目录

[1. 系统管理 1](#_Toc413788393)

[1.1 用户管理 1](#_Toc413788394)

[1.1.1 GET /user 获取用户列表 1](#_Toc413788395)

[~~1.1.2~~ ~~POST /user 新增账户信息~~ 2](#_Toc413788396)

[1.1.3 GET /user/{uid} 获取账户信息 2](#_Toc413788397)

[1.1.4 PUT /user/{uid} 修改基本账户信息 3](#_Toc413788398)

[1.1.5 PUT /user/{uid}/password 修改用户密码 3](#_Toc413788399)

[~~1.1.6 GET /user/{uid}/role 获取用户角色信息列表~~ 3](#_Toc413788400)

[1.1.7 PUT /user/{uid}/role 修改某个用户的角色 4](#_Toc413788401)

[~~1.1.8 DELETE /user/{uid}/role/{rid} 用户删除一个角色~~ 4](#_Toc413788402)

[1.1.9 DELETE /user/{uid} 删除账户信息 4](#_Toc413788403)

[1.2 养老院的角色管理（管理员权限） 4](#_Toc413788404)

[1.2.1 GET /gero/{gid}/role 获取养老院角色列表 4](#_Toc413788405)

[1.2.2 POST /gero/{gid}/role 新增养老院角色 5](#_Toc413788406)

[1.2.3 GET /gero/{gid}/role/{rid} 角色信息 5](#_Toc413788407)

[1.2.4 PUT /gero/{gid}/role/{rid} 修改角色的基本信息 5](#_Toc413788408)

[1.2.5 DELETE /gero/{gid}/role/{rid} 删除养老院角色 6](#_Toc413788409)

[1.2.6 PUT /gero/{gid}/role/{rid}/privilege 给养老院角色修改权限 6](#_Toc413788410)

[1.3 默认角色管理（超级管理员） 6](#_Toc413788411)

[1.3.1 GET /role 6](#_Toc413788412)

[1.3.2 POST /role 6](#_Toc413788413)

[1.3.3 GET /role/{rid} 6](#_Toc413788414)

[1.3.4 PUT /role/{rid} 修改默认角色的信息 6](#_Toc413788415)

[1.3.5 DELETE /role/{rid} 删除默认角色的信息 6](#_Toc413788416)

[1.3.6 PUT /role/{rid}/privilege 修改默认角色的权限 7](#_Toc413788417)

[1.4 权限管理(超级管理员) 7](#_Toc413788418)

[1.4.1 GET /privilege 权限列表 7](#_Toc413788419)

[1.4.2 POST /privilege 新增权限 7](#_Toc413788420)

[1.4.3 PUT /privilege/{pid} 修改权限 7](#_Toc413788421)

[1.4.4 DELETE /privilege/{pid} 删除权限 8](#_Toc413788422)

[1.4.5 GET /role/{rid}/privilege 获取角色的权限 8](#_Toc413788423)

[2. 养老院信息 8](#_Toc413788424)

[2.1 养老院管理（超管） 8](#_Toc413788425)

[2.1.1 GET /gero 获取养老院列表 8](#_Toc413788426)

[2.1.2 POST /gero 新增养老院基本信息 9](#_Toc413788427)

[2.1.3 GET /gero/{gid} 获取养老院基本信息 9](#_Toc413788428)

[2.1.4 PUT /gero/{gid} 修改养老院基本信息 10](#_Toc413788429)

[2.1.5 DELETE /gero/{gid} 删除养老院 10](#_Toc413788430)

[2.2 养老院区域管理 10](#_Toc413788431)

[2.2.1 GET /gero/{gid}/area 获取养老院所有的区域列表 10](#_Toc413788432)

[2.2.2 POST /gero/{gid}/area 创建一个养老院区域 11](#_Toc413788433)

[2.2.3 GET /gero/{gid}/area/{aid} 获取某区域及其子区域 11](#_Toc413788434)

[2.2.4 PUT /gero/{gid}/area/{aid} 修改某区域 12](#_Toc413788435)

[2.2.5 DELETE /gero/{gid}/area/{aid} 修改某区域 12](#_Toc413788436)

[2.3 养老院房间和床位 12](#_Toc413788437)

[3.2.1 GET /room/{area\_id} 房间信息+老人 13](#_Toc413788438)

[2.4 默认老人项目（超管） 13](#_Toc413788439)

[2.4.1 GET /care\_item 获取老人护理项目总表 13](#_Toc413788440)

[2.4.2 POST /care\_item 添加老人护理项目 14](#_Toc413788441)

[2.4.3 GET /care\_item/{cid} 获取总表中某个老人项目 14](#_Toc413788442)

[2.4.4 PUT /care\_item/{cid} 修改总表中某个老人项目 14](#_Toc413788443)

[2.4.5 DELETE /care\_item/{cid} 删除总表中某个老人项目 15](#_Toc413788444)

[2.5 默认房间项目（超管） 15](#_Toc413788445)

[2.5.1 GET /area\_item 获取房间项目总表 15](#_Toc413788446)

[2.5.2 POST /area\_item 添加房间项目 16](#_Toc413788447)

[2.5.3 GET /area\_item/{aid} 获取总表中某个房间项目 16](#_Toc413788448)

[2.5.4 PUT /area\_item/{aid} 修改总表中房间项目 16](#_Toc413788449)

[2.5.5 DELETE /area\_item/{aid} 删除总表中某个房间项目 17](#_Toc413788450)

[2.6 养老院包含的老人项目 17](#_Toc413788451)

[2.6.1 GET /gero/{gid}/care\_item 获取养老院护理项目列表 17](#_Toc413788452)

[2.6.2 POST /gero/{gid}/care\_item 添加养老院护理项目 18](#_Toc413788453)

[2.6.3 GET /gero/{gid}/care\_item/{itemid} 获取某个养老院的老人项目 18](#_Toc413788454)

[2.6.4 PUT /gero/{gid}/care\_item/{itemid} 修改某个养老院的老人项目 19](#_Toc413788455)

[2.6.5 DELETE /gero/{gid}/care\_item/{itemid} 删除养老院某个老人项目 19](#_Toc413788456)

[2.7 养老院包含的房间项目 20](#_Toc413788457)

[2.7.1 GET /gero/{gid}/area\_item 获取养老院房间项目列表 20](#_Toc413788458)

[2.7.2 POST /gero/{id}/area\_item 添加养老院房间项目 20](#_Toc413788459)

[2.7.3 GET /gero/{gid}/area\_item/{aid} 获取养老院房间项目 21](#_Toc413788460)

[PUT /gero/{gid}/area\_item/{aid} 更新养老院房间项目 21](#_Toc413788461)

