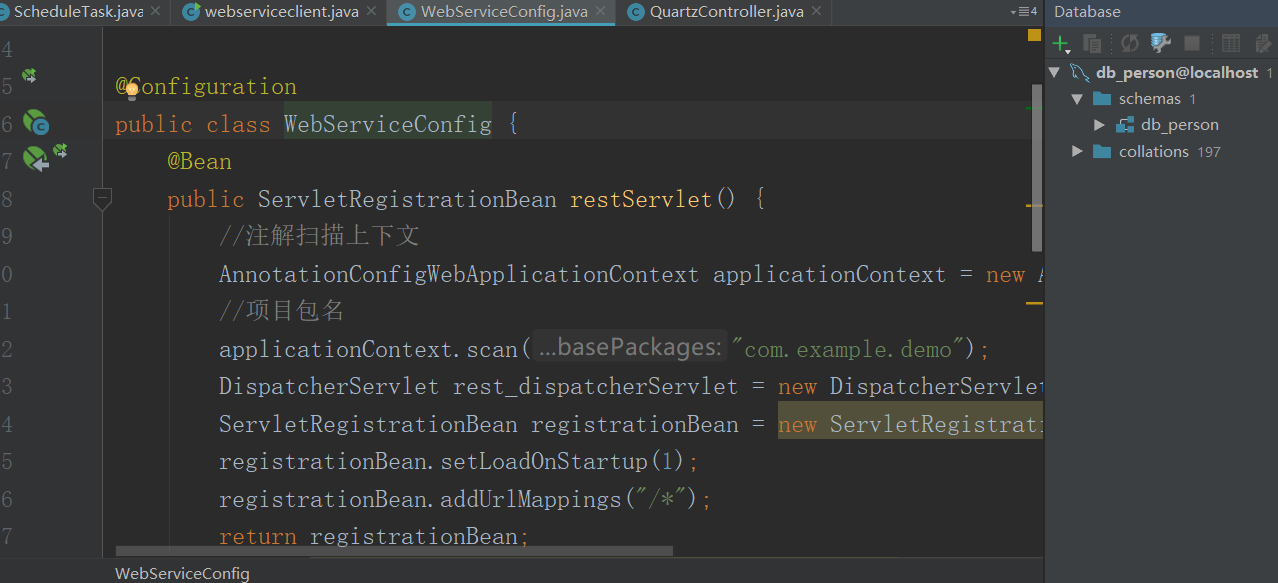
ECS相关前后端框架整合

1. spring boot+mybatis+quartz+cxf后端框架配置

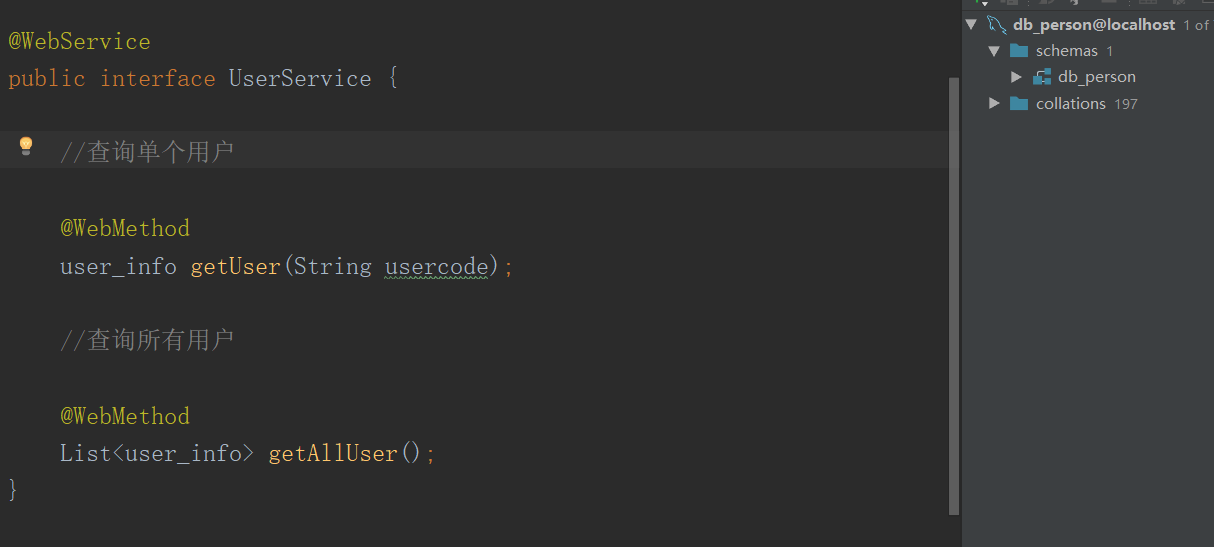
Maven配置以及mybatis生成代码

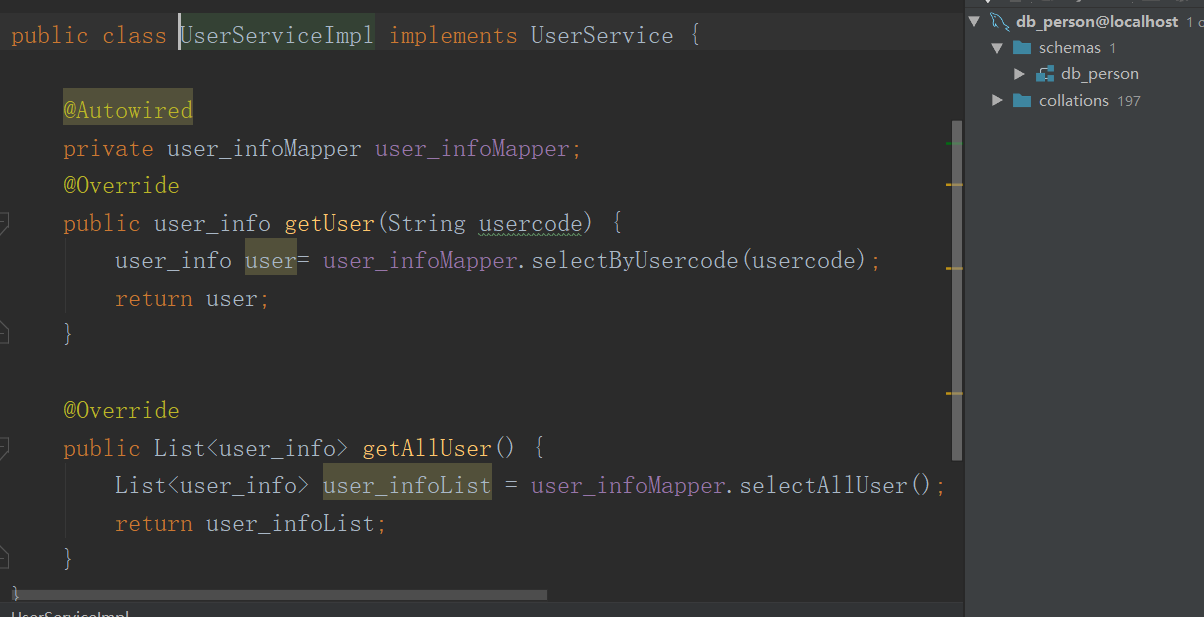
<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-jdbc</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-thymeleaf</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web-services</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.mybatis.spring.boot</groupId>  
 <artifactId>mybatis-spring-boot-starter</artifactId>  
 <version>2.1.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!--Quartz依赖-->  
 <dependency>  
 <groupId>org.quartz-scheduler</groupId>  
 <artifactId>quartz</artifactId>  
 <version>2.2.1</version>  
 <exclusions>  
 <exclusion>  
 <artifactId>slf4j-api</artifactId>  
 <groupId>org.slf4j</groupId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
 <!--WerbService CXF依赖-->  
 <dependency>  
 <groupId>org.apache.cxf</groupId>  
 <artifactId>cxf-rt-frontend-jaxws</artifactId>  
 <version>3.2.6</version>  
 </dependency>  
 <dependency>  
 <groupId>org.apache.cxf</groupId>  
 <artifactId>cxf-rt-transports-http</artifactId>  
 <version>3.2.6</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context-support</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context-support</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context-support</artifactId>  
 <version>5.0.5.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>com.vaadin.external.google</groupId>  
 <artifactId>android-json</artifactId>  
 <version>0.0.20131108.vaadin1</version>  
 <scope>compile</scope>  
 </dependency>  
 </dependencies>  
  
<build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 <!-- Mybatis-Generator插件，自动生成代码 -->  
 <plugin>  
 <groupId>org.mybatis.generator</groupId>  
 <artifactId>mybatis-generator-maven-plugin</artifactId>  
 <version>1.3.5</version>  
 <configuration>  
 <configurationFile>${project.basedir}/src/main/resources/generatorConfig.xml</configurationFile>  
 <verbose>true</verbose>  
 <overwrite>true</overwrite>  
 </configuration>  
 </plugin>  
 </plugins>  
</build>

