

实现课程订单页面

目录

目标任务	2
相关知识点	2
思路分析	2
● 前端开发思路分析	2
● 后端 API	3
● 订单列表的展示	3
● 订单状态类别的切换查询	4
编码步骤	4
● 课程列表的展示实现	4
● 订单状态类别的切换查询	错误!未定义书签。
源码	10
回马枪总结	11

目标任务

- 使用 Vue 的条件渲染、列表绑定、过滤器、Class 与 Style 绑定知识实现订单列表的展示
- 使用 Element 模板、条件渲染、事件处理等知识实现订单状态类别的切换查询和 Tabs 的切换

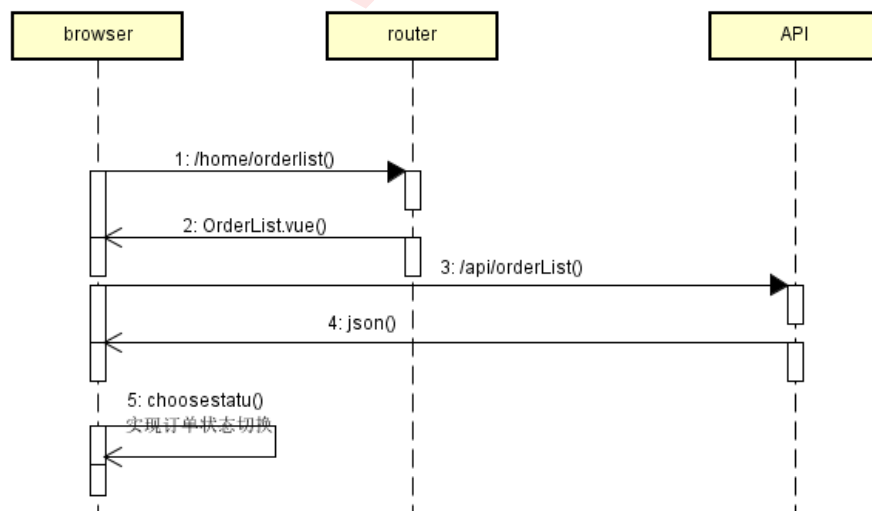
相关知识点

- Element 模板样式中按钮的使用
- Vue 的条件渲染
- Vue 的列表渲染
- Vue 的 Class 绑定
- Vue 的事件处理
- Vue 的过滤器使用
- Vue 的 router 的使用

思路分析

- 前端开发思路分析

浏览器发送 “/home/orderlist” 请求后，Vue 的路由接受 并处理返回 OrderList.vue.vue 组件显示，然后异步发送 “/api/orderList” 请求到后台查询结果，返回订单数据到浏览器显示