[DELETE /gero/{gid}/area\_item/{aid} 删除养老院房间项目 22](#_Toc413788462)

[3. 老人相关 22](#_Toc413788463)

[3.1 老人基本信息 22](#_Toc413788464)

[3.1.1 GET /gero/{gid}/elder 获取老人列表 22](#_Toc413788465)

[3.1.2 POST /gero/{gid}/elder/ 增加老人详细信息 23](#_Toc413788466)

[3.1.3 GET /gero/{gid}/elder/{eid} 查单个老人详细信息 24](#_Toc413788467)

[3.1.4 PUT /gero/{gid}/elder/{eid} 修改单个老人详细信息 25](#_Toc413788468)

[3.1.5 DELETE /gero/{gid}/elder/{eid} 老人离院登记 26](#_Toc413788469)

[3.2老人表单信息 26](#_Toc413788470)

[3.3 老人健康数据 26](#_Toc413788471)

[3.3.1 GET /elder/{eid}/temperature 获取老人的体温数据 26](#_Toc413788472)

[3.3.2 POST /elder/{eid}/temperature 输入老人的体温数据 27](#_Toc413788473)

[3.3.3 GET /elder/{eid}/blood\_pressure 获取老人的血压数据 27](#_Toc413788474)

[3.3.4 POST /elder/{eid}/blood\_pressure 输入老人的血压数据 28](#_Toc413788475)

[3.3.5 GET /elder/{eid}/heart\_rate 获取老人的心率数据 28](#_Toc413788476)

[3.3.6 POST /elder/{eid}/heart\_rate 获取老人的心率数据 29](#_Toc413788477)

[3.4老人专属的项目 29](#_Toc413788478)

[3.4.1 GET /elder/{eid}/care\_item 获取养老院老人的项目列表 29](#_Toc413788479)

[3.4.2 POST /elder/{eid}/care\_item 添加养老院某个老人的护理项目 30](#_Toc413788480)

[3.4.3 GET /elder/{eid}/care\_item/{itemid} 获取老人的某个项目 31](#_Toc413788481)

[3.4.4 PUT /elder/{eid}/care\_item/{itemid} 修改老人的某个项目 31](#_Toc413788482)

[3.4.5 DELETE /elder/{eid}/care\_item/{itemid} 删除老人某个项目 32](#_Toc413788483)

[4. 员工相关 33](#_Toc413788484)

[4.1员工基本信息 33](#_Toc413788485)

[4.1.1 GET /gero/{gid}/staff 获取员工列表 33](#_Toc413788486)

[4.1.2 POST /gero/{gid}/staff 新增员工信息 33](#_Toc413788487)

[4.1.3 GET /gero/{gid}/staff/{sid} 查询员工信息 34](#_Toc413788488)

[4.1.4 PUT /gero/{gid}/staff/{sid} 修改员工信息 35](#_Toc413788489)

[4.1.5 DELETE /gero/{gid}/staff/{sid} 员工离职 35](#_Toc413788490)

[4.2老人护工的记录 35](#_Toc413788491)

[4.2.1 GET /carer/{cid}/elder/{eid}/records 获取老人护工负责的老人的项目记录列表 35](#_Toc413788492)

[4.2.2 POST /carer/{cid}/elder/{eid}/records 添加老人护工负责的某个老人的项目记录 36](#_Toc413788493)

[4.2.3 GET /carer/{cid}/elder/records 获取老人护工的项目记录列表（所有老人） 37](#_Toc413788494)

[4.3 房间护工的房间记录 38](#_Toc413788495)

[4.3.1 GET /carer/{cid}/area/{aid}/records 获取房间护工的房间项目记录列表 38](#_Toc413788496)

[4.3.2 POST /carer/{cid}/area/{aid}/records 添加房间护工的房间项目记录 38](#_Toc413788497)

[4.3.3 GET /carer/{cid}/area/records 获取房间护工的项目记录列表（所有房间） 39](#_Toc413788498)

[5. 排班相关 40](#_Toc413788499)

[5.1 职工上班时间安排（单个员工维度） 40](#_Toc413788500)

[5.1.1 GET /staff/{sid}/schedule 获取某员工的上班信息列表（批量） 40](#_Toc413788501)

[5.1.2 POST(PUT) /staff/{sid}/schedule 添加员工的排班信息（批量） 40](#_Toc413788502)

[5.1.3 GET /staff/{sid}/schedule/{date} 获取某员工的某天是否上班 41](#_Toc413788503)

[5.1.4 POST(PUT) /staff/{sid}/schedule/{date} 增加/修改员工的某一天排班信息 41](#_Toc413788504)

[5.1.5 DELETE /staff/{sid}/schedule/{date} 删除员工的某一天排班信息 42](#_Toc413788505)

[5.2 职工上排时间安排（养老院维度） 42](#_Toc413788506)

[5.2.1 GET /gero/{gid}/schedule 获取养老院员工排班信息 42](#_Toc413788507)

[5.2.2 PUT /gero/{gid}/schedule 修改养老院护工排班信息(多员工) 42](#_Toc413788508)

[5.3 老人护工职责 43](#_Toc413788509)

[5.3.1 GET /carer/{cid}/elder 获取老人护工负责的老人列表 43](#_Toc413788510)

[5.3.2 GET /carer/{cid}/elder/{eid}/duty 获取老人护工负责的某个老人的相关信息 （历史信息） 44](#_Toc413788511)

[5.3.3 DELETE /carer/{cid}/elder/{eid}/duty 终止老人护工对某个老人的负责关系 44](#_Toc413788512)

[5.3.4 PUT /carer/{cid}/elder/{eid}/duty 建立老人护工对某个老人的负责关系，同时终止这个老人与其他护工的负责关系 45](#_Toc413788513)

[5.4 房间护工职责 46](#_Toc413788514)

[5.4.1 GET /carer/{sid}/area 获取房间护工负责的房间列表 46](#_Toc413788515)

[5.4.2 GET /carer/{sid}/area/{aid}/duty 获取房间护工负责的某个房间的相关信息 46](#_Toc413788516)

[5.4.3 DELETE /carer/{cid}/area/{aid}/duty 终止房间护工对某个房间的负责关系 47](#_Toc413788517)

[5.4.4 POST /carer/{cid}/area/{aid}/duty 建立房间护工对某个房间的负责关系 47](#_Toc413788518)

[5.4.5 PUT /carer/{cid}/area/{aid}/duty 建立房间护工对某个房间的负责关系，同时终止这个房间与其他护工的负责关系 48](#_Toc413788519)

[5.5 排班结果 48](#_Toc413788520)