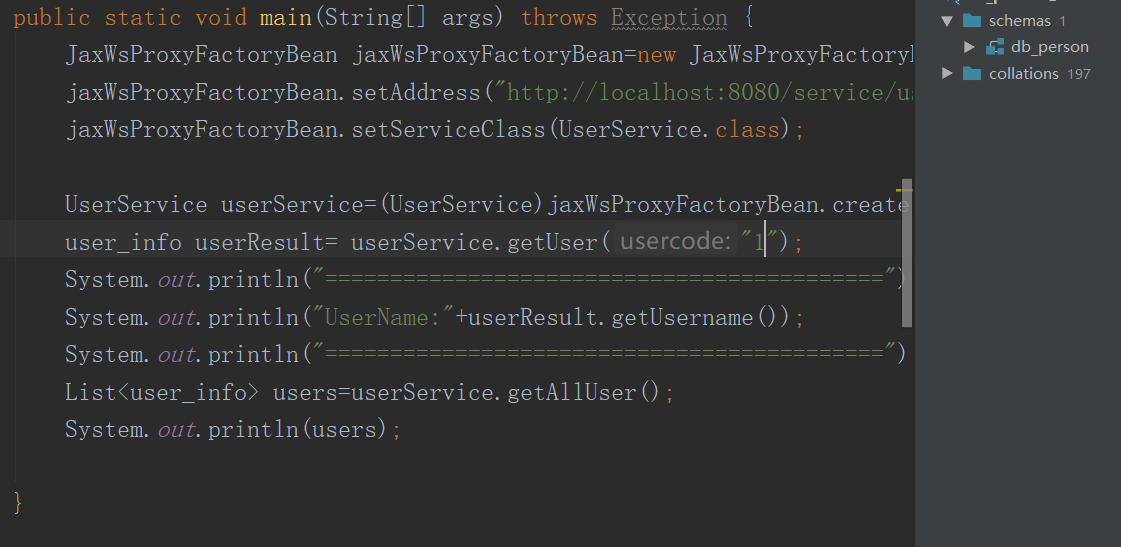
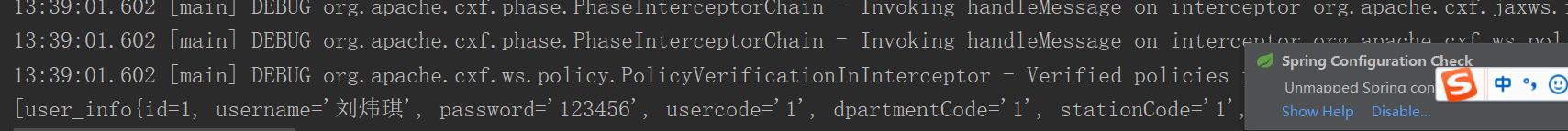
Cxf框架的配置

