

目录

目录	1
1.1 介绍	4
huicmf-webman	4
源码地址：	4
后台演示	4
后台基本模块	4
如需更改redis配置，请查看1.2 安装的最后面的教程	4
后台登录安全验证	4
前端技术栈	4
huicmf-webman 目标	4
为应用插件提供统一的后台	4
统一用户体系	4
减少重复的开发工作	4
1.2 安装	7
安装	7
1.首先clone代码到项目目录：	7
2.在项目根目录执行：	7
3.导入并配置数据库：	7
1.命令行导入	7
2.原始sql导入	7
4.服务器放行端口	7
5.执行启动命令	9
6.愉快的访问后台吧：	9
7.配置伪静态(nginx转发[需要配置nginx反代])：	10
8.配置网站的运行目录为：public	10
9.绑定域名	10
10.可以愉快的使用绑定的域名访问啦啦啦	10
11. 更改redis配置	10
1.3 启动与env环境	11
启动方式（命令）	11
windows开发环境	11
Linux生产环境	11
平滑重启	11
.env环境	11
orm使用think的。数据库链接配置文件在：	11
nginx代理	11
伪静态	11
1.4 目录结构	13
2.1 数据库管理	14
数据库管理	14
2.2 控件及参数	19
控件及参数	19
控件参数格式	19
控件类型及参数	19
文本框/数字文本框 input	19
多行文本 textArea	20
下拉单选/下拉多选 Select	20
树形单选/树形多选 treeSelect/treeSelectMulti	20
日期选择/日期时间选择 datePicker	20

上传文件 upload	21
上传图片 uploadImage	21
图标 iconPicker	21
2.3 一键CURD	22
一键CURD（菜单）	22
说明	23
2.4 权限节点	25
权限节点	25
2.5 定时任务	26
定时任务	26
项目地址：	27
定时器格式说明:	27
简单使用	27
任务分类	27
计划任务列表	27
方法名	27
请求参数	27
返回数据	27
计划任务日志列表	28
请求参数	29
返回数据	29
添加任务	30
请求参数	30
返回数据	30
重启任务	30
请求参数	30
返回数据	30
修改任务	31
请求参数	31
返回数据	31
删除任务	31
请求参数	31
返回数据	32
2.6 菜单管理	33
菜单管理	33
字段含义	33
菜单名称	33
类型	33
排序	33
3.1 注解权限开发	34
注解权限开发	34
首先引入注解权限方法：	34
书写控制器注解内容	34
书写方法注解内容	34
开发完成后：	34
3.2 API接口开发	35
API接口开发	35
接口说明	35
api.php配置文件	35
新增一个接口	35

公共接口	35
接口调用案例（以UserController为例）：	36
3.3 后台上传开发	37
上传使用说明：	37
单图上传：	37
参数说明：	37
多图上传：	38
参数说明：	38
4.1 huicmf-webman升级pear-admin笔记	39
一、备份：	39
二、覆盖：	39
三、修改：	39
1、对比修改：pear.config.json	39
2、对比修改：pear.config.yml	39
3、复制备份中的module-js文件到对应文件夹	40
4、修改module-js部分文件	40
1、admin.js	40
2、menu.js	40
2、修改其他文件	40
四、检查：	40
五、提交：	40
1.21、nginx代理示例	41

## 1.1 介绍

### huicmf-webman

huicmf-webman 是基于webman+mysql开发的一个开源免费php后台权限管理系统。

源码地址：

[https://gitee.com/xianrenqh/huicmf\\_webman](https://gitee.com/xianrenqh/huicmf_webman) (欢迎点星星收藏)

后台演示

<http://webman.xiaohuihui.net/admin> 账号：admin 密码：admin888

### 后台基本模块

1. 管理员管理
2. 权限节点管理
3. 角色组管理
4. 后台菜单管理
5. 定时任务管理
6. 数据库管理
7. 支持一键CURD

如需更改redis配置，请查看1.2 安装的最后面的教程

### 后台登录安全验证

1. 5分钟内连续错误超过5次，限制此ip再次访问。（需要在后台设置里（或者数据库中）解除ip）
2. N分钟内连续N次输入密码错误，此账号将禁止登录。（过去N分钟之后才能再次登录）
3. 2中的N可以在环境变量.env中设置（默认均为10）

## 前端技术栈

前端使用pear-admin-layui，开发者需要熟悉Layui组件库

[Layui官方手册](#) [Layui官方示例](#) [Pear Admin Layui](#) [Pear Admin Layui文档](#)

## huicmf-webman 目标

### 为应用插件提供统一的后台

应用插件作为webman的一个重要生态，需要有一个统一的admin后台，所有应用插件可以接入这个后台，实现各个应用插件功能统一在一个后台管理

### 统一用户体系

huicmf-webman 提供一个用户表，各个应用插件基于此表开发应用插件，用户数据会自然打通，实现应用插件之间的互通。

### 减少重复的开发工作

huicmf-webman 提供通用的功能，如数据库管理、一键菜单、权限管理、用户管理等，使用huicmf-webman开发者无需重复开发这些模块，便可以快速拥有一个简单的管理后台

预览

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理应用管理开发辅助

清除缓存

admin

首页

快速方式

清除缓存

临时文件

系统配置

个人信息

系统官网

联系客服

加入群聊

开发手册

数据统计

管理0

会员0

文章0

商品0

个人信息

运行环境

服务器环境

Linux

上传限制50M

PHP版本8.0.20

数据库版本5.7.37-log

服务器域名/IPwebman.xiaohuihui.net

服务器时间2022年12月29日 11:20:18

剩余空间42420.89M

登录名称: admin

昵称: 超级管理员

角色组名: 超级管理员

邮箱地址: admin1@admin.com

登录地址: 1.193.37.99

登录时间: 2022-12-29 11:20:15

您好! admin

Released under the MIT license.

Copyright © 2021-2022 huicmf

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理应用管理开发辅助

清除缓存

admin

数据库管理

创建表

立即备份

优化表

修复表

数据还原

表名

备注

类型

编码

数据量

操作

casbin\_rule策略规则表InnoDButf8mb4\_general\_ci25

cmf\_admin管理员表InnoDButf8mb4\_general\_ci1

cmf\_auth\_group分组表InnoDButf8mb4\_general\_ci2

cmf\_auth\_menu系统菜单表InnoDButf8mb4\_general\_ci21

cmf\_auth\_rule节点表InnoDButf8mb4\_general\_ci71

cmf\_config系统配置InnoDButf8mb4\_general\_ci30

cmf\_login\_log后台登录记录表InnoDButf8mb4\_general\_ci625

cmf\_message站内消息InnoDButf8mb4\_general\_ci42

cmf\_options选项表InnoDButf8mb4\_general\_ci17

cmf\_plugin\_linkInnoDButf8mb4\_general\_ci0

cmf\_system\_log后台操作日志表InnoDButf8mb4\_general\_ci3086

cmf\_upload\_fileInnoDButf8mb4\_general\_ci0

Released under the MIT license.

Copyright © 2021-2022 huicmf

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理应用管理开发辅助

清除缓存

admin

后台菜单管理

+ 新增菜单

	权限名称	图标	菜单url	类型	状态	排序	操作
1	系统管理		/admin/Auth	模块	启用	0	添加下级 编辑 删除
2	管理员管理		/admin/system/admin	菜单	启用	90	添加下级 编辑 删除
3	管理员列表		/admin/system/admin/index	菜单	启用	100	添加下级 编辑 删除
4	角色组管理		/admin/system/authgroup	菜单	启用	90	添加下级 编辑 删除
5	角色组列表		/admin/system/authgroup/l...	菜单	启用	100	添加下级 编辑 删除
6	后台菜单管理		/admin/system/authmenu	菜单	启用	90	添加下级 编辑 删除
7	后台菜单列表		/admin/system/authmenu/l...	菜单	启用	100	添加下级 编辑 删除
8	权限节点管理		/admin/system/node	菜单	启用	91	添加下级 编辑 删除
9	权限节点列表		/admin/system/node/index	菜单	启用	100	添加下级 编辑 删除
10	系统设置		/admin/system/config	菜单	启用	97	添加下级 编辑 删除
11	系统设置		/admin/system/config/index	菜单	启用	100	添加下级 编辑 删除
12	自定义设置		/admin/system/config/cust...	菜单	启用	97	添加下级 编辑 删除

Released under the MIT license. Copyright © 2021-2022 huicmf

## 1.2 安装

### 安装

注意 要求 php>=8.0.2 ; webman>=1.4 ; mysql>=5.5 ; 开启redis

#### 1.首先clone代码到项目目录：

```
https://gitee.com/xianrenqh/huicmf_webman.git
```

#### 2.在项目根目录执行：

```
composer update
```

#### 3.导入并配置数据库：

导入数据并配置数据库文件方式有2种。任意一种即可。

##### 1.命令行导入

在命令输入以下命令并执行：

```
# 简洁命令
php webman config:mysql --u root --p 123456 --d databasename
```

```
#完整命令
php webman config:mysql --h 127.0.0.1 --port 3306 --u root --p 123456 --d databasename --in 0
```

