[**#**](https://pig4cloud.com/#%E4%B8%BA%E4%BB%80%E4%B9%88%E4%BD%BF%E7%94%A8lombok) **为什么使用lombok**

还在编写无聊枯燥又难以维护的POJO吗？ 洁癖者的春天在哪里？请看Lombok！在过往的Java项目中，充斥着太多不友好的代码：POJO的getter/setter/toString；异常处理；I/O流的关闭操作等等，这些样板代码既没有技术含量，又影响着代码的美观，Lombok应运而生。首先说明一下：任何技术的出现都是为了解决某一类问题的，如果在此基础上再建立奇技淫巧，不如回归Java本身。应该保持合理使用而不滥用。

[**#**](https://pig4cloud.com/#%E5%A6%82%E4%BD%95%E5%AE%89%E8%A3%85) **如何安装**

当前你使用的ide未安装lombok. lombok能够达到的效果就是在源码中不需要写一些通用的方法，但是在编译生成的字节码文件中会帮我们生成这些方法,减少代码冗余.

[IDEA安装方法](https://blog.csdn.net/zhglance/article/details/54931430) | [eclipse安装方法](https://blog.csdn.net/arkblue/article/details/52608213)

[**#**](https://pig4cloud.com/#pig-%E4%B8%AD%E7%9A%84%E4%BD%BF%E7%94%A8%E4%BE%8B%E5%AD%90%EF%BC%8C%E4%B8%8D%E8%AE%B2%E5%B8%B8%E8%A7%81%E7%9A%84%E3%80%81%E6%8A%80%E5%B7%A7%E6%80%A7%E7%9A%84) **pig 中的使用例子，不讲常见的、技巧性的**

* @AllArgsConstructor 替代@Autowired构造注入,多个bean 注入时更加清晰L

@Slf4j

@Configuration

@AllArgsConstructor

public class RouterFunctionConfiguration {

private final HystrixFallbackHandler hystrixFallbackHandler;

private final ImageCodeHandler imageCodeHandler;

}

@Slf4j

@Configuration

public class RouterFunctionConfiguration {

@Autowired

private HystrixFallbackHandler hystrixFallbackHandler;

@Autowired

private ImageCodeHandler imageCodeHandler;

}

* @SneakyThrows

@SneakyThrows

private void checkCode(ServerHttpRequest request) {

String code = request.getQueryParams().getFirst("code");

if (StrUtil.isBlank(code)) {

throw new ValidateCodeException("验证码不能为空");

}

redisTemplate.delete(key);

}

// 不使用就要加这个抛出

private void checkCode(ServerHttpRequest request) throws ValidateCodeException {

String code = request.getQueryParams().getFirst("code");

if (StrUtil.isBlank(code)) {

throw new ValidateCodeException("验证码不能为空");

}

}

* @UtilityClass 工具类再也不用定义static的方法了，直接就可以Class.Method 使用

@UtilityClass

public class Utility {

public String getName() {

return "name";

}

}

public static void main(String[] args) {

System.out.println(Utility.getName());

}

* @CleanUp: 清理流对象,不用手动去关闭流，多么优雅

@Cleanup

OutputStream outStream = new FileOutputStream(new File("text.txt"));

@Cleanup

InputStream inStream = new FileInputStream(new File("text2.txt"));

byte[] b = new byte[65536];

while (true) {

int r = inStream.read(b);

if (r == -1) break;

outStream.write(b, 0, r);

}

[**#**](https://pig4cloud.com/#%E6%80%BB%E7%BB%93) **总结**

Lombok 常用的注解就那么几个，@Data 、@Getter/Setter ，Pig 使用例子中的几个可以让代码的更加优雅，建议在你的工程中使用