

短域名服务

本项目的目的是提供将长域名转短域名，项目要求如下：

撰写两个 API 接口：

- 短域名存储接口：接受长域名信息，返回短域名信息
- 短域名读取接口：接受短域名信息，返回长域名信息。

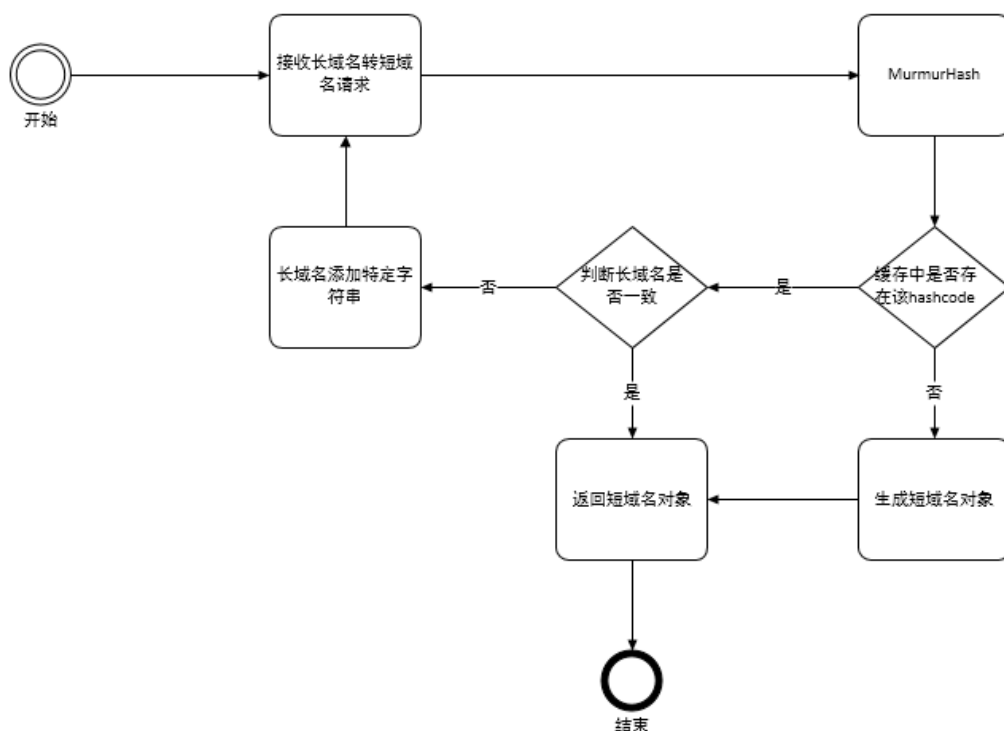
限制：

- 短域名长度最大为 8 个字符
- 采用SpringBoot，集成Swagger API文档；
- JUnit编写单元测试, 使用Jacoco生成测试报告(测试报告提交截图即刻)；
- 映射数据存储在JVM内存即可，防止内存溢出；

短域名服务实现过程如下：

通过长域名生成短域名

1. 通过murmurhash，计算长域名的hashcode，返回8位hashcode。
2. 查询缓存中是否存在该hashcode对应的短域名对象：
 - 2.1 存在短域名对象，再比较长域名是否一致，如果一致，直接返回；如果不一致，说明hashcode重复，在长域名后增加特定字符串，重新执行第一步。
 - 2.2 不存在短域名对象，生成新的短域名，并将对象保存到spring cache中，为了防止数据过多导致内存泄漏，新增了CachingConfig.reportCacheEvict()，定时刷新缓存，目前暂定的是10分钟为失效时间。可以根据实际生产业务情况再调整该时间。
3. 长域名生成短域名的业务流程图如下：