命令行中参数意义：

参数	是否必填	默认值	描述
--h	否	127.0.0.1	host地址
--port	否	3306	端口号
--u	是		数据库用户名
--p	是		数据库密码
--d	是		数据库名
--in	否	0	是否强制安装数据库：是=1

注意：

1. 如果数据库已存在，同时想要强制（覆盖）安装，请携带参数 "--in 1" 。否则请更换数据库名
2. 如果地址和端口一致，可以不写其参数
3. 如果携带参数，那么参数后面的值必须存在。
4. 如果数据库的密码为空，此方法无法安装，原因同3。

#### 2.原始sql导入

- 1.根目录找到 install.sql 并导入数据库
- 2.设置 config/thinkorm.php 文件中的数据库配置
- 3.如果使用.env文件，请把根目录下.example.env重命名为：.env并设置.env中的数据库配置
- 4.【注：】 如果使用.env配置数据库，则 config/thinkorm.php 文件的数据库配置项不生效。

#### 4.服务器放行端口

## 1.2 安装

默认端口:8787

更改默认端口方法：

如果使用了.env文件，请在env文件中更改端口【APP\_PORT】



## 1.2 安装

如果没有使用.env文件：

请在\config\server.php 文件中更改：

```
'listen' => 'http://0.0.0.0:8787',
```

## 5.执行启动命令

(不知道命令的可以继续往下看)

## 6.愉快的访问后台吧：

<http://127.0.0.1:8787/admin>

用户名：admin

密码：123456

## 7.配置伪静态(nginx转发[需要配置nginx反代])：

```
location / {  
    proxy_set_header X-Real-IP $remote_addr;  
    proxy_set_header Host $host;  
    proxy_http_version 1.1;  
    proxy_set_header Connection "";  
    if (!-f $request_filename){  
        proxy_pass http://webman;  
    }  
}
```

## 8.配置网站的运行目录为：public

## 9.绑定域名

(不绑定域名请使用 ip+端口访问)

## 10.可以愉快的使用绑定的域名访问啦啦啦

## 11. 更改redis配置

如需更改redis配置，需要更改以下2个地方：

(个人如果另行安装其他带有redis的插件的，需要另行更改其配置)

1. 环境变量.env文件对应的redis配置
2. webman-redis-queue配置： config/plugin/webman/redis-queue/redis.php

如果没有使用.env环境变量，则另需更改以下3个地方的配置

1. \config\redis.php
2. \config\session.php
3. \config\thinkcache.php

## 1.3 启动与env环境

### 启动方式（命令）

#### windows开发环境

```
php windows.php start
```

#### Linux生产环境

```
php start.php start
php start.php start -d
```

#### 平滑重启

```
php start.php reload
```

### .env环境

请复制或者重命名一份.example.env 为 .env文件。并编辑里面的数据库配置 如果是使用命令行安装的，则无需复制并编辑数据

### orm使用think的。 数据库链接配置文件在：

```
config\thinkorm.php
```

### nginx代理

#### 伪静态

```
location / {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header Host $host;
    proxy_http_version 1.1;
    proxy_set_header Connection "";
    if (!-f $request_filename){
        proxy_pass http://webman;
    }
}
```

当webman需要直接提供外网访问时，建议在webman前增加一个nginx代理，这样有以下好处。

- 静态资源由nginx处理，让webman专注业务逻辑处理
- 让多个webman共用80、443端口，通过域名区分不同站点，实现单台服务器部署多个站点
- 能够实现php-fpm与webman架构共存
- nginx代理ssl实现https，更加简单高效
- 能够严格过滤外网一些不合法请求
- 

```
upstream webman {
    server 127.0.0.1:8787;
    keepalive 10240;
}
```

```
server {
```

### 1.3 启动与env环境

```
server_name 站点域名;
listen 80;
root /your/webman/public;

location / {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header Host $host;
    if (!-f $request_filename){
        proxy_pass http://webman;
    }
}
```

## 1.4 目录结构

```

huicmf_webman |—— app |—— admin |—— annotation |—— ControllerAnnotation.php |——
NodeAnotation.php |—— controller |—— CommonController.php |—— CurdController.php |—— dev |——
|—— IndexController.php |—— LoginController.php |—— plugin |—— system |—— middleware |——
|—— AccessControl.php |—— AuthCheckAccess.php |—— SystemLog.php |—— service |——
NodeService.php |—— view |—— admin |—— auth_group |—— auth_menu |—— config |—— crontab
|—— database |—— dev |—— file_list.html |—— footer.html |—— head.html |—— index.html |——
|—— login.html |—— message.html |—— node |—— plugin |—— welcome.html |—— api |——
controller |—— ApiController.php |—— CommonController.php |—— IndexController.php |——
UserController.php |—— UserControllerBak.php |—— middleware |—— AccessControl.php |—— BaseController.php
|—— command |—— ConfigMySQLCommand.php |—— common |—— controller |—— UploadController.php |——
|—— crontab |—— ClearLogCrontab.php |—— functions.php |—— middleware |—— model |——
Admin.php |—— AuthGroup.php |—— AuthMenu.php |—— AuthRule.php |—— Common.php |——
|—— Config.php |—— Crontab.php |—— LoginLog.php |—— Message.php |—— Options.php |——
|—— SystemLog.php |—— UploadFile.php |—— UploadGroup.php |—— User.php |—— UserLog.php |——
|—— UserToken.php |—— service |—— CacheService.php |—— MenuService.php |—— functions.php |——
|—— index |—— controller |—— IndexController.php |—— model |—— Test.php |—— view |——
middleware |—— AccessControl.php |—— AuthCheckAccess.php |—— StaticFile.php |—— queue |—— redis |——
|—— OptimizeTables.php |—— Request.php |—— composer.json |—— composer.lock |—— config |—— api.php |——
app.php |—— autoload.php |—— bootstrap.php |—— container.php |—— database.php |—— dependence.php |——
exception.php |—— huicmf.php |—— log.php |—— middleware.php |—— plugin |—— casbin |—— webman-
permission |—— tinywan |—— webman |—— console |—— push |—— redis-queue |——
xianrenqh |—— task |—— yzh52521 |—— filesystem |—— process.php |—— redis.php |—— route.php |——
|—— server.php |—— session.php |—— static.php |—— thinkcache.php |—— thinkorm.php |—— translation.php |——
|—— ueditor_icon.php |—— view.php |—— exception |—— install.sql |—— plugin |—— link |—— app |—— controller
|—— functions.php |—— model |—— view |—— config |—— app.php |—— container.php |——
middleware.php |—— view.php |—— config.html |—— config.json |—— data |—— menu.php |—— install.sql |——
|—— Link.php |—— uninstall.sql |—— process |—— Monitor.php |—— public |—— 404.html |—— databak |——
backup-20220914105649.zip |—— backup-20220914142943.zip |—— backup-20220914142948.zip |—— backup-
20221122103217.zip |—— favicon.ico |—— static |—— admin |—— error.svg |—— images |——
js |—— style |—— success.svg |—— errors |—— 404.html |—— 500.html |—— css
|—— images |—— lib |—— iceEditor |—— layui |—— pear |—— ueditor-plus-2.6 |——
|—— vue |—— water |—— mark.png |—— storage |—— pic |—— 20221227 |—— tpl |——
|—— plugin |—— app |—— config |—— config.html |—— data |—— install.sql |——
|—— uninstall.sql |—— think_jump.tpl |—— uploads |—— crawl |—— 20221227 |—— README.md |——
ssh.exe.stackdump |—— start.php |—— support |—— bootstrap |—— ThinkOrm.php |—— bootstrap.php |—— exception
|—— Handler.php |—— helpers.php |—— lib |—— Backup.php |—— Crontab.php |—— DebugTrait.php |——
|—— Env.php |—— Form.php |—— IpToAddress.php |—— Layui.php |—— Random.php |—— Sql.php |——
|—— TimeHui.php |—— Tree.php |—— Plugin.php |—— Request.php |—— Response.php |—— webman |——
windows.bat |—— windows.php

```

## 2.1 数据库管理

### 数据库管理

huicmf-webman 自带一个数据库管理，类似小型的phpmyadmin系统。提供表的浏览、创建、更改、删除，以及记录的浏览、添加、更新、删除等功能。

创建表格 点击创建表格

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

清除缓存

admin

数据库管理

创建表

立即备份

优化表

修复表

数据还原

表名	备注	类型	编码	数据量	操作
casbin_rule	策略规则表	InnoDB	utf8mb4_general_ci	25	优化 修复 编辑 一键CURD
cmf_admin	管理员表	InnoDB	utf8mb4_general_ci	1	优化 修复 编辑 一键CURD
cmf_auth_group	分组表	InnoDB	utf8mb4_general_ci	2	优化 修复 编辑 一键CURD
cmf_auth_menu	系统菜单表	InnoDB	utf8mb4_general_ci	21	优化 修复 编辑 一键CURD
cmf_auth_rule	节点表	InnoDB	utf8mb4_general_ci	71	优化 修复 编辑 一键CURD
cmf_config	系统配置	InnoDB	utf8mb4_general_ci	30	优化 修复 编辑 一键CURD
cmf_login_log	后台登录记录表	InnoDB	utf8mb4_general_ci	625	优化 修复 编辑 一键CURD
cmf_message	站内消息	InnoDB	utf8mb4_general_ci	42	优化 修复 编辑 一键CURD
cmf_options	选项表	InnoDB	utf8mb4_general_ci	17	优化 修复 编辑 一键CURD
cmf_plugin_link		InnoDB	utf8mb4_general_ci	0	优化 修复 编辑 一键CURD
cmf_system_log	后台操作日志表	InnoDB	utf8mb4_general_ci	3086	优化 修复 编辑 一键CURD
cmf_upload_file		InnoDB	utf8mb4_general_ci	0	优化 修复 编辑 一键CURD

