package com.jwcq.config;

*/\*\**

*\* Created by liuma on 2017/6/7.*

*\*/*

public class Config {

public final static String *ADMIN*="admin";*//管理员*

public final static String *USER*="user";*//商务部普通用户*

public final static String *VIEWER*="viewer";*//未登录的用户*

public final static String *REVIEWER*="reviewer";*//审核员*

public final static String *NOTIFY*="notify";*//知会者，给他发短信*

*//1.暂存、2.审核中，3.审核通过4.审核驳回，*

public final static String *TEMP*="暂存";

public final static String *REVIEWING*="审核中";

public final static String *PASS*="审核通过";

public final static String *REJECT*="审核驳回";

}

package com.jwcq.config;

import org.apache.shiro.cache.MemoryConstrainedCacheManager;

import org.apache.shiro.cas.CasFilter;

import org.apache.shiro.cas.CasRealm;

import org.apache.shiro.cas.CasSubjectFactory;

import org.apache.shiro.spring.LifecycleBeanPostProcessor;

import org.apache.shiro.spring.web.ShiroFilterFactoryBean;

import org.apache.shiro.web.filter.authc.LogoutFilter;

import org.apache.shiro.web.mgt.DefaultWebSecurityManager;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.boot.web.servlet.FilterRegistrationBean;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.filter.DelegatingFilterProxy;

import javax.servlet.Filter;

import java.util.HashMap;

import java.util.LinkedHashMap;

import java.util.Map;

*/\*\**

*\* Created by willwu on 16-4-14.*

*\*/*

@Configuration

public class ShiroCasConfiguration {

private static final String *casFilterUrlPattern* = "/shiro-cas";

@Bean

public FilterRegistrationBean filterRegistrationBean() {

FilterRegistrationBean filterRegistration = new FilterRegistrationBean();

filterRegistration.setFilter(new DelegatingFilterProxy("shiroFilter"));

filterRegistration.addInitParameter("targetFilterLifecycle", "true");

filterRegistration.setEnabled(true);

filterRegistration.addUrlPatterns("/\*");

return filterRegistration;

}

@Bean(name = "lifecycleBeanPostProcessor")

public LifecycleBeanPostProcessor getLifecycleBeanPostProcessor() {

return new LifecycleBeanPostProcessor();

}

@Bean(name = "securityManager")

public DefaultWebSecurityManager getDefaultWebSecurityManager(@Value("${shiro.cas}") String casServerUrlPrefix, @Value("${shiro.server}") String shiroServerUrlPrefix) {

DefaultWebSecurityManager securityManager = new DefaultWebSecurityManager();

CasRealm casRealm = new CasRealm();

casRealm.setDefaultRoles("ROLE\_USER");

casRealm.setCasServerUrlPrefix(casServerUrlPrefix);

casRealm.setCasService(shiroServerUrlPrefix + *casFilterUrlPattern*);

securityManager.setRealm(casRealm);

securityManager.setCacheManager(new MemoryConstrainedCacheManager());

securityManager.setSubjectFactory(new CasSubjectFactory());

return securityManager;

}

*/\*\**

*\* 加载shiroFilter权限控制规则（从数据库读取然后配置） \**

*\*/*

private void loadShiroFilterChain(ShiroFilterFactoryBean shiroFilterFactoryBean) {

Map<String, String> filterChainDefinitionMap = new LinkedHashMap<>();

filterChainDefinitionMap.put(*casFilterUrlPattern*, "casFilter");

filterChainDefinitionMap.put("/logout","logout");

filterChainDefinitionMap.put("/\*\*", "authc");

shiroFilterFactoryBean.setFilterChainDefinitionMap(filterChainDefinitionMap);

}

*/\*\**

*\* CAS 过滤器*

*\*/*

@Bean(name = "casFilter")

public CasFilter getCasFilter(@Value("${shiro.cas}") String casServerUrlPrefix,

@Value("${shiro.server}") String shiroServerUrlPrefix) {

CasFilter casFilter = new CasFilter();

casFilter.setName("casFilter");

casFilter.setEnabled(true);

String loginUrl = casServerUrlPrefix + "/login?service=" + shiroServerUrlPrefix + *casFilterUrlPattern*;

casFilter.setFailureUrl(loginUrl);

return casFilter;

}

*/\*\**

*\* shiroFilter*

*\*/*

@Bean(name = "shiroFilter")

public ShiroFilterFactoryBean getShiroFilterFactoryBean(DefaultWebSecurityManager securityManagerCasFilter casFilter, @Value("${shiro.cas}") String casServerUrlPrefix,

@Value("${shiro.server}") String shiroServerUrlPrefix) {

ShiroFilterFactoryBean shiroFilterFactoryBean = new ShiroFilterFactoryBean();

shiroFilterFactoryBean.setSecurityManager(securityManager);

String loginUrl = casServerUrlPrefix + "/login?service=" + shiroServerUrlPrefix + *casFilterUrlPattern*;

shiroFilterFactoryBean.setLoginUrl(loginUrl);

shiroFilterFactoryBean.setSuccessUrl("/");

Map<String, Filter> filters = new HashMap<>();

filters.put("casFilter", casFilter);

LogoutFilter logoutFilter = new LogoutFilter();

logoutFilter.setRedirectUrl(casServerUrlPrefix + "/logout?service=" + shiroServerUrlPrefix);

filters.put("logout",logoutFilter);

shiroFilterFactoryBean.setFilters(filters);

loadShiroFilterChain(shiroFilterFactoryBean);

return shiroFilterFactoryBean;

}

}

package com.jwcq.controller;

import com.jwcq.entity.result.Response;

import com.jwcq.service.PriceService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

*/\*\**

*\* Created by luotuo on 17-5-19.*

*\*/*

@Controller

@RequestMapping("/")

@EnableAutoConfiguration

public class PriceController {

@Autowired

private PriceService priceService;

@RequestMapping(value = "/price")

@ResponseBody

public Response getPrice(@RequestParam(value = "city", required = true) String city) {

return priceService.getPrice(city);

}

@RequestMapping(value = "/getAll")

@ResponseBody

public Response getPrice() {

return priceService.getAll();

}

@RequestMapping(value = "/updateCenter", method = RequestMethod.*POST*)

@ResponseBody

public Response updateCenter(@RequestParam(value = "id", required = true) Long id,

@RequestParam(value = "json4", required = true) String json4) {

Response response = new Response();

return response;

}

@RequestMapping(value = "/getCenters", method = RequestMethod.*POST*)

@ResponseBody

public Response getCenters(@RequestParam(value = "id", required = true) Long id) {

Response response = new Response();

return response;

}

@RequestMapping(value = "/getCities")

@ResponseBody

public Response getCities() {

return priceService.getCities();

}

@RequestMapping(value = "/getAllowanceByLevel", method = RequestMethod.*POST*)

@ResponseBody

public Response getAllowanceByLevel(@RequestParam(value = "city", required = true) String city,

