

简历里面常见的模块

1	商品管理
2	权限模块
3	单点登录
4	
5	
6	数据的导入导出
7	搜索模块
8	
9	
10	购物车模块
11	订单模块
12	支付模块
13	
14	
15	秒杀模块
16	
17	详情页面模块
18	首页广告
19	图片上传模块
20	
21	

模块常见的问题

1	用到了哪些技术？
2	怎么考虑技术选型？
3	涉及几张表？
4	涉及几个接口？
5	开发流程、开发步骤是怎样的？
6	哪些地方有难度或者印象深刻？
7	开发过程中有没有遇到什么问题？
8	开发了多久？

商品管理-->新增

1	用到了哪些技术？
2	5个请求
3	post-->json->Bean
4	post-->file-->Multipart-->fastDFS/7牛云
5	单张
6	多张
7	get-->分类-->3级联动

```

8         单表->自关联
9         3个表->关联查询
10        get->品牌选择-->关联查询
11        get->模板选择-->关联查询
12    存db
13        存tb_sku,tb_spu-->多表-->事务
14        编程事务
15        声明事务
16        注解
17        xml
18    id的生成
19    时间问题-->统一
20        db:dateTime
21        Bean:Date
22        前端:var
23
24
25        我们jdbc连接数据的时候
26        url: jdbc:mysql://localhost:3306/changgou_oauth?
useUnicode=true&characterEncoding=utf-
8&useSSL=false&allowMultiQueries=true&serverTimezone=GMT%2B8
27        Bean(Date)-->springboot(jackson)-->json-->前端
28        统一配置
29        spring:
30        jackson:
31            date-format: java.text.SimpleDateFormat
32            time-zone: GMT+8
33        单个配置
34        @JsonFormat(pattern = "yyyy-MM-dd HH:mm:ss", locale = "zh",
timezone="GMT+8")
35        private Date createTime;//添加日期
36
37
38    怎么考虑技术选型?
39        不涉及
40    涉及几张表?
41        数据来源-->5张表
42        数据存储-->2张表
43    涉及几个接口?
44        6个get
45        2个post
46    开发流程、开发步骤是怎样的?
47        图片上传单独完成
48        完成数据来源相关的接口-->6个
49        完成add接口-->6个
50    哪些地方有难度或者印象深刻? 开发过程中有没有遇到什么问题?
51        fastDfs图片上传过大,可能有问题
52        声明事务到底用注解/xml 有去思考
53        时区问题
54    开发了多久?
55        一周

```

商品管理-->修改

- 1 | 数据回显-->提交

商品管理-->删除

- 1 | 物理删除/逻辑删除
- 2 | 权限-->需要使用spring security进行权限的控制

商品管理-->查询

- 1 | 分页

商品管理-->审核

- 1 | 简单
- 2 | status-->0/1

权限模块

- 1 | 用到了哪些技术?
- 2 | spring security
- 3 | shiro
- 4 |
- 5 | 怎么考虑技术选型?
- 6 | spring security 对比shiro
- 7 |
- 8 | 涉及几张表?
- 9 | 5张表
- 10 | user
- 11 | role
- 12 | permission/menu
- 13 |
- 14 | 7张表-->最主流
- 15 | user
- 16 | user_role
- 17 | role
- 18 | permission
- 19 | role_permission
- 20 | menu
- 21 | role_menu
- 22 |
- 23 | 涉及几个接口?
- 24 |

