public class BaseServlet extends HttpServlet {

    @Override

    protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        System.out.println("BaseServlet被执行");

        String uri = req.getRequestURI();

        String methodName = uri.substring(uri.lastIndexOf('/') + 1);

        try {

            /\*获取method对象\*/

            Method method = this.getClass().getMethod(methodName, HttpServletRequest.class, HttpServletResponse.class);

            method.invoke(this,req,resp);

        } catch (NoSuchMethodException e) {

            e.printStackTrace();异常都应该删掉

        } catch (IllegalAccessException e) {

            e.printStackTrace();

        } catch (InvocationTargetException e) {

            e.printStackTrace();

        }

    }

    public void writeValue(Object obj,HttpServletResponse response) throws IOException {

        ObjectMapper mapper = new ObjectMapper();

        response.setContentType("application/json;charset=utf-8");

        mapper.writeValue(response.getOutputStream(),obj);

    }

    public String writeValueString(Object obj) throws JsonProcessingException {

        ObjectMapper mapper = new ObjectMapper();

        return mapper.writeValueAsString(obj);

    }

}

UserServlet注册

//验证校验

        String check = request.getParameter("check");

        //从sesion中获取验证码

        HttpSession session = request.getSession();

        String checkcode\_server = (String) session.getAttribute("CHECKCODE\_SERVER");

        session.removeAttribute("CHECKCODE\_SERVER");//为了保证验证码只能使用一次

        //比较

        if(checkcode\_server == null || !checkcode\_server.equalsIgnoreCase(check)){

            //验证码错误

            ResultInfo info = new ResultInfo();

            //注册失败

            info.setFlag(false);

            info.setErrorMsg("验证码错误");

            //将info对象序列化为json

           writeValue(info,response);

            return;

        }

        //1.获取数据

        Map<String, String[]> map = request.getParameterMap();

        //2.封装对象

        User user = new User();

        try {

            BeanUtils.populate(user,map);

        } catch (IllegalAccessException e) {

        } catch (InvocationTargetException e) {

        }

        //3.调用service完成注册

        //UserService service = new UserServiceImpl();

        boolean flag = service.regist(user);

        ResultInfo info = new ResultInfo();

        //4.响应结果

        if(flag){

            //注册成功

            info.setFlag(true);

        }else{

            //注册失败

            info.setFlag(false);

            info.setErrorMsg("注册失败!");

        }

      writeValue(info,response);

UserServiceImpl

 public boolean regist(User user) {

        User u = userDao.findByUsername(user.getUsername());

        if(u!=null){

            return false;

        }

        user.setCode(UuidUtil.getUuid());

        user.setStatus("N");

       userDao.sava(user);

        String content ="<a href='http://localhost/travel/user/active?code="+user.getCode()+"'>点击激活【黑马旅游网】</a>";

        MailUtils.sendMail(user.getEmail(),content,"激活邮件");

        return true;

}

UserDao

  public User findByUsername(String username) {

        User user = null;

        try {

            //1.定义sql

            String sql = "select \* from tab\_user where username = ?";

            //2.执行sql

            user = template.queryForObject(sql, new BeanPropertyRowMapper<User>(User.class), username);

        } catch (Exception e) {

        }

        return user;

    }

    @Override

    public void sava(User user) {

        String sql = "insert into tab\_user(username,password,name,birthday,sex,telephone,email,status,code) values(?,?,?,?,?,?,?,?,?)";

        //2.执行sql

        template.update(sql, user.getUsername(),

                user.getPassword(),

                user.getName(),

                user.getBirthday(),

                user.getSex(),

                user.getTelephone(),

                user.getEmail(),

                user.getStatus(),

                user.getCode()

        );

    }

Regist

function checkUserName() {

            var username = $("#username").val()

            var reg\_username = /^\w{8,20}$/

            var flag = reg\_username.test(username)

            if (flag) {

                $("#username").css("border", "")

            } else {

                $("#username").css("border", "1px solid red")

            }

            return flag

        }

        function checkPassWord() {

            var password = $("#password").val()

            var reg\_password = /^\w{8,20}$/

            var flag = reg\_password.test(password)

            if (flag) {

                $("#password").css("border", "")

            } else {

                $("#password").css("border", "1px solid red")

            }

            return flag

        }

        function checkEmail(){

            var email = $("#email").val()

            var reg\_email = /^\w+@\w+\.\w+$/

            var flag = reg\_email.test(email)

            if (flag){

                $("#email").css("border","")