Released under the MIT license. Copyright © 2021-2022 huicmf

设置字段等信息

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

清除缓存

admin

数据库管理

编辑

表名: cmf\_test 注释: cmf\_test

字段属性 表单属性 索引 表结构

+新增

删除

字段名称	字段备注	字段类型	长度/值	默认值	主键	自增	为空
id	主键	unsignedInteger	10	默认值 (为空啥也不填)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
username	用户名	string	255	"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
create_time	创建时间	integer	11	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
update_time	更新时间	integer	11	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
nickname	昵称	string	50	"	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

提交

重置

Released under the MIT license. Copyright © 2021-2022 huicmf

设置表单信息

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

清除缓存

admin

数据库管理

编辑

表名: cmf\_test 注释: cmf\_test

字段属性 表单属性 索引 表结构

字段名称	字段备注	控件类型	控件参数	表单显示	列表显示	支持排序	支持查询	查询类型
id	主键	数字文本框	控件参数	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	普通查询
username	用户名	文本框	lay-verify:required,place	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	范围查询
create_time	创建时间	日期时间选择	控件参数	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	普通查询
update_time	更新时间	日期时间选择	控件参数	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	普通查询
nickname	昵称	文本框	控件参数	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	普通查询
password		文本框	控件参数	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	普通查询

提交

重置

Released under the MIT license. Copyright © 2021-2022 huicmf

设置索引

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

清除缓存

admin

数据库管理

编辑

表名

cmf\_test

注释

cmf\_test

字段属性

表单属性

索引

表结构

+新增

删除

☐

索引名称

索引字段

索引类型

☐

字段名称

请选择

normal

提交

重置

Released under the MIT license. Copyright © 2021-2022 huicmf

查看表结构

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

清除缓存

admin

数据库管理

编辑

表名

cmf\_test

注释

cmf\_test

字段属性

表单属性

索引

表结构

```
CREATE TABLE `cmf_test` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键',
  `username` varchar(255) NOT NULL DEFAULT '' COMMENT '用户名',
  `create_time` int(11) NOT NULL DEFAULT '0' COMMENT '创建时间',
  `update_time` int(11) NOT NULL DEFAULT '0' COMMENT '更新时间',
  `nickname` varchar(50) NOT NULL DEFAULT '' COMMENT '昵称',
  `password` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8mb4 COMMENT='cmf_test';
```

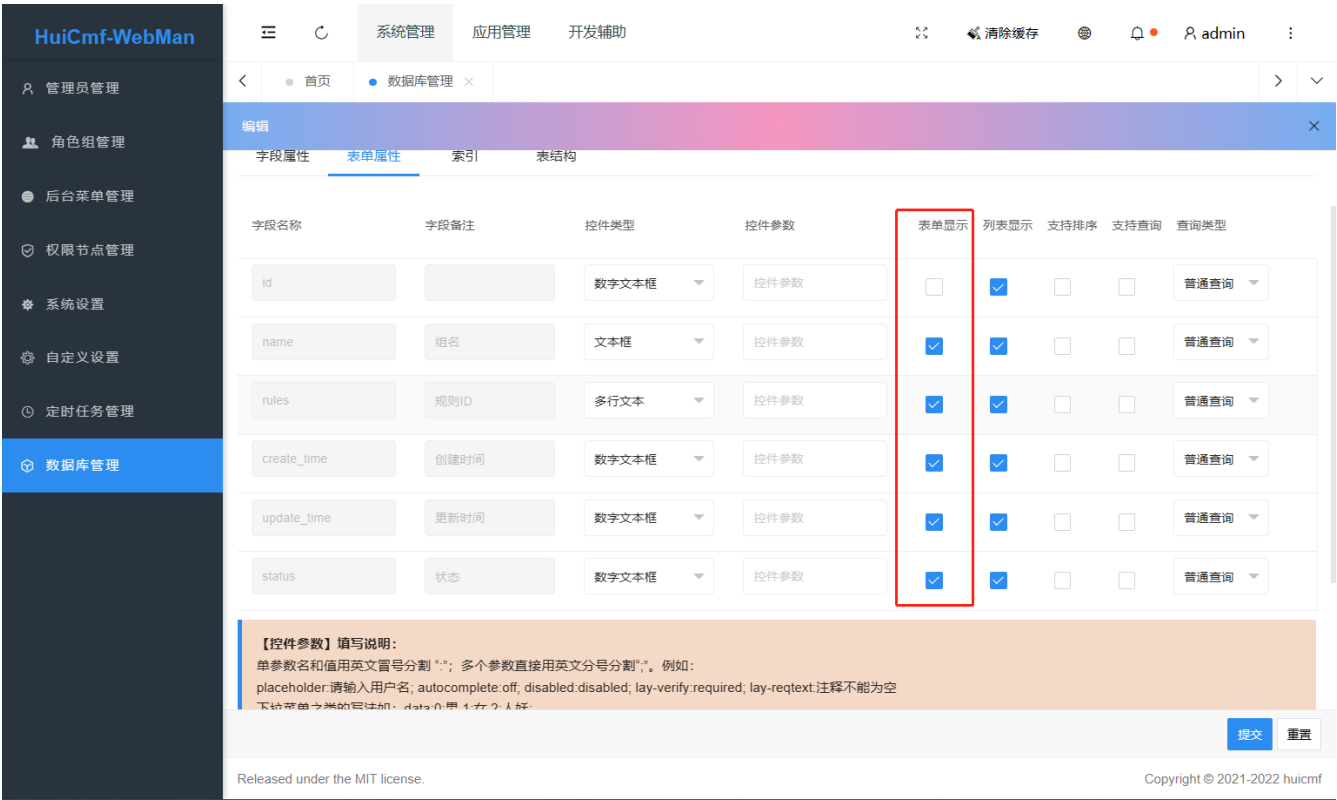
提交

重置

Released under the MIT license. Copyright © 2021-2022 huicmf

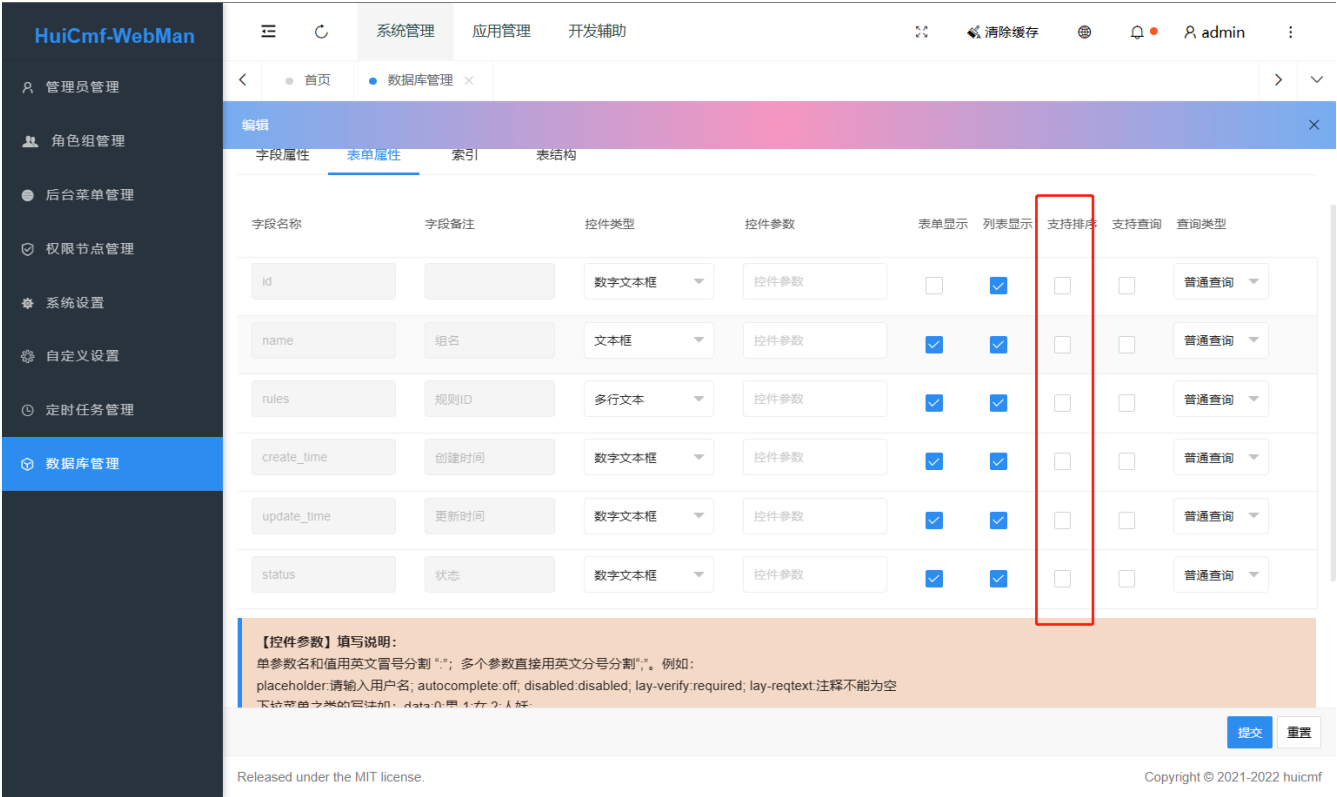
设置表单字段 有时候有些字段有默认值不需要我们在表单中填写，例如创建时间和更新时间是系统自动填写的，这时候我们需要再次到表编辑页面的“表单属性”一栏中设置



