

springboot-offcn

- 1.springboot的maven依赖问题
- 2.关于configureProperties的报错问题
- 3. 关于yml及其元数据书写问题
- 4. springboot的多环境支持
- 5.静态资源匹配问题
- 6. swagger接口整合
- 7.boot整合mybatis出错点
- 8.springboot整合mvc逻辑错误点
- 9.关于单元测试逻辑错误点
- 10.boot配置文件加载顺序和位置
- 12.springboot整合redis或者缓存
- 13.fastjson使用小技巧
- 14.springboot整合dubbo
- 15.如何创建前端的无脑要求

与手册参考配合使用： `source_everyday\springboot+vue+axios\offcn`

1.springboot的maven依赖问题

Java Version: 8

Version: 0.0.1-SNAPSHOT

Name: day16_springbootDemo01 2 maven名

Description: Demo project for Spring Boot

Package: com.offcn.demo 1 入口位置

```
<modelVersion>4.0.0</modelVersion>
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.6.8</version> 1 继承就是springboot项目
  <relativePath/> <!-- Lookup parent from repository -->
</parent>
```

war代表是使用外置的tomcat

2.关于configureProperties的报错问题

自动配置关联配置：

- configureProperties+ compent
- configureProperties+EnableAutoConfigureProperties(自动配置)

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-configuration-processor</artifactId>
  <version>${version.boot}</version>
  <optional>true</optional>
</dependency>
```

```
@ConfigurationProperties(prefix = "yunai.server")
public class YunaiServerProperties {
  /**
   * 默认端口
   */
  private static final Integer DEFAULT_PORT = 8000;
  /**
```

- yml 没提示： 打包+重启+才好使

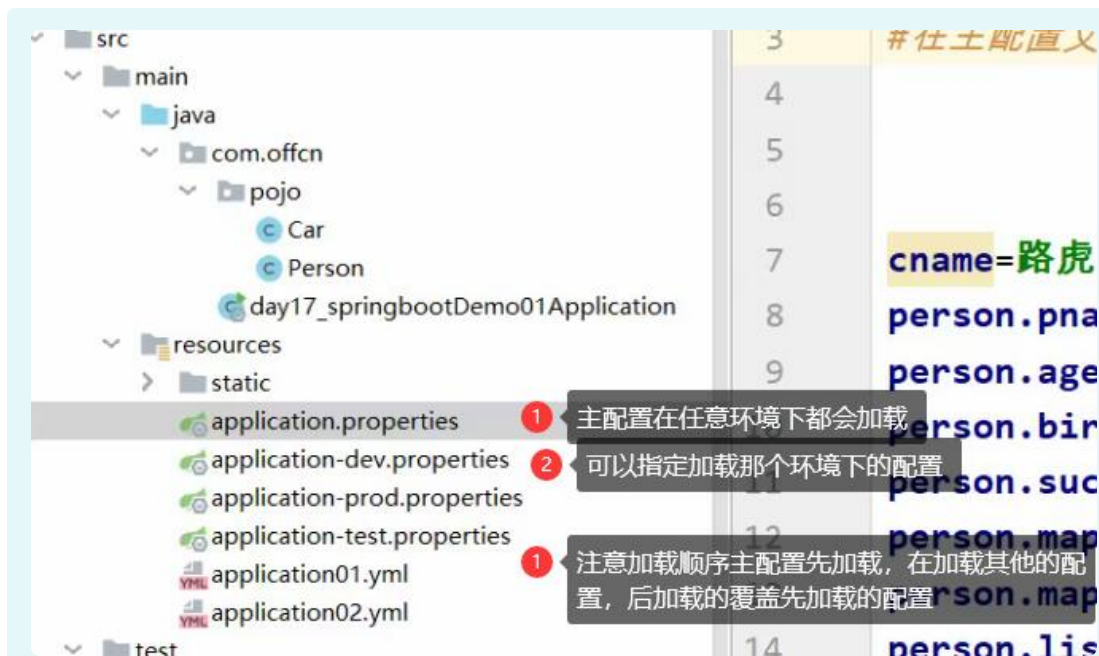
3. 关于yml及其元数据书写问题

```
name: "我是帅哥 \n 吗?"
name02: '我是帅哥 \n 吗?'
name03: 我是帅哥 \n 吗 #
```

- ""： \n代表换行
- '': \n代表真正的\n没有换行的意思

```
# 引入外包
person:
  pname: machao
  age: 18
  success: true
  birth: 2020/12/23
  car:
    cid: 10001
    cname: baoshijie
    cprice: 10000
  # k1: v1
  # k2: v2
  maps: {k1: v1,k2: v2}
  # ~ 1
  # ~ 2
  lists: [1,2,3,4]
spring:
  profiles:
    active: dev
```

4. springboot的多环境支持



- `spring.profiles.active=dev` 或者

