

RuoYi项目常见问题

来源：<http://doc.ruoyi.vip/ruoyi/other/faq.html>

#如何不登录直接访问

在 ShiroConfig 中设置filterChainDefinitionMap 配置url=anon

```
/admins/**=anon           # 表示该 uri 可以匿名访问
/admins/**=auth           # 表示该 uri 需要认证才能访问
/admins/**=authcBasic     # 表示该 uri 需要 httpBasic 认证
/admins/**=perms[user:add:]* # 表示该 uri 需要认证用户拥有 user:add:* 权限才能访问
/admins/**=port[8080]     # 表示该 uri 需要使用 8080 端口
/admins/**=roles[admin]   # 表示该 uri 需要认证用户拥有 admin 角色才能访问
/admins/**=ssl            # 表示该 uri 需要使用 https 协议
/admins/**=user           # 表示该 uri 需要认证或通过记住我认证才能访问
/logout=logout            # 表示注销,可以当作固定配置

注意：
anon, authcBasic, authc, user 是认证过滤器。
perms, roles, ssl, rest, port 是授权过滤器。
```

#如何更换项目包路径

1、更换目录名称

```
├── xxxxx
│   ├── xxxxx-admin
│   ├── xxxxx-common
│   ├── xxxxx-framework
│   ├── xxxxx-generator
│   ├── xxxxx-quartz
│   ├── xxxxx-system
│   └── pom.xml
```

2、更换顶级目录中的pom.xml

```
<modules>
    <module>xxxxx-admin</module>
    <module>xxxxx-framework</module>
    <module>xxxxx-system</module>
    <module>xxxxx-quartz</module>
    <module>xxxxx-generator</module>
    <module>xxxxx-common</module>
</modules>
```

3、更换项目所有包名称 com.ruoyi.xxx 换成 com.xxxxx.xxx

提示
DataScopeAspect，DataSourceAspect，LogAspect 这三个类@Pointcut注解上面的包路径也需要替换com.xxxxx
CaptchaConfig 这个类验证码文本生成器参数KAPTCHA_TEXTPRODUCER_IMPL的包路径也需要替换com.xxxxx
ApplicationConfig 这个类@MapperScan注解上面的包路径也需要替换com.xxxxx

4、更换application.yml指定要扫描的Mapper类的包的路径 typeAliasesPackage包路径名称替换com.xxxxx

5、更换mapper文件的namespace包路径

```
ruoyi-system/resources/mapper/system/*
ruoyi-quartz/resources/mapper/quartz/*
ruoyi-generator/resources/mapper/generator/*
```

包路径名称替换com.xxxxx

6、更换pom文件内容

提示

以下pom.xml文件中包含ruoyi的关键字替换成xxxxx

```
├── xxxxx
│   ├── xxxxx-admin      pom.xml
│   ├── xxxxx-common     pom.xml
│   ├── xxxxx-framework pom.xml
│   ├── xxxxx-generator  pom.xml
│   ├── xxxxx-quartz     pom.xml
│   ├── xxxxx-system     pom.xml
│   └── pom.xml
```

7、更换日志路径

- 更换application.yml文件logging属性为com.xxxxx: debug
- 更换logback.xml文件为com.xxxxx

8、启动项目验证

提示

到此步骤如能正常启动，表示更换完成。剩余的小细节可以自行调整。

#如何使用多数据源

1. 在 resources 目录下修改application-druid.yml

```
# 从库数据源
slave:
  # 开启从库
  enabled: true
  url: 数据源
  username: 用户名
  password: 密码
```

1. 在Service实现中添加DataSource注解

```
@DataSource(value = DataSourceType.SLAVE)
public List<User> selectUserList()
{
    return mapper.selectUserList();
}
```

#如何更换主题皮肤

- 1、项目主页-个人信息中选择切换主题
- 2、修改主框架页-默认皮肤，在菜单参数设置修改参数键名sys.index.skinName支持如下几种皮肤
 - 蓝色 skin-blue
 - 绿色 skin-green
 - 紫色 skin-purple
 - 红色 skin-red
 - 黄色 skin-yellow
- 3、修改主框架页-侧边栏主题，在菜单参数设置修改参数键名sys.index.sideTheme支持如下几种主题
 - 深色主题theme-dark
 - 浅色主题theme-light