25 1. 登录接口->在登录的时候会查询
26 用户-->role
27 ->权限
28
29 2. 用户所拥有的菜单
30 用户-->role
31 ->菜单
32
33 需求是怎样的?
34 1.没有登录,不能访问任何资源(url)
35 2.不用用户,看到的菜单不同-->不用用户是不同角色
36 3.不用用户,可以操作的controller/controller里面方法/controller里面方法的具体代码块
37 4.需要自定义登录界面-->spring security默认登录界面太丑了
38
39 开发流程、开发步骤是怎样的?
40 1.登录接口完成接口
41 2.用户所拥有的菜单接口
42 3.没有登录,不能访问任何资源-->spring security进行配置
43 4.自定义登录界面-->
44 1.完成界面(div+css)
45 2.spring security配置
46 1.loginPage
47 2.loginProcessingUrl
48 5.不用用户,可以操作的controller/controller里面方法/controller里面方法的具体代码块
49 1.登录的时候-->赋值权限-->需要自定义userDetail-->loadUserByUsername
50 2.应用权限
51 1.开启注解支持-->
52 @EnableGlobalMethodSecurity
53 2.使用相关注解-->
54 @PreAuthorize("hasAnyAuthority('CHECKITEM_DELETE')")
55 6.不用用户,看到的菜单不同-->只有管理系统才是
56 进入页面的时候,去加载
57 vue-->mounted()-->加载
58
59 哪些地方有难度或者印象深刻?
60 spring-security方式也多
61 spring-security-->xml
62 spring-security-->bean
63 spring-security配置项多
64 1.放行配置
65 2.拦截配置
66 3.loginPage/loginProcessingUrl
67 4.自定义认证器-->找我们自己的userDetail
68 5.告诉db里面的密码使用的什么加密方式
69 6.csrf
70 7.httpBasic
71
72 开发过程中有没有遇到什么问题?
73 管理系统使用了iframe,只能
74 <security:frame-options policy="SAMEORIGIN"/>
75
76
77 csrf
78
79 httpBasic
80
81
82 •

83 开发了多久?
84 • 2周

单点登录

```
1  用到了哪些技术?
2
3  1.sso-->效果-->方案
4  2.spring security 框架
5      认证
6      授权
7  3.oauth2.0 协议-->规范
8      authorization-->授权-->accessToken
9      1.颁发accessToken-->提到了可以有4种模式
10         密码-->oauth/token
11         授权码模式-->oauth/authorize
12         2.校验accessToken-->oauth/check_token
13         3.刷新accessToken-->oauth/token
14             refresh_token-->xxxxxx
15             grant_type-->refresh_token
16  4.jwt-->规范
17      jwt由base64(头部).base64(载荷).sign(base64(头部).base64(载荷))
18      签名算法有两种，分别是HS256和RS256。一个是对称加密，一个是非对称加密
19
20  5.加密方式
21      对称加密
22          1个密码
23          aes des HS256
24      非对称加密
25          1个密钥对
26          rsa HS256
27      非对称加密可以做3个事情
28          1.公钥加密-->私钥解密
29          2.私钥加密-->公钥解密
30          3.私钥签名-->中间人修改-->公钥校验
31
32  6.spring security oauth
33      方便基于oauth2.0协议完成授权和认证
34      开发授权服务器
35      开发资源服务器
36
37  7.第三方登录
38      qq, 微博, 微信, github-->登录我们的系统
39
40
41
42  给大家提问的方式梳理?
43  解释什么是sso?
44      password.jd.com
45      card.jd.com
46      user.jd.com
47  怎么理解spring security ?
48      WebSecurityConfigurerAdapter
49      @EnableWebSecurity
```

```

50     AuthenticationManager
51     UserDetailsService
52 怎么理解spring security oauth?
53     @EnableResourceServer
54     @EnableAuthorizationServer
55     AuthorizationServerConfigurerAdapter
56     TokenStore
57     AccessTokenConverter
58
59 怎么理解oauth?
60     authorization-->授权-->accessToken
61         1.颁发accessToken-->提到了可以有4种模式
62             密码-->oauth/token
63             授权码模式-->oauth/authorize
64         2.校验accessToken-->oauth/check_token
65         3.刷新accessToken-->oauth/token
66             refresh_token-->xxxxxx
67             grant_type-->refresh_token
68 jwt有几种签名方式?
69     签证算法有两种, 分别是HS256和RS256。一个是对称加密, 一个是非对称加密
70 谈一下你对非对称加密了解?
71     1.公钥加密-->私钥解密
72     2.私钥加密-->公钥解密
73     3.私钥签名-->中间人修改-->公钥校验===>主要是这里
74
75 如果网页/h5/app要实现三方登录, 后台开发需要做什么事情?
76     参考开放平台
77
78     https://wiki.open.qq.com/wiki/%E3%80%90%E7%99%BB%E5%BD%95%E3%80%91%E7%BD%91%E7%AB%99%E6%8E%A5%E5%85%A5
79
80     https://developers.weixin.qq.com/doc/oplatform/Website_App/WeChat_Login/Wechat_Login.html
81     1.用户在我们的网页-->点击qq登录按钮(前端)-->要么前端跳/后台跳(redirect_url)-->跳转qq登录页面-->登录完成-->拿到accessToken(qq颁发的)
82     2.回调自己服务器
83         1.走到我们的一个controller方法里面
84         2.有了accessToken
85             1.头像, 昵称, 邮箱-->对用户可见
86             2.三方平台的唯一标识-->openUserId
87                 -->静默注册了一个新用户
88                 -->找到与之对应的自己平台的userId
89
90         t_user      userid      openUserId      otherinfo
91         0001        0000000001
92
93 我们自己的系统要基于oauth2.0协议去完成授权(颁发token)用什么授权模式?
94     密码模式
95     简单
96     自己的系统, 自己的user, 自己user对应的password
97
98 spring security oauth可以对accessToken持久化, 持久化方式有哪些, 为什么我们用JwtTokenStore?
99     redis-->持久化到服务器了
100    jdbc-->持久化到服务器了
101    jwt-->持久化道我们的浏览器客户端