[5.5.1 GET /elder/{eid}/duty\_carer 获取老人的负责护工 48](#_Toc413788521)

[5.5.2 GET /area/{aid}/duty\_carer 获取房间的负责护工 48](#_Toc413788522)

版本号： 1.5.2

1. API接口输入输出通用规则（这部分先不考虑，可以后面总结）
2. 所有接口地址均以<https://www.resthousecare.com/> 为前缀，实现时不必考虑，由API前端路由进行地址转换。
3. URL通用参数

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| limit | 否 | 分页大小，仅支持分页的API需要 |
| page | 否 | 页标记，仅支持分页的API需要。不提供时默认为第一页，下一页的offset由返回数据中的nextToken给出 |
| order\_by | 否 | 排序标记，单独约定 |

1. API返回数据

返回的结果数据为JSON格式，会带有以下属性

|  |  |  |
| --- | --- | --- |
| **属性** | **类型** | **说明** |
| status | int | http status代码，200表示调用成功 |
| error | string | 可选，错误信息/返回信息 |
| total | int | 可选，数据总长度，仅支持分页的api提供 |
| next\_page | string | 可选，查询下一页数据时使用的页码，对页码的解释由提供页码的api进行 |
| entities | object[] | 返回对象。即使返回单条数据，也以数组形式返回 |

1. API接口定义

# 1. 系统管理

1.用户登陆

…

## 1.1 用户管理

### 1.1.1 GET /user 获取用户列表

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
|  |  |  |

返回数据:

{

id:{int},

name:{string},

username:{string}

user\_type:{int} \\3种用户：员工、老人、管理员

user\_id:{int} \\ staff\_id, elder\_id, gero\_id

role\_list:[name,…] \\用于显示

photo\_url:{string}

}

### ~~1.1.2 POST /user 新增账户信息~~

~~提交数据：~~

~~{~~

~~name:{string}~~

~~username:{string}~~

~~password:{string}~~

~~user\_type:{int} \\3种用户：员工、老人、管理员~~

~~user\_id:{int} \\ staff\_id, elder\_id, gero\_id~~

~~photo\_url:{strng} \\optional~~

~~}~~

### 1.1.3 GET /user/{uid} 获取账户信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **uid** | **是** | **账户id** |

返回数据：

{

id:{int},

name:{string},

username:{string}

user\_type:{int} \\3种用户：员工、老人、管理员

user\_id:{int} \\ staff\_id, elder\_id, gero\_id

role\_list:[{

id:{int},

name:{string},

},…

],

privilege\_list:[{

id

name

parent\_id

parent\_ids

permission

href

icon

},…

]

register\_date:{int}

cancel\_date:{int}

photo\_url:{string}

}

### 1.1.4 PUT /user/{uid} 修改基本账户信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **uid** | **是** | **账户id** |

提交数据：

{

请参考现在的数据库填入，下面几个信息不能修改

\username:{string}

\password:{string}

\user\_type:{int} \\3种用户：员工、老人、管理员

\user\_id:{int} \\ staff\_id, elder\_id, gero\_id

}

返回新user

### 1.1.5 PUT /user/{uid}/password 修改用户密码

### ~~1.1.6 GET /user/{uid}/role 获取用户角色信息列表~~

~~（比用户信息中更详细）~~

~~返回数据：~~

~~[{~~

~~id:{int},~~

~~name:{string},~~

~~notes:{},~~

~~privilege\_list:[{~~

~~id~~

~~name~~

~~parent\_id~~

~~parent\_ids~~

~~permission~~

~~href~~

~~icon~~

~~},…]~~

~~},…]~~

### 1.1.7 PUT /user/{uid}/role 修改某个用户的角色

[ids:{int},…]

因为仅仅是对user-role的关系进行操作，而且数据量很小，POST和DELETE也可以通过PUT完成。当然了POST和DELETE这两个接口也是可以提供的，使得操作更细化一点。

### ~~1.1.8 DELETE /user/{uid}/role/{rid} 用户删除一个角色~~

### 1.1.9 DELETE /user/{uid} 删除账户信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **uid** | **是** | **删除账户id** |

数据库中填入cancel\_date

## 1.2 养老院的角色管理（管理员权限）

### 1.2.1 GET /gero/{gid}/role 获取养老院角色列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

返回数据：

[{

id:{int},

name:{string},

notes:{string}

}]

### 1.2.2 POST /gero/{gid}/role 新增养老院角色

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

提交数据:

{

name:{string},

notes: {string},

}

### 1.2.3 GET /gero/{gid}/role/{rid} 角色信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **rid** | **是** | **角色id** |

返回数据：

{

name:{string},

type: {int},

note:{string},

privilege\_list:[

{

id:{int},

name:{string},

notes:{string}

}]

}

### 1.2.4 PUT /gero/{gid}/role/{rid} 修改角色的基本信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **rid** | **是** | **角色id** |

提交数据:

{

name:{string},

notes:{string}

}

### 1.2.5 DELETE /gero/{gid}/role/{rid} 删除养老院角色

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **rid** | **是** | **角色id** |

### 1.2.6 POST /gero/{gid}/role/{rid}/privilege 给养老院角色添加权限

提交数据:

{insert\_privilege\_ids[{id}…]}

**1.2.7 DELETE /gero/{gid}/role/{rid}/privilege 给养老院角色删除权限**

提交数据：{delete\_privilege\_ids[{id}…]}

### 1.2.8 PUT /gero/{gid}/role/{rid} 为某个角色关联员工

提交数据：[user\_id:{int},……]