@RequestParam(value = "level", required = true) int level) {

return priceService.getAllowanceByLevel(city, level);

}

@RequestMapping(value = "/getAllowance", method = RequestMethod.*POST*)

@ResponseBody

public Response getAllowance(@RequestParam(value = "city", required = true) String city) {

return priceService.getAllowance(city);

}

@RequestMapping(value = "/getTravellingExpense", method = RequestMethod.*POST*)

@ResponseBody

public Response getTravellingExpense(@RequestParam(value = "city", required = true) String city) {

return priceService.getTravellingExpense(city);

}

}

ProjectController.java

package com.jwcq.controller;

import com.jwcq.config.Config;

import com.jwcq.entity.Project;

import com.jwcq.entity.Sponsor;

import com.jwcq.entity.User;

import com.jwcq.service.ProjectService;

import com.jwcq.service.SponsorService;

import com.jwcq.service.UserService;

import com.taobao.api.ApiException;

import com.taobao.api.DefaultTaobaoClient;

import com.taobao.api.TaobaoClient;

import com.taobao.api.request.AlibabaAliqinFcSmsNumSendRequest

import com.taobao.api.response.AlibabaAliqinFcSmsNumSendResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.PageRequest;

import org.springframework.data.domain.Sort;

import org.springframework.format.annotation.DateTimeFormat;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

import java.sql.Timestamp;

import java.text.SimpleDateFormat;

import java.util.\*;

*/\*\**

*\* Created by liuma on 2017/5/19.*

*\*/*

@Controller

public class ProjectController {

@Autowired

ProjectService projectService;

@Autowired

SponsorService sponsorService;

@Autowired

UserService userService;

@RequestMapping("/")

public String home() {

return "redirect:/view/home.html";

}

@RequestMapping("/logout")

public String logout(){

return "redirect:/logout";

}

@RequestMapping("/login")

public String login(){

return "redirect:/view/home.html";

}

*//测试jpa*

@RequestMapping(value = "/test")

@ResponseBody

Map test() {

Map results = new HashMap();

PageRequest pageRequest = new PageRequest(0, 20);

results.put("results", projectService.findTest(pageRequest));

return results;

}

*//获取项目列表*

@RequestMapping(value = "/projectList")

@ResponseBody

Map getAllList() {*//获取所有列表*

List<Project> result = projectService.findAll();

Map results = new HashMap();

results.put("result", result);

results.put("message", "succeed");

results.put("success", 1);

return results;

}

*//获取项目列表*

@RequestMapping(value = "/projectListPage")

@ResponseBody

Map getAllListPage(@RequestParam(value = "page", required = false) String page,

@RequestParam(value = "number", required = false) String number) {*//获取分页*

if(page==null)page="0";

if(number==null)number="20";

PageRequest pageRequest = new PageRequest(Integer.*valueOf*(page), Integer.*valueOf*(number));

Iterable<Project> result = projectService.findAll(pageRequest);

Map results = new HashMap();

results.put("result", result);

results.put("message", "succeed");

results.put("success", 1);

return results;

}

*//项目搜索*

@RequestMapping(value = "/search")

@ResponseBody

Map search(@RequestParam(value = "serial", required = false) String serial\_number,

@RequestParam(value = "name", required = false) String name,

@RequestParam(value = "sponsor", required = false) String sponsor,

@RequestParam(value = "user",required = false)String user,

@RequestParam(value = "state",required = false)String state,

@RequestParam(value = "start", required = false) @DateTimeFormat(pattern = "yyyy-MM-dd HH:mm:ss") Date start,

@RequestParam(value = "end", required = false) @DateTimeFormat(pattern = "yyyy-MM-dd HH:mm:ss") Date end) {*//搜索功能*

if (start == null) {

start = new Date();

start.setTime(0);

}

if (end == null) end = new Date();

if(serial\_number==null)serial\_number="";

if(sponsor==null)sponsor="";

if(user==null)user="";

if(state==null)state="";

if(userService.getRole().equals(Config.*ADMIN*)){

*//当前用户是超级管理员*

System.*out*.println("超级管理员"+"start time " + start + " end time" + end);

}else if(userService.getRole().equals(Config.*REVIEWER*)){

*//商务部审核人员*

System.*out*.println("商务部人员"+"start time " + start + " end time" + end);

}else if(userService.getRole().equals(Config.*USER*)){

*//普通用户人员*

System.*out*.println("普通用户人员"+"start time " + start + " end time" + end);

user=userService.getUserInfo().getName();

}

List<Project> result = projectService.findByProjectLike(serial\_number, name, sponsor,user,state,start, end);

Map results = new HashMap();

if (result == null || result.size() == 0) {

results.put("message", "查询为空，请更新查询条件");

results.put("success", 0);

} else {

results.put("result", result);

results.put("message", "succeed");

results.put("success", 1);

}

return results;

}

*//项目搜索*

@RequestMapping(value = "/searchPage")

@ResponseBody

Map searchPage(@RequestParam(value = "serial", required = false) String serial\_number,

@RequestParam(value = "name", required = false) String name,

@RequestParam(value = "sponsor", required = false) String sponsor,

@RequestParam(value = "user",required = false)String user,

@RequestParam(value = "state",required = false)String state,

@RequestParam(value = "start", required = false) @DateTimeFormat(pattern = "yyyy-MM-dd HH:mm:ss") Date start,

@RequestParam(value = "end", required = false) @DateTimeFormat(pattern = "yyyy-MM-dd HH:mm:ss") Date end,

@RequestParam(value = "page", required = false) String page,

@RequestParam(value = "number", required = false) String number) {*//搜索功能*

if (start == null) {

start = new Date();

start.setTime(0);

}

if (end == null) end = new Date();

if(name==null)name="";

if(serial\_number==null)serial\_number="";

if(sponsor==null)sponsor="";

if(user==null)user="";

if(page==null)page="0";

if(number==null)number="20";

if(state==null)state="";

if(userService.getRole().equals(Config.*ADMIN*)){

*//当前用户是超级管理员*

System.*out*.println("超级管理员"+"start time " + start + " end time" + end);

}else if(userService.getRole().equals(Config.*REVIEWER*)){

*//商务部审核人员*

System.*out*.println("商务部人员"+"start time " + start + " end time" + end);

}else if(userService.getRole().equals(Config.*USER*)){

*//普通用户人员*

System.*out*.println("普通用户人员"+"start time " + start + " end time" + end);

user=userService.getUserInfo().getName();

}

System.*out*.println(

"serial"+serial\_number

+ "name="+name

+ " name "+name

+ " sponsor"+sponsor

+ " user"+user

+" state"+state

+" start" + start

+ " end" + end

+" number="+number

+" page="+page);

Sort sort = new Sort(Sort.Direction.*DESC*, "id");

PageRequest pageRequest = new PageRequest(Integer.*valueOf*(page), Integer.*valueOf*(number), sort);

Iterable<Project> result = projectService.findByProjectLike(serial\_number, name, sponsor, user,state,start, end, pageRequest);