```

102 spring security oauth为什么对token的签名使用非对称加密方式?
103 校验方便
104 怎么校验?
105 框架校验spring security oauth
106 怎么方便
107 1.放入public.key
108 2.给spring security oauth传入public.key
109 @Bean
110 public JwtAccessTokenConverter jwtAccessTokenConverter() {
111 JwtAccessTokenConverter converter = new
112 JwtAccessTokenConverter();
113 converter.setVerifierKey(getPubKey());
114 return converter;
115 }
116 安全性
117 非对称加密,安全性高于对称加密
118 为什么生成token,我们要基础oauth2.0协议?
119 方便
120 关于token相关的操作规范都提供了
121 如何颁发
122 如何校验
123 如何刷新
124
125 为什么我们做sso,我们要基础oauth2.0协议?
126 因为oauth2.0有那么一套关于accessToken的颁发,校验,刷新
127
128 如果要完成qq登录,是否需要了解oauth2.0协议?
129 不怎么了解都行
130 围绕accessToken
131 什么时候让用户登录
132 网关控制
133 请求需要token的一些url的时候有可能有跳
134 订单
135 支付
136 加入购物车
137 地址列表
138 什么情况会让你跳
139 cookie/header/params没有
140 什么时候颁发token
141 登录成功之后-->基于oauth2.0协议的密码模式请求成功后,spring security oauth就会把accessToken给我们。
142 什么时候校验token,如何校验,谁去校验
143 spring-security-oauth框架校验
144 受保护的资源服务器/资源服务器校验/加了@EnableResourceServer注解微服务校验/使用了public.key的微服务校验
145
146
147 spring-security-oauth框架从哪里去取出token?
148 header
149 key:Authorization
150 value:[bearer token]
151
152
153 •
154
155 用户客户端-->网关-->微服务-->我们需要怎么处理token
156 假设用户客户端携带了正确的token-->网关过滤器(AuthorizeFilter)-->加到请求头

```
157
158     用户客户端-->网关-->微服务-->feign-->微服务-->我们需要怎么处理token
159     假设用户客户端携带了正确的token-->拦截器(MyFeignInterceptor implements
RequestInterceptor)-->加到请求头
160
161
162
163 怎么考虑技术选型?
164 sessionId+redis-->下去自己看一下
165     https://blog.csdn.net/qq_22172133/article/details/82291112
166 cas
167 jwt+拦截器
168 jwt+(spring security)+(spring security oauth)+(spring cloud gateway)
169
170
171 涉及几张表?
172 1.t_user
173 2.oauth_client_details
174
175 涉及几个接口?
176 1.登录
177 2.颁发-->登录 , 刷新 ,校验
178
179 开发流程、开发步骤是怎样的?
180     自己想下
181
182 哪些地方有难度或者印象深刻?
183     概念多,容易混
184 开发过程中有没有遇到什么问题?
185     概念多,容易混
186 开发了多久?
187     2个星期
188     1.网关-->修改
189     2.授权中心-->独立
190     3.资源服务器还需要修改
191         a1->修改
192         a2->修改
193         a3->修改
194         a4->修改
```