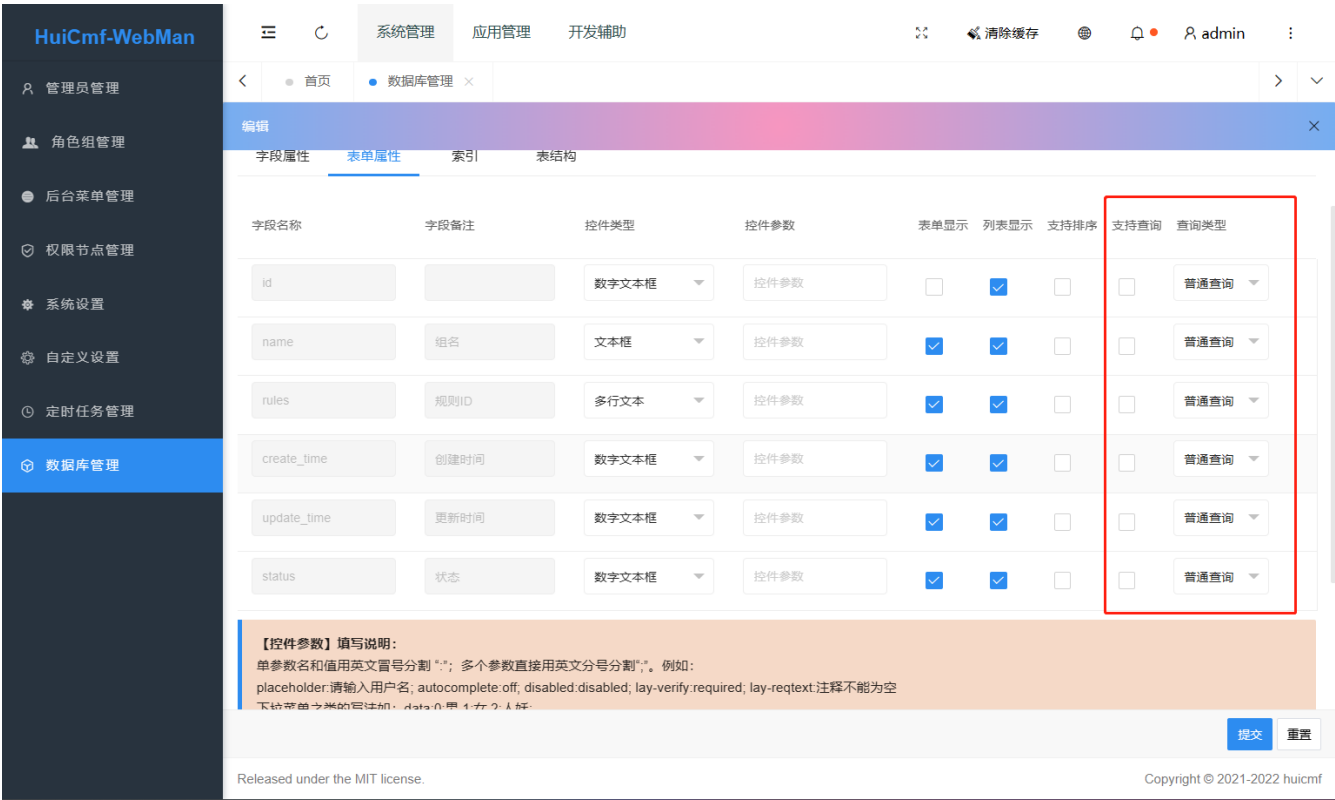


同时我们可以为表单的某个字段设置控件及控件参数，支持的控件及参数参考控件

列表排序 有时候我们希望列表能够按照某个字段排序，例如按照id排序，我们只需要在表格的表单属性里设置支持排序字段即可，例如



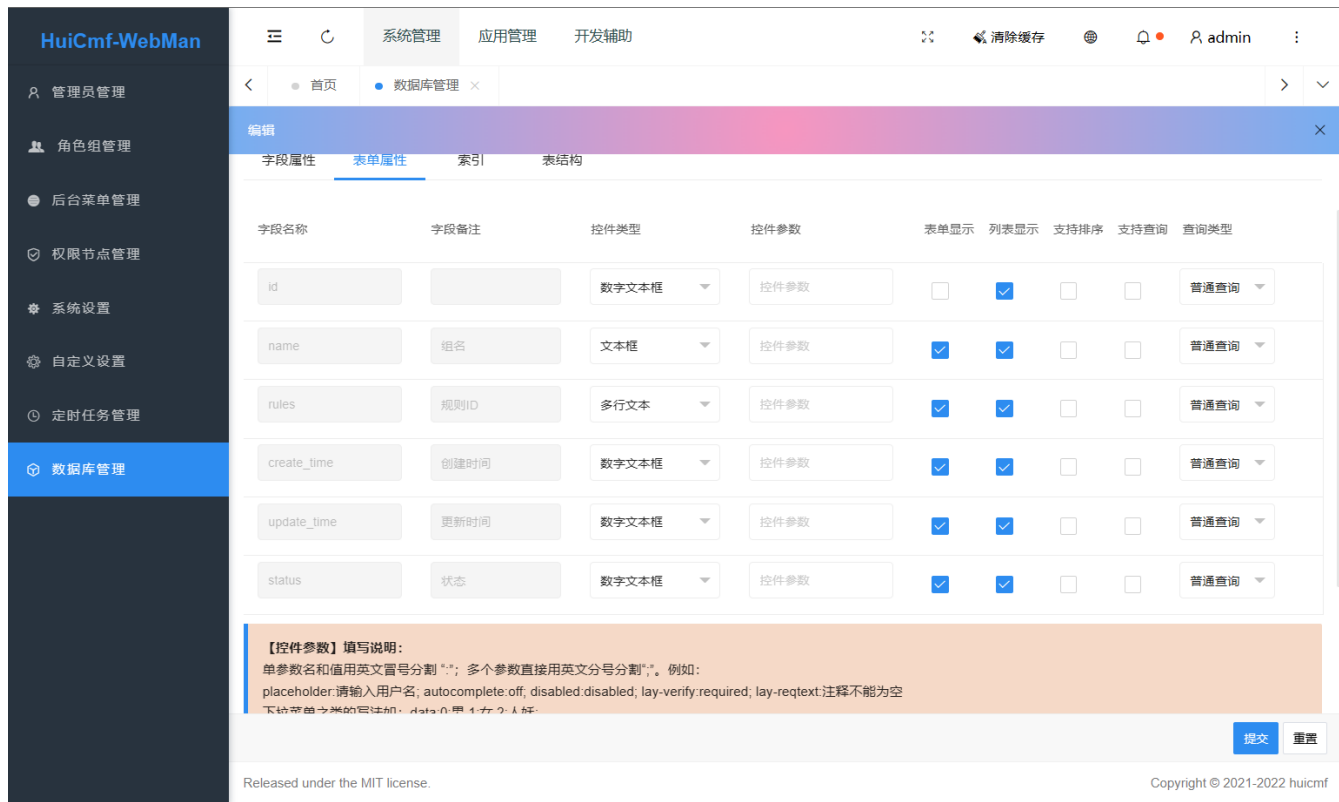
列表搜索



同样在表编辑页面的"表单属性"中可以设置某些字段是否支持查询，如上图所示，点击确定后列表页将显示创建时间、标题、用户id的搜框，如下图所示

## 2.2 控件及参数

### 控件及参数



为了方便向表中插入记录，插入表单支持设置控件类型及参数。本节将讲述表单支持哪些控件及参数。

### 控件参数格式

选项与值之间以冒号：分割，多个控件参数之间以分号；分割。例如上传控件参数可能为 url:/app/admin/upload/file;exts:zip

### 控件类型及参数

#### 文本框/数字文本框 input

参数	说明	类型	默认值
lay-verify	验证类型 required (必填项) phone (手机号) email (邮箱) url (网址) number (数字) date (日期) identity(身份证)	string	
placeholder	placeholder占位符	string	
disabled	只读	int	0
autocomplete	是否自动完成表单	int	1
type	input类型 number 或 text	string	text
value	初始值	string	