注：如需新增修改皮肤主题可以在skins.css中调整

#如何获取用户登录信息

1. 第一种方法

```
// 获取当前的用户信息
User currentUser = ShiroUtils.getSysUser();
// 获取当前的用户名称
String userName = currentUser.getUserName();
```

1. 第二种方法（子模块可使用）

```
// 获取当前的用户名称
String userName = (String) PermissionUtils.getPrincipalProperty("userName");
```

- 3、界面获取当前用户信息（支持任意th标签）

```
<input th:value="@permission.getPrincipalProperty('userName')">
```

- 4、js中获取当前用户信息

```
var userName = [[${@permission.getPrincipalProperty('userName')}]];
```

#如何防止请求重复提交

1. 前端通过js控制

```
// 禁用按钮
$.modal.disable();
// 启用按钮
$.modal.enable();
```

1. 后端通过@RepeatSubmit注解控制

```
/**
 * 在对应方法添加注解 @RepeatSubmit
 */
@RepeatSubmit
public AjaxResult editSave()
```

#如何配置允许跨域访问

现在开发的项目一般都是前后端分离的项目，所以跨域访问会经常使用。

- 1、单个控制器方法CORS注解

```
@RestController
@RequestMapping("/system/test")
public class TestController {

    @CrossOrigin
    @GetMapping("/{id}")
    public AjaxResult getUser(@PathVariable Integer userId) {
        // ...
    }

    @DeleteMapping("/{userId}")
    public AjaxResult delete(@PathVariable Integer userId) {
        // ...
    }
}
```

- 2、整个控制器启用CORS注解

```
@CrossOrigin(origins = "http://ruoyi.vip", maxAge = 3600)
@RestController
@RequestMapping("/system/test")
public class TestController {

    @GetMapping("/{id}")
    public AjaxResult getUser(@PathVariable Integer userId) {
        // ...
    }
}
```

```

    }

    @DeleteMapping("/{userId}")
    public AjaxResult delete(@PathVariable Integer userId) {
        // ...
    }
}

```

3、全局CORS配置（在ResourcesConfig重写addCorsMappings方法）

```

/**
 * web跨域访问配置
 */
@Override
public void addCorsMappings(CorsRegistry registry)
{
    // 设置允许跨域的路径
    registry.addMapping("/**")
        // 设置允许跨域请求的域名
        .allowedOrigins("*")
        // 是否允许证书
        .allowCredentials(true)
        // 设置允许的方法
        .allowedMethods("GET", "POST", "DELETE", "PUT")
        // 设置允许的header属性
        .allowedHeaders("*")
        // 跨域允许时间
        .maxAge(3600);
}

```

#日期插件精确到时分秒

1、界面设置时间格式data-format

```

<li class="select-time">
    <label>创建时间: </label>
    <input type="text" class="time-input" id="startTime" placeholder="开始时间" name="params[beginTime]" data-format="yyyy-MM"/>
    <span>-</span>
    <input type="text" class="time-input" id="endTime" placeholder="结束时间" name="params[endTime]" data-format="yyyy-MM"/>
</li>

```

2、通过js函数设置 datetimpicker日期控件可以设置format

```

$('.input-group.date').datetimepicker({
    format: 'yyyy-mm-dd hh:ii:ss',
    autoclose: true,
    minView: 0,
    minuteStep:1
});

```

laydate日期控件可以设置common.js 配置type=datetime

```

layui.use('laydate', function() {
    var laydate = layui.laydate;
    var startDate = laydate.render({
        elem: '#startTime',
        max: $('#endTime').val(),
        theme: 'molv',
        trigger: 'click',
        type: 'datetime',
        done: function(value, date) {
            // 结束时间大于开始时间
            if (value !== '') {

```

```

        endDate.config.min.year = date.year;
        endDate.config.min.month = date.month - 1;
        endDate.config.min.date = date.date;
    } else {
        endDate.config.min.year = '';
        endDate.config.min.month = '';
        endDate.config.min.date = '';
    }
}

});
var endDate = laydate.render({
    elem: '#endTime',
    min: $('#startTime').val(),
    theme: 'molv',
    trigger: 'click',
    type: 'datetime',
    done: function(value, date) {
        // 开始时间小于结束时间
        if (value !== '') {
            startDate.config.max.year = date.year;
            startDate.config.max.month = date.month - 1;
            startDate.config.max.date = date.date;
        } else {
            startDate.config.max.year = '';
            startDate.config.max.month = '';
            startDate.config.max.date = '';
        }
    }
});
});
});

