



链滴

Spring Cache 缓存注解实现

作者: [ixiaozhi](#)

原文链接: <https://ld246.com/article/1501948820396>

来源网站: 链滴

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

要使用 Spring Cache 前提是 Spring jar 的版本在 3.1 及以上，及引入 spring-context-*.jar，这个其他的环境及 jar 包不作介绍。

缓存用的实体类

缓存用的实体类必须具有 getter 和 setter 方法。如下：

```
public class ErrorEntity {
    private int code;
    private String msg;
    public ErrorEntity(int code, String msg) {
        this.code = code;
        this.msg = msg;
    }

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

缓存类实现

在缓存 Manage 管理实现类中，使用 @Cacheable 来缓存。如下：

```
public class ErrorCacheManager {
    @Cacheable(value = "errorCacheManager")
    public ErrorEntity get(String name) {
        //省略读取过程，直接返回结果
        System.out.println("reading from disk");
        return new ErrorEntity(1,"成功");
    }
}
```

修改 Spring 的配置文件

配置的话，在 spring.xml 添加对 Manager 类的实例化，以及把该类加入缓存类管理。

```
<bean id="errorCacheManager" class="com.ixiaozhi.ErrorCacheManager"/>
<cache:annotation-driven/>
<bean id="cacheManager" class="org.springframework.cache.support.SimpleCacheManager
```

```
>
  <property name="caches">
    <set>
      <bean class="org.springframework.cache.concurrent.ConcurrentMapCacheFactoryBea
" p:name="default"/>
      <bean class="org.springframework.cache.concurrent.ConcurrentMapCacheFactoryBea
"
          p:name="errorCacheManager"/>
    </set>
  </property>
</bean>
```

测试

接下来，我们可以写测试类了，这里使用 JUnit。

```
@Test
public test() {
    ApplicationContext ctx = new ClassPathXmlApplicationContext("spring.xml");
    ErrorCacheManager cache = (ErrorCacheManager) ctx.getBean("errorCacheManager");
    cache.get("test");
    cache.get("test");
    cache.get("test");
    cache.get("test");
}
```

运行上面测试代码，只会输出一次

reading from disk，其他都是从 Spring Cache 里取得。

后篇：[Spring Cache 使用 GuavaCache 进行管理](#)