示例 lay-verify:required;placeholder:请输入

多行文本 textArea

参数	说明	类型	默认值
lay-verify	验证类型 required (必填项) phone (手机号) email (邮箱) url (网址) number (数字) date (日期) identity(身份证)	string	
placeholder	placeholder占位符		
disabled	只读	int	0
value	初始值	string	

示例 `lay-verify:required;placeholder:请输入`

下拉单选/下拉多选 Select

参数	说明	类型	默认值
url	获取options数据的地址	string	
data	options数据	string	
value	初始值	string	
示例	<code>data:0:正常,1:禁用,2:审核;value:2</code>		

如果下拉选择是静态的 则填写格式类似为 `data:0:正常,1:禁用,2:审核`

如果下拉选择是动态的 则填写格式为 `url:/app/admin/admin/select?format=select` 返回格式参考 `plugin/admin/app/controller/Crud.php` `formatSelect` 方法

如果是字典数据 则填写格式为 `url:/app/admin/dict/get/{字典名}`

树形单选/树形多选 treeSelect/treeSelectMulti

参数	说明	类型	默认值
url	返回树形结构的接口地址，格式参考表格底部说明	string	
value	初始值	string	
示例	<code>url:/app/admin/rule/select?format=tree</code>		
返回格式参考	<code>plugin/admin/app/controller/Crud.php</code> <code>formatTree</code> 方法		
注意			
由于pear-admin自带的select下拉多选组件官方不再维护，huicmf-webman下拉选择与树形选择使用的是xm-select组件，这两种组件都不是layui官方组件。			

日期选择/日期时间选择 datePicker

参数	说明	类型	默认值
lay-verify	验证类型 required (必填项)	string	
type	选择器类型可选值有 year month date time datetime	string	

format	日期格式可选值有yyyy y MM M dd d HH H mm m ss s	string	
weekStart	0周日 1周一	int	
min	最小值 如2017-1-1	string	
max	最大值 如2017-12-31	string	
position	定位方式 absolute fixed static	string	
zIndex	是否显示底部栏	int	1
calendar	是否显示公历节日	int	0
value	初始值	string	

示例 `format:date;max:2022-12-12`

上传文件 upload

参数	说明	类型	默认值	
url	上传接口	string		
accept	指定允许上传时校验的文件类型，可选值有 images(图片) file(所有文件) video(视频) audio(音频)	string		
acceptMime	MIME 类型列表 如image/*(只显示图片文件) image/jpg, image/png(只显示 jpg 和 png 文件)	string		
exts	允许上传的文件后缀 如zip	rar	7z	string
field	设定文件域的字段名	string	file_	
size	设置文件最大可允许上传的大小，单位 KB	int		
multiple	是否允许多文件上传	int	0	
value	初始值	string		

示例 `url:/app/admin/upload/file;exts:zip`

上传图片 uploadImage

参数	说明	类型	默认值	
url	上传接口	string		
exts	允许上传的文件后缀 如zip	rar	7z	string
field	设定文件域的字段名	string	file	
multiple	是否允许多文件上传	int	0	
size	设置文件最大可允许上传的大小，单位 KB	int		
value	初始值	string		

示例 `url:/app/admin/upload/avatar;size:2000`

图标 iconPicker

参数	说明	类型	初始值	
value	初始值	string		
示例	<code>value:layui-icon-set</code>			

## 2.3 一键CURD

### 一键CURD（菜单）

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

首页

定时任务管理

数据库管理

创建表

立即备份

优化表

修复表

数据还原

表名	备注	类型	编码	数据量	数据大小	操作
casbin_rule	策略规则表	InnoDB	utf8mb4_general_ci	16	16 KB	优化 修复 编辑 一键CURD
cmf_admin	管理员表	InnoDB	utf8mb4_general_ci	2	16 KB	优化 修复 编辑 一键CURD
cmf_auth_group	分组表	InnoDB	utf8mb4_general_ci	2	16 KB	优化 修复 编辑 一键CURD
cmf_auth_menu	系统菜单表	InnoDB	utf8mb4_general_ci	23	16 KB	优化 修复 编辑 一键CURD
cmf_auth_rule	节点表	InnoDB	utf8mb4_general_ci	70	16 KB	优化 修复 编辑 一键CURD
cmf_config	系统配置	InnoDB	utf8mb4_general_ci	29	16 KB	优化 修复 编辑 一键CURD
cmf_login_log	后台登录记录表	InnoDB	utf8mb4_general_ci	35	16 KB	优化 修复 编辑 一键CURD
cmf_message	站内消息	InnoDB	utf8mb4_general_ci	36	16 KB	优化 修复 编辑 一键CURD
cmf_options	选项表	InnoDB	utf8mb4_general_ci	16	64 KB	优化 修复 编辑 一键CURD
cmf_system_log	后台操作日志表	InnoDB	utf8mb4_general_ci	0	16 KB	优化 修复 编辑 一键CURD
cmf_test	cmf_test	InnoDB	utf8mb4_general_ci	6	16 KB	优化 修复 编辑 一键CURD
cmf_upload_file	附件表	InnoDB	utf8mb4_general_ci	0	16 KB	优化 修复 编辑 一键CURD

Released under the MIT license. Copyright © 2021-2022 huicmf

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

首页

定时任务管理

数据库管理

一键CURD

提示:

强制覆盖, 会强制覆盖已有的控制器和方法, 请谨慎勾选。

创建成功后请自行更新一下权限节点。

路径: 系统管理->权限节点->(强制)更新节点。

菜单名

菜单图标

上级菜单

控制器

模型

强制覆盖

提交

重置

标题

1255.25px × 828.68px

内容

段落格式

字体

字号

加粗

斜体

下划线

有序列表

无序列表

链接

插入

删除

撤销

重做

全屏

自定义标题

帮助

代码语言

代码块

表格

公式

图表

流程图

思维导图

插入

删除

撤销

重做

全屏

自定义标题

帮助

元素路径

字数统计

提交

重置

说明

支持手动指定控制器、模型的类名及位置。可将位置指定到主项目，例如控制器位置指定为/app/admin/controller/QuestionController.php，模型指定为/app/common/model/Question.php则会在主项目生成如下的文件结构

```
app/admin/ |—— controller |   —— QuestionController.php   —— view   —— question |—— index.html |—— add.html |—— edit.html
app/common/ |—— model   —— Question.php
```

提示 在一键菜单之前，请预先在表格编辑->表单属性里设置好各个字段的控件及参数，这样在生代码时，可以将控件参数一同生成，避免再次开发。表单属性编辑如下图

HuiCmf-WebMan

管理员管理

角色组管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

1473.51px × 828.68pxadmin

编辑

表名cmf\_test注释cmf\_test

字段属性表单属性索引表结构

字段名称	字段备注	控件类型	控件参数	表单显示	列表显示	支持排序	支持查询	查询类型
id	主键	数字文本框	控件参数	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	普通查询
username	用户名	文本框	lay-verify:required;placeholder:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	范围查询
create_time	创建时间	日期时间选择	控件参数	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	普通查询
update_time	更新时间	日期时间选择	控件参数	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	普通查询
nickname	昵称	文本框	控件参数	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	普通查询
password		文本框	控件参数	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	普通查询

提交重置

Released under the MIT license.Copyright © 2021-2022 huicmf

###【控件参数】填写说明：单参数名和值用英文冒号分割“:”；多个参数直接用英文分号分割“;”。例如：placeholder:请输入用户名; autocomplete:off; disabled:disabled; lay-verify:required; lay-reqtext:注释不能为空 下拉菜单之类的写法如：data:0:男,1:女,2:人妖;

###【查询类型】选择说明：1、普通查询：即，等于查询（完全匹配查询）2、范围查询：支持特殊查询，例如：<,>,,=,<>,like,in,not in。其中第一个输入框输入以上查询方法，第二个输入框输入查询关键词。3、特别说明 like查询方式：第二个字段必须加上“%”，例如：%用户%



## 2.4 权限节点

### 权限节点

权限节点，用于对角色组进行权限控制的操作。权限节点无需手工填写，使用的是控制器注释的方式（注解权限）来进行更新。注释写好后，需要手工点击更新节点或者强制更新节点来更新数据。

更新节点：普通更新节点，此方式只会更新新增的数据 强制更新节点：（推荐）强制性更新节点，此方式会更新所有新增、编辑（修改）后的数据节点 清除失效节点：更新节点不会删除已失效的（控制器的方法被删除了）的节点，需要使用此按钮进行更新失效的数据。