## 1.3 默认角色管理（超级管理员）

与上面很类似

### 1.3.1 GET /role

### 1.3.2 POST /role

说明：新增默认角色后，用函数向所有养老院角色中复制该角色

### 1.3.3 GET /role/{rid}

说明：可以查看privilege信息

### 1.3.4 PUT /role/{rid} 修改默认角色的信息

### 1.3.5 DELETE /role/{rid} 删除默认角色的信息

问题：可以修改养老院的吗，

说明：不影响已存在的养老院角色副本，仅影响新养老院创建时的副本

### 1.3.6 PUT /role/{rid}/privilege 修改默认角色的权限

说明：此URL不要GET方法，因为在/role/{id}时已经get过了

问题：但不要post和delete方法了，行得通吗

## 1.4 权限管理(超级管理员)

### 1.4.1 GET /privilege 权限列表

返回数据：

{

id:{int}

name:{string}

parent\_id:{int}

parent\_ids:{string}

permission:{string}

href:{string}

icon:{string}

notes:{string}

}

### 1.4.2 POST /privilege 新增权限

返回数据：

{

name:{string}

parent\_id:{int}

parent\_ids:{string}

permission:{string}

href:{string}

icon:{string}

notes:{string}

}

### 1.4.3 PUT /privilege/{pid} 修改权限

返回数据：

{

name:{string}

permission:{string}

href:{string}

icon:{string}

notes:{string}

}

### 1.4.4 DELETE /privilege/{pid} 删除权限

应该逻辑删除

### 1.4.5 GET /role/{rid}/privilege 获取角色的权限

# 2. 养老院信息

## 2.1 养老院管理（超管）

### 2.1.1 GET /gero 获取养老院列表

路径参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //养老院id

name:{string}, //养老院全称

city:{string}, //所在城市

district{string}, //所在行政区

level:{int}, //养老院最高护理等级，可取1,2,3,0（专护），默认护理等级为3

address:{string}, //地址

contact:{string}, //联系方式

contact\_id{int} //联系人staff\_id

license :{string}, //许可证

scale:{int} //养老院人数

}

### 2.1.2 POST /gero 新增养老院基本信息

URL参数：无

提交数据：

{

name:{string}, //养老院全称

city:{string}, //所在城市

district{string}, //所在行政区

level:{int}, //养老院最高护理等级，可取1,2,3,0（专护），默认护理等级为3

address:{string}, //地址

contact:{string}, //联系方式

contact\_id{int} //联系人staff\_id

license :{string}, //许可证

scale:{int} //养老院人数

logo：{string}, //养老院logo图片的url地址

image:{string}, //养老院封面图片的url地址

}

返回数据：无

### 2.1.3 GET /gero/{gid} 获取养老院基本信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //养老院id

name:{string}, //养老院全称

city:{string}, //所在城市

district{string}, //所在行政区

level:{int}, //养老院最高护理等级，可取1,2,3,0（专护），默认护理等级为3

address:{string}, //地址

contact:{string}, //联系方式

contact\_id{int} //联系人staff\_id

license :{string}, //许可证

scale:{int} //养老院人数

logo：{string}, //养老院logo图片的url地址

image:{string}, //养老院封面图片的url地址

}

### 2.1.4 PUT /gero/{gid} 修改养老院基本信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：无

提交数据：

{

name:{string}, //养老院全称

city:{string}, //所在城市

district{string}, //所在行政区

level:{int}, //养老院最高护理等级，可取1,2,3,0（专护），默认护理等级为3

address:{string}, //地址

contact:{string}, //联系方式

contact\_id{int} //联系人staff\_id

license :{string}, //许可证

scale:{int} //养老院人数

logo：{string}, //养老院logo图片的url地址

image:{string}, //养老院封面图片的url地址

}

返回数据：无

### 2.1.5 DELETE /gero/{gid} 删除养老院

## 2.2 养老院区域管理

### 2.2.1 GET /gero/{gid}/area 获取养老院所有的区域列表

返回数据：[{

id

parent\_id

parent\_ids

type

level

name

}]

### 2.2.2 POST /gero/{gid}/area 创建一个养老院区域

提交数据：{

parent\_id

//parent\_ids

type

level

name

}

### 2.2.3 GET /gero/{gid}/area/{aid} 获取某区域及其子区域

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sub\_level** | **否** | **表示向下延伸的子区域的层数，默认为1层** |

返回数据：{

id

parent\_id

parent\_ids

type

level

name

area\_list:[{

id

parent\_id

parent\_ids

type

level

name

},…

]

}

### 2.2.4 PUT /gero/{gid}/area/{aid} 修改某区域

提交数据：{

name:

}

### 2.2.5 DELETE /gero/{gid}/area/{aid} 修改某区域

## 2.3 养老院房间和床位

有点问题，先不做

GET /gero/{gid}/building 楼栋列表信息

返回数据：[{

id

name:

type:

level:

},…]

GET /gero/{gid}/building/{aid}/floor 某一栋+楼层信息

返回数据：{

building\_name:{string}

type:

level

floor\_list:[{

id

name:

type:

level:

},…]

}

GET /gero/{gid}/building/{aid}/floor/{aid}/room某一层+房间信息

返回数据：{

building\_name:{string}

floor\_name:{string}

type:

level

room\_list:[{

id

name:

type:

level:

},…]

}

GET /gero/{gid}/building/{aid}/floor/{aid}/room/{aid}/bed 某一间+床位信息

返回数据：{

building\_name:{string}

floor\_name:{string}

room\_name:{string}

type:

level

bed\_list:[{

id

name:

type:

level:

},…]

}

### 3.2.1 GET /room/{area\_id} 房间信息+老人

## 2.4 默认老人项目（超管）

### 2.4.1 GET /care\_item 获取老人护理项目总表

路径参数：无

URL参数：无

返回数据：json, entities中包含的对象如下

[{

id: {int}, //id

name:{string}, //名称

level:{int}, //基于护理标准值为1,2,3,0（专护），默认护理等级为3

notes:{string}, //解释

},…]

### 2.4.2 POST /care\_item 添加老人护理项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

name:{string}, //名称

level:{int}, //基于护理标准值为1,2,3,0（专护），默认护理等级为3

notes:{string}, //解释

}

返回数据： entities中不包含对象

说明：

### 2.4.3 GET /care\_item/{cid} 获取总表中某个老人项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **护理项目id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //单个老人项目id

name:{string}, //名称

level:{int}, //基于护理标准值为1,2,3,0（专护），默认护理等级为3

notes:{string}, //解释

}

### 2.4.4 PUT /care\_item/{cid} 修改总表中某个老人项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **护理项目id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

id:{int}, //老人项目id

name:{string}, //名称

level:{int}, //基于护理标准值为1,2,3,0（专护），默认护理等级为3

notes:{string}, //解释

}

返回数据： entities中不包含对象

说明：

### 2.4.5 DELETE /care\_item/{cid} 删除总表中某个老人项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **护理项目cid** |
|  |  |  |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

返回数据： entities中不包含对象

说明：删除的项目在养老院的老人项目列表中不能保留，否则不能删

## 2.5 默认房间项目（超管）

### 2.5.1 GET /area\_item 获取房间项目总表

路径参数：无

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, // 房间项目id

name:{string}, //名称

notes:{string}, //解释

}

### 2.5.2 POST /area\_item 添加房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

name:{string}, //名称

notes:{string}, //解释

}

返回数据： entities中不包含对象

说明：

### 2.5.3 GET /area\_item/{aid} 获取总表中某个房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **rid** | **是** | **房间项目id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //房间项目id

name:{string}, //名称

notes:{string}, //解释

}

### 2.5.4 PUT /area\_item/{aid} 修改总表中房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **aid** | **是** | **房间项目id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

name:{string}, //名称

notes:{string}, //解释

}

返回数据： entities中不包含对象

说明：

### 2.5.5 DELETE /area\_item/{aid} 删除总表中某个房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **aid** | **是** | **房间项目id** |
|  |  |  |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

