Spring参考博文

1. Spring-Redis

1.1 集成

- Implementation of Redis in a Microservice/Spring Boot Application

1.2 Request缓存

- Spring Boot Redis: Ultimate Guide to Redis Cache with Spring Boot 2

> 博文内容：
>
> 1. Setting up the Redis Cache on your machine Writing a Spring Boot Application
> 2. Use Spring's Integrated @Cacheable Annotation to cache results of method invocations using Spring Data Redis
> 3. Gain more fine granular control by using the other available Annotations
> 4. Create dynamic CacheKeys: Cache depending on the input parameters of our methods
> 5. Define after what time (TTL=time-to-live) our cached Entries are not valid anymore Define different TTLs

2. lombok

2.1 注解

- @NoArgsConstructor, @RequiredArgsConstructor, @AllArgsConstructor

3. Response

3.1 rest-response类型解析

枚举/日期 date enum json 解析类型 返回数字 或者自定义文字

4. 日志

细说Java主流日志工具库 Java常用的日志框架对比 Log4J 2 Configuration: Using YAML

5. CORS

SpringBoot 实现前后端分离的跨域访问（CORS）

> 我们都知道浏览器的同源策略，就是出于安全考虑，浏览器会限制从脚本发起的跨域HTTP请求，像XMLHttpRequest和Fetch都遵循同源策略。 浏览器限制跨域请求一般有两种方式：
>
> 1. 浏览器限制发起跨域请求
> 2. 跨域请求可以正常发起，但是返回的结果被浏览器拦截了
>
> 一般浏览器都是第二种方式限制跨域请求，那就是说请求已到达服务器，并有可能对数据库里的数据进行了操作，但是返回的结果被浏览器拦截了，那么我们就获取不到返回结果，这是一次失败的请求，但是可能对数据库里的数据产生了影响。
>
> 为了防止这种情况的发生，规范要求，对这种可能对服务器数据产生副作用的HTTP请求方法，浏览器必须先使用OPTIONS方法发起一个预检请求，从而获知服务器是否允许该跨域请求：如果允许，就发送带数据的真实请求；如果不允许，则阻止发送带数据的真实请求。

> 在浏览器中，script, img, iframe, link等标签都可以加载跨域资源，而不受同源限制

同源策略 多级域名下cookie共享

> 设置cookie的domain:
>
> ```
> document.cookie='key=value;domain=xxx.com'
> ```

浏览器同源政策及其规避方法

6. HTTP状态码

http状态码204理解

> HTTP的状态码有很多种,主要有1xx（临时响应）、2xx（成功）、3xx（已重定向）、4xx（请求错误）以及5xx（服务器错误）五个大类 HTTP状态码 201 304 404 500等代表的含义

7. controller

spring-mvc自定义数据绑定