数据的导入导出

```
1  用到了哪些技术?
2      poi-->读写xls,pdf,word
3      org.apache.poi->xls
4      jasperReports->pdf
5      自己查->docx doc
6
7
8
9      easypoi
10
11 怎么考虑技术选型?
12      org.apache.poi->xls
```


13
14 涉及几张表?
15 看具体的报表
16 涉及几个接口?
17 导入-->批量添加
18 1. 下载一个模板
19 java实现文件下载有两者方式
20 方式一:window.location.href =
21 ".../template/ordersetting_template.xlsx";
22 方式二:res.setHeader("Content-Disposition","attachment;filename=" + filename);
23 两者区别是什么?
24 自己下去看一下
25 2. 上传数据
26 文件上传
27 1. 前端content-type
28 2. 如果用springmvc 需要配置多媒体视图解析器,如果是springboot.默认就有了
29 3. 可以通过配置,限制文件的大小
30 4. 后台通过multipartFile接收
31 5. poi读取收到的xls
32
33 导出
34 1. 查数据
35 2. 根据模板,写入内容
36 3. 文件的下载
37
38 开发流程、开发步骤是怎样的?
39 分导入导出自己再去分析下
40 哪些地方有难度或者印象深刻?
41 1. 文件的上传下载
42 2. xls读取以及写入
43 1. 读取的数据量很大-->可能有问题
44 2. 写入的数据量很大-->100w记录-->xls--
45 ><https://www.cnblogs.com/swordfall/p/8298386.html>
46 开发过程中有没有遇到什么问题?
47 cell-->文本-->782300-->poi读出来的值是-->7.823E5
48 开发了多久?
49 一周

搜索模块

1 用到了哪些技术?
2 es
3 怎么考虑技术选型?
4 es
5 涉及几张表?
6 一张
7 涉及几个接口?
8 2个
9 开发流程、开发步骤是怎样的?
10 参考面试宝带你
11 哪些地方有难度或者印象深刻?