返回数据： entities中不包含对象

说明：删除的项目在养老院的房间项目列表中不能保留，否则不能删

逻辑删除，设置del\_flag=1

## 2.6 养老院包含的老人项目

### 2.6.1 GET /gero/{gid}/care\_item 获取养老院护理项目列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //养老院护理项目id

care\_item:{ //总表中护理项目

id:{int},

name:{string}

level:{int}

}

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

### 2.6.2 POST /gero/{gid}/care\_item 添加养老院护理项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

care\_item\_id:{int} //护理项目id (总表)

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

返回数据： entities中不包含对象

说明：

### 2.6.3 GET /gero/{gid}/care\_item/{itemid} 获取某个养老院的老人项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **itemid** | **是** | **护理项目id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //养老院老人护理项目id

care\_item:{ //总表中护理项目

id:{int},

name:{string}

level:{int}

}

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

### 2.6.4 PUT /gero/{gid}/care\_item/{itemid} 修改某个养老院的老人项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **itemid** | **是** | **护理项目id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

care\_item\_id: {int}, //老人护理项目id（总表）

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

返回数据： entities中不包含对象

说明：

### 2.6.5 DELETE /gero/{gid}/care\_item/{itemid} 删除养老院某个老人项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **itemid** | **是** | **护理项目id** |

返回数据： entities中不包含对象

说明：删除的项目在老人项目列表中不能保留，否则不能删。

删除的话，是逻辑删除

## 2.7 养老院包含的房间项目

### 2.7.1 GET /gero/{gid}/area\_item 获取养老院房间项目列表

路径参数：无

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //养老院房间项目id

area\_item:{ //总表中房间项目

id:{int},

name:{string}

}

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

### 2.7.2 POST /gero/{id}/area\_item 添加养老院房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

area\_item\_id:{int} //房间项目id（总表）

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

返回数据： entities中不包含对象

说明：

### 2.7.3 GET /gero/{gid}/area\_item/{aid} 获取养老院房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **aid** | **是** | **房间项目id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //养老院房间项目id

area\_item:{ //总表中房间项目

id:{int},

name:{string}

}

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

### PUT /gero/{gid}/area\_item/{aid} 更新养老院房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **aid** | **是** | **房间项目id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

area\_item\_id: {int}, //房间项目id(总表)

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

返回数据： entities中不包含对象

说明：

### DELETE /gero/{gid}/area\_item/{aid} 删除养老院房间项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **aid** | **是** | **房间项目id** |
|  |  |  |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

返回数据： entities中不包含对象

说明：逻辑删除

# 3. 老人相关

## 3.1 老人基本信息

### 3.1.1 GET /gero/{gid}/elder 获取老人列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **name** | **no** | **姓名的部分或全部** |
| **gender** | **no** | **0:male 1:female** |
| **age** | **no** | **?age=min,max** |
| **care\_level** | **no** | **护理等级，0专护,1一级,2同理,3.同理** |

返回数据：json, entities中包含的对象如下

[{

id:{int}, \\老人ID

name:{string}, \\老人姓名

gender:{bit}, \\0代表男性，1代表女性

address:{string}, \\现居住地

age:{int}, \\老人年龄

identity\_no:{string}, \\老人身份证号码

nssf\_id:{string}, \\老人社保卡号

archive\_id:{string}, \\档案编号

area\_id:{int}, \\老人入住床号，关联T\_AREA表

nationality:{string}, \\民族

marriage:{int}, \\0:未婚 1：已婚 2：离异 3:丧偶

native\_place:{string}, \\老人籍贯

birthday:{string}, \\老人出身年月日

political\_status:{string}, \\老人政治面貌

education:{string}, \\老人受教育水平

residence:{string}, \\户口所在地

care\_level:{int}, \\老人护理等级

checkin\_date:{string}, \\入院日期

checkout\_date:{string}, \\离院日期

pad\_mac:{string}, 老人房间pad的mac

}]

}

### 3.1.2 POST /gero/{gid}/elder/ 增加老人详细信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

提交数据：

{

name:{string}, \\老人姓名

gender:{int}, \\0代表男性，1代表女性

address:{string}, \\现居住地

age:{int}, \\老人年龄

identity\_no:{string}, \\老人身份证号码

nssf\_id:{string}, \\老人社保卡号

archive\_id:{string}, \\档案编号

photo\_url:{string}, \\老人照片地址

area\_id:{int}, \\老人入住床号，关联T\_AREA表

nationality:{string}, \\民族

marriage:{int}, \\0:未婚 1：已婚 2：离异 3:丧偶

native\_place:{string}, \\老人籍贯

birthday:{string}, \\老人出身年月日

political\_status:{string}, \\老人政治面貌

education:{string}, \\老人受教育水平

residence:{string}, \\户口所在地

care\_level:{int}, \\老人护理等级

checkin\_date:{string}, \\入院日期

checkout\_date:{string}, \\离院日期

apply\_url:{string}, \\申请表url（图片）

survey\_url:{string}, \\调防表url（图片）

assess\_url:{string}, \\审批表url（图片）

track\_url:{string}, \\7天跟踪记录表url（图片）

pad\_mac:{string}, \\老人房间pad的mac

}

### 3.1.3 GET /gero/{gid}/elder/{eid} 查单个老人详细信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |
| **gid** | **是** | **养老院id** |

返回数据：json, entities中包含的对象如下

{

id:{int}, \\老人ID

name:{string}, \\老人姓名

gender:{bit}, \\0代表男性，1代表女性

address:{string}, \\现居住地

age:{int}, \\老人年龄

identity\_no:{string}, \\老人身份证号码

nssf\_id:{string}, \\老人社保卡号

archive\_id:{string}, \\档案编号

photo\_url:{string}, \\老人照片地址

area\_id:{int}, \\老人入住床号，关联T\_AREA表

nationality:{string}, \\民族

marriage:{int}, \\0:未婚 1：已婚 2：离异 3:丧偶

native\_place:{string}, \\老人籍贯

birthday:{string}, \\老人出身年月日

political\_status:{string}, \\老人政治面貌

education:{string}, \\老人受教育水平

residence:{string}, \\户口所在地

care\_level:{int}, \\老人护理等级

checkin\_date:{string}, \\入院日期

checkout\_date:{string}, \\离院日期

apply\_url:{string}, \\申请表url（图片）

survey\_url:{string}, \\调防表url（图片）

assess\_url:{string}, \\审批表url（图片）

track\_url:{string}, \\7天跟踪记录表url（图片）

pad\_mac:{string}, \\老人房间pad的mac

}

### 3.1.4 PUT /gero/{gid}/elder/{eid} 修改单个老人详细信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |
| **gid** | **是** | **养老院id** |

