# POJO

## 一：什么是POJO

POJO 按照Martin Fowler的解释是“Plain Old Java Object”， “简单java对象”。 POJO的内在含义是指那些没有从任何类继承、也没有实现任何接口，更没有被其它框架侵入的java对象。

## 二：为什么会有POJO？

主要是Java的开发者被EJB的繁杂搞怕了，经过反思，又回归“纯洁老式”的JavaBean，即有无参构造函数，每个字段都有getter和setter的java类。

## 三：POJO的意义

POJO让开发者可专注于业务逻辑和脱离框架的单元测试。

POJO的意义就在于它的简单而灵活性，使得POJO能够任意扩展，从而胜任多个场合，也就让一个模型贯穿多个层成为现实。

先写一个核心POJO，然后实现业务逻辑接口和持久化接口，就成了Domain Model； UI需要使用时，就实现数据绑定接口，变成VO（View Object）。

## 四：POJO与PO、VO的区别

POJO是指简单java对象。

PO是指持久对象（persistant object持久对象）。

VO是指值对象或者View对象（Value Object、View Object）。注意，本文的VO特指View Object。

持久对象实际上必须对应数据库中的entity，所以和POJO有所区别。比如说POJO是由new创建，由GC回收。但是持久对象是insert数据库创建，由数据库delete删除的。基本上持久对象生命周期和数据库密切相关。另外持久对象往往只能存在一个数据库Connection之中，Connnection关闭以后，持久对象就不存在了，而POJO只要不被GC回收，总是存在的。

## 五：POJO的扩展

POJO仅包含最简单的字段属性，没有多余的东西，它本质上就是一个普通的JavaBean。 但是在POJO的基础上，能够扩展出不同的对象。 为POJO增加了持久化的方法（Insert、Update、Delete……）之后，POJO就变成了PO。 为POJO增加了数据绑定功能之后，POJO就变成了View Object，即UI Model。 为POJO增加业务逻辑的方法（比如单据审核、转帐……）之后，POJO就变成了Domain Model。 POJO还可以当作DTO使用。

### Javabean，POJO，PO，VO，DTO

javabean: 在Java程序或JSP程序中，其实有很多的代码段是可以重复使用的，比如对数据库的操作、用户的有效性检查及某些项特定功能的实现等。为了很好的解决这个问题，提高开发效率，Sun公司推出了JavaBean，

简言之，JavaBean这是一个实现了某个功能且供重复使用的Java类。 在JSP中使用JavaBean后，可实现HTML与Java代码的分离，使JSP更易于开发和维护。因此JavaBean成了JSP程序员的必备利器之一。 虽然JavaBean是Java类，同时它也具有自己的一些特征：

1、 JavaBean是一个public的类

2、 JavaBean有一个不带参数的构造方法

3、 JavaBean中的属性通过getXXX()和setXXX()进行操作 按着Sun公司的定义，JavaBean是一个可重复使用的软件组件。

实际上JavaBean是一种Java类，通过封装属性和方法成为具有某种功能或者处理某个业务的对象，简称bean。

javabean是基于java语言的，不依赖平台，具有以下特点：

1． 可以实现代码的重复利用

2． 易编写、易维护、易使用

3． 可以在任何安装了Java运行环境的平台上的使用，而不需要重新编译。 比如我们编写一个计算圆面积的java类CircleArea.java 以后需要计算圆面积的地方直接调用就行了。

---来自百度对POJO的解释：简单的Java对象（Plain Ordinary Java Objects）实际就是普通JavaBean,使用POJO名称是为了避免和EJB混淆起来, 而且简称比较直接. 其中有一些属性及其getter setter方法的类,有时可以作为value object或dto(Data Transform Object)来使用.

对于初识POJO的人来说，它很像JavaBean，但按照其应与其他组件易组合的设计目的，它不应包含业务逻辑的处理，这也是与JavaBean不同的地方。

PO －Persistent Object ，即＂持久化对象＂． PO是位于EIMS层数据库中的实体，生命周期和数据库密切相关．在向数据库插入记录时创建该实体，删除或关闭数据库时该实体随之消亡．

很多优秀的开源框架都实现了将数据库中的PO通过ORM用POJO来实际操作，如Hibernate,JDO等。

最后我们总结一下：

1.VO/DTO 就是Value Object和Data Transform object就是作为数据使用的对象

2.POJO，是Plain Ordinary（Old）Java Object的缩写，是不包含业务逻辑的JavaBean，通常情况下可以作为VO/DTO来使用

3.PO，是Persistence Object的缩写，来自各种各个JPA ORM(Java Persistent API Object Relational Mapping)提供商比如Hibernate toplink等厂商，是跟数据库交互的时候存在的中间对象，而且可以有各种逻辑操作。

pojo与实体bean： pojo 应用于页面数据的操作，是来保存数据的。（你也可以不这样做，但是这已经成了一种规范。） 实体bean与数据库表关联，是对数据表的操作。

其它： POJO和JavaBean是我们常见的两个关键字， POJO普通Java类，具有一部分getter/setter方法的那种类就可以称作POJO，但是JavaBean则比 POJO复杂很多，JavaBean是一种组件技术，就好像你做了一个扳子，而这个扳子会在很多地方被拿去用，这个扳子也提供多种功能(你可以拿这个扳子 扳、锤、撬等等)，而这个扳子就是一个组件。

### Pojo和JavaBean的区别

POJO(Plain Old Java Object)这个名字用来强调它是一个普通java对象，而不是一个特殊的对象。 2005年11月时，“POJO”主要用来指代那些没用遵从特定的Java对象模型，约定或框架如EJB的Java对象.

理想地讲，一个POJO是一个不受任何限制的Java对象（除了Java语言规范）。例如一个POJO不应该是

* 扩展预定的类，如 public class Foo extends javax.servlet.http.HttpServlet { ...
* 实现预定的接口，如 public class Bar implements javax.ejb.EntityBean { ...
* 包含预定的标注，如 @javax.ejb.Entity public class Baz{ ...

JavaBean 是一种JAVA语言写成的可重用组件。它的方法命名，构造及行为必须符合特定的约定：

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，其他方法遵从标准命名规范。

这个类应是可序列化的。

因为这些要求主要是靠约定而不是靠实现接口，所以许多开发者把JavaBean看作遵从特定命名约定的POJO。

简而言之，当一个Pojo可序列化，有一个无参的构造函数，使用getter和setter方法来访问属性时，他就是一个JavaBean。