            }else {

                $("#email").css("border","1px solid red")

            }

            return flag

        }

        $(function () {

            $("#registerForm").submit(function () {

                if(checkUserName()&&checkPassWord()&&checkEmail()){

                    $.post("user/regist",$(this).serialize(),function (data) {

                        if(data.flag){

                            location.href="register\_ok.html"

                        }else {

                            $("#errorMsg").html("注册失败")

                        }

                    })

                }

                return false 不要写错位置了

            })

            $("#username").blur(checkUserName)

            $("#password").blur(checkPassWord)

            $("#email").blur(checkEmail)

        })

激活

 String code = request.getParameter("code");

        if(code != null){

            //2.调用service完成激活

            //UserService service = new UserServiceImpl();

            boolean flag = service.active(code);

            //3.判断标记

            String msg = null;

            if(flag){

                //激活成功

                msg = "激活成功，请<a href='http://localhost/travel/login.html'>登录</a>";

            }else{

                //激活失败

                msg = "激活失败，请联系管理员!";

            }

            response.setContentType("text/html;charset=utf-8");

            response.getWriter().write(msg);

        }

UserServiceImpl

  User user = userDao.findByCode(code);

        if(user!=null){

            userDao.updataStatus(user);

            return  true;

        }else {

            return false;

        }

 <a href="javascript:location.href='exitServlet';">退出</a>

 request.getSession().invalidate();

        //2.跳转登录页面

 response.sendRedirect(request.getContextPath()+"/login.html");

login

$(function () {

            $("#btn\_sub").click(function () {

                $.post("user/login",$("#loginForm").serialize(),function (data) {

                    if(data.flag){

                        location.href="index.html"

                    }else {

                        $("#errorMsg").html(data.errorMsg)

                    }

                })

            })

        })

  //1.获取用户名和密码数据

        Map<String, String[]> map = request.getParameterMap();

        //2.封装User对象

        User user = new User();

        try {

            BeanUtils.populate(user,map);

        } catch (IllegalAccessException e) {

            e.printStackTrace();

        } catch (InvocationTargetException e) {

            e.printStackTrace();

        }

        //3.调用Service查询

        // UserService service = new UserServiceImpl();

        User u  = service.login(user);

        ResultInfo info = new ResultInfo();

        //4.判断用户对象是否为null

        if(u == null){

            //用户名密码或错误

            info.setFlag(false);

            info.setErrorMsg("用户名密码或错误");

        }

        //5.判断用户是否激活

        if(u != null && !"Y".equals(u.getStatus())){

            //用户尚未激活

            info.setFlag(false);

            info.setErrorMsg("您尚未激活，请激活");

        }

        //6.判断登录成功

        if(u != null && "Y".equals(u.getStatus())){

            request.getSession().setAttribute("user",u);//登录成功标记

            //登录成功

            info.setFlag(true);

        }

        //响应数据

       writeValue(info,response);

 return userDao.findByUsernameAndPassword(user.getUsername(),user.getPassword());

前端

$.get("user/findOne",{},function (data) {

            var msg ="欢迎回来"+data.name

            $("#span\_username").html(msg)

        })

 //从session中获取登录用户

        Object user = request.getSession().getAttribute("user");

        writeValue(user,response);

前端

$.get("category/findAll", {}, function (data) {

            var lis = "<li class=\"nav-active\"><a href=\"index.html\">首页</a></li>"

            for (var i = 0; i < data.length; i++) {

                lis += ' <li><a href="route\_list.html?rid=' + data[i].cid + '&rname=">' + data[i].cname + '</a></li>'

            }

            lis += "<li><a href=\"favoriterank.html\">收藏排行榜</a></li>"

            $("#category").html(lis)

        })

Servlet

        List<Category> cs = service.findAll();

        writeValue(cs,response);

service

  Jedis jedis = JedisUtil.getJedis();

        Set<Tuple> categorys = jedis.zrangeWithScores("category", 0, -1);

        List<Category> cs =null;

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

            System.out.println("从数据库里查询");

             cs = categoryDao.findAll();

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

                jedis.zadd("category",cs.get(i).getCid(),cs.get(i).getCname());

            }

        }else {

            System.out.println("从redis里查询");

            cs = new ArrayList<Category>();

            for (Tuple tuple :categorys){

                Category category = new Category();

                category.setCname(tuple.getElement());

                category.setCid((int)tuple.getScore());

                cs.add(category);