HuiCmf-WebMan

管理員管理

角色組管理

后台菜单管理

权限节点管理

系统设置

自定义设置

定时任务管理

数据库管理

系统管理

应用管理

开发辅助

清除缓存

admin

权限节点管理

更新节点

强制更新节点

清除失效节点

ID	类型	系统节点	节点名称	添加时间	更新时间
1	控制器	/admin/system/node	系统节点管理	2022-06-17 14:08:12	2022-06-17 14:08:12
2	节点	/admin/system/node/refr...	系统节点更新	2022-06-17 14:08:12	2022-06-17 14:08:12
3	节点	/admin/system/node/index	节点列表	2022-06-17 14:13:10	2022-06-17 14:13:10
5	节点	/admin/system/node/clea...	清除失效节点	2022-06-17 14:13:31	2022-06-17 14:13:31
6	控制器	/admin/system/admin	管理员管理	2022-06-17 14:54:29	2022-06-17 14:54:29
7	节点	/admin/system/admin/ind...	管理员列表	2022-06-17 14:54:29	2022-06-17 14:54:29
8	节点	/admin/system/admin/add	添加管理员	2022-06-17 14:54:29	2022-06-17 14:54:29
9	节点	/admin/system/admin/info	获取管理员信息	2022-06-17 14:54:29	2022-06-17 14:54:29
10	节点	/admin/system/admin/edit	更新管理员	2022-06-17 14:54:29	2022-06-17 14:54:29
11	节点	/admin/system/admin/del...	删除管理员	2022-06-17 14:54:29	2022-06-17 14:54:29
12	节点	/admin/system/admin/sta...	更改状态	2022-06-17 14:54:29	2022-06-17 14:54:29
13	节点	/admin/system/admin/pass	修改密码	2022-06-17 14:54:29	2022-06-17 14:54:29

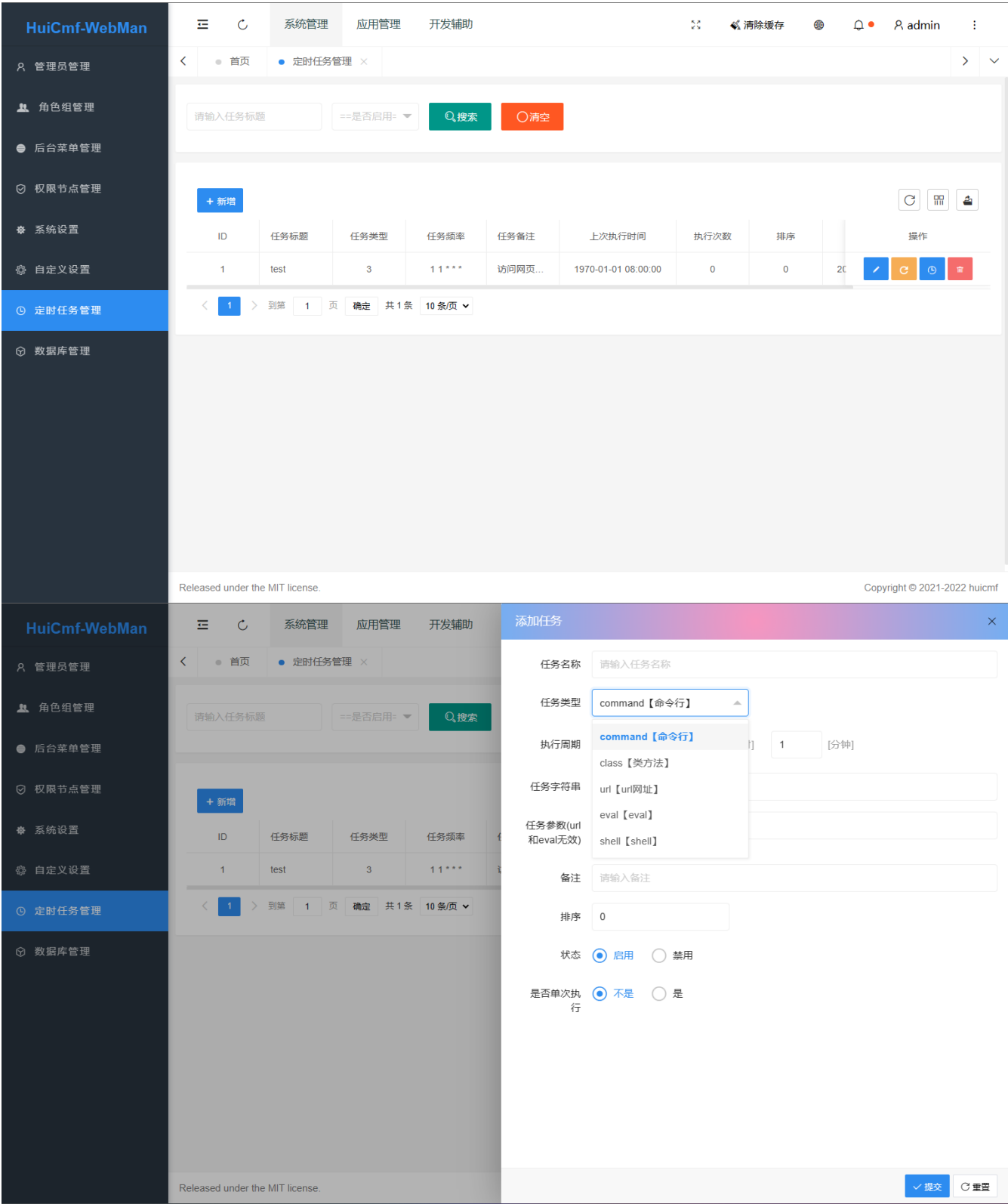
Released under the MIT license.

Copyright © 2021-2022 huicmf

## 2.5 定时任务

### 定时任务

定时任务，使用的是后台自带的定时任务，类似宝塔定时任务管理功能



项目地址：

<https://github.com/xianrenqh/webman-crontab-task>

定时器格式说明:

```
0 1 2 3 4 5
| | | | |
| | | | | +----- day of week (0 - 6) (Sunday=0)
| | | | | +----- month (1 - 12)
| | | | | +----- day of month (1 - 31)
| | | | | +----- hour (0 - 23)
| | | | | +----- min (0 - 59)
| | | | | +----- sec (0-59)[可省略，如果没有0位,则最小时间粒度是分钟]
```

简单使用

```
$param = [
    'method' => 'crontabIndex',//计划任务列表
    'args' => ['limit' => 10, 'page' => 1]//参数
];
$result= xianrenqh\Task\Client::instance()->request($param);
return json($result);
```

任务分类

- url 任务可以指定一个url地址来请求，没有什么可解释的。
- eval 任务可以直接写入 php 代码，但代码请不要带 ，具体查阅 eval 函数
- Class 任务必须指定带有 命名空间的类名，并且实现一个 public 属性的方法：方法返回值为 bool / string 类型
- Command 任务请先按照 webman 官方文档定义好执行命令，在新增任务，输入定义的 命令 即可 例如：php webman version 或者 version
- Shell 任务 在新增任务，输入定义的 shell命令 即可 例如：ps -ef | grep php

计划任务列表

方法名

method：crontabIndex

请求参数

args

参数名称	是否必须	示例	备注
page	是	1	页码
limit	是	15	每页条数

返回数据

```
{
  "code": 200,
  "msg": "ok",
  "data": {
    "total": 4,
    "per_page": 15,
    "current_page": 1,
    "last_page": 1,
    "data": [
```

```

{
  "id": 6,
  "title": "class任务 每月1号清理所有日志",
  "type": 2,
  "rule": "0 0 1 * *",
  "target": "app\\common\\crontab\\ClearLogCrontab",
  "parameter": "",
  "running_times": 71,
  "last_running_time": 1651121710,
  "remark": "",
  "sort": 0,
  "status": 1,
  "create_time": 1651114277,
  "update_time": 1651114277,
  "singleton": 1
},
{
  "id": 5,
  "title": "eavl任务 输出 hello world",
  "type": 4,
  "rule": "* * * * *",
  "target": "echo 'hello world';",
  "parameter": "",
  "running_times": 25,
  "last_running_time": 1651121701,
  "remark": "",
  "sort": 0,
  "status": 1,
  "create_time": 1651113561,
  "update_time": 1651113561,
  "singleton": 0
},
{
  "id": 3,
  "title": "url任务 打开 workerman 网站",
  "type": 3,
  "rule": "*/20 * * * *",
  "target": "https://www.workerman.net/",
  "parameter": "",
  "running_times": 39,
  "last_running_time": 1651121700,
  "remark": "请求workerman网站",
  "sort": 0,
  "status": 1,
  "create_time": 1651112925,
  "update_time": 1651112925,
  "singleton": 0
},
{
  "id": 1,
  "title": "command任务 输出 webman 版本",
  "type": 1,
  "rule": "*/20 * * * *",
  "target": "version",
  "parameter": null,
  "running_times": 112,
  "last_running_time": 1651121700,
  "remark": "20秒",
  "sort": 0,
  "status": 1,
  "create_time": 1651047480,
  "update_time": 1651047480,
  "singleton": 1
}
}
}
}