● 后端 API

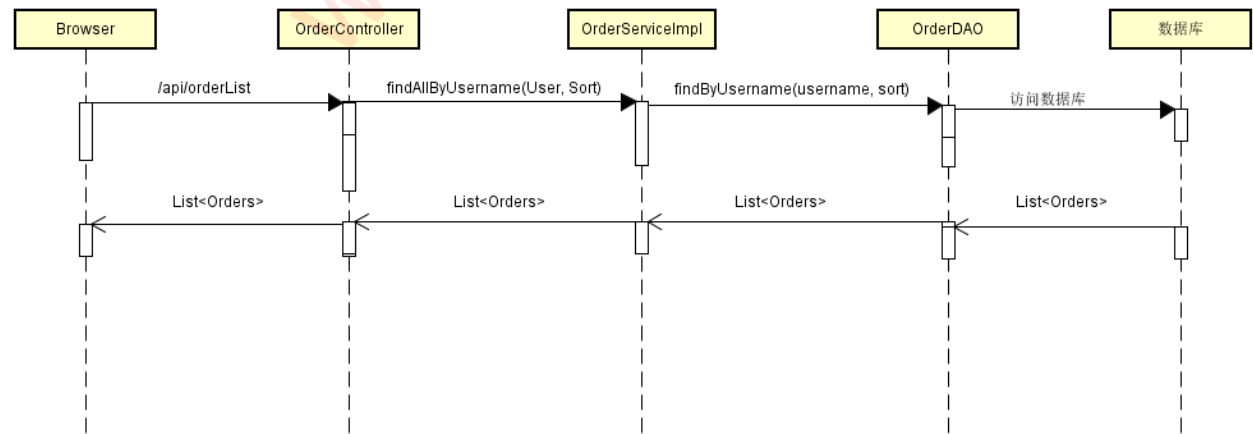
1.Get: /api/orderList

逻辑：查询全部订单

参数：无

返回值：

参数名	类型	是否必须	说明
orderId	Integer	Y	
courseName	String	Y	
courseNo	String	Y	
coursePic	Integer	N	
coursePrice	Integer	N	
creatBy	String	N	
createDate	String	N	
payDate	String	N	
payPrice	Integer	N	
payStatus	String	Y	
payment	String	N	
updateBy	String	N	
updateDate	String	N	
userName	String	Y	



● 订单列表的展示

Step1. 进入页面，展示基本元素，tabs 默认选中“全部”