            }

        }

        /\*return categoryDao.findAll();\*/

        return cs;

dao

 public List<Category> findAll() {

        String sql = "select \* from tab\_category ";

        return template.query(sql,new BeanPropertyRowMapper<Category>(Category.class));

    }

Flushall清除缓存

$.get("category/findAll", {}, function (data) {  
 var lis = "<li class=\"nav-active\"><a href=\"index.html\">首页</a></li>"  
 for (var i = 0; i < data.length; i++) {  
 lis += ' <li><a href="route\_list.html?rid=' + data[i].cid + '&rname=">' + data[i].cname + '</a></li>'  
 }  
 lis += "<li><a href=\"favoriterank.html\">收藏排行榜</a></li>"  
 $("#category").html(lis)  
})

前端

 $(function () {

            var cid = getParameter("cid")

            var rname =getParameter("rname")

            if (rname){

                rname = window.decodeURIComponent(rname)

            }

            load(cid,null,rname)

        })

       function load(cid,currentPage,rname) {

           $.get("route/pageQuery",{cid:cid,currentPage:currentPage,rname:rname},function (pb) {

               $("#totalPage").html(pb.totalPage)

               $("#totalCount").html(pb.totalCount)

               var lis=""

               var firstPage='<li onclick="javascript:load('+cid+',1,\''+rname+'\')"><a href="javascript:void(0)">首页</a></li>'

               var beforeNum = pb.currentPage-1

               if (beforeNum<1){

                   beforeNum=1

               }

               var beforePage='<li onclick="javascript:load('+cid+','+beforeNum+',\''+rname+'\')" class="threeword"><a href="javascript:void(0)">上一页</a></li>'

               lis+=firstPage

               lis+=beforePage

               var begin

               var end

               if(pb.totalPage<10){

                   begin=1

                   end=pb.totalPage

               }else {

                   begin=pb.currentPage-5

                   end=pb.currentPage+4

                   /\*如果前面没有5个\*/

                   if(begin<1){

                       begin=1

                       end=begin+9

                   }

                   if(end>pb.totalPage){

                       end=pb.totalPage

                       begin=end-9

                   }

               }

               for (var i = begin; i <=end; i++) {

                    var li

                   if(currentPage==i){

                       li='<li onclick="javascript:load('+cid+','+i+',\''+rname+'\') "class="curPage"><a href="javascript:void(0)">'+i+'</a></li>'

                   }else {

                       li='<li onclick="javascript:load('+cid+','+i+',\''+rname+'\')"><a  href="javascript:void(0)">'+i+'</a></li>'

                   }

                   lis+=li

               }

               var nextNum = pb.currentPage+1

               if (nextNum>pb.totalPage){

                   nextNum=pb.totalPage

               }

               var nextPage=' <li onclick="javascript:load('+cid+','+nextNum+',\''+rname+'\')" class="threeword"><a href="javascript:void(0)">下一页</a></li>'

               var lastPage=' <li onclick="javascript:load('+cid+','+pb.totalPage+',\''+rname+'\')" class="threeword"><a href="javascript:void(0)">末页</a></li>'

               lis+=nextPage

               lis+=lastPage

               $("#pageNum").html(lis)

               var route\_list=""

               for (var i = 0; i < pb.list.length; i++) {

                   var route=pb.list[i]

                   var li ='<li style="height: 200px">\n' +

                       '                            <div class="img"><img src="'+route.rimage+'" alt="" width="299px"></div>\n' +

                       '                            <div class="text1">\n' +

                       '                                <p>'+route.rname+'</p>\n' +

                       '                                <br/>\n' +

                       '                                <p style="margin-top: -10px">'+route.routeIntroduce+'</p>\n' +

                       '                            </div>\n' +

                       '                            <div class="price">\n' +

                       '                                <p class="price\_num">\n' +

                       '                                    <span>&yen;</span>\n' +

                       '                                    <span>'+route.price+'</span>\n' +

                       '                                    <span>起</span>\n' +

                       '                                </p>\n' +

                       '                                <p><a href="route\_detail.html?rid='+route.rid+'">查看详情</a></p>\n' +

                       '                            </div>\n' +

                       '                        </li>'

                   route\_list+=li

               }

               $("#route").html(route\_list)

               window.scrollTo(0,0)

           })

       }

Servlet

WebServlet("/route/\*")

public class RouteServlet extends BaseServlet {

    private RouteService routeService = new RouteServiceImpl();

    private FavoriteService favoriteService = new FavoriteServiceImpl();