```

## 计划任务日志列表

method : crontabLog

请求参数

args

参数名称	是否必须	示例	备注
page	是	1	页码
limit	是	15	每页条数
crontab_id	否	1	计划任务ID

返回数据

```
{
  "code": 200,
  "msg": "ok",
  "data": {
    "total": 97,
    "per_page": 15,
    "current_page": 1,
    "last_page": 7,
    "data": [
      {
        "id": 257,
        "crontab_id": 1,
        "target": "version",
        "parameter": "",
        "exception": "Webman-framework v1.3.11",
        "return_code": 0,
        "running_time": "0.834571",
        "create_time": 1651123800,
        "update_time": 1651123800
      },
      {
        "id": 251,
        "crontab_id": 1,
        "target": "php webman version",
        "parameter": "",
        "exception": "Webman-framework v1.3.11",
        "return_code": 0,
        "running_time": "0.540384",
        "create_time": 1651121700,
        "update_time": 1651121700
      },
      {
        "id": 246,
        "crontab_id": 1,
        "target": "php webman version",
        "parameter": "{}",
        "exception": "Webman-framework v1.3.11",
        "return_code": 0,
        "running_time": "0.316019",
        "create_time": 1651121640,
        "update_time": 1651121640
      },
      {
        "id": 244,
        "crontab_id": 1,
        "target": "php webman version",
        "parameter": "{}",
        "exception": "Webman-framework v1.3.11",
        "return_code": 0,
        "running_time": "0.493848",
        "create_time": 1651121580,
        "update_time": 1651121580
      }
    ]
  }
}
```

2.5 定时任务

```
    }
  }
}
```

添加任务

method : crontabCreate

请求参数

args

参数名称	参数类型	是否必须	示例	备注
title	text	是	输出 webman 版本	任务标题
type	text	是	1	任务类型 (1 command, 2 class, 3 url, 4 eval ,5 shell)
rule	text	是	*/3 * * * * *	任务执行表达式
target	text	是	php webman version/ version	调用任务字符串
parameter	text	否	{}	调用任务参数(url和eval无效)
remark	text	是	每3秒执行	备注
sort	text	是	0	排序
status	text	是	1	状态[0禁用; 1启用]
singleton	text	否	1	是否单次执行 [0 是 1 不是]

返回数据

```
{
  "code": 200,
  "msg": "ok",
  "data": {

  }
}
```

重启任务

method : crontabReload

请求参数

args

参数名称	参数类型	是否必须	示例	备注
id	text	是	1,2	计划任务ID 多个逗号隔开

返回数据

```
{
  "code": 200,
  "msg": "ok",
  "data": {

  }
}
```

修改任务

method：crontabUpdate

请求参数

args

参数名称	参数类型	是否必须	示例	备注
id	text	是	1	
title	text	否	输出 webman 版本	任务标题
type	text	否	1	任务类型 (1 command, 2 class, 3 url, 4 eval , 5 shell)
rule	text	否	*/3 * * * * *	任务执行表达式
target	text	否	php webman version/version	调用任务字符串
parameter	text	否	{}	调用任务参数(url和eval无效)
remark	text	否	每3秒执行	备注
sort	text	否	0	排序
status	text	否	1	状态[0禁用; 1启用]
singleton	text	否	1	是否单次执行 [0 是 1 不是]

```
//使用类【type=2】的添加案例
$request = [
    'method' => 'crontabCreate',
    'args' => [
        'title' => '检测设备心跳',
        'type' => 2,
        'rule' => '*/3 * * * * ',
        'target' => 'app\api\crontab\HeartCrontab\checkheart',
        'status' => 1,
        'remark' => '每3秒执行',
    ]
];
$result = \xianrenqh\Task\Client::instance()->request($request);

return json($result);
```

返回数据

```
{
  "code": 200,
  "msg": "ok",
  "data": {
  }
}
```

删除任务

method：crontabDelete

请求参数

args

参数名称	参数类型	是否必须	示例	备注
------	------	------	----	----

id	text	是	1,2	计划任务ID 多个逗号隔开
----	------	---	-----	---------------

返回数据

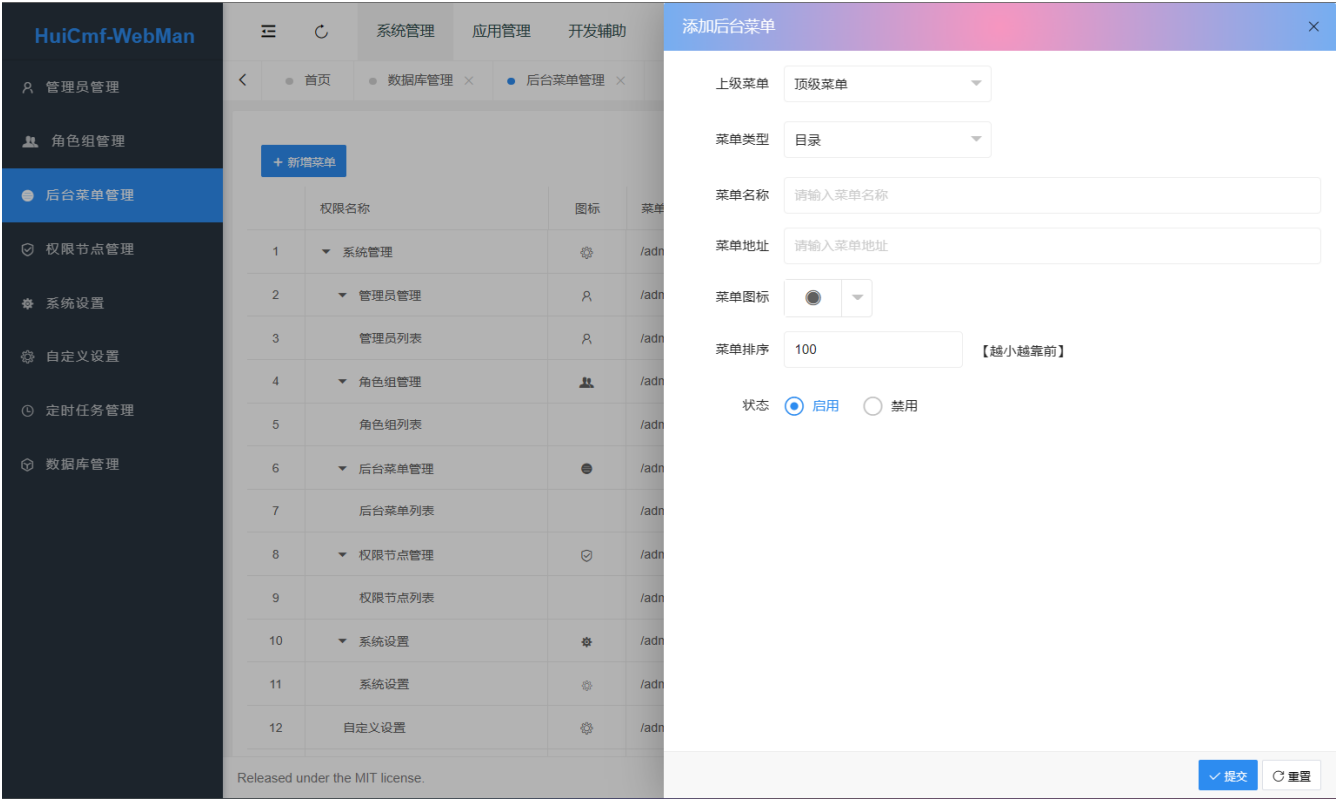
```
{
  "code": 200,
  "msg": "ok",
  "data": {

  }
}
```



## 2.6 菜单管理

### 菜单管理



### 字段含义

#### 菜单名称

也就是显示在菜单栏的名字

###上级菜单 选择上级菜单

#### 类型

分为目录、菜单

#### 排序

数字越大，位置也靠前

## 3.1 注解权限开发

### 注解权限开发

即：控制器里书写注解权限

新建控制器例如：TestController.php

首先引入注解权限方法：

```
use app\admin\annotation\ControllerAnnotation;  
use app\admin\annotation\NodeAnotation;
```

书写控制器注解内容

```
/**  
 * @ControllerAnnotation(title="测试管理")  
 * Class Node  
 * @package app\admin\controller\dev  
 */
```

title：当前控制器名字 @package：当前文件的路径位置

书写方法注解内容

```
/**  
 * @NodeAnotation(title="上传测试")  
 */  
public function upload_test(Request $request)  
{  
    return view('dev/upload_test');  
}
```

开发完成后：

开发完成后记得在后台 权限节点里 进行权限节点更新 完成。

## 3.2 API接口开发

### API接口开发

(如果不喜欢此接口开发，可以删掉api文件夹里的代码，自行开发自己的接口)

#### 接口说明

接口的访问地址为：<http://127.0.0.1:8787/api>

请求方法：post

参数格式为：method 调用具体方法

其它参数：根据每个参数来定义。

需要登录的接口，可以用\$this->userId获取当前访问用户id

```
$this->userId
```

**\*\*需要登录的接口的接口必须传递参数 token \*\***

判断登录与否的标准就是是否在本地图保存了token，如果保存了，就是登录状态，如果没有保存，就是未登录状态，需要登录的接口会自动带上token来进行请求数据。

### api.php配置文件

config/api.php

新增接口均需要在此文件中增加接口配置

#### 新增一个接口

新增接口时，需要先在api模块中增加一个接口控制器，所有接口控制器均要继承\app\api\controller\ApiController 控制器，如：

控制器名称需要使用完整名称，需要带上"Controller"，如：TestController

例如：

```
namespace app\api\controller;
class TestController extends ApiController{
    public function list(){}
}
```