Map results = new HashMap();

if (result == null || (!result.iterator().hasNext())) {

results.put("message", "查询为空，请更新查询条件");

results.put("success", 0);

} else {

results.put("result", result);

results.put("message", "succeed");

results.put("success", 1);

}

return results;

}

*//项目编辑*

@RequestMapping(value = "/edit")

@ResponseBody

Map search(@RequestParam("id") String id) {

Project result = projectService.findProjectById(Integer.*valueOf*(id));

Map results = new HashMap();

if (result != null) {

results.put("result", result);

results.put("message", "succeed");

results.put("success", 1);

} else {

results.put("message", "项目id出现问题，请联系管理员");

results.put("success", 0);

}

return results;

}

*//更新备注*

@RequestMapping(value = "/updateRemark")

@ResponseBody

Map updateRemark(@RequestParam("id") String id, @RequestParam("remark") String remark) {

Map results = new HashMap();

Project project = projectService.findProjectById(Integer.*valueOf*(id));

if (project == null) {

results.put("message", "项目id出现问题，请联系管理员");

results.put("success", 0);

} else {

project.setRemarks(remark);

projectService.updateProject(project);

results.put("message", "备注更新成功");

results.put("success", 1);

}

return results;

}

*//新建项目信息，返回新建项目的project信息*

@RequestMapping(value = "/create")

@ResponseBody

Map create(@RequestParam(value = "serial\_number") String serial\_number,

@RequestParam(value = "version") String version,

@RequestParam(value = "name", required = false) String name,

@RequestParam(value = "center\_number", required = false) String center\_number,

@RequestParam(value = "budget\_price", required = false) String budget\_price,

@RequestParam(value = "sponsor", required = false) String sponsor,

@RequestParam(value = "type", required = false) String type,

@RequestParam(value = "json1", required = false) String json1,

@RequestParam(value = "json2", required = false) String json2,

@RequestParam(value = "json3", required = false) String json3,

@RequestParam(value = "json4", required = false) String json4,

@RequestParam(value = "json5", required = false) String json5

) {

Project project = new Project();

project.setSerial\_number(serial\_number);

project.setVersion(version);

project.setName(name);

project.setCenter\_number(center\_number);

project.setBudget\_price(budget\_price);

project.setSponsor(sponsor);

project.setSponsor\_code(sponsorService.findSponsorsByName(sponsor).getCode());

project.setSponsor(type);

project.setJson1(json1);

project.setJson2(json2);

project.setJson3(json3);

project.setJson4(json4);

project.setJson4(json5);

project.setState(Config.*TEMP*);

project.setUser(userService.getUserInfo().getName());

Map results = new HashMap();

Project result = projectService.createProject(project);

if (result == null) {

results.put("message", "项目编号和版本号重复，请更新");

results.put("success", 0);

} else {

results.put("result", result);

results.put("message", "添加成功");

results.put("success", 1);

}

return results;

}

*//更新信息 都返回项目id信息*

@RequestMapping(value = "/update")

@ResponseBody

Map update(@RequestParam(value = "id") String id,

@RequestParam(value = "serial\_number", required = false) String serial\_number,

@RequestParam(value = "version", required = false) String version,

@RequestParam(value = "name", required = false) String name,

@RequestParam(value = "center\_number", required = false) String center\_number,

@RequestParam(value = "budget\_price", required = false) String budget\_price,

@RequestParam(value = "sponsor", required = false) String sponsor,

@RequestParam(value = "type", required = false) String type,

@RequestParam(value = "json1", required = false) String json1,

@RequestParam(value = "json2", required = false) String json2,

@RequestParam(value = "json3", required = false) String json3,

@RequestParam(value = "json4", required = false) String json4,

@RequestParam(value = "json5", required = false) String json5

) {

Project project = projectService.findProjectById(Integer.*valueOf*(id));

Map results = new HashMap();

if (project == null) {

results.put("message", "项目id出现问题，请联系管理员");

results.put("success", 0);

} else {

project.setName(name);

project.setCenter\_number(center\_number);

project.setBudget\_price(budget\_price);

project.setSponsor(sponsor);

project.setSponsor\_code(sponsorService.findSponsorsByName(sponsor).getCode());

project.setSponsor(type);

project.setJson1(json1);

project.setJson2(json2);

project.setJson3(json3);

project.setJson4(json4);

project.setJson4(json5);

projectService.updateProject(project);

}

results.put("message", "保存成功");

results.put("success", 1);

return results;

}

*//保存或者新建*

@RequestMapping(value = "/save")

@ResponseBody

Map save(@RequestParam(value = "id", required = false) String id,

@RequestParam(value = "serial\_number", required = true) String serial\_number,

@RequestParam(value = "version", required = true) String version,

@RequestParam(value = "name", required = false) String name,

@RequestParam(value = "center\_number", required = false) String center\_number,

@RequestParam(value = "budget\_price", required = false) String budget\_price,

@RequestParam(value = "sponsor", required = false) String sponsor,

@RequestParam(value = "type", required = false) String type,

@RequestParam(value = "json1", required = false) String json1,

@RequestParam(value = "json2", required = false) String json2,

@RequestParam(value = "json3", required = false) String json3,

@RequestParam(value = "json4", required = false) String json4,

@RequestParam(value = "json5", required = false) String json5

) {

Map results = new HashMap();

if(id==null){

results.put("message", "项目id为空，请联系前台开发工程师~");

results.put("success", 0);

return results;

}

if(name==null)name="";

if(center\_number==null)center\_number="";

if(budget\_price==null)budget\_price="";

if(type==null)type="";

if(json1==null)json1="";

if(json2==null)json2="";

if(json3==null)json3="";

if(json4==null)json4="";

if(json5==null)json5="";

serial\_number = serial\_number.trim();

version = version.trim();

Project project;

project = new Project();

project.setSerial\_number(serial\_number);*//新建的时候可以修改项目编号*

project.setVersion(version);*//新建的时候可以修改项目版本号*

project.setName(name);

project.setCenter\_number(center\_number);

project.setBudget\_price(budget\_price);

project.setSponsor(sponsor);

project.setSponsor\_code(sponsorService.findSponsorsByName(sponsor).getCode());

project.setType(type);

project.setJson1(json1);

project.setJson2(json2);

project.setJson3(json3);

project.setJson4(json4);

project.setJson5(json5);

*//新建或者复制*

if (Integer.*valueOf*(id) <= 0) {

*//新建一个项目*

if (Integer.*valueOf*(id) == 0) {

project.setVersion("V1.0");

project.setVersion\_num(0);

project.setState(Config.*TEMP*);

project.setUser(userService.getUserInfo().getName());

if (projectService.findMaxProjectId(project.getSerial\_number()) > 0) {

results.put("message", "项目编号已存在，请修改项目编号或者进行复制项目操作！");

results.put("success", 0);

return results;

