

POJO、entity、domain、DTO、VO 的简单说明

POJO (Plain Ordinary Java Object) 简单的Java对象，实际就是普通JavaBeans (规范), pojo是业务称呼。

entity: 实体类, 一个类, 属性严格对应一个表格的行数据

domain: 封装数据库的javabean对象, 一个对象可以封装多个表格数据. (domain的范围大于entity)

DTO: 专门负责接收前端表格数据的对应结构的javabean.

vo: view object, 视图对象, ajax要什么, 就封装什么

POJO与JavaBean的比较:

pojo: 用于数据的临时传递, 它只能装载数据, 作为数据存储的载体, 不具有业务逻辑的处理能力。

javaBean: 与pojo相比, 它里面可以有其他方法。

JavaBean是可重用组件, 需要符合以下条件:

这个类必须有一个公共的缺省构造函数

这个类的属性使用getter和setter来访问设置数据, 其他方法遵从标准命名规范

这个类可以序列化

Entity, DTO, SpringMVC 参数 类型

对于Entity里的数值字段类型最好使用包装类, 不要使用int, long, 或者BigInteger, 使用Long比较好;

对于DTO的数值也是最好使用包装类, 但如果是一些计数器count的话, 可以使用int, 或者Integer, 其他的使用Long;

对于SpringMVC 方法的参数, 最好使用包装类, 因为使用包装类, 即使前端不传参数, 也会映射一个null对象, 但如果使用原始数据类型, 如下:

```
@RequestMapping(value = "/test")
@ResponseBody
```

```
public String test(int i) {  
    return "data=" + i;  
}
```

如果前端不传参数i的话，或抛出异常，但如果使用包装类

```
@RequestMapping(value = "/test")  
@ResponseBody  
public String test(Integer i) {  
    return "data=" + i;  
}
```

如果前端不传参数i的话，也不会抛出异常，i 会赋值为null