```

#代码生成不显示新建表

默认条件需要表注释，特殊情况可在GenMapper.xml去除table_comment条件

```

<select id="selectTableByName" parameterType="String" resultMap="TableInfoResult">
    <include refid="selectGenVo"/>
    where table_comment <> '' and table_schema = (select database())
</select>

```

#提示您没有数据的权限

这种情况都属于权限标识配置不对在菜单管理配置好权限标识（菜单&按钮）

1. 确认此用户是否已经配置角色
2. 确认此角色是否已经配置菜单权限
3. 确认此菜单权限标识是否和后台代码一致

如参数管理

后台配置@RequiresPermissions("system:config:view")对应参数管理权限标识为system:config:view

注：如需要角色权限，配置角色权限字符 使用@RequiresRoles("admin")

#富文本编辑器文件上传

富文本控件采用的summernote，图片上传处理需要设置callbacks函数

```

$('#.summernote').summernote({
    height: '220px',
    lang: 'zh-CN',
    callbacks: {

```

```

        onImageUpload: function(files, editor, $editable) {
            var formData = new FormData();
            formData.append("file", files[0]);
            $.ajax({
                type: "POST",
                url: ctx + "common/upload",
                data: data,
                cache: false,
                contentType: false,
                processData: false,
                dataType: 'json',
                success: function(result) {
                    if (result.code == web_status.SUCCESS) {
                        $(obj).summernote('editor.insertImage', result.url, result.fileName);
                    } else {
                        $.modal.alertError(result.msg);
                    }
                },
                error: function(error) {
                    $.modal.alertWarning("图片上传失败。");
                }
            });
        }
    });
});

```

#富文本编辑器底部回弹

富文本控件采用的summernote，如果不需要底部回弹设置followingToolbar: false

```

$($('.summernote').summernote({
    placeholder: '请输入公告内容',
    height : 192,
    lang : 'zh-CN',
    followingToolbar: false,
    callbacks: {
        onImageUpload: function (files) {
            sendFile(files[0], this);
        }
    }
}));

```

#如何创建新的菜单页签

建新新的页签有以下两种方式（js&html）

```

// 方式1 打开新的选项卡
function dept() {
    var url = ctx + "system/dept";
    $.modal.openTab("部门管理", url);
}

// 方式2 选卡页同一页签打开
function dept() {
    var url = ctx + "system/dept";
    $.modal.parentTab("部门管理", url);
}

// 方式3 html创建
<a class="menuItem" href="/system/dept">部门管理</a>

```

#表格数据进行汇总统计

对于某些数据需要对金额，数量等进行汇总，可以配置showFootertrue表示尾部统计

```
// options 选项中添加尾部统计
showFooter: true,
// columns 中添加
{
    field : 'balance',
    title : '余额',
    sortable: true,
    footerFormatter:function (value) {
        var sumBalance = 0;
        for (var i in value) {
            sumBalance += parseFloat(value[i].balance);
        }
        return "总金额: " + sumBalance;
    }
},
```

#表格设置行列单元格样式

1、options参数中配置属性

```
rowStyle: rowStyle,
1
```

2、对应js添加响应方法（根据row或index定义规则）即可

```
function rowStyle(row, index) {
    var style = { css: { 'color': '#ed5565' } };
    return style;
}
```

#如何去除数据监控广告

服务监控中使用的Driud，默认底部有阿里的广告。如果是一个商业项目这个是很不雅也是不允许的

1. 找到本地maven库中的对应的druid-1.1.xx.jar文件，用压缩包软件打开
2. 找到support/http/resource/js/common.js, 打开找到 buildFooter 方法

```
this.buildFooter();
buildFooter : function() {
    var html ='此处省略一些相关JS代码';
    $(document.body).append(html);
},
```

1. 删除此函数和及初始方法后覆盖文件
2. 重启项目后，广告就会消失了

#如何支持多类型数据库

对于某些特殊需要支持不同数据库，参考以下支持oracle mysql配置

```
<!--oracle驱动-->
<dependency>
    <groupId>com.oracle</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.2.0.3</version>
</dependency>
```

```
# 数据源配置
spring:
    datasource:
```

```

type: com.alibaba.druid.pool.DruidDataSource
druid:
  # 主库数据源
  master:
    url: jdbc:mysql://127.0.0.1:3306/ry?useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=true&serv
    username: root
    password: password
  # 从库数据源
  slave:
    # 从数据源开关/默认关闭
    enabled: true
    url: jdbc:oracle:thin:@127.0.0.1:1521:oracle
    username: root
    password: password