然后在接口配置文件 config\api.php 中定义下新增的接口

```
'test'=>[
    'code' => 'Test',
    'method' => [
        'getlist' => [
            'code' => 'list',
            'is_login' => false
        ]
    ]
]
```

最外面的test是url地址调用的第一个参数 code对应的是接口控制器类名 method里面定义的是参数 getlist是外部访问的方法名，getlist里面code是Test控制器里面的具体方法名 is\_login 含义为是否需要登录，当is\_login 为true时，必须传token。可使用\$this->userId 获取当前登录用户

#### 公共接口

如果是一个公共数据接口，直接在 app\api\controller\CommonController.php 文件中写方法，任意请求方式访问：

<http://你的域名/api.html/common/test> 即可。

接口调用案例（以UserController为例）：

基本信息

- 接口URL：https://你的域名/api
- 请求方式：POST
- Content-Type：multipart/form-data

请求参数：

接口请求参数见下表：

Body参数说明 (multipart/form-data)

参数名	示例值	是否必填	参数描述
method	user.login	必填	接口方法
username	admin	必填	登录名
password	admin	必填	密码

响应示例

正确响应示例

```
{
  "code": 200,
  "data": "c9d2343fd754ca12a9be33e957574cce",
  "msg": ""
}
```

错误响应示例

```
{
  "code": 0,
  "msg": "没有找到此账号",
  "data": ""
}
```

## 3.3 后台上传开发

上传使用说明：

单图上传：

```
<div class="layui-form-item huicmf-upload">
  <label class="layui-form-label">头像</label>
  <div class="layui-input-inline" style="width: 50%">
    <input type="text" name="avatar" value=""
      onmouseover="hui_img_preview('avatar-select-input',this.value)"
      onmouseout="layer.closeAll();" id="avatar-select-input" autocomplete="off"
      class="layui-input">
  </div>
  <div class="layui-input-inline" style="width: 120px">
    <a class="layui-btn"
      data-open-pic="{:_url('index/file_list',{'type'=>'one','select_id'=>'avatar-select-input'})}">
      <i class="layui-icon">&#xe67c;</i>选择图片
    </a>
  </div>
</div>
```

参数说明：

1. 最顶部需要有class类名：huicmf-upload
2. input回调框的代码中，hui\_img\_preview 里面的值和id值需要相同
3. 点击按钮 a 标签 里面：

打开选择框属性：data-open-pic="{:\_url('index/file\_list',{'type'=>'one','select\_id'=>'avatar-select-input'})}"，其中：

type=>'one' 意思是：单张图

select\_id' 意思是：需要传递回调数据表单的id字段

多图上传：

```
<div class="layui-form-item huicmf-upload">
  <label class="layui-form-label">多图选择</label>
  <div class="layui-input-block" style="margin-top: 15px;">
    <a class="layui-btn" data-open-pic="{:_url('index/file_list',{'type'=>'more','select_id'=>'uploader-list'})}"><i class="layui-icon">&#xe67c;<
    <div style="margin-top: 15px;">
      <small>(可拖拽图片调整显示顺序 )</small>
    </div>
    <!--回调显示选择的图片-->
    <div class="uploader-list am-cf" id="uploader-list" data-nums="3" ></div>
  </div>
</div>
```

注意：由于增加了多选图的数量判断（限制），需要注意以下代码：

```
<div class="uploader-list am-cf" data-nums="0"></div>
```

其中：data-nums参数为多图的数量（如果是添加，默认为0，如果是编辑，请传递后台图片数量到此值）

参数说明：

1. 最顶部需要有class类名：huicmf-upload
2. input回调框的代码中，hui\_img\_preview 里面的值和id值需要相同
3. 点击按钮 a 标签 里面：

打开选择框属性：data-open-pic="{:\_url('index/file\_list',{'type'=>'more','select\_id'=>'uploader-list'})}"，其中：

type=>'more' 意思是：单张图

select\_id' 意思是：需要传递回调数据的id

## 4.1 huicmf-webman升级pear-admin笔记

鉴于友友们有升级到最新版pear-admin-layui的需求，特整理此升级文档，以备不时之需。

注意：

自行升级前一定要先做对应文件或者文件夹的备份操作。

### 一、备份：

分别备份如下文件或者文件夹：

`\public\static\lib\layui`

`\public\static\lib\pear`

### 二、覆盖：

将pear-admin的部分文件或者文件夹放置到对应的目录下\*\*（放置前请先删除原文件[已做备份]）\*\*：

1、将 `\Pear-Admin-Layui-main\component\layui` 文件夹整体放置到：`\public\static\lib`

2、【重点】

将 `\Pear-Admin-Layui-main\component\pear` 文件夹下整体放置到：`\public\static\lib`

`\public\static\lib\pear` 文件夹下的文件，一定要注意，绝大部分的升级替换是这个文件夹下的内容且有一定的修改。

此重点说明在第三部分。

3、复制备份文件夹：

layuimini文件夹 到 `\public\static\lib\pear\module`

webuploader文件夹到 `\public\static\lib\pear\module`

复制备份文件夹pear\module下的文件：

`ddSort.js`

`fileLibrary.js`

`upload_hui.js`

`webuploader.js`

`xmSelect.js`

### 三、修改：

打开文件夹：`\public\static\lib\pear`

（如果没有对应的文件，请先到自己的备份文件夹里查找）

1、对比修改：`pear.config.json`

官方默认配置文件，理论上可以不修改

2、对比修改：`pear.config.yml`

自行配置文件，这个要注意修改，有可能pear-admin会有新版本改动（新增字段等）

四、检查：

### 3、复制备份中的module-js文件到对应文件夹

因直接删除替换的官方pear-admin文件夹，所以部分自定义的module-js文件就会有丢失，需要找到自己的备份文件夹对应的文件，粘贴进去。

目录文件夹：==\public\static\lib\pear\module==

1. ddSort.js
2. fileLibrary.js
3. upload\_hui.js
4. webuploader.js

### 4、修改module-js部分文件

#### 1、admin.js

找到314-316行，注释掉

#### 2、menu.js

搜索 `getData(option.url).then(function (data) {` （大概在41行左右吧）：

修改第42行和47行：

```
option.data = data
```

为：

```
option.data = data.data;
```

=====

#### 2、修改其他文件

目前版本升级调整以上后，基本正常，如果有其他报错或者新功能新版本，请酌情对照修改。

## 四、检查：

## 五、提交：



## 1.21、nginx代理示例

官方案例：

```
upstream webman {
    server 127.0.0.1:8787;
    keepalive 10240;
}

server {
    server_name 站点域名;
    listen 80;
    access_log off;
    root /your/webman/public;

    location / {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        if (!-f $request_filename){
            proxy_pass http://webman;
        }
    }
}
```

自己整理案例：

```
upstream webman {
    server 127.0.0.1:8787;
    keepalive 10240;
}

server
{
    listen 80;
    listen [::]:80;
    server_name webman.xiaohuihui.net;
    index index.php index.html index.htm default.php default.htm default.html;
    root /www/wwwroot/webman.xiaohuihui.net/public;
    access_log off;

    #SSL-START SSL相关配置，请勿删除或修改下一行带注释的404规则
    #error_page 404/404.html;
    ssl_protocols TLSv1.1 TLSv1.2 TLSv1.3;
    ssl_ciphers EECDH+CHACHA20:EECDH+CHACHA20-draft:EECDH+AES128:RSA+AES128:EECDH+AES256:RSA+AES256:EECDH+3DES:RSA+3DES;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;
    add_header Strict-Transport-Security "max-age=31536000";
    error_page 497 https://$host$request_uri;

    #SSL-END

    #ERROR-PAGE-START 错误页配置，可以注释、删除或修改
    #error_page 404 /404.html;
    #error_page 502 /502.html;
    #ERROR-PAGE-END

    #PHP-INFO-START PHP引用配置，可以注释或修改
    include enable-php-74.conf;
    #PHP-INFO-END

    #REWRITE-START URL重写规则引用,修改后将导致面板设置的伪静态规则失效
    include /www/server/panel/vhost/rewrite/webman.xiaohuihui.net.conf;
    #REWRITE-END
```

## 1.21、nginx代理示例

```
#禁止访问的文件或目录
location ~ ^/(\.user.ini|\.htaccess|\.git|\.svn|\.project|LICENSE|README.md)
{
    return 404;
}

#一键申请SSL证书验证目录相关设置
location ~ \.well-known{
    allow all;
}

location /app
{
    proxy_pass http://127.0.0.1:3131;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "Upgrade";
    proxy_set_header X-Real-IP $remote_addr;
}

location ~ .*\.?(gif|jpg|jpeg|png|bmp|swf)$
{
    expires 30d;
    # 这里是重点
    proxy_pass http://webman;
    error_log /dev/null;
    access_log /dev/null;
}

location ~ .*\.?(js|css)?$
{
    expires 12h;
    # 这里是重点
    proxy_pass http://webman;
    error_log /dev/null;
    access_log /dev/null;
}

}
```

以上代码中，`_location ~_` 中的 一定要注意，如果没处理好这里， 可能会出现部分静态资源无法访问到情况