提交数据：

{

name:{string}, \\老人姓名

gender:{bit}, \\0代表男性，1代表女性

address:{string}, \\现居住地

age:{int}, \\老人年龄

identity\_no:{string}, \\老人身份证号码

nssf\_id:{string}, \\老人社保卡号

archive\_id:{string}, \\档案编号

photo\_url:{string}, \\老人照片地址

area\_id:{int}, \\老人入住床号，关联T\_AREA表

nationality:{string}, \\民族

marriage:{int}, \\0:未婚 1：已婚 2：离异 3:丧偶

native\_place:{string}, \\老人籍贯

birthday:{string}, \\老人出身年月日

political\_status:{string}, \\老人政治面貌

education:{string}, \\老人受教育水平

residence:{string}, \\户口所在地

care\_level:{int}, \\老人护理等级

checkin\_date:{string}, \\入院日期

checkout\_date:{string}, \\离院日期

apply\_url:{string}, \\申请表url（图片）

survey\_url:{string}, \\调防表url（图片）

assess\_url:{string}, \\审批表url（图片）

track\_url:{string}, \\7天跟踪记录表url（图片）

pad\_mac:{string}, \\老人房间pad的mac

}

### 3.1.5 DELETE /gero/{gid}/elder/{eid} 老人离院登记

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |
| **eid** | **是** | **老人id** |

## 3.2老人表单信息

/elder/{eid}/sheet/xx

/elder/{eid}/sheet/yy

## 3.3 老人健康数据

### 3.3.1 GET /elder/{eid}/temperature 获取老人的体温数据

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //老人id

name:{string}, //老人姓名

photo:{string}, //老人头像

tp\_list：[{

id:{int}, //测量记录id

temperature:{string},//体温值

time:{string} //测量时间，注意格式

doctor:{

id:{int}, //测量医生id

name:{string} //测量医生姓名

photo:{string} //测量医生头像

}

}…]

}

### 3.3.2 POST /elder/{eid}/temperature 输入老人的体温数据

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

传输数据：json中包含的对象如下

{

doctor\_id:{int} //医生id

temperature:{string}, //测得体温

time:{string} //请注意时间格式

}

### 3.3.3 GET /elder/{eid}/blood\_pressure 获取老人的血压数据

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //老人id

name:{string}, //老人姓名

photo:{string}, //老人头像

bp\_list：[{

id:{int}, //测量记录id

diastolic\_pressure:{float},//舒张压

systolic\_pressure:{float},//收缩压

time:{string} //测量时间，注意格式

doctor:{

id:{int}, //测量医生id

name:{string} //测量医生姓名

photo:{string} //测量医生头像

}

}…]

}

### 3.3.4 POST /elder/{eid}/blood\_pressure 输入老人的血压数据

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

传输数据：json中包含的对象如下

{

doctor\_id:{int} //医生id

diastolic\_pressure:{float}, //舒张压

systolic\_pressure:{float},//收缩压

time:{string} //测量时间

}

### 3.3.5 GET /elder/{eid}/heart\_rate 获取老人的心率数据

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //老人id

name:{string}, //老人姓名

photo:{string}, //老人头像

hr\_list：[{

id:{int}, //测量记录id

heart\_rate:{int}, //心率值

time:{string} //测量时间，注意格式

doctor:{

id:{int}, //测量医生id

name:{string} //测量医生姓名

photo:{string} //测量医生头像

}

}…]

}

### 3.3.6 POST /elder/{eid}/heart\_rate 获取老人的心率数据

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

传输数据：json中包含的对象如下

{

doctor\_id:{int} //医生id

heart\_rate:{string}, //测得心率值

time:{string} //测量时间

}

## 3.4老人专属的项目

### 3.4.1 GET /elder/{eid}/care\_item 获取养老院老人的项目列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id:{int} //老人专属项目id

gero\_care\_item: { //养老院护理项目

id: {int}, //养老院老人护理项目id

care\_item:{ //总表中护理项目

id:{int},

name:{string}

level:{int}

}

level:{int}, //基于养老院实际情况，所提供的最高服务等级

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

level:{int}, //基于老人实际情况，所提供的服务等级

period:{int}, //周期，间隔多少天做一轮，optional，默认取1

start\_time:{datetime} //项目开始时间，optional

end\_time:{datetime} //项目结束时间，optional

}

### 3.4.2 POST /elder/{eid}/care\_item 添加养老院某个老人的护理项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

gero\_care\_item\_id:{int} //养老院护理项目id

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮，optional，默认取1

start\_time:{datetime} //项目开始时间，optional

end\_time:{datetime} //项目结束时间，optional

}

返回数据： entities中不包含对象

说明：

### 3.4.3 GET /elder/{eid}/care\_item/{itemid} 获取老人的某个项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |
| **itemid** | **是** | **护理项目id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id:{int} //老人专属项目id

gero\_care\_item: { //养老院护理项目

id: {int}, //养老院老人护理项目id

care\_item:{ //总表中护理项目

id:{int},

name:{string}

level:{int}

}

level:{int}, //基于养老院实际情况，所提供的最高服务等级

period:{int}, //周期，间隔多少天做一轮

frequency:{int} //频率，一天做几次

}

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮，optional，默认取1

start\_time:{datetime}, //项目开始时间，optional

end\_time:{datetime} //项目结束时间，optional

}

### 3.4.4 PUT /elder/{eid}/care\_item/{itemid} 修改老人的某个项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **eid** | **是** | **老人id** |
| **itemid** | **是** | **护理项目id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

提交数据：json

{

id:{int} //老人专属项目id

level:{int}, //基于养老院实际情况，所提供的最高等级服务

period:{int}, //周期，间隔多少天做一轮，optional，默认取1

start\_time:{datetime}, //项目开始时间，optional

end\_time:{datetime} //项目结束时间，optional

}

返回数据： entities中不包含对象

说明：

### 3.4.5 DELETE /elder/{eid}/care\_item/{itemid} 删除老人某个项目

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **id** | **是** | **老人id** |
| **itemid** | **是** | **护理项目id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |

返回数据： entities中不包含对象

说明：逻辑删除

# 4. 员工相关

## 4.1员工基本信息

### 4.1.1 GET /gero/{gid}/staff 获取员工列表

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **name** | **no** | **姓名的部分或全部** |
| **gender** | **no** | **0:male 1:female** |
| **identity\_no** | **no** | **身份证号的部分或全部** |
| **role** | **no** | **以权限查询 role=doctor/area\_carer/…,联合查询用，分割** |