```

对于不同数据源造成的驱动问题，可以删除driverClassName。会自动识别驱动

如需要针对不同数据源分页需要操作application.yml中的pagehelper配置 删除helperDialect: mysql 会自动识别数据源 新增autoRuntimeDialect=true 表示运行时获取数据源

#如何实现翻页保留选择

1. 配置checkbox选项field属性为state

```

{
  field: 'state',
  checkbox: true
},

```

1. 表格选项options添加rememberSelected

```
rememberSelected: true,
```

#如何实现跳转至指定页

1. 表格选项options添加showPageGo

```
showPageGo: true,
```

#如何自定义查询条件参数

- 1、在options中添加queryParams参数

```

var options = {
  url: prefix + "/list",
  queryParams: queryParams,
  columns: [{
    field: 'id',
    title: '主键'
  },
  {
    field: 'name',
    title: '名称'
  }]
};
$.table.init(options);

```

- 2、在当前页添加queryParams方法设置自定义查询条件如userName

```

function queryParams(params) {
  var search = $.table.queryParams(params);
  search.userName = $("#userName").val();
  return search;
}

```

请求后台参数为: pageSize、pageNum、searchValue、orderByColumn、isAsc、userName

3、如果是表格树，添加参数ajaxParams参数

```
var options = {
    code: "deptId",
    parentCode: "parentId",
    uniqueId: "deptId",
    url: prefix + "/list",
    ajaxParams: {
        "userId": "1",
        "userName": "ruoyi"
    },
    columns: [{
        field: 'id',
        title: '主键'
    },
    {
        field: 'name',
        title: '名称'
    }]
};
$.treeTable.init(options);
```

#如何降低mysql驱动版本

1、在pom.xml中properties新增节点如：

```
<mysql.version>6.0.6</mysql.version>
```

2、单应用可以不添加，多模块需要在dependencyManagement声明依赖

```
<!-- Mysql驱动包 -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>${mysql.version}</version>
</dependency>
```

注意：如果是6以下的版本需要修改application-druid.yml中driverClassName

com.mysql.jdbc.Driver 是 mysql-connector-java 5中的

com.mysql.cj.jdbc.Driver 是 mysql-connector-java 6中的

#如何配置tomcat访问日志

1、修改application.yml中的server开发环境配置

```
# 开发环境配置
server:
    # 服务器的HTTP端口，默认为80
    port: 80
    servlet:
        # 应用的访问路径
        context-path: /
    tomcat:
        # 存放Tomcat的日志目录
        basedir: D:/tomcat
        accesslog:
            # 开启日志记录
            enabled: true
            # 访问日志存放目录
            directory: logs
        # tomcat的URI编码
        uri-encoding: UTF-8
```

```
# tomcat最大线程数，默认为200
max-threads: 800

# Tomcat启动初始化的线程数，默认值25
min-spare-threads: 30
```

2、重启项目后，在D:/tomcat/logs目录就可以看到服务器访问日志了

#如何汉化系统接口Swagger

想必很多小伙伴都曾经使用过Swagger，但是打开UI界面是纯英文的界面并不太友好，作为国人还是习惯中文界面。

1. 找到m2/repository/io/springfox/springfox-swagger-ui/x.x.x/springfox-swagger-ui-x.x.x.jar
2. 修改对应springfox-swagger-ui-x.x.x.jar包内resources目录下swagger-ui.html，添加如下JS代码

```
<!-- 选择中文版 -->
<script src='webjars/springfox-swagger-ui/lang/translator.js' type='text/javascript'></script>
<script src='webjars/springfox-swagger-ui/lang/zh-cn.js' type='text/javascript'></script>
```

1. 本地修改结束后，在覆盖压缩包文件重启就实现汉化了

#如何在html页面格式化日期

Thymeleaf主要使用org.thymeleaf.expression.Dates 这个类来处理日期，在模板中使用"#dates"来表示这个对象。

1、格式化日期

```
[[#{#dates.format(date)}]] 或 th:text="${#dates.format(date)}"
[[#{#dates.formatISO(date)}]] 或 th:text="${#dates.formatISO(date)}"
[[#{#dates.format(date, 'yyyy-MM-dd HH:mm:ss')}]] 或 th:text="${#dates.format(date, 'yyyy-MM-dd HH:mm:ss')}"
```