Step2. 异步请求数据，获取用户全部订单信息

Step 3. 渲染数据或者展示无订单信息

- 订单状态类别的切换查询

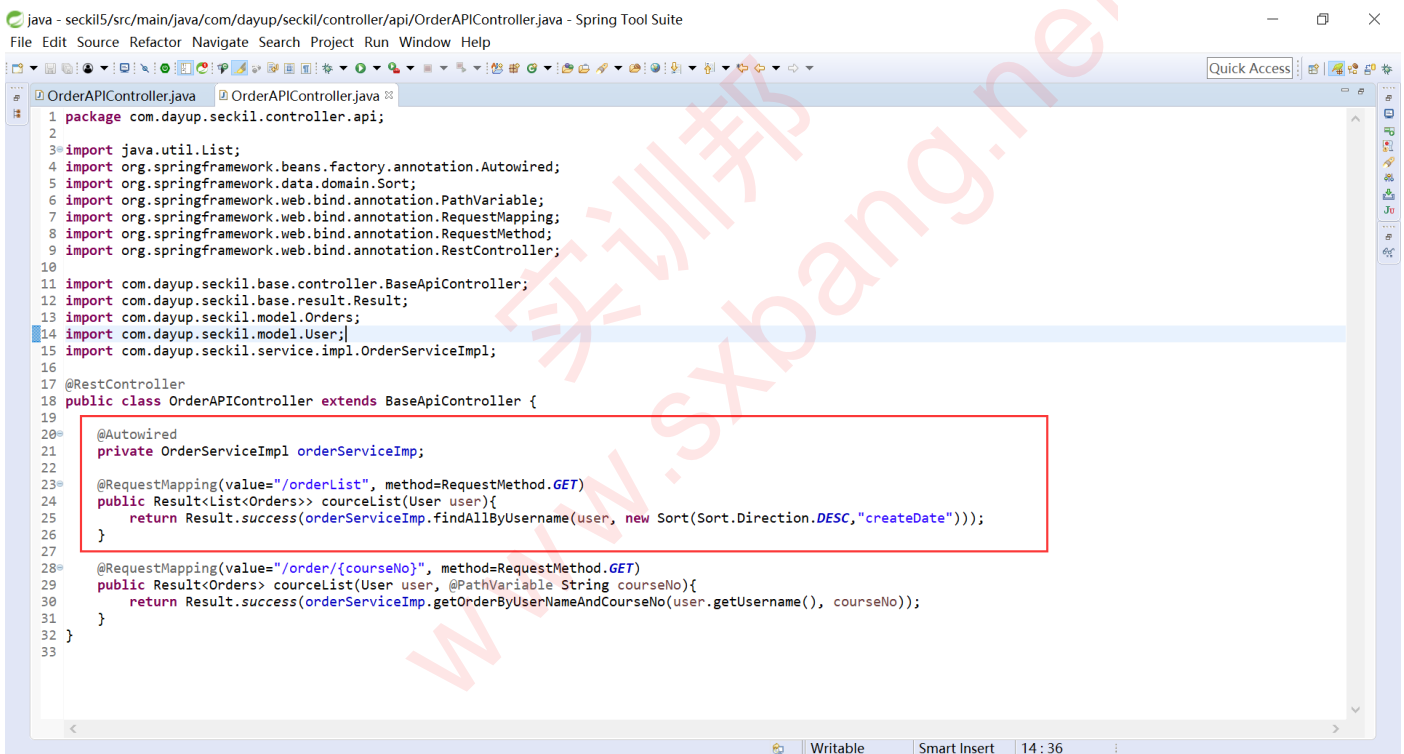
Step 1. 切换 tabs 后只显示对应状态的订单

Step 2. 切换 tabs 后样式同时做按钮样式切换

编码步骤

- 后端开发-订单列表

编写 OrderController



```
java - seckil5/src/main/java/com/dayup/seckil/controller/api/OrderApiController.java - Spring Tool Suite
File Edit Source Refactor Navigate Search Project Run Window Help

OrderApiController.java OrderApiController.java
1 package com.dayup.seckil.controller.api;
2
3 import java.util.List;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.data.domain.Sort;
6 import org.springframework.web.bind.annotation.PathVariable;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RequestMethod;
9 import org.springframework.web.bind.annotation.RestController;
10
11 import com.dayup.seckil.base.controller.BaseApiController;
12 import com.dayup.seckil.base.result.Result;
13 import com.dayup.seckil.model.Orders;
14 import com.dayup.seckil.model.User;
15 import com.dayup.seckil.service.impl.OrderServiceImpl;
16
17 @RestController
18 public class OrderApiController extends BaseApiController {
19
20     @Autowired
21     private OrderServiceImpl orderServiceImp;
22
23     @RequestMapping(value="/orderList", method=RequestMethod.GET)
24     public Result<List<Orders>> courcelist(User user){
25         return Result.success(orderServiceImp.findAllByUsername(user, new Sort(Sort.Direction.DESC,"createDate")));
26     }
27
28     @RequestMapping(value="/order/{courseNo}", method=RequestMethod.GET)
29     public Result<Orders> courcelist(User user, @PathVariable String courseNo){
30         return Result.success(orderServiceImp.getOrderByUserNameAndCourseNo(user.getUsername(), courseNo));
31     }
32 }
33
```

编写 OrderService

java - seckil5/src/main/java/com/dayup/seckil/service/impl/OrderServiceImpl.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

```
1 package com.dayup.seckil.service.impl;
2
3 import java.util.List;
4
5 @Service
6 @Transactional
7 public class OrderServiceImpl implements IOrderService{
8
9     @Autowired
10    private OrderRepository orderDAO;
11
12
13    public Orders getOrderByNameAndCourseNo(String username, String courseNo) {
14        return orderDAO.findByUsernameAndCourseNo(username, courseNo);
15    }
16
17    public Orders saveAndFlush(Orders orders) {
18        return orderDAO.saveAndFlush(orders);
19    }
20
21    public List<Orders> findAllByUsername(User user, Sort sort) {
22        return orderDAO.findByUsername(user.getUsername(), sort);
23    }
24 }
```

Quick Access

Writable Smart Insert 19:38

编写 OrderRepository

java - seckil5/src/main/java/com/dayup/seckil/repository/OrderRepository.java - Spring Tool Suite

File Edit Source Refactor Navigate Search Project Run Window Help

```
1 package com.dayup.seckil.repository;
2
3 import java.util.List;
4
5 @Repository
6 public interface OrderRepository extends JpaRepository<Orders, String>{
7
8     public Orders findByUsernameAndCourseNo(String username, String courseNo);
9
10    public List<Orders> findByUsername(String username, Sort sort);
11 }
```

