**Domain短域名服务设计文档**

## 一、假设

1、假设了服务的地址域名为<https://d.cn/>

对外提供服务的服务器都应有一个对应的域名，要求中只给出了8位短域名的要求，为了更接近真实的服务，所以，假设了服务的地址。

2、假设了短域名过期时间为5分钟

为了避免内存溢出，设计了定时任务清除过期的域名，为了能够验证服务，所以假设了过期时间为5分钟，定时任务每次清除与当前时间差大于5分钟的短域名映射关系。

## 二、设计思路

#### 0、整体思路

1）、设计一个递增的发号器， 每次接口请求通过数组下标的递增（超过数组最大值则循环）从而获取不同的字符串。

2）、在存储方面，K/V结构存储能够实现快速查找，考虑线程安全，使用ConcurrentHashMap存储映射关系。

3）、为避免内存耗尽，对映射关系设置过期时间，并通过定时任务进行清除。

#### 1、8位短域名字符的生成

用8个线程安全的计数器分别记录数组8个下标值（经过求余计算），为适应能集群部署，第一个下标设计为“机器编号”（短域名使用字母和数字，共62个字符，所以机器编号取值应0~61），其余7位，每次接口调用，通过随机的方式，判定哪个位置上的下标加1，这样每次调用都会有个下标取值不同，避免重复。

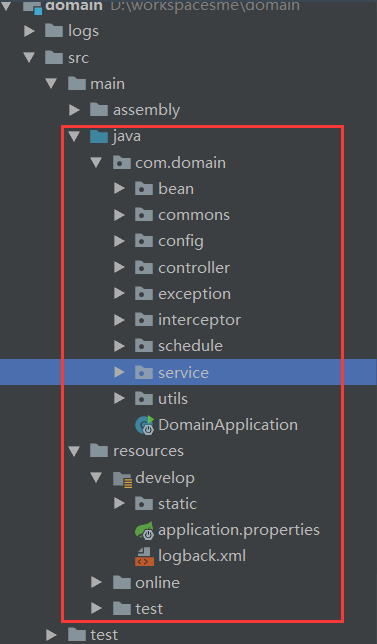
2、数据存储

由于数据直接在JVM中缓存，考虑查询性能，KV类型数据结构最合适，加上线程安全问题ConcurrentHashMap用于存储数据最为合适。

3、避免内存溢出

避免内存溢出，可以从两个方面进行考虑，一个是限制数据存储的数量，一个是适当的时候对数据进行清除。限制数量，比较容易实现，但是数量的多少无法准确预估仍然需要容错的方案；所以，本项目中使用了数据清除的方案，设计定时任务，对过期的数据进行清除，避免内存溢出。

## 三、项目目录说明



**Java目录下包说明**

bean包：封装request请求或者用于参数传递的Bean。

commons包：一些公共的类。

config包：系统配置类均在这个包下，Swagger配置、MVC配置、全局异常配置、自定义配置。

controller包：对外接口。

exception包：自定义异常。

interceptor包：拦截器，本项目中配置了用于接口请求日志打印的拦截器。

schedule包：定时任务。

utils包：共用的工具类。

**resource下文件说明**

develop为开发环境配置

online为生产环境配置

test为测试环境配置

static文件夹为404页面和网站favicon.ico图标。