Atitit 最佳实践 返回null还是返回空

目录

[1.1. 返回一个空的collections（empty list),而不要返回null 1](#_Toc10349)

[1.2. 使用Null Object pattern（空对象模式） 1](#_Toc10512)

## 返回一个空的collections（empty list),而不要返回null

**这里给一些实践建议：**  
1、假如方法的返回类型是collections，当返回结果是空时，你可以返回一个空的collections（empty list),而不要返回null，这样调用侧就能大胆地处理这个返回，例如调用侧拿到返回后，可以直接print list.size()，又无需担心空指针问题。（什么？想调用这个方法时，不记得之前实现该方法有没按照这个原则？所以说，代码习惯很重要！如果你养成习惯，都是这样写代码（返回空collections而不返回null)，你调用自己写的方法时，就能大胆地忽略判空）

## 使用Null Object pattern（空对象模式）

2、返回类型不是collections，又怎么办呢？  
那就返回一个空对象（而非null对象），下面举个“栗子”，假设有如下代码

public interface Action {  
 void doSomething();}  
   
public interface Parser {  
 Action findAction(String userInput);}

其中，Parse有一个接口FindAction，这个接口会依据用户的输入，找到并执行对应的动作。假如用户输入不对，可能就找不到对应的动作（Action），因此findAction就会返回null，接下来action调用doSomething方法时,就会出现空指针。

解决这个问题的一个方式，就是使用Null Object pattern（空对象模式）

、Java8或者guava lib中，提供了Optional类，这是一个元素容器，通过它来封装对象，可以减少判空。不过代码量还是不少。不爽。  
3、如果你想返回null，请停下来想一想，这个地方是否更应该抛出一个异常