}

}

*//复制一个项目*

else if (Integer.*valueOf*(id) < 0) {

int version\_num = 1 + projectService.findMaxProjectId(project.getSerial\_number());

project.setVersion\_num(version\_num);

project.setVersion("V1." + version\_num);

project.setState(Config.*TEMP*);*//复制一个项目的状态还是暂存状态*

project.setUser(userService.getUserInfo().getName());

}

if (projectService.createProject(project) == null) {

results.put("message", "项目编号与版本号已存在，请查明修改！");

results.put("success", 0);

return results;

} else {

results.put("result", project);

results.put("message", "保存成功");

results.put("success", 1);

return results;

}

} else {

*//保存一个项目*

project = projectService.findProjectById(Integer.*valueOf*(id));

if (project == null) {

results.put("message", "项目id出现问题，请联系管理员");

results.put("success", 0);

return results;

}

}

project.setName(name);

project.setCenter\_number(center\_number);

project.setBudget\_price(budget\_price);

project.setSponsor(sponsor);

project.setSponsor\_code(sponsorService.findSponsorsByName(sponsor).getCode());

project.setType(type);

project.setJson1(json1);

project.setJson2(json2);

project.setJson3(json3);

project.setJson4(json4);

project.setJson5(json5);

projectService.updateProject(project);

results.put("result", project);

results.put("message", "保存成功");

results.put("success", 1);

return results;

}

*//复制*

@RequestMapping(value = "/copy")

@ResponseBody

Map copy(@RequestParam("id") String id) {

Project project = projectService.findProjectById(Integer.*valueOf*(id));

Map results = new HashMap();

if (project == null) {

results.put("message", "项目id出现问题，请联系管理员");

results.put("success", 0);

} else {

project.setId(-1);

project.setPversion(project.getVersion());

int version\_num = 1 + projectService.findMaxProjectId(project.getSerial\_number());

project.setVersion\_num(version\_num);

project.setVersion("V1." + version\_num);

project.setState(Config.*TEMP*);*//复制一个项目的状态时，返回的暂存状态*

project.setUser(userService.getUserInfo().getName());

results.put("result", project);

results.put("message", "复制成功");

results.put("success", 1);

}

return results;

}

*//删除*

@RequestMapping(value = "/delete")

@ResponseBody

Map delete(@RequestParam("id") String id) {

Map results = new HashMap();

if (projectService.deleteProjectById(Integer.*valueOf*(id))) {

results.put("message", "删除成功");

results.put("success", 1);

} else {

results.put("message", "删除成功");

results.put("success", 0);

}

return results;

}

*//提交项目审核*

@RequestMapping(value = "/submit")

@ResponseBody

Map submit(@RequestParam("id") String id) {

Map results = new HashMap();

if(id==null||id.equals("")||Integer.*valueOf*(id)<0){

results.put("message", "项目未保存成功，请先保存项目！");

results.put("success", 0);

}

else if (projectService.submitProjectById(Integer.*valueOf*(id))) {

results.put("message", "提交审核成功");

results.put("success", 1);

try {

List<User>notifiers=userService.getNotifier();

String sponsor=projectService.findProjectById(Integer.*valueOf*(id)).getSponsor();

if(sponsor.length()>10)sponsor=sponsor.substring(0,8)+"...";

String name=userService.getUserInfo().getName();

for(int i=0;i<notifiers.size();i++){

String notifier=notifiers.get(i).getName();

String phone=notifiers.get(i).getContact();

sendMessage(notifier,name,sponsor,phone);

}

} catch (ApiException e) {

System.*out*.println(" sendMessage erro ");

e.printStackTrace();

}

} else {

results.put("message", "提交审核失败");

results.put("success", 0);

}

return results;

}

*//审核通过*

@RequestMapping(value = "/passaudit")

@ResponseBody

Map pass(@RequestParam("id") String id) {

Map results = new HashMap();

if (projectService.passProjectById(Integer.*valueOf*(id))) {

results.put("message", "通过审核");

results.put("success", 1);

} else {

results.put("message", "审核出现问题");

results.put("success", 0);

}

return results;

}

*//审核通过*

@RequestMapping(value = "/reject")

@ResponseBody

Map reject(@RequestParam("id") String id) {

Map results = new HashMap();

if (projectService.rejectProjectById(Integer.*valueOf*(id))) {

results.put("message", "审核驳回");

results.put("success", 1);

} else {

results.put("message", "审核出现问题出现问题");

results.put("success", 0);

}

return results;

}

*/\*\**

*\* 获取所有申办方/委托方列表*

*\* \*/*

@RequestMapping(value = "/getSponsors")

@ResponseBody

Map getSponsors() {

Map results = new HashMap();

List<Sponsor> sponsors=sponsorService.findAllSponsor();

if (sponsorService==null || sponsors.size()==0) {

results.put("message", "申办方列表为空");

results.put("success", 0);

} else {

results.put("result",sponsors);

results.put("message", "申办方列表获取成功");

results.put("success", 1);

}

return results;

}

*/\*\**

*\* 新建委托方或者申办方*

*\* \*/*

@RequestMapping(value = "/saveSponsor")

@ResponseBody

Map saveSponsor(@RequestParam(value = "name", required = false) String name,

@RequestParam(value = "code", required = false) String code

) {

Map results = new HashMap();

if (sponsorService.saveSponsor(name,code)) {

results.put("message", "申办方添加成功");

results.put("success", 1);

} else {

results.put("message", "申办方添加失败，名称或者编号重复");

results.put("success", 0);

}

return results;

}

*/\*\**

*\* 获取当前用户信息*

*\* \*/*

@RequestMapping(value = "/getUserInfo")

@ResponseBody

Map getUserInfo() {

Map results = new HashMap();

*//TODO sso之后更换为当天用户信息*

results.put("result", userService.getUserInfo());

results.put("message", "用户信息");

results.put("success", 1);

return results;

}

private void sendMessage(String notifier,String name,String sponsor,String phone) throws ApiException{

String url="http://gw.api.taobao.com/router/rest";

String appkey="24360449";

String secret="0e70fa9c42a0762bb08c756acc7d66eb";

TaobaoClient client = new DefaultTaobaoClient(url, appkey, secret);

AlibabaAliqinFcSmsNumSendRequest req = new AlibabaAliqinFcSmsNumSendRequest();

req.setExtend( "" );

req.setSmsType( "normal" );

req.setSmsFreeSignName("内部报价待审信息");

req.setSmsParamString("{\n" +

" \"notifier\": \""+notifier+"\",\n" +

" \"name\": \""+name+"\",\n" +

" \"sponsor\": \""+sponsor+"\"\n" +

"}");

req.setRecNum(phone);

req.setSmsTemplateCode("SMS\_71380304");

AlibabaAliqinFcSmsNumSendResponse rsp = client.execute(req);

System.*out*.println("response"+rsp.getBody());

}

}

package com.jwcq.entity.result;

*/\*\**

*\* Created by luotuo on 17-5-23.*

*\*/*

public class Response {

private int success;

private String message;

private Object result;

public Response(int success, String message, Object obj) {

this.success = success;

this.message = message;

this.result = obj;