查询用户的接口



查询用户的实现类

Cxf通过客户端接口协议获取数据 

1. react前端框架的搭建及页面显示

首先系统执行main.js方法



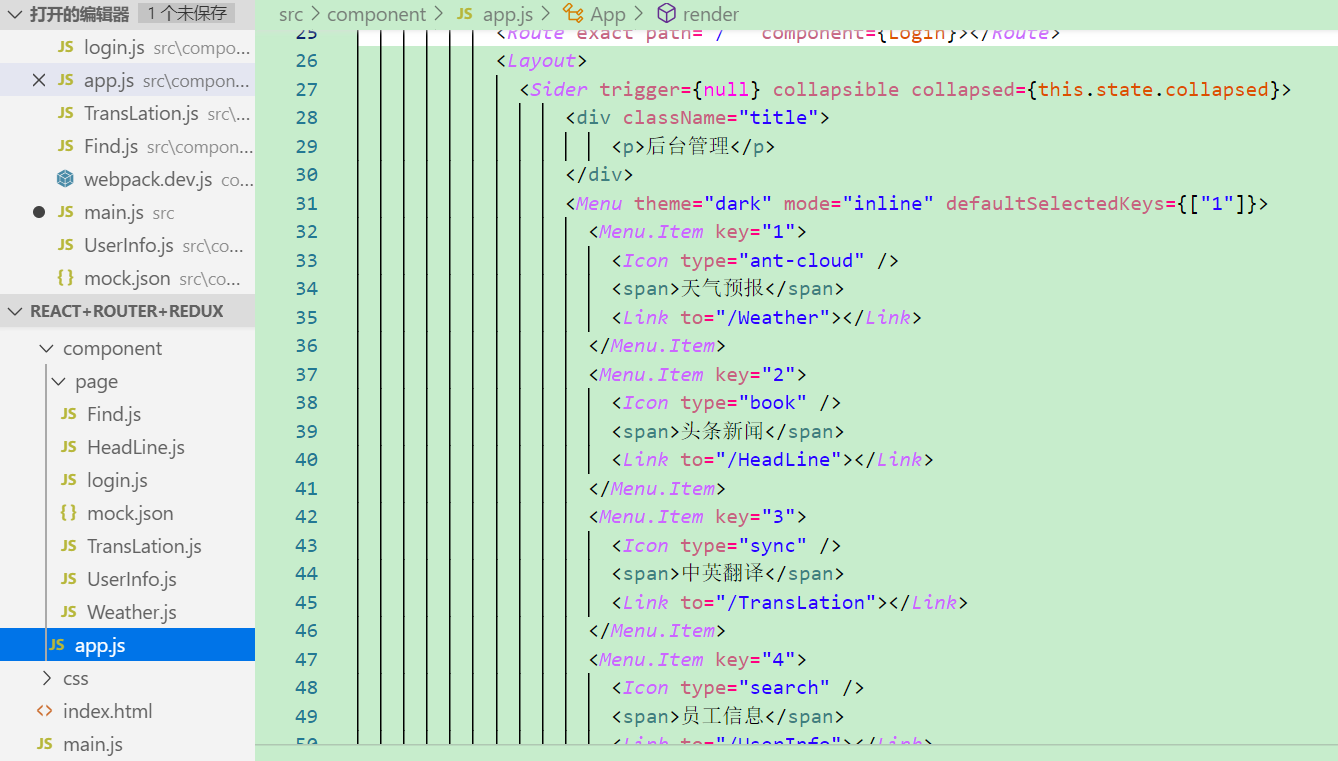
之后执行app.js组件

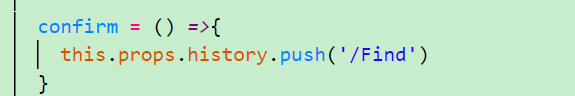


首先渲染login.js的组件，执行login.js组件



通过页面输入的值通过Axios配置url，将数据从前台传到后端，后端处理后的数据将返回到前端，前端通过后端的响应结果来实现是否登录成功，如果登录成功，则跳转到UserInfo组件，到了后台管理模块。

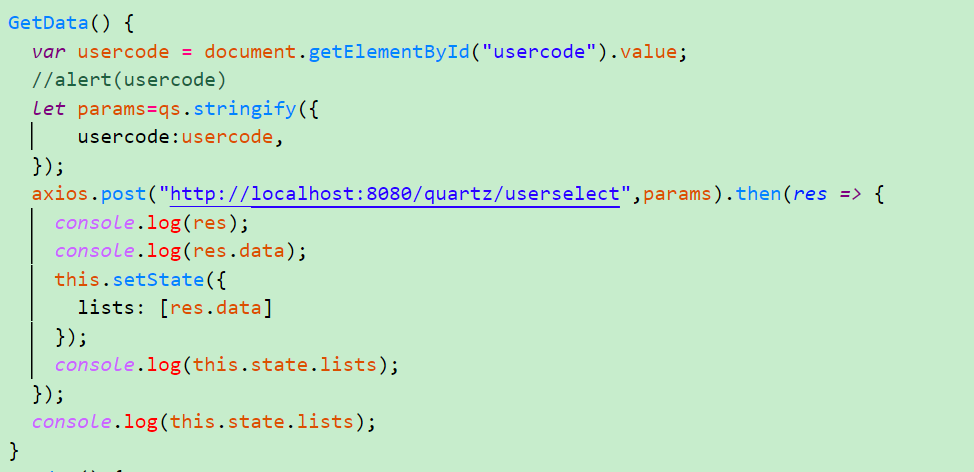


用户列表的显示，当点击用户信息时，会渲染页面，直接执行Axios的post请求，将后台的响应结果返回到前端页面，点击查找用户的按钮，会出现组件的跳转，会跳转到Find.js页面。



Find.js组件对于页面的渲染



通过获取员工编号的输入框中的值，通过axios的post方法将url和用户编号一起发送到后端，后端接收请求后，会执行对应的方法，将查找到的数据返回到前端页面。

1. 页面的展示