12 1.es建立索引的时候,映射建立需要思考
13 1.哪些分词
14 2.哪些不分词
15 3.规格怎么存到es
16 开发过程中有没有遇到什么问题?
17 skuMapper.saveAll(skuInfos);
18 skuInfos东西多
19 会不会内存溢出
20 会不会太慢
21 spring-data-es.jar
22 如何导入大量数据
23 查询不够快?
24 官方:query和filter,query比较慢,filter比较快-->不是很理解
25
26
27 我们能感受到的
28 1.query-->filter+filter+filter-->快
29 2.query-->query+query+query-->快
30
31 最基本结论--->dsl结构可能引入效率不一样
32
33 开发了多久?
34 一周
35
36 #购物车模块
37 用到了哪些技术?
38 redis
39 hash
40 key:cart_userId
41 value:
42 key:skuId
43 value:OrderItem
44 feign的调用
45 hytrix-->fallback
46 Ribbon
47
48
49 怎么考虑技术选型?
50 cookie+redis--><https://blog.csdn.net/QuietHRH/article/details/83692332>
51 redis-->选择
52 涉及几张表?
53 没表
54 涉及几个接口?
55 crud
56 开发流程、开发步骤是怎样的?
57 分别完成crud
58 哪些地方有难度或者印象深刻?
59 1.购物车的商品数量,我们需要简单的检验
60 开发过程中有没有遇到什么问题?
61 开发了多久?
62
63 #订单模块
64 用到了哪些技术?
65 雪花算法生成orderId
66 时间统一问题
67 分布式事务
68 Feign
69 Redis

```

70  怎么考虑技术选型?
71      暂无
72  涉及几张表?
73      插入了2个表
74          t_order
75          t_orderItem
76      修改了2个表
77          t_spu
78          t_user
79  涉及几个接口?
80      1.order/add
81      2.2个Feign接口
82          skuFeign.decCount(orderItem.getSkuId(),orderItem.getNum());
83          userFeign.addPoints(order.getUsername(),10);
84      3.收件人信息拉取
85      4.优惠券拉取
86  开发流程、开发步骤是怎样的?
87      1.先开发feign接口
88      2.在开发order/add
89  哪些地方有难度或者印象深刻?
90      1.分布式事务
91      2.避免前端模拟请求修改价格
92          1.价格应该交给后台计算
93  开发过程中有没有遇到什么问题?
94      1.库存数量问题
95  开发了多久?
96      3天

```

支付模块

```

1  用到了哪些技术?
2      微信支付
3      支付宝支付
4      Ping++ 聚合支付
5      二维码生成
6  怎么考虑技术选型?
7      单个集成
8  涉及几张表?
9      t_pay_way-->可以前端写死
10  涉及几个接口?
11      1.拉取支付列表-->可以前端写死
12      2.支付接口/去支付/立即支付
13          http://localhost:18092/weixin/pay/create/native
14              out_trade_no:订单号
15              total_fee:价格
16              from:1 普通 2秒杀
17              payType: 1 微信 2 支付宝 3其他网银
18      3.不断查询支付状态-->课程里面没有体现 ==>主动查询
19          交给前端轮询
20          http://localhost:18092/weixin/pay/status/query
21              orderId:
22      4.接口支付结果的接口 ==>被动通知
23          http://localhost:18092/weixin/pay/notify/url
24

```

25 什么时候微信就不通知了呢?

26 5s以内收到了商户服务器的回复报文报文

27

28

29 开发流程、开发步骤是怎样的?

30 1. 微信开放平台 ==>2,3天

31 注册-->提交资料-->审核 ==>可能不是开发做

32 结果==>会拿到一些东西

33 appId:

34 partner:

35 partnerkey:

36 2. 开发一个一个的接口

37

38 哪些地方有难度或者印象深刻?

39 1. 支付流程有点复杂-->https://pay.weixin.qq.com/wiki/doc/api/native.php?chapter=6_5

40 2. 会考虑的多一些

41 1. 和钱相关-->是否安全的问题

42 1. 微信如何保证我们服务器传递过去的的数据没有被修改

43 利用md5数字摘要算法

44 3. 统一下单接口里面需要传递的参数比较多

45 有10个必传的参数

46 appid-->不用思考

47 mch_id-->不用思考

48 nonce_str-->不用思考

49 sign-->不用思考

50 sign_type-->不用思考

51 body-->写死的

52 out_trade_no-->雪花id

53 total_fee-->动态的

54 spbill_create_ip-->固定的

55 notify_url-->写死的

56 trade_type-->固定的

57 假如支付服务宕机,没有收到支付notify,如何处理此类订单?

58 什么意思呢?