}

public Response(){}

public int getSuccess() {

return success;

}

public void setSuccess(int success) {

this.success = success;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public Object getResult() {

return result;

}

public void setResult(Object result) {

this.result = result;

}

public void buildResponse(int success, String message, Object obj) {

this.success = success;

this.message = message;

this.result = obj;

}

}

package com.jwcq.entity;

*/\*\**

*\* Created by luotuo on 17-5-23.*

*\*/*

public class Allowance extends AllowanceByLevel{

protected int level;

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

}

package com.jwcq.entity;

*/\*\**

*\* Created by luotuo on 17-5-23.*

*\*/*

public class AllowanceByLevel{

protected float rate;

protected float technical\_allowance;

protected float food\_allowance;

protected float communication\_allowance;

protected float transportation\_allowance;

public float getRate() {

return rate;

}

public void setRate(float rate) {

this.rate = rate;

}

public float getTechnical\_allowance() {

return technical\_allowance;

}

public void setTechnical\_allowance(float technical\_allowance) {

this.technical\_allowance = technical\_allowance;

}

public float getFood\_allowance() {

return food\_allowance;

}

public void setFood\_allowance(float food\_allowance) {

this.food\_allowance = food\_allowance;

}

public float getCommunication\_allowance() {

return communication\_allowance;

}

public void setCommunication\_allowance(float communication\_allowance) {

this.communication\_allowance = communication\_allowance;

}

public float getTransportation\_allowance() {

return transportation\_allowance;

}

public void setTransportation\_allowance(float transportation\_allowance) {

this.transportation\_allowance = transportation\_allowance;

}

}

package com.jwcq.entity;

import javax.persistence.\*;

*/\*\**

*\* Created by luotuo on 17-5-19.*

*\*/*

@Entity

@Table(name = "price")

public class Price {

@Id

@GeneratedValue

private long id;

private String city;

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getTransportation() {

return transportation;

}

public void setTransportation(String transportation) {

this.transportation = transportation;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

public float getRate() {

return rate;

}

public void setRate(float rate) {

this.rate = rate;

}

public float getTechnical\_allowance() {

return technical\_allowance;

}

public void setTechnical\_allowance(float technical\_allowance) {

this.technical\_allowance = technical\_allowance;

}

public float getFood\_allowance() {

return food\_allowance;

}

public void setFood\_allowance(float food\_allowance) {

this.food\_allowance = food\_allowance;

}

public float getCommunication\_allowance() {

return communication\_allowance;

}

public void setCommunication\_allowance(float communication\_allowance) {

this.communication\_allowance = communication\_allowance;

}

public float getTransportation\_allowance() {

return transportation\_allowance;

}

public void setTransportation\_allowance(float transportation\_allowance) {

this.transportation\_allowance = transportation\_allowance;

}

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

private String transportation;

private int level;

private float rate;

private float technical\_allowance;

private float food\_allowance;

private float communication\_allowance;

private float transportation\_allowance;

private float price;

}

package com.jwcq.entity;

*/\*\**

*\* Created by luotuo on 17-5-23.*

*\*/*

public class Price1{

private String city;

private float price;

private String transportation;

private int level;

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

public String getTransportation() {

return transportation;

}

public void setTransportation(String transportation) {

this.transportation = transportation;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

}

package com.jwcq.entity;

import javax.persistence.\*;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Size;

import java.util.Date;

*/\*\**

*\* Created by liuma on 2017/5/18.*

*\*/*

@Entity

@Table(name="budget\_summary")

public class Project {

@Id

@GeneratedValue

private int id;

@NotNull

@Size(min = 1, max = 64, message = "serial\_number have to be grater than 1 characters and less than 64 characters")

private String serial\_number;

private int version\_num;

@NotNull

@Size(min = 1, max = 64, message = "version have to be grater than 1 characters and less than 64 characters")

private String version;

private String pversion;

private String name;

private String center\_number;

private String type;

private String budget\_price;

private String sponsor;

private String sponsor\_code;

private String remarks;

private Date create\_time;

private String json1;

private String json2;

private String json3;

private String json4;

private String json5;

private String state;

private String user;

public int getIsDelete() {

return isDelete;

}

public void setIsDelete(int isDelete) {

this.isDelete = isDelete;

}

@Column(name = "is\_delete",nullable=false,length=1,columnDefinition="INT default 0")

private int isDelete;*//是否入报告,0没有删除，1删除*

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getSerial\_number() {

return serial\_number;

}

public void setSerial\_number(String serial\_number) {

this.serial\_number = serial\_number;

}

public String getVersion() {

return version;

}

public void setVersion(String version) {

this.version = version;

}

public String getPversion() {

return pversion;

}

public void setPversion(String pversion) {

this.pversion = pversion;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCenter\_number() {

return center\_number;

}

public void setCenter\_number(String center\_number) {

this.center\_number = center\_number;

}

public String getBudget\_price() {

return budget\_price;

}

public void setBudget\_price(String budget\_price) {

this.budget\_price = budget\_price;

}

public String getSponsor() {

return sponsor;

}

public void setSponsor(String sponsor) {

this.sponsor = sponsor;

}

public String getRemarks() {

return remarks;

}

public void setRemarks(String remarks) {

this.remarks = remarks;

}

public String getJson1() {

return json1;

}

public void setJson1(String json1) {

this.json1 = json1;

}

public String getJson2() {

return json2;

}

public void setJson2(String json2) {

this.json2 = json2;

}

public String getJson3() {

return json3;

}

public void setJson3(String json3) {

this.json3 = json3;

}

public String getJson4() {

return json4;

}

public void setJson4(String json4) {

this.json4 = json4;

}

public Date getCreate\_time() {

return create\_time;

}

public int getVersion\_num() {

return version\_num;

}

public void setVersion\_num(int version\_num) {

this.version\_num = version\_num;

}

public void setCreate\_time(Date create\_time) {

this.create\_time = create\_time;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getUser() {

return user;

}

public void setUser(String user) {

this.user = user;

}

public String getSponsor\_code() {

return sponsor\_code;

}

public void setSponsor\_code(String sponsor\_code) {

this.sponsor\_code = sponsor\_code;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getJson5() {

return json5;

}

public void setJson5(String json5) {

this.json5 = json5;

}

}

package com.jwcq.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

*/\*\**

*\* Created by liuma on 2017/6/7.*

*\*/*

@Entity

@Table(name="sponsor")

public class Sponsor {

@Id

@GeneratedValue

private int id;

private String name;

private String code;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

}

package com.jwcq.entity;

*/\*\**

*\* Created by luotuo on 17-5-23.*

*\*/*

public class TravellingExpense extends Allowance{

private String city;

private float price;

private String transportation;

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

public String getTransportation() {

return transportation;

}

public void setTransportation(String transportation) {

this.transportation = transportation;

