



链滴

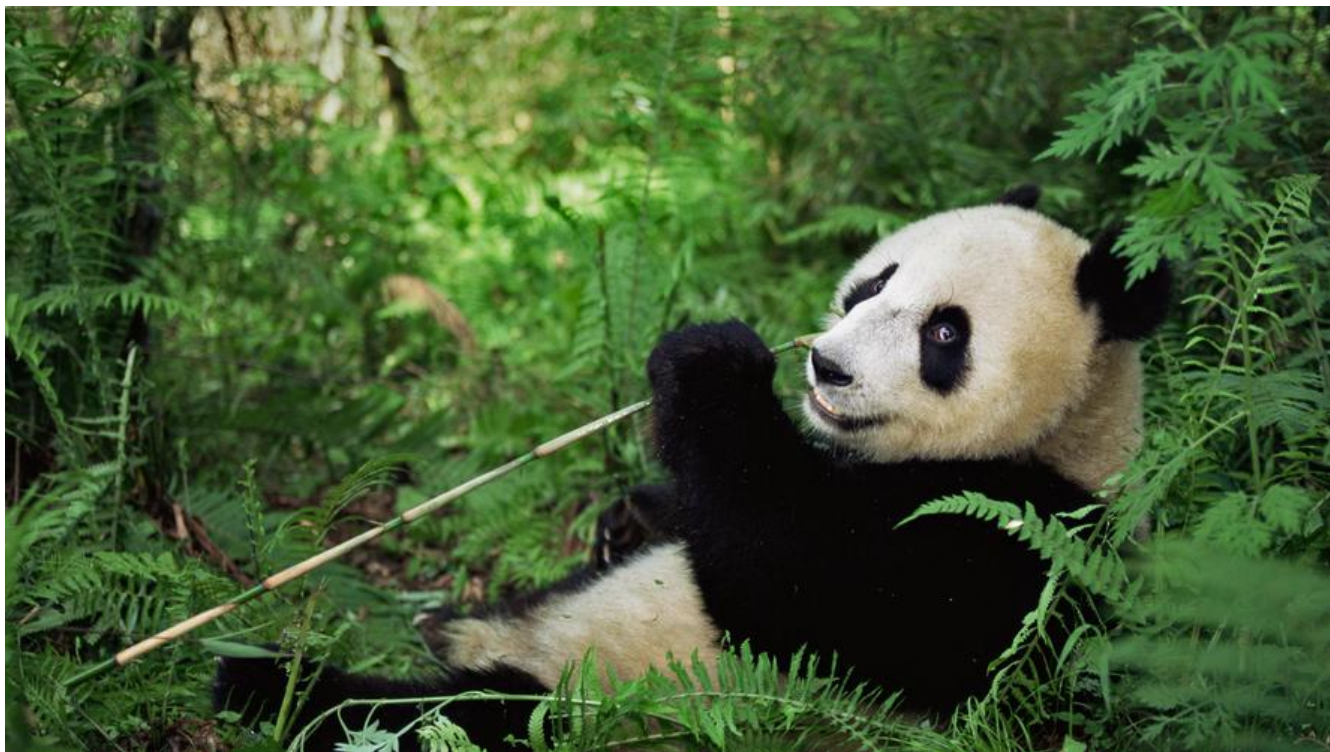
# 模仿 JAVA DTO 概念的实现的包 PyDto

作者: [Gaoshengyue](#)

原文链接: <https://ld246.com/article/1598516803085>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## PyDto实现记录

### 实现背景

因为python语言的特性，在业务数据返回方面，没有类似JAVA DTO的概念实现主体，苦于目前返回数据结构都需要自己写代码整合，所以需要模仿JAVA DTO的概念写一个工具包，方便定义返回结构。

### 诉求

例如在返回一个用户数据的时候，python的sqlalchemy或者某一个类直接返回所有的结构数据，甚至会带上token等敏感数据，这时候通常会在返回前使用代码拼装整合，变动结构也需要变动代码。理解AVA DTO概念后想到是否有类似的包，可以在数据层与业务层之间进行拦截，根据定义好的结构直返回相应数据结构。

### 思路

如果要方便使用，肯定不能在sqlalchemy或者字典或者对象本身进行操作，需要进行隔离，那么选用方式就是类继承的形式，将数据解析以及自动拼装的方法封装进去。

字典与类的形式相对简单，容易解析多层结构，但是遇到sqlalchemy需要进行调用返回链表数据结构所以需要特殊的方式进行处理。

```
if type(obj) != dict and not hasattr(obj, "__iter__"):
    for key_name, k_type in obj_dict_class.items():
        if obj.__getattr__(key_name):
            if k_type.__dict__.get("__annotations__") and self.data_mode == "sqlalchemy":
                result[key_name] = self.MapSchema(obj.__getattr__(key_name),
                                                  k_type.__dict__.get("__annotations__"))
```

```
else:  
    result[key_name] = self.todict(obj.__getattr__(key_name))
```

需要判断链表后的数据是单结构还是迭代对象，所以在进行判断，并根据数据体结构，传入到下结构中。

## 项目

[PyDto实现](#)

这里是实现的源码，整体思路可以到代码中详细查看