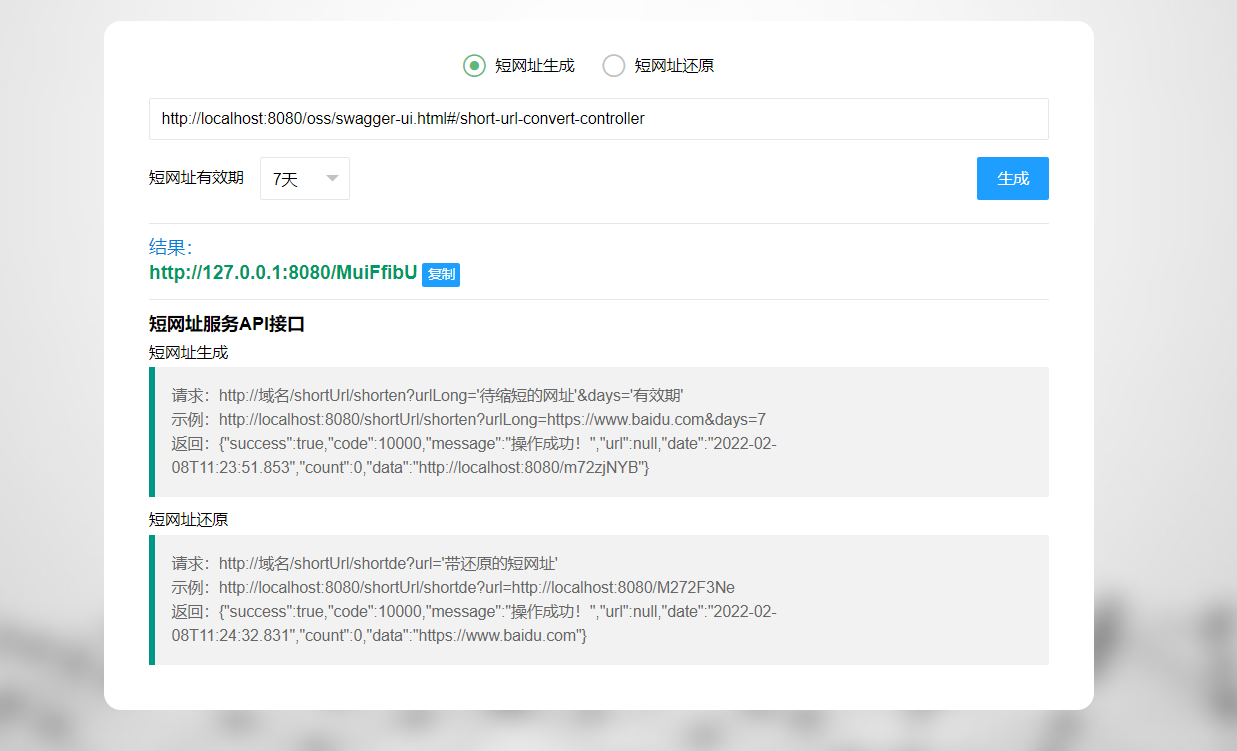
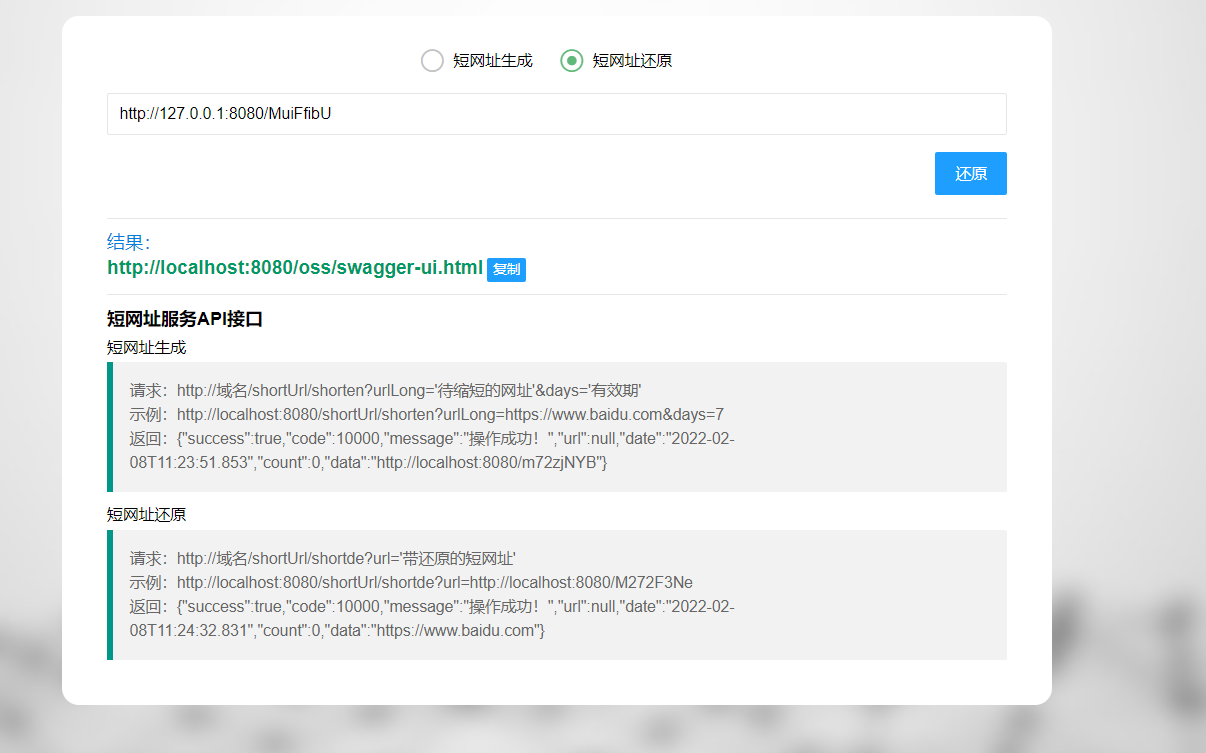
**整个交互流程如下：**

**启动项目，访问**[**http://localhost:8080/**](http://localhost:8080/)



1. 用户输入长链接，返回短链接



2. 用户输入短链接，返回长连接

3. 用户访问短链接，浏览器重新向



**设计要点：**

—**短链接生成算法**

1）对传入网址进行MD5加密,把加密字符按照8位一组,16进制与0x3FFFFFFF进行位与运算,把得到的值与 0x0000003D 进行位与运算，取得字符数组chars,索引把取得的字符相加

每次循环按位右移3位把字符串存入对应索引的输出数组

2）每次输出4组短链接，如果生成的短链接在redis中不存在，则以短链接为key,长链接为value,存入redis,如果生成的短链接在redis中存在，则根据短链接为key,从redis中获取对应的value值长连接，比较与新传入的网址是否一致，如果一致则覆盖，如果不一致，则跳过取出第二组短连接比较，直到不重复为止。

—**重定向过程**

生成短链接之后，需要存储短链接到长链接的映射关系，即短链接 -> 长链接，浏览器访问短链接服务器时，获取用户域名后面传入的短链接，从redis上查询到长连接，然后进行重定向。

**优化方案：**

—**算法优化**

短链接每次生成了4组，如果需要转换的链接非常多，可能4组以后还有重复的短链接，则需要将生成算法进一步优化，参考布隆过滤器。同时也能压缩数据大小

—**可伸缩和高可用**

如果将短链接生成服务单机部署，缺点一是性能不足，不足以承受海量的并发访问，二是成为系统单点，如果这台机器宕机则整套服务不可 用，为了解决这个问题，可以将系统集群化。