}

}

package com.jwcq.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

*/\*\**

*\* Created by liuma on 2017/6/7.*

*\*/*

@Entity

@Table(name="user")

public class User {

@Id

@GeneratedValue

private int id;

private String name;

private String password;

private String email;

private String contact;

private String role;

private String account;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getContact() {

return contact;

}

public void setContact(String contact) {

this.contact = contact;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public String getAccount() {

return account;

}

public void setAccount(String account) {

this.account = account;

}

}

package com.jwcq.repository;

import com.jwcq.entity.Price;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import java.util.List;

import javax.persistence.Table;

*/\*\**

*\* Created by luotuo on 17-5-19.*

*\*/*

@Repository

@Table(name = "price")

@Qualifier("priceRepository")

public interface PriceRepository extends JpaRepository<Price, Long> {

@Query(value = "SELECT city, price, transportation, `level` from price where city = ?1", nativeQuery = true)

public List<Object[]> getPrice(String city);

@Query(value = "SELECT city FROM price GROUP BY city", nativeQuery = true)

public List<Object[]> getCities();

@Query(value = "SELECT rate, technical\_allowance, food\_allowance, communication\_allowance, transportation\_allowance from price where city = ?1 and `level` = ?2", nativeQuery = true)

public List<Object[]> getAllowanceByLevel(String city, int le);

@Query(value = "SELECT rate, technical\_allowance, food\_allowance, communication\_allowance, transportation\_allowance, `level` from price where city = :city", nativeQuery = true)

public List<Object[]> getAllowance(@Param("city") String city);

@Query(value = "SELECT rate, technical\_allowance, food\_allowance, communication\_allowance, transportation\_allowance, `level`, city, price, transportation from price where city = :city", nativeQuery = true)

public List<Object[]> getTravellingExpense(@Param("city") String city);

@Query(value = "select city, price, transportation from price", nativeQuery = true)

public List<Object[]> getAll();

}

package com.jwcq.repository;

import com.jwcq.entity.Project;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.PagingAndSortingRepository;

import java.util.Date;

import java.util.List;

*/\*\**

*\* Created by liuma on 2017/5/19.*

*\*/*

public interface ProjectPageRepository extends PagingAndSortingRepository<Project, Long> {

Project findById(int id);

List<Project> findByNameLike(String name);

@Query(value="select bean from Project bean where serial\_number like %?1% and name like %?2% and (sponsor like %?3% or sponsor\_code like %?3%) and user like %?4% and state like %?5% and create\_time>?6 and create\_time<?7 and isDelete=0",nativeQuery = false)

Page findProject(String serial, String name, String sponsor, String user,String state,Date start, Date end, Pageable page);

@Query(value="select bean.id,bean.center\_number from Project bean",nativeQuery = false)

Page getProjects(Pageable page);

@Query(value="select \* from budget\_summary where serial\_number=?1 and version=?2 and is\_delete=0",nativeQuery = true)

List<Project> findBySerialNumberAndVersion(String serial, String version);

@Query(value="select serial\_number, version from budget\_summary where id>0",nativeQuery = true)

List findObjectById();

}

package com.jwcq.repository;

import com.jwcq.entity.Project;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import java.sql.Timestamp;

import java.util.Date;

import java.util.List;

*/\*\**

*\* Created by liuma on 2017/5/19.*

*\*/*

public interface ProjectRepository extends JpaRepository<Project, Long> {

Project findById(int id);

List<Project> findByNameLike(String name);

@Query(value="select *\** from budget\_summary where serial\_number like %?1% and name like %?2% and (sponsor like %?3% or sponsor\_code like %?3%) and user like %?4% and state like %?5% and create\_time>?6 and create\_time<?7 and isDelete=0 order by id desc",nativeQuery = true)

List<Project>findProject(String serial,String name,String sponsor,String user,String state,Date start,Date end);

@Query(value="select *\** from budget\_summary where serial\_number=?1 and version=?2 and is\_delete=0 order by id desc",nativeQuery = true)

List<Project> findBySerialNumberAndVersion(String serial,String version);

@Query(value="select serial\_number, version from budget\_summary where id>0",nativeQuery = true)

List findObjectById();

@Query(value="select *\** from budget\_summary where serial\_number=?1 order by version\_num desc",nativeQuery = true)

List<Project> findProjectsBySerial(String serial);

}

package com.jwcq.repository;

import com.jwcq.entity.Project;

import com.jwcq.entity.Sponsor;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import java.util.Date;

import java.util.List;

*/\*\**

*\* Created by liuma on 2017/5/19.*

*\*/*

public interface SponsorRepository extends JpaRepository<Sponsor, Long> {

Sponsor findById(int id);

List<Sponsor> findByName(String name);

List<Sponsor> findByCode(String code);

List<Sponsor> findByNameOrCode(String name,String code);

}

package com.jwcq.repository;

import com.jwcq.entity.Sponsor;

import com.jwcq.entity.User;

import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

*/\*\**

*\* Created by liuma on 2017/5/19.*

*\*/*

public interface UserRepository extends JpaRepository<User, Long> {

User findById(int id);

List<User> findByRole(String role);

List<User> findByName(String name);

List<User> findByAccount(String account);

List<User> findByContact(String contact);

}

package com.jwcq.service;

import com.jwcq.entity.result.Response;

import com.jwcq.repository.PriceRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

import com.jwcq.entity.\*;

*/\*\**

*\* Created by luotuo on 17-5-19.*

*\*/*

@Service

public class PriceService {

@Autowired

private PriceRepository priceRepository;

public Response getPrice(String city) {

int success = 1;

String message = "成功";

List<Object[]> obj = priceRepository.getPrice(city);

Price1 price;

Response response;

if (obj.size() > 0) {

price = new Price1();

price.setCity((String)obj.get(0)[0]);

price.setPrice(Float.*valueOf*(obj.get(0)[1].toString()));

price.setTransportation((String)obj.get(0)[2]);

price.setLevel(0);

response = buildResponse(success, message, price);

} else {

success = 0;

message = "结果为空";

response = buildResponse(success, message, obj);

}

return response;

}

public Response updateCenter() {

int success = 1;

String message = "成功";

Object obj = new Object();

return buildResponse(success, message, obj);

}

public Response getCenters() {

int success = 1;

String message = "成功";

Object obj = new Object();

return buildResponse(success, message, obj);

}

public Response getCities() {

int success = 1;

String message = "成功";

List<Object[]> obj = priceRepository.getCities();

if (obj.size() <= 0) {

success = 0;

message = "结果为空";

}

return buildResponse(success, message, obj);

}

public Response getAllowanceByLevel(String city, int level) {

int success = 1;

String message = "成功";

List<Object[]> obj = priceRepository.getAllowanceByLevel(city, level);

AllowanceByLevel ab;

Response response;

if (obj.size() > 0) {

ab = new AllowanceByLevel();

response = buildResponse(success, message, ab);

} else {

success = 0;

message = "结果为空";

response = buildResponse(success, message, obj);

}

return response;

}

public Response getAllowance(String city) {

int success = 1;

String message = "成功";