返回数据：

[{

id:{int},

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

identity\_no:{string}, \\身份证号码

nssf:{string}, \\社保卡账号

birthday:{string}, \\员工生日

gender:{string}, \\员工性别

residence\_address:{string}, \\员工户籍地址

household\_address:{string}, \\员工居住地址

leave\_date:{string}, \\离职时间

archive\_id:{string}, \\纸质档案编号

role\_list:[{

user\_id:{int},

username:{string},

},…],

},…]

### 4.1.2 POST /gero/{gid}/staff 新增员工信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **gero\_id** |

提交数据：

{

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

identity\_no:{string}, \\身份证号码

nssf:{string}, \\社保卡账号

birthday:{string}, \\员工生日

gender:{string}, \\员工性别

basic\_url:{string}, \\员工基本信息表扫描件的地址

residence\_address:{string}, \\员工户籍地址

household\_address:{string}, \\员工居住地址

leave\_date:{string}, \\离职时间

archive\_id:{string}, \\纸质档案编号

photo\_url:{string}, \\照片url

}

### 4.1.3 GET /gero/{gid}/staff/{sid} 查询员工信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **gero\_id** |
| **sid** | **是** | **staff\_id** |

返回数据：

{

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

identity\_no:{string}, \\身份证号码

nssf:{string}, \\社保卡账号

gero\_id:{int}, \\养老院ID

birthday:{string}, \\员工生日

gender:{string}, \\员工性别

basic\_url:{string}, \\员工基本信息表扫描件的地址

residence\_address:{string}, \\员工户籍地址

household\_address:{string}, \\员工居住地址

leave\_date:{string}, \\离职时间

archive\_id:{string}, \\纸质档案编号

user\_id:{int},

username:{string},

role\_list:[{

user\_id:{int},

username:{string},

},…],

privilege\_list:[{

id,

name

parent\_id

parent\_ids

permission

href

icon

},…

]

register\_date:{int}

cancel\_date:{int}

photo\_url:{string}

}

### 4.1.4 PUT /gero/{gid}/staff/{sid} 修改员工信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **gero\_id** |
| **sid** | **是** | **staff\_id** |

提交数据：

{

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

identity\_no:{string}, \\身份证号码

nssf\_id:{string}, \\社保卡账号

birthday:{string}, \\员工生日

gender:{string}, \\员工性别

basic\_url:{string}, \\员工基本信息表扫描件的地址

residence\_address:{string}, \\员工户籍地址

household\_address:{string}, \\员工居住地址

leave\_date:{string}, \\离职时间

archive\_id:{string}, \\纸质档案编号

photo\_url:{string}, \\照片url

}

### 4.1.5 DELETE /gero/{gid}/staff/{sid} 员工离职

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **gero\_id** |
| **sid** | **是** | **staff\_id** |

## 4.2老人护工的记录

### 4.2.1 GET /carer/{cid}/elder/{eid}/records 获取老人护工负责的老人的项目记录列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //记录id

elder\_item\_id:{int}

item\_name:{int}

finish\_time:{datetime}, //记录时间

}

说明：若护工与老人无对应关系，需要有错误提示

### 4.2.2 POST /carer/{cid}/elder/{eid}/records 添加老人护工负责的某个老人的项目记录

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **eid** | **是** | **老人id** |

URL参数：无

提交数据：json

{

elder\_item:[{ //老人专属项目id

id:{int};

name:{int}

}, ……],

finish\_time:{datetime} //optional，完成时间，默认为提交时间

}

### 4.2.3 GET /carer/{cid}/elder/records 获取老人护工的项目记录列表（所有老人）

暂不实现

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **date** | **否** | **date,查询某天的数据** |
| **……** |  |  |

返回数据：json, entities中包含的对象如下

{

id: {int}, //老人记录 id

elder:{ //老人信息

id: {int}, //老人id

name:{string}, //姓名

age:{int} //年龄

gender:{string}, //性别

photo\_url:{url} //照片url

care\_level:{string }, //护理等级

}

elder\_items:[ //老人专属项目列表

{ id:{int}, //专属项目id

name:{string},

level:{int}

start\_time:{time} //规定开始时间

end\_time:{time} //规定结束时间

finish\_time:{datetime}, //实际记录时间

}，

……..

]

}

## 4.3 房间护工的房间记录

### 4.3.1 GET /carer/{cid}/area/{aid}/records 获取房间护工的房间项目记录列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **aid** | **是** | **区域id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //房间项目记录id

area\_item\_id:{int} //养老院房间项目id

area\_name:{string} //项目名称

finish\_time:{datetime} //记录时间

}

### 4.3.2 POST /carer/{cid}/area/{aid}/records 添加房间护工的房间项目记录

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **aid** | **是** | **区域id** |

URL参数：无

提交数据：json

{

area\_item:[{

id:{int}, //房间项目id

name:{int} //项目名称

}

…

]

finish\_time:{datetime} //optional，完成时间，默认为提交时间

}

返回数据： entities中不包含对象

说明：

### 4.3.3 GET /carer/{cid}/area/records 获取房间护工的项目记录列表（所有房间）

暂不实现

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **date** | **否** | **date,查询某天的数据** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //房间项目记录id

room:{string}, //房间名称

floor:{string} //楼层

building:{string} //楼号

area\_items:[ //养老院房间项目信息列表

{

id:{int} //养老院房间项目id

name:{string}

finish\_time:{datetime}, //记录时间

},

….

]

}

# 5. 排班相关

## 5.1 职工上班时间安排（单个员工维度）

包括医生和护工

### 5.1.1 GET /staff/{sid}/schedule 获取某员工的上班信息列表（批量）

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //排班 id

work\_date:[{date}, …] //工作日期

}

### 5.1.2 POST(PUT) /staff/{sid}/schedule 添加员工的排班信息（批量）

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |

URL参数：无

提交数据：json

{

work\_date: [

{ date}, //排班日期

…

],

nowork\_date: [

{ date}, //取消排班的日期

…

]

}

说明：一天不能同时出现在work\_date和nowork\_date中，

### 5.1.3 GET /staff/{sid}/schedule/{date} 获取某员工的某天是否上班

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |
| **date** | **是** | **排班日期** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //排班 id