    /\*

     \* 分页查询

     \*

     \* \*/

    public void pageQuery(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String currentPageStr = request.getParameter("currentPage");

        String pageSizeStr = request.getParameter("pageSize");

        String cidStr = request.getParameter("cid");

        String rname = request.getParameter("rname");

        rname = new String(rname.getBytes("iso-8859-1"), "utf-8");

        /\*类别，也没有o的cid啊  难怪可以传入null 不传也没事\*/

        int cid = 0;

        if (cidStr != null && cidStr.length() > 0 && !"null".equals(cidStr)) {

            cid = Integer.parseInt(cidStr);

        }

        /\*当前页码\*/

        int currentPage = 0;

        if (currentPageStr != null && currentPageStr.length() > 0) {

            currentPage = Integer.parseInt(currentPageStr);

        } else {

            currentPage = 1;

        }

        /\*每页显示条数\*/

        int pageSize = 0;

        if (pageSizeStr != null && pageSizeStr.length() > 0) {

            currentPage = Integer.parseInt(pageSizeStr);

        } else {

            pageSize = 5;

        }

        PageBean<Route> pb = routeService.pageQuery(cid, currentPage, pageSize, rname);

        System.out.println("我进了后台");

        /\*区别就是要不要返回json 返回有什么用\*/

        writeValue(pb, response);

    }

Service

        PageBean<Route> pb = new PageBean<>();

        /\*设置当前页码\*/

        pb.setCurrentPage(currentPage);

        /\*设置每页数目\*/

        pb.setPageSize(pageSize);

        /\*设置cid为这个的总记录数\*/

        int totalCount = routeDao.findTotalCount(cid,rname);

        pb.setTotalCount(totalCount);

        /\*当前页的数据\*/

        int start =(currentPage -1)\*pageSize;

        List<Route> list = routeDao.findByPage(cid,start,pageSize,rname);

        pb.setList(list);

        /\*设置总页数\*/

        int totalPage = totalCount%pageSize==0?totalCount/pageSize:(totalCount/pageSize)+1;

        pb.setTotalPage(totalPage);

        return pb;

dao

    @Override

    public int findTotalCount(int cid, String rname) {

        /\*String sql = "select count(\*) from tab\_route where cid =?";\*/

        String sql = "select count(\*) from tab\_route where 1 = 1 ";

        /\*为了方便拼接\*/

        StringBuilder sb = new StringBuilder(sql);

        /\*加条件\*/

        ArrayList params = new ArrayList();

        if (cid != 0) {

            sb.append(" and cid = ? ");

            params.add(cid);

        }

        if (rname != null && rname.length() > 0) {

            sb.append(" and rname like ? ");

            /\*模糊查询  这才是真的传入的\*/

            params.add("%" + rname + "%");

        }

        sql = sb.toString();

        /\*转数组\*/

        return template.queryForObject(sql, Integer.class, params.toArray());

    }

    @Override

    public List<Route> findByPage(int cid, int start, int pageSize, String rname) {

        /\*String sql = "select \* from tab\_route where cid =? limit ? , ?";\*/

        String sql = "select \* from tab\_route where 1 = 1 ";

        StringBuilder sb = new StringBuilder(sql);

        /\*加条件\*/

        ArrayList params = new ArrayList();

        if (cid != 0) {

            sb.append(" and cid = ? ");

            params.add(cid);

        }

        if (rname != null && rname.length() > 0) {

            sb.append(" and rname like ? ");

            /\*模糊查询  这才是真的传入的\*/

            params.add("%" + rname + "%");

        }

        sb.append(" limit ? , ? ");

        sql = sb.toString();

        params.add(start);

        params.add(pageSize);

        return template.query(sql, new BeanPropertyRowMapper<Route>(Route.class), params.toArray());

    }

 $("#search-button").click(function () {

            var cid = getParameter("cid")

            var rname = $("#search\_input").val()

            location.href = "http://localhost/travel/route\_list.html?cid="+cid+"&rname=" + rname

        })

Servlet

   /\*rid查询旅游详细信息\*/

    public void findOne(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String rid = request.getParameter("rid");

        Route route = routeService.findOne(rid);

        writeValue(route,response);

    }

Service

    @Override

    public Route findOne(String rid) {

        Route route = routeDao.findOne(Integer.parseInt(rid));

        List<RouteImg> routeImgList = routeImgDao.findByRid(Integer.parseInt(rid));

        route.setRouteImgList(routeImgList);

