"/testtwopersonbean"** ,method = RequestMethod.***GET***)  
 **public** String TestApplication(){  
 **return person**.getName();  
 }

## 4.测试失败

# 7、更多的生产空间

## 1.多建立几个配置文件（复制application.yml的内容，并进行属性的重新赋值）



## 2.application.yml重新写入，这个就是相当于总控制器，下面的代码就是让dev的配置文件执行（如果是prod则是另一个配置文件）

**spring:  
 profiles:  
 active:** dev

## 3.那为什么还要使用两个配置文件，

### 不同环境使用不同配置，(中间通过更改了spring:profiles:active)比如一个是在mvn中启动的，一个是在idea重启动的，在浏览器中出入两个不同的端口都是可以实现的