59 微信不保证通知最终能成功(通知频率为
15s/15s/30s/3m/10m/20m/30m/30m/60m/3h/3h/3h/6h/6h - 总计 24h4m)

60 官方方案-->https://pay.weixin.qq.com/wiki/doc/api/native.php?chapter=23_9&index=1

61 1. 类似每隔5秒/30秒/1分钟/3分钟/5分钟/10分钟/30分钟

62 2. 定时任务每隔30秒启动一次,找出最近10分钟内创建并且未支付的订单

63

64 开发过程中有没有遇到什么问题?

65 没问题

66 开发了多久?

67 1,2周

68

秒杀模块

1 用到了哪些技术?

2 redis

3 存hash

4 存list

```
5         increment
6         redlock
7 怎么考虑技术选型?
8         redis mq实现
9 涉及几张表?
10        1.mysql-->秒杀商品
11        2.mysql-->order
12 涉及几个接口?
13        1.秒杀时间段-->简单
14        2.秒杀列表-->简单,从redis里面查询指定时间的商品列表
15        3.详情-->简单-->从redis里面查询指定时间,指定商品id的详情信息
16        4.排队-->重点
17        5.查询抢单状态-->简单-->查redis
18        6.生成支付二维码-->简单
19 开发流程、开发步骤,开发思路是怎样的?
20        开发一个一个的接口
21        开发思路是怎样的?
22            1.如何提高并发
23                数据流转都在redis里面进行
24
25
26                UserQueueCount-->hash->increment()->为了防止超买
27
28                SeckillOrderQueue-->list-->leftPush(),rightPop()->保证公平性
29
30
31        •
32        •
33
34                SeckillGoods_20200831-->秒杀商品-->提高并发
35
36                UserQueueStatus-->hash-->提高并发-->记录用户针对秒杀商品状态(1:排队
37                中, 2:秒杀等待支付,3:支付超时, 4:秒杀失败,5:支付完成)
38
39                SeckillOrder-->秒杀订单-->提高并发--->在未支付之前放到redis,支付之后会
40                从redis同步到mysql,因为支付之后的并发就不会太大
41        •
42
43                下单的时候时候多线程下单
44                消耗redis的list队列快一些(rightPop)
45        2.如何防止超买
46            什么是超买?
47            1个用户,即使发送1w次请求,也只能买一个
48            如何实现
49            使用了redis里面的increment方法
50            redis的计数器功能
51            为什么这么强?-->https://www.cnblogs.com/jian-ge/p/10097244.html
52        3.如何防止超卖
53            1w个商品,不能出现卖出去1w+
54            如何实现?
55            redlock
56            为什么用?
57            redis get和put操作必须是一个完整的整体,不然就有并发问题
58            多线程下的redis的get和put操作有并发问题
59            怎么用?
60            //加锁
```

```
61         RLock mylock = redissonClient.getLock("Mylock");
62         try {
63             mylock.lock(20, TimeUnit.SECONDS);
64             dercount(time, id);//已经减库存了
65         } catch (Exception e) {
66             e.printStackTrace();
67             //报错返回
68             return false;
69         } finally {
70             //释放锁
71             mylock.unlock();
72         }
73     4.保证公平性
74         先排队成功了,优先生成订单
75         如何实现
76             redis->list->leftPush,rightPop
77     5.结果
78         jmeter
79             1s中以内5000个请求,1000个商品,请求失败率23%,超买超卖都没问题
80             1s中以内2000个请求,1000个商品,请求失败率0%,超买超卖都没问题
81
82             0s中以内1000个请求,500个商品,请求失败率0%-->没有tomcat,没有redis集群
83 的情况
84
85             0s中以内3000个请求,500个商品,请求失败率0%-->没有tomcat,没有redis集群
86 的情况
87
88 哪些地方有难度或者印象深刻?
89     1.redis存了5个key
90     2.用到了redis 计数器(increment),redLock
91     3.为什么get和put操作会有并发问题有点难理解
92 开发过程中有没有遇到什么问题?
93     没啥问题
94     开发了多久?
95     一周
```