        Seller seller = sellerDao.findById(route.getSid());

        route.setSeller(seller);

        int count = favoriteDao.findCountByRid(route.getRid());

        route.setCount(count);

        return route;

    }

Sellerdao

    @Override

    public Seller findById(int id) {

        String sql = "select \* from tab\_seller where sid = ? ";

        return template.queryForObject(sql, new BeanPropertyRowMapper<Seller>(Seller.class), id);

    }

Routeimg

    @Override

    public List<RouteImg> findByRid(int rid) {

        String sql = "select \* from tab\_route\_img where rid = ? ";

        return template.query(sql,new BeanPropertyRowMapper<RouteImg>(RouteImg.class),rid);

    }

RouteDao

    @Override

    public Route findOne(int rid) {

        String sql = "select \* from tab\_route where rid = ?";

        return template.queryForObject(sql, new BeanPropertyRowMapper<Route>(Route.class), rid);

前端

 $(document).ready(function () {

        goImg()

        var timer = setInterval("auto\_play()", 1000);

    });

 $(function () {

        var rid = getParameter("rid")

        $.get("route/findOne", {rid: rid}, function (route) {

            $("#rname").html(route.rname)

            $("#routeIntroduce").html(route.routeIntroduce)

            $("#sname").html(route.sname)

            $("#consphone").html(route.consphone)

            $("#address").html(route.address)

            $("#price").html("￥" + route.price)

            $("#count").html("已收藏" + route.count + "次")

            var route\_list = '<a class="up\_img up\_img\_disable"></a>'

            for (var i = 0; i < route.routeImgList.length; i++) {

                if(i>4){

                    var li = ' <a style="display: none" title="" class="little\_img" data-bigpic="' + route.routeImgList[i].bigPic + '">\n' +

                        '                    <img src="' + route.routeImgList[i].smallPic + '">\n' +

                        '                </a>'

                }else {

                    var li = ' <a title="" class="little\_img" data-bigpic="' + route.routeImgList[i].bigPic + '">\n' +

                        '                    <img src="' + route.routeImgList[i].smallPic + '">\n' +

                        '                </a>'

                }

                route\_list += li

            }

            $("#dd").html(route\_list)

        })

        goImg()

    })

前端

 $(function () {

       var rid = getParameter("rid")

       $.get("route/isFavorite",{rid:rid},function (flag) {

           if (flag){

               $("#favorite").addClass("already")

               $("#favorite").attr("disabled","disabled")

               $("#favorite").removeAttr("onclick")

           }else {

           }

       })

   })

Servlet

 public void isFavorite(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String rid = request.getParameter("rid");

        User user = (User) request.getSession().getAttribute("user");

        int uid;

        if(user!=null){

            uid=user.getUid();

        }else {

            uid = 0;

        }

        boolean flag = favoriteService.isFavorite(rid, uid);

        writeValue(flag,response);

    }

Service

    @Override

    public boolean isFavorite(String rid, int uid) {

        Favorite favorite = favoriteDao.findByRidAndUid(Integer.parseInt(rid), uid);

        return favorite != null;

    }

   @Override

    public Favorite findByRidAndUid(int rid, int uid) {

        Favorite favorite = null;

        try {

            String sql = " select \* from tab\_favorite where rid = ? and uid = ?";

            favorite = template.queryForObject(sql, new BeanPropertyRowMapper<Favorite>(Favorite.class), rid, uid);

        } catch (DataAccessException e) {

        }

        return favorite;

    }

前端

    function addFavorite(){

        var rid = getParameter("rid")

        $.get("user/findOne",{},function (user){

            if (user){

                $.get("route/addFavorite",{rid:rid},function () {

                    location.reload()

                })

            }else {

                alert("您没登入，请登入")

                location.href="login.html"

            }

        })

    }

    public void addFavorite(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String rid = request.getParameter("rid");

        User user = (User) request.getSession().getAttribute("user");

        int uid;

        if (user == null){

            return;

        }else {

            uid=user.getUid();

        }

        favoriteService.add(rid,uid);

}

  public void add(String rid, int uid) {

            favoriteDao.add(Integer.parseInt(rid),uid);

    }

    @Override

    public int findCountByRid(int rid) {

        String sql = "select count(\*) from tab\_favorite where rid = ? ";

        return template.queryForObject(sql,Integer.class,rid);

    }

    @Override

    public void add(int rid, int uid) {

        String sql = "insert into tab\_favorite values(?,?,?)";

        template.update(sql,rid,new Date(),uid);

    }