Quick Access

Writable Smart Insert 14:39

● 课程列表的展示实现

1. 页面效果展示

我的订单

全部

未支付

已完成

已失效

Q 搜索关键字

订单编号: 201800003 2018-04-28 13:11:29



JAVA 开发

¥ 0

已完成

订单编号: 201800002 2018-04-27 13:11:26



JAVA 开发

¥ 0

未支付

我的订单

全部

未支付

已完成

已失效

Q 搜索关键字

订单编号: 201800002 2018-04-27 13:11:26



JAVA 开发

¥ 0

未支付

订单编号: 201800005 2018-04-26 15:44:44



JAVA WEB开发

¥ 0

未支付

订单编号: 201800001 2018-04-25 12:17:18


```
<div class="order-list-title-filter">
  <!--      -1:已失效, 0:未支付, 1: 已完成, 2:全部 -->
  <el-button class="el-button--primary"
    size="mini" round>全部</el-button>
  <el-button class="el-button--text"
    size="mini" round>未支付</el-button>
  <el-button class="el-button--text"
    size="mini" round>已完成</el-button>
  <el-button class="el-button--text"
    size="mini" round>已失效</el-button>
</div>

<div class="order-list-search" >
  <el-input placeholder="搜索关键字" prefix-icon="el-icon-search" ></el-input>
</div>
</div>

<!-- 筛选结果 -->
<div class="order-list-body">
  <div v-for="(item, index) in 6" :key='index' class="order-info-container"
    ><!-- 2 == orderStatus || item.payStatus == orderStatus ? '' : 'hide' -->
    <div class="order-nubmer">
      <span>订单编号:&nbsp;&nbsp;   xxxx</span>
      <span class="order-datetime"><time>2018-06-01 11:11</time></span>
    </div>

    <div class="order-info">
      
      <!--  -->

      <div class="course-info">
        <span class="course-info-name">课程名</span>
        <span class="course-info-price" > ¥2000 </span>
      </div>
      <div class="order-pay-info">
        <div >
          <span>已完成</span><br/>
          <span>支付方式</span>
          <!-- <span v-if="item.payment == 0 ">支付宝支付</span>
          <span v-else-if="item.payment == 1">微信支付</span>
          <span v-else>其他支付</span> -->
        </div>
        <!-- <div v-else> -->
        <!-- <span v-if="item.payStatus == -1 ">已失效</span> -->
        <!-- <span v-else>未支付</span> <br/> -->
        <!-- <span >未支付</span> -->
        <!-- </div> -->
      </div>
    </div>
  </div>
```




订单源码.zip

回马枪总结

1 , Element 模板样式中按钮的使用 :

`<el-button type="text" size="mini" round></el-button>`,主要是利用了它的模板样式

参考链接 : <http://element.eleme.io/#/zh-CN/component/button>

2 , 使用了 Vue 的条件渲染 : `vi-if` , `v-else` , `v-else-if`

参考链接 : <https://cn.vuejs.org/v2/guide/conditional.html>

3 , 使用了 Vue 的列表渲染 : `v-for`

参考链接 : <https://cn.vuejs.org/v2/guide/list.html>

4 , Vue 的 Class 绑定 : `v-bind:class`

参考链接 : <https://cn.vuejs.org/v2/guide/class-and-style.html>

5 , Vue 的事件处理 : `v-on:click`

参考链接 : <https://cn.vuejs.org/v2/guide/events.html>

6 , Vue 的过滤器使用 : `filters`

参考链接 : <https://cn.vuejs.org/v2/guide/filters.html>

```
<span class="order-datetime"><time>{{item.createDate | date-format}}</time></span>
```

7 , Vue 的 router 的使用

参考链接 : <https://cn.vuejs.org/v2/guide/routing.html>