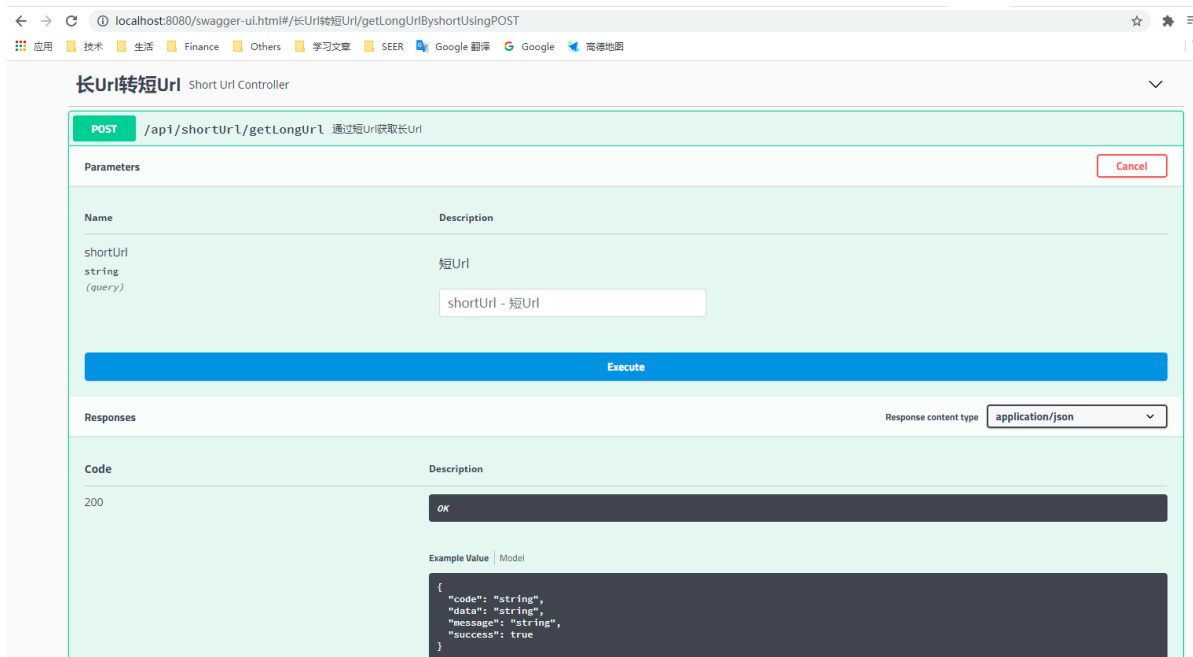


通过短域名查询长域名

1. 截取短域名的hashCode，查询缓存
2. 如果缓存返回对应的对象，替换长链接里可能存在的特殊字符后，返回短域名对象。
3. 如果缓存找不到对象，返回404，提示找不到对应的长域名或长域名已经失效。

测试案例

Swagger测试截图如下：



swagger测试了两个接口，并测试了输入参数为空时，系统会自动返回错误code和提示信息，见下面截图



业务逻辑测试类：ShortUrlConvertServiceTests

案例1：testLong2Short()，通过长域名返回短域名

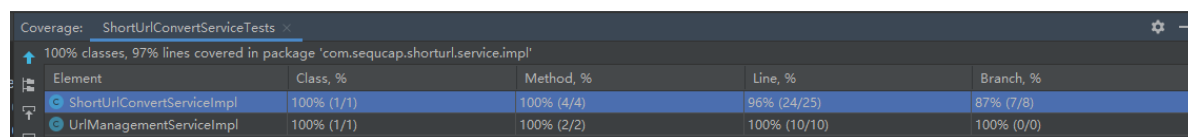
案例2：testLong2ShortRepeatHashCode()，当长域名的hashCode重复时，模拟hashCode重复的情况，在长域名后添加指定字符串，重新生成新的hashCode，并返回对应的短域名；

案例3：testGetLongUrlByShort()，通过短域名返回长域名，正常返回

案例4：testGetLongUrlByShortWithNullResult()，通过短域名返回长域名，找不到对应的短域名

案例5：testGetLongUrlByShortWithErrorUrl()，通过短域名返回长域名，短域名格式不正确，返回错误代码和提示

jacoco的报告截图如下：



Coverage: ShortUrlConvertServiceTests				
100% classes, 97% lines covered in package 'com.sequicap.shorturl.service.impl'				
Element	Class, %	Method, %	Line, %	Branch, %
ShortUrlConvertServiceImpl	100% (1/1)	100% (4/4)	96% (24/25)	87% (7/8)
UriManagementServiceImpl	100% (1/1)	100% (2/2)	100% (10/10)	100% (0/0)

压力测试类：ShortUrlConvertServiceThreadTests

这里直接模拟多线程并发调用ShortUrlConvertService，由于受限于本地开发笔记本的性能问题（CPU: Intel core(R) i5-5200, 内存：12G），每新生成100个子线程，程序会暂停100毫秒，然后继续。

压力测试的方案为循环多次（ $\text{int } i < 100000$ ）调用ShortUrlConvertService，当 $i \% 3 == 0$ 时，返回同样的长域名，模拟域名调用重复，直接返回缓存里的短域名对象。

目前测试的循环次数分别为10万次，50万次，100万次，目前3个批次的测试，在本地开发笔记本上均顺利完成，完成时间如下：

10万次，耗时121秒

50万次，耗时540秒

100万次，耗时1098秒