List<Object[]> obj = priceRepository.getAllowance(city);

List<Allowance> list = new ArrayList<Allowance>();

if (obj.size() > 0) {

Allowance a = new Allowance();

} else {

success = 0;

message = "结果为空";

}

return buildResponse(success, message, list);

}

public Response getTravellingExpense(String city) {

int success = 1;

String message = "成功";

List<Object[]> obj = priceRepository.getTravellingExpense(city);

TravellingExpense te;

Response response;

if (obj.size() > 0) {

te = new TravellingExpense();

response = buildResponse(success, message, te);

} else {

success = 0;

message = "结果为空";

response = buildResponse(success, message, obj);

}

return response;

}

public Response getAll() {

int success = 1;

String message = "成功";

List<Object[]> obj = priceRepository.getAll();

Response response = buildResponse(success, message, obj);;

return response;

}

Response buildResponse(int success, String message, Object obj) {

Response response = new Response(success, message, obj);

return response;

}

}

package com.jwcq.service;

import com.jwcq.config.Config;

import com.jwcq.entity.Project;

import com.jwcq.repository.ProjectPageRepository;

import com.jwcq.repository.ProjectRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.Pageable;

import org.springframework.data.domain.Sort;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import java.sql.Timestamp;

import java.util.Collection;

import java.util.Date;

import java.util.List;

*/\*\**

*\* Created by liuma on 2017/5/18.*

*\*/*

@Service

public class ProjectService {

@Autowired

private ProjectRepository projectRepository;

@Autowired

private ProjectPageRepository projectPageRepository;

public List<Project> findByProjectLike(String serial, String name, String sponsor,String user,String state,Date start, Date end) {

*//return null;*

return projectRepository.findProject(serial,name,sponsor,user,state,start,end);

}

public Iterable<Project> findByProjectLike(String serial, String name, String sponsor,String user,String state, Date start, Date end, Pageable pageable) {

Iterable<Project>projects= projectPageRepository.findProject(serial,name,sponsor,user,state,start,end,pageable);

return projects;

}

public Iterable<Project> findTest(Pageable page){

return projectPageRepository.getProjects(page);

}

public Project createProject(Project project) {

if(projectRepository.findBySerialNumberAndVersion(project.getSerial\_number(),project.getVersion()).size()==0){

return projectRepository.save(project);

}else{

return null;

}

}

public Project updateProject(Project project) {

return projectRepository.save(project);

}

public List<Project> copyProject(int id) {

return null;

}

public Project findProjectById(int id) {

return projectRepository.findById(id);

}

public boolean deleteProjectById(int id) {

Project project= projectRepository.findById(id);

if(project==null)return false;

else {

project.setIsDelete(1);

projectRepository.save(project);

return true;

}

}

public List<Project> findAll() {

Sort sort = new Sort(Sort.Direction.*DESC*, "id");

List<Project> projects=projectRepository.findAll(sort);

return projects;

}

public Iterable<Project> findAll(Pageable pageable) {

Iterable<Project>projects= projectPageRepository.findAll(pageable);

return projects;

}

public List test(){

return projectRepository.findObjectById();

}

public int findMaxProjectId(String serial) {

List<Project>projects= projectRepository.findProjectsBySerial(serial);

if(projects==null||projects.size()==0)return 0;

else return projects.get(0).getVersion\_num();

}

*/\*\**

*\* 提交项目审核*

*\* 提交前是暂存、提交后是审核中，审核后是审核通过/审核驳回*

*\* \*\*/*

public boolean submitProjectById(int id) {

Project project= projectRepository.findById(id);

if(project==null)return false;

else {

project.setState(Config.*REVIEWING*);

projectRepository.save(project);

return true;

}

}

*/\*\**

*\* 审核驳回*

*\* \*\*/*

public boolean rejectProjectById(int id) {

Project project= projectRepository.findById(id);

if(project==null)return false;

else {

project.setState(Config.*REJECT*);

projectRepository.save(project);

return true;

}

}

*/\*\**

*\* 审核通过*

*\* \*\*/*

public boolean passProjectById(int id) {

Project project= projectRepository.findById(id);

if(project==null)return false;

else {

project.setState(Config.*PASS*);

projectRepository.save(project);

return true;

}

}

}

package com.jwcq.service;

import com.jwcq.entity.Sponsor;

import com.jwcq.repository.SponsorRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.logging.SocketHandler;

*/\*\**

*\* Created by liuma on 2017/6/7.*

*\*/*

@Service

public class SponsorService {

@Autowired

private SponsorRepository sponsorRepository;

*/\*\**

*\* 查找申办方/委托方*

*\* \*/*

public Sponsor findSponsorById(int id){

return sponsorRepository.findById(id);

}

*/\*\**

*\* 通过名称查找申办方/委托方*

*\* \*/*

public Sponsor findSponsorsByName(String name){

List<Sponsor> sponsors=sponsorRepository.findByName(name);

if(sponsors!=null&&sponsors.size()>0){

return sponsors.get(0);

}else{

Sponsor sp= new Sponsor();

sp.setName("未知");

sp.setCode("UNKNOWN");

return sp;

}

}

*/\*\**

*\* 通过代码查找申办方/委托方*

*\* \*/*

public Sponsor findSponsorsByCode(String code){

List<Sponsor> sponsors=sponsorRepository.findByCode(code);

if(sponsors!=null&&sponsors.size()>0){

return sponsors.get(0);

}else{

Sponsor sp= new Sponsor();

sp.setName("");

sp.setCode("");

return sp;

}

}

*/\**

*\* 获取所有申办方、委托方列表*

*\* \*\*/*

public List<Sponsor> findAllSponsor(){

return sponsorRepository.findAll();

}

*/\*\**

*\* 新建委托方*

*\* \*/*

public boolean saveSponsor(String name,String code){

List<Sponsor> sp= sponsorRepository.findByNameOrCode(name,code);

if(sp==null||sp.size()==0){

Sponsor sponsor=new Sponsor();

sponsor.setName(name);

sponsor.setCode(code);

sponsorRepository.save(sponsor);

return true;

}

return false;

}

}

package com.jwcq.service;

import com.jwcq.config.Config;

import com.jwcq.entity.Sponsor;

import com.jwcq.entity.User;

import com.jwcq.repository.SponsorRepository;

import com.jwcq.repository.UserRepository;

import org.apache.shiro.SecurityUtils;

import org.apache.shiro.subject.Subject;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

*/\*\**

*\* Created by liuma on 2017/6/7.*

*\*/*

@Service

public class UserService {

@Autowired

UserRepository userRepository;

*/\*\**

*\* 根据email获取当前用户的角色*

*\* \*/*

public String getRole(){

List<User> users=userRepository.findByAccount(getUserAccount());

if(users==null||users.size()==0){

return Config.*VIEWER*;

}else if(users.get(0).getRole().contains(Config.*ADMIN*)){

return Config.*ADMIN*;

}else if(users.get(0).getRole().contains(Config.*USER*)){

return Config.*USER*;