2、获取日期字段

```
获取当前的年份: [[#{#dates.year(date)}]]
获取当前的月份: [[#{#dates.month(date)}]]
获取当月的天数: [[#{#dates.day(date)}]]
获取当前的小时: [[#{#dates.hour(date)}]]
获取当前的分钟: [[#{#dates.minute(date)}]]
获取当前的秒数: [[#{#dates.second(date)}]]
获取当前的毫秒: [[#{#dates.millisecond(date)}]]
获取当前的月份名称: [[#{#dates.monthName(date)}]]
获取当前是星期几: [[#{#dates.dayOfWeek(date)-1}]]
```

#如何在表格中实现图片预览

对于某些图片需要在表格中显示，可以使用imageView方法

```
// 在columns中格式化对应相关的列属性
{
    field: 'avatar',
    title: '用户头像',
    formatter: function(value, row, index) {
        return $.table.imageView(value, '/profile/avatar');
    }
},
```

#如何去掉页脚及左侧菜单栏

1、去除页脚修改style.css

```
#content-main {
    height: calc(100%);
    overflow: hidden;
}
```

2、去左侧菜单栏（收起时隐藏左侧菜单）修改style.css

```
body.fixed-sidebar.mini-navbar #page-wrapper {
```

```

    margin: 0 0 0 0px;
}

body.body-small.fixed-sidebar.mini-navbar #page-wrapper {
    margin: 0 0 0 0px;
}

```

3、去左侧菜单栏（收起时隐藏左侧菜单）修改index.js

```

function() {
    if ($(this).width() < 769) {
        $('body').addClass('mini-navbar');
        $('.navbar-static-side').fadeIn(); // 换成 $('.navbar-static-side').hide();
        $(".sidebar-collapse .logo").addClass("hide");
    }
});

function SmoothlyMenu() {
    if (!$('body').hasClass('mini-navbar')) {
        $(".navbar-static-side").show(); // 添加显示这一行
        $('#side-menu').hide();
        $(".sidebar-collapse .logo").removeClass("hide");
        setTimeout(function() {
            $('#side-menu').fadeIn(500);
        },
        100);
    } else if ($('body').hasClass('fixed-sidebar')) {
        $(".navbar-static-side").hide(); // 添加隐藏这一行
        $('#side-menu').hide();
        $(".sidebar-collapse .logo").addClass("hide");
        setTimeout(function() {
            $('#side-menu').fadeIn(500);
        },
        300);
    } else {
        $('#side-menu').removeAttr('style');
    }
}

```

4、隐藏左侧菜单，需要添加.canvas-menu到body元素

```
<body class = "canvas-menu">
```

#如何Excel导出子对象多个字段

```

// 单个字段导出
@Excel(name = "部门名称", targetAttr = "deptName", type = Type.EXPORT)
private Dept dept;

// 多个字段导出
@Excel({
    @Excel(name = "部门名称", targetAttr = "deptName", type = Type.EXPORT),
    @Excel(name = "部门负责人", targetAttr = "leader", type = Type.EXPORT)
})
private Dept dept;

```

#单元格内容过长显示处理方法

1、使用系统自带的方法格式化处理

```

{
    field: 'remark',

```

```

        title: '备注',
        align: 'center',
        formatter: function(value, row, index) {
            return $.table.tooltip(value);
        }
    },

```

2、添加css控制

```

.select-table table {
    table-layout:fixed;
}

.select-table .table td {
    /* 超出部分隐藏 */
    overflow:hidden;
    /* 超出部分显示省略号 */
    text-overflow:ellipsis;
    /*规定段落中的文本不进行换行 */
    white-space:nowrap;
    /* 配合宽度来使用 */
    height:40px;
}

```

#表格禁用某列复选框选择方法

条件成立禁用checkbox返回（disabled:true）即可。

```

{
    checkbox: true,
    formatter: function (value, row, index) {
        if($.common.equals("ry", row.loginName)){
            return { disabled : true}
        } else {
            return { disabled : false}
        }
    }
},

```

#Linux系统验证码乱码解决方法

在云服务器（少许），或者干净的服务器上，服务器没有安装字体。

1、上传本地的 [Arial.ttf](#) 字体

2、此时执行以下三个命令：（建立字体索引信息，更新字体缓存）

```
mkfontscale mkfontdir fc-cache -fv
```

3、重新刷新你的页面