}

说明：返回id表示存在，不存在返回空

### 5.1.4 POST(PUT) /staff/{sid}/schedule/{date} 增加/修改员工的某一天排班信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |

URL参数：无

说明：若原来不上班，状态变为上班；否则不变

### 5.1.5 DELETE /staff/{sid}/schedule/{date} 删除员工的某一天排班信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |

URL参数：无

说明：1.必须存在 ; 2.当前日期必须小于排班日期才能删除

## 5.2 职工上排时间安排（养老院维度）

### 5.2.1 GET /gero/{gid}/schedule 获取养老院员工排班信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **start\_date非空时，执行start\_date开始的范围查询；为空时，执行end\_date那天的点查询。若end\_date也为空，默认为当天** |
| **end\_date** | **否** |
| **role** | **否** | **0:elder\_carer; 1:area\_carer；2：doctor；默认返回所有** |

返回数据：json, entities中包含的对象如下

{

id: {int}, // id

staff\_id:{int} //员工id

work\_date: { object[date]}, //记录时间

}

### 5.2.2 PUT /gero/{gid}/schedule 修改养老院护工排班信息(多员工)

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **gid** | **是** | **养老院id** |

提交数据：json

[{

staff\_id:{int},

work\_date: [{ date}, … ], //排班日期

nowork\_date: [{ date}, ....] //取消排班的日期

},…]

## 5.3 老人护工职责

（其实这部分api存储的是一种关系）

### 5.3.1 GET /carer/{cid}/elder 获取老人护工负责的老人列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **date** | **否** | **查询日期，默认为当前日期。** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //老人id

name:{string}, //姓名

age:{int} //年龄

gender:{string}, //性别

photo\_url:{url} //照片url

care\_level:{string }, //护理等级

}

### 5.3.2 GET /carer/{cid}/elder/{eid}/duty 获取老人护工负责的某个老人的相关信息 （历史信息）

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **history** | **否** | **默认history=false，history=true时返回负责的历史时间段** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //老人id

name:{string}, //姓名

age:{int} //年龄

gender:{string}, //性别

photo\_url:{url} //照片url

care\_level:{string }, //护理等级

work\_date:[ //负责时间段，列表，以start\_date降序排列

{

start\_date:{date}

end\_date:{date}

},

……

]

}

说明：目前对work\_date区间的个数暂无限制

### 5.3.3 DELETE /carer/{cid}/elder/{eid}/duty 终止老人护工对某个老人的负责关系

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **end\_date** | **否** | **护工对老人负责关系终止的日期，默认为当天** |

返回数据： entities中不包含对象

说明：并不直接在数据库中删除记录，只是在end\_date填入日期，表明关系的终止。此时没有老人对其负责了，需要post一个新关系。建议直接用PUT方法维护这个一致性

* + 1. POST /carer/{cid}/elder/{eid}/duty 建立老人护工对某个老人的负责关系

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **护工对老人负责关系终止的日期，默认为明天** |
| **end\_date** | **否** | **护工对老人负责关系终止的日期，默认为空，表示到永远** |

返回数据： entities中不包含对象

说明：注意与put方法的区别，post允许一个老人对应多个护工

### 5.3.4 PUT /carer/{cid}/elder/{eid}/duty 建立老人护工对某个老人的负责关系，同时终止这个老人与其他护工的负责关系

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |
| **eid** | **是** | **老人id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **护工对老人负责关系终止的日期，默认为明天** |
| **end\_date** | **否** | **护工对老人负责关系终止的日期，默认为空，表示到永远** |

返回数据： entities中不包含对象

说明：终止这个老人与其他护工的负责关系时，旧记录中的end\_date改为url参数中start\_date的值减一天。

考虑不存在旧关系的情况，则与post相同

## 5.4 房间护工职责

### 5.4.1 GET /carer/{sid}/area 获取房间护工负责的房间列表

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **date** | **否** | **查询日期，默认为当前日期。** |

返回数据：json, entities中包含的对象如下

{

id: {int}, //房间id

room:{sting}, //房间名称

floor:{string}, //层名称,optional

building:{string}, //楼名称,optional

}

### 5.4.2 GET /carer/{sid}/area/{aid}/duty 获取房间护工负责的某个房间的相关信息

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **员工id** |
| **aid** | **是** | **房间id** |

URL参数：无

返回数据：json, entities中包含的对象如下

{

id: {int}, //房间id

room:{string}, //房间名称

floor:{string} //楼层

building:{string} //楼号

work\_date:[ //负责时间段，列表，以start\_date降序排列

{

start\_date:{date}

end\_date:{date}

},

……

]

}

### 5.4.3 DELETE /carer/{cid}/area/{aid}/duty 终止房间护工对某个房间的负责关系

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **aid** | **是** | **房间id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **end\_date** | **否** | **护工对房间负责关系终止的日期，默认为当天** |

返回数据： entities中不包含对象

说明：并不直接在数据库中删除记录，只是更改end\_date的日期，表明关系的终止。但是并没有新建end\_date时间后的关系，要用POST新建。建议慎用次api

### 5.4.4 POST /carer/{cid}/area/{aid}/duty 建立房间护工对某个房间的负责关系

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **rid** | **是** | **房间id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **护工对房间负责关系终止的日期，默认为明天** |
| **end\_date** | **否** | **护工对房间负责关系终止的日期，默认为空，表示到永远** |

返回数据： entities中不包含对象

说明：注意与put方法的区别，这里允许一个房间对应多个护工

### 5.4.5 PUT /carer/{cid}/area/{aid}/duty 建立房间护工对某个房间的负责关系，同时终止这个房间与其他护工的负责关系

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **cid** | **是** | **员工id** |
| **aid** | **是** | **房间id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **start\_date** | **否** | **护工对房间负责关系终止的日期，默认为明天** |
| **end\_date** | **否** | **护工对房间负责关系终止的日期，默认为空，表示到永远** |

返回数据： entities中不包含对象

说明：终止这个房间与其他护工的负责关系时，旧记录中的end\_date改为url参数中start\_date的值减一天

考虑不存在旧关系的情况，则与post相同

## 5.5 排班结果

### 5.5.1 GET /elder/{eid}/duty\_carer 获取老人的负责护工

### 5.5.2 GET /area/{aid}/duty\_carer 获取房间的负责护工

路径参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **sid** | **是** | **老人id或房间id** |

URL参数：

|  |  |  |
| --- | --- | --- |
| **参数** | **必须** | **说明** |
| **date** | **否** | **查询日期，默认为当前日期。** |

返回数据：json, entities中包含的对象如下

[{

id: {int}, //员工（护工）id

gero\_id:{int}, //养老院ID

name:{string}, //姓名

phone:{string}, //员工电话 optional

gender:{string}, //员工性别 optional

photo\_url:{url}, //头像图片 optional

work\_date:{date}, //工作日期

}]

说明：护工app只需要用点查询即可，暂时还是先不支持范围查询了