



链滴

java 反射的应用场景

作者: [moloee](#)

原文链接: <https://ld246.com/article/1505382718866>

来源网站: [链滴](#)

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

问题

网上很多将反射的，举n多例子说明反射怎么用，有啥用，能咋用。然而很搞不明白如下这样的例子啥用？

```
class = Class.forName("cn.lee.demo.Person");
//得到一系列构造函数集合
Constructor[] constructors = class.getConstructors();

person = (Person) constructors[0].newInstance();
person.setAge( );
person.setName("leeFeng");

person = (Person) constructors[1].newInstance(,"leeFeng");

System.out.println("Demo " + person.getName() + " : " + person.getAge()
    + " , " + person.getName() + " : " + person.getAge()
    );
```

感觉就是吃饱了撑的。。。

使用场景

所以找了一下，终于知道实际的使用场景了。。出自[这里](#)

总的来说是

1. 框架开发中的动态配置
2. 插件开发中的持续集成
3. 应用扩展
4. ...

1) Java的反射机制在做基础框架的时候非常有用，有一句话这么说来着：**反射机制是很多Java框架基石**。而一般应用层面很少用。经典的就是在**xml文件或者properties**里面写好了配置，然后在Java里面解析xml或properties里面的内容，得到一个字符串，然后用反射机制，根据这个字符串获得某类的Class实例，这样就可以动态配置一些东西，不用每一次都要在代码里面去new或者做其他的事，以后要改的话直接改配置文件，代码维护起来就很方便了，同时有时候要适应某些需求，Java类里不一定能直接调用另外的方法，这时候也可以通过反射机制来实现。

总的来说，自己写的很少，具体什么时候要用那要看需求，反射机制无非就是根据一个String来得到要的实体对象，然后调用它原来的东西。但是如果是要自己写框架的话，那就会用得比较多了。

2) 当你做一个软件可以安装插件的功能，你连插件的类型名称都不知道，你怎么实例化这个对象呢？因为程序是支持插件的（第三方的），在开发的时候并不知道。所以无法在代码中 New 出来，但可以，通过反射，动态加载程序集，然后读出类，检查标记之后再实例化对象，就可以获得正确的类例。

3) 在编码阶段不知道那个类名,要在运行期从配置文件读取类名,这时候就没有办法硬编码new Class ame(),而必须用到反射才能创建这个对象.反射的目的就是为了扩展未知的应用。比如你写了一个程序这个程序定义了一些接口，只要实现了这些接口的dll都可以作为插件来插入到这个程序中。那么怎么现呢？就可以通过反射来实现。就是把dll加载进内存，然后通过反射的方式来调用dll中的方法。很多厂模式就是使用的反射。

缺点

一般并不建议在业务代码中用反射

1. 代码可读性变差，如果出错只能在运行时才会报错。
2. 性能相比直接写的java代码差很多。