```
-Dserver.port=8086 -Dspring.profiles.active=test
```

5. 静态资源匹配问题

找：自动配置类

`webMvcAutoConfiguration` ----- `staticLocations`

6. swagger接口整合

- 依赖

`<!--swagger-->`

访问路径: <http://localhost:8080/swagger-ui.html>

`<dependency>`

`<groupId>com.spring4all</groupId>`

`<artifactId>swagger-spring-boot-starter</artifactId>`

`<version>1.9.0.RELEASE</version>`

`</dependency>`

`<dependency>`

`<groupId>org.springframework.boot</groupId>`

`<artifactId>spring-boot-starter-validation</artifactId>`

`</dependency>`

`<!--swagger皮肤-->`

访问路径: <http://localhost:8080/doc.html>

`<dependency>`

`<groupId>com.github.xiaoymin</groupId>`

`<artifactId>swagger-bootstrap-ui</artifactId>`

`<version>1.9.1</version>`

`</dependency>`

- 资料参考

参考资料一: <https://github.com/dyc87112/SpringBoot-Learning/tree/master/2.x>

参考资料二: <https://blog.csdn.net/Yearningforthefuture/article/details/117421390>

- yml配置: 几乎不用变

`#swagger整合`

swagger:

title: spring-boot-starter-swagger

description: Starter for swagger 2.x

version: 1.9.0.RELEASE

license: Apache License, Version 2.0

license-url: <https://www.apache.org/licenses/LICENSE-2.0.html>

terms-of-service-url: <https://github.com/dyc87112/spring-boot-starter-swagger>

contact:

name: guihaole

url: http://blog.didispace.com

email: 2033443140@qq.com

base-package: com.edu.controller

base-path: /**

- 注解作用理解

@Api(tags = "账户管理")

@ApiOperation(value = "查询所有账户信息", notes = "参数可以传也可以不传")

@ApiImplicitParam(paramType = "path", dataType = "Integer",
name = "id", value = "用户编号", required = true, example = "1")

7.boot整合mybatis出错点

```
4 mybatis:
5   type-aliases-package: com.edu.bean
6   mapper-locations: classpath:mappers/*Mapper.xml
7   configuration:
8     cache-enabled: true #开启二级缓存
9     lazy-loading-enabled: true #开启延时加载开关
10    aggressive-lazy-loading: false #关闭立即加载，实施按需加载
11  #分页插件的配置
```

① 不加/mappers, 要不然会报找不到映射错误

mapper.xml写sql时不要带分号；分页可能会报错

8.springboot整mvc逻辑错误点

- 2.7.0以后springboot的自动配置类放在了spring文件下，没在spring.factories
- 视图解析器默认走static下，前缀和后缀
- templates配置前缀和后缀templates下

<https://github.com/dyc87112/SpringBoot-Learning/tree/master/2.x> 各种连接池配置

9.关于单元测试逻辑错误点

- springboot-start整合时默认是**不支持junit4的**
- 导报要到对
- 如何兼容junit4 看中公boot文档

```
import com.edu.bean.Person;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
public class BeanTest {
    @Autowired
    private Person person;
    @Test
    public void test1(){
        System.out.println(person);
    }
}
```

1 相当于整合了spring和junit

10.boot配置文件加载顺序和位置

- **spring会加载所有的配置文件，后加载的覆盖新加载的**
- **properties和yml都存在，都加载，yml先加载，后覆盖先加载的**
- 什么位置：看源码怎么写即可 spring.config.location可以自定义

12.springboot整合redis或者缓存

参考资料一：source_everyday\redis开发手册 boot-api

参考资料二： <https://github.com/dyc87112/SpringBoot-Learning/tree/master/2.x>

类关系，用哪个都可以，子类功能更强大

```
@SpringBootApplication
public class RedisApi {
    @Autowired
    private StringRedisTemplate stringRedisTemplate;
    @Autowired
    private RedisTemplate<Object, Object> redisTemplate;

    public void stringRedis(){
        BoundValueOperations<String, String> boundValueOps = stringRedisTemplate.boundValueOps(key: "123");
        boundValueOps.set("123456");

        ValueOperations<Object, Object> opsForValue = redisTemplate.opsForValue();
        opsForValue.set(1, 1);
    }
}
```

① 父子关系，string~是子类

② 存的key相当于密钥

③ 设置一对值

13.fastjson使用小技巧

```
2 <dependency>
3 <groupId>com.alibaba</groupId>
4 <artifactId>fastjson</artifactId>
5 <version>1.2.73</version>
6 </dependency>
7
```

2.将对象转化为json数据

JSON.toJSONString("对象")

3.将json数据转化为对象

JSON.parseObject("json字符串", new TypeReference<要转化的对象>())


```

@Test
void test03() {
    ValueOperations<String, String> opsForValue = redisTemplate.opsForValue();
    String userJson = opsForValue.get("user");
    JSON.parseObject(userJson, new TypeReference<User>(){});
}

```

1 万能转换

```

String userJson = opsForValue.get("userList");
List<User> userList1 = JSON.parseObject(userJson, new TypeReference<List<User>>(){});

```

- 万能转换公式

14.springboot整合dubbo

- 关键点：实体类一定要序列化
- 参考资料：后端开发\dubbo
- 参考资料2：source_everyday\springboot+vue+axios\offcn\整合文档
- 参考资料三：上github找资源

15.如何创建前端的无脑要求

```

70  @test
71  void test05() {
72      List<Object> list=new ArrayList<>();
73      Map<String,Object> map=new HashMap<String, Object>();
74      map.put("id",1);map.put("name",1);map.put("birth",new Date());map.put("age",18);
75      String s = JSON.toJSONString(map);
76      System.out.println(s);
77  }
78
79

```

1 <[[k1,v1],[k1,v1]]

2 不让我创建实体类怎么搞

1 前端无脑需求