}else if(users.get(0).getRole().contains(Config.*REVIEWER*)){

return Config.*REVIEWER*;

}

else return Config.*VIEWER*;

}

*/\*\**

*\* 根据当前用户的信息*

*\* \*/*

public User getUserInfo(){

List<User> users=userRepository.findByAccount(getUserAccount());

if(users==null||users.size()==0){

User user=new User();

user.setName("未登录测试");

user.setEmail("未登录@未登录.com");

user.setContact("15556123456");

user.setRole(Config.*ADMIN*);

return user;

}else{

*//users.get(0).setPassword("");*

return users.get(0);

}

}

*/\*\**

*\* 获取知会者*

*\* \*/*

*/\*\**

*\* 根据当前用户的信息*

*\* \*/*

public List<User> getNotifier(){

List<User> users=userRepository.findByRole(Config.*NOTIFY*);

if(users==null||users.size()==0){

User user=new User();

user.setName("未登录测试");

user.setEmail("未登录@未登录.com");

user.setContact("15011488150");

user.setRole(Config.*VIEWER*);

users=new ArrayList<User>();

users.add(user);

return users;

}else{

return users;

}

}

*/\*\**

*\* 获取用户email*

*\* \*/*

public String getUserAccount() {

*// shiro-cas*

Subject subject = SecurityUtils.*getSubject*();

if (subject == null || subject.getPrincipals() == null) {

return null;

}

return (String) subject.getPrincipals().getPrimaryPrincipal();

*// shiro-cas*

*// return "未登录@未登录.com";*

}

}

package com.jwcq;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.boot.web.support.SpringBootServletInitializer;

import org.springframework.context.annotation.Bean;

import org.springframework.web.cors.CorsConfiguration;

import org.springframework.web.cors.UrlBasedCorsConfigurationSource;

import org.springframework.web.filter.CorsFilter;

@SpringBootApplication

public class BudgetApplication extends SpringBootServletInitializer{

public static void main(String[] args) {

SpringApplication.*run*(BudgetApplication.class, args);

}

private CorsConfiguration buildConfig() {

CorsConfiguration corsConfiguration = new CorsConfiguration();

corsConfiguration.addAllowedOrigin("\*");

corsConfiguration.addAllowedHeader("\*");

corsConfiguration.addAllowedMethod("\*");

return corsConfiguration;

}

*/\*\**

*\* 跨域过滤器*

*\* @return*

*\*/*

@Bean

public CorsFilter corsFilter() {

UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();

source.registerCorsConfiguration("/\*\*", buildConfig()); *// 4*

return new CorsFilter(source);

}

}

*#修改tomcat的默认的端口号，将8080改为8888*

server.port=84

server.context-path=/budget

*#jsp支持*

spring.mvc.view.prefix: /WEB-INF/jsp/

spring.mvc.view.suffix: .jsp

application.message: Hello Phil

*# 数据库访问配置*

*# 主数据源，默认的*

*#spring.datasource.type=com.alibaba.druid.pool.DruidDataSource*

spring.datasource.url=jdbc:mysql://118.190.132.68:3306/budget?characterEncoding=utf8&useSSL=true

spring.datasource.username=root

spring.datasource.password=tiger

spring.datasource.driverClassName = com.mysql.jdbc.Driver

*# 下面为连接池的补充设置，应用到上面所有数据源中*

*# 初始化大小，最小，最大*

spring.datasource.initialSize=5

spring.datasource.minIdle=5

spring.datasource.maxActive=20

*# 配置获取连接等待超时的时间*

spring.datasource.maxWait=60000

*# 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒*

spring.datasource.timeBetweenEvictionRunsMillis=60000

*# 配置一个连接在池中最小生存的时间，单位是毫秒*

spring.datasource.minEvictableIdleTimeMillis=300000

spring.datasource.validationQuery=SELECT 1 FROM DUAL

spring.datasource.testWhileIdle=true

spring.datasource.testOnBorrow=false

spring.datasource.testOnReturn=false

*# 打开PSCache，并且指定每个连接上PSCache的大小*

spring.datasource.poolPreparedStatements=true

spring.datasource.maxPoolPreparedStatementPerConnectionSize=20

*# 配置监控统计拦截的filters，去掉后监控界面sql无法统计，'wall'用于防火墙*

spring.datasource.filters=stat,wall,log4j

*# 通过connectProperties属性来打开mergeSql功能；慢SQL记录*

spring.datasource.connectionProperties=druid.stat.mergeSql=true;druid.stat.slowSqlMillis=5000

*# 合并多个DruidDataSource的监控数据*

spring.datasource.useGlobalDataSourceStat=true

*#JPA Configuration:*

spring.jpa.database=MYSQL

*# Show or not log for each sql query*

spring.jpa.show-sql=true

spring.jpa.generate-ddl=true

*# Hibernate ddl auto (create, create-drop, update)*

*#spring.jpa.hibernate.ddl-auto=create*

*#spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect*

spring.jpa.hibernate.naming\_strategy=org.hibernate.cfg.ImprovedNamingStrategy

*#spring.jpa.database=org.hibernate.dialect.MySQL5InnoDBDialect*

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

*# 这是CAS服务的地址*

*#更改为CAS服务器的地址*

shiro.cas=http://118.190.132.68/cas-server

*#应用地址*

shiro.server=http://118.190.132.68:84/budget

*#shiro.server=http://192.168.1.67:84/budget*

*#spring.mvc.view.prefix=/WEB-INF/view/*

*#spring.mvc.view.suffix=.jsp*

*#spring.resources.static-locations=classpath:/resources/,classpath:/static/*

*<?*xml version="1.0" encoding="UTF-8"*?>*

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.jwcq</groupId>

<artifactId>budget</artifactId>

<version>0.0.2-SNAPSHOT</version>

<packaging>jar</packaging>

<name>budget</name>

<description>jwcq budget project</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.3.RELEASE</version>

<relativePath/> *<!-- lookup parent from repository -->*

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.7</java.version>

<shiro.version>1.2.4</shiro.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

*<!--<exclusions>-->*

*<!--<exclusion>-->*

*<!--<groupId>org.springframework.boot</groupId>-->*

*<!--<artifactId>spring-boot-starter-tomcat</artifactId>-->*

*<!--</exclusion>-->*

*<!--</exclusions>-->*

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

*<!-- 下面两个引入为了操作数据库 -->*

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<scope>provided</scope>

</dependency>

*<!--alibaba 插件-->*

<dependency>

<groupId>com</groupId>

<artifactId>alibaba</artifactId>

<version>1.0.1.RELEASE</version>

</dependency>

*<!--支持shiro-->*

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-spring</artifactId>

<version>${shiro.version}</version>

</dependency>

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-cas</artifactId>

<version>${shiro.version}</version>

</dependency>

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-ehcache</artifactId>

<version>${shiro.version}</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<configuration> <useSystemClassLoader>false</useSystemClassLoader>

</configuration>

</plugin>

</plugins>

</build>

</project>