

并发的压力在与数据库，为了减少对数据库的访问，应当采取的策略是缓存

1、页面缓存

将页面源代码存入到缓存中

调用to_list 这个接口时首先不渲染再返回，先取缓存中的页面，有就返回没有再进行渲染，并且要存入redis中以便下次使用。

```
@RequestMapping(value = "to_list", produces = "text/html")
@ResponseBody
public String toLogin(Model model, MiaoshaUser user, HttpServletResponse response, HttpServletRequest request) {
    // 取缓存
    String html = redisService.get(GoodsKey.getGoodsList, key: "", String.class);
    if (!StringUtils.isEmpty(html)) {
        return html;
    }

    model.addAttribute("user", user);

    // 查询商品列表
    List<GoodsVo> goodsList = goodsService.listGoodsVo();

    model.addAttribute("goodsList", goodsList);
    // 手动渲染
    SpringWebContext springWebContext = new SpringWebContext(request, response, request.getServletContext(),
        request.getLocale(), model.asMap(), applicationContext);
    html = thymeleafViewResolver.getTemplateEngine().process(templateName: "goods_list", springWebContext);
    if (StringUtils.isEmpty(html)) {
        redisService.set(GoodsKey.getGoodsList, key: "", html);
    }
    return html;
}
```

Run Dashboard
Multiple Spring Boot run configur

缓存保留时间为60秒

```
public class GoodsKey extends BasePrefix {
    private GoodsKey(int expireSeconds, String prefix) { super(expireSeconds, prefix); }
    public static GoodsKey getGoodsList = new GoodsKey(60, "gl");
    public static GoodsKey getGoodsDetail = new GoodsKey(60, "gd");
}
```

优化后1778

原先是896

2、对象缓存

将对象缓存到redis中这样就无须取查数据库，减轻数据库的压力。

3、页面静态化

即前后端分离

前端页面都是html 通过ajax来调用后端的接口实现页面的渲染，后端需要返回的是json格式的数据即可。

#static

spring.resources.add-mappings=true

spring.resources.cache-period= 3600

spring.resources.chain.cache=true

spring.resources.chain.enabled=true

spring.resources.chain.gzippped=true

spring.resources.chain.html-application-cache=true

spring.resources.static-locations=classpath:/static/

这段配置可以让客户端缓存页面3600秒，不必向服务端去发送请求返回整个页面，动态的部分通过ajax去调用接口实现。

4、静态资源的优化

1. JS/CSS压缩，减少流量

2. 多个JS/CSS组合，减少连接数

3. CDN就近访问