版本号： 1.4

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 |
| pageToken | 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pageToken由返回数据中的nextToken给出 |

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** |
| errno | int | 错误代码，0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 |
| nextToken | 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
2. 管理相关

1.1养老院信息

1.1.1 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

}

1.1.2 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

1.1.3 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

}

1.1.4 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

1.1.5 DELETE /gero/{gid} 删除养老院

1.2 养老院区域

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

}]

POST /gero/{gid}/area 创建一个养老院区域

提交数据：{

parent\_id

//parent\_ids

type

level

name

}

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

},…

]

}

PUT /gero/{gid}/area/{aid} 修改某区域

返回数据：{

name:

}

DELETE /gero/{gid}/area/{aid} 修改某区域

1.3 养老院房间和床位

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:

},…]

}

1. Item项目相关
   1. 老人项目总表
      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

}

* + 1. 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：

* 1. 房间项目总表
     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

}

* + 1. 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：

* 1. 总表中的单个老人项目
     1. 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

}

* + 1. 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：

* + 1. 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

* 1. 总表中单个房间项目
     1. 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

}

* + 1. 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：

* + 1. 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：逻辑删除，设置del\_flag=1

* 1. 养老院的老人项目总表
     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

}

* + 1. 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：

* 1. 房间项目总表
     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

}

* + 1. 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

{

room\_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：

* 1. 养老院的单个老人项目
     1. 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

}

* + 1. POST /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：

* + 1. 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：逻辑删除

* 1. 养老院的单个房间项目
     1. 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

}

* + 1. POST /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：

* + 1. 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

* 1. 老人专属的项目列表
     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

}

* + 1. 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：

* 1. 老人专属的单个项目
     1. 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

}

* + 1. 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：

* + 1. 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

1. 老人相关
   1. 老人列表

/elders

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** |
| **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

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

}]

}

* 1. 单个老人信息

3.2.1 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

}

3.2.2 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

}

3.2.3 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:{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

}

3.2.4 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** |

* 1. 老人表单信息

/elder/{eid}/sheet/xx

/elder/{eid}/sheet/yy

* 1. 老人项目记录
     1. GET /elder/{eid}/records 给家属用

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

carer: { //护工信息

id: {int}, //员工（护工）id

name:{string}, //姓名

photo\_url:{url} //照片url

phone:{string}, //手机号

leave\_date:{date} //离职日期

}

elder\_item\_list:{string}, //项目名称列表

finish\_time:{datetime } //记录日期

}

说明：不需要提供修改和删除的功能

1. 员工相关
   1. 员工列表

GET

POST /gero/{gid}/staff

考虑如何区分已经离职的和仍然在职的区别

* 1. 单个员工

GET PUT DELETE gero/{gid}/staff/{sid}

* 1. 护工列表
     1. GET gero/{gid}/elder\_carer 获取老人护工列表

GET gero/{gid}/area\_carer 获取房间护工列表

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** |
| **date** | **否** | **查询日期，在checkin\_date和checkout\_date之间则返回；**  **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

name:{string}, //姓名

gender:{string}, //员工性别 optional

photo\_url:{url} // 照片url optional

phone:{string}, //手机号

}

说明：增加护工的操作应在staff相关api中实现，此处仅供查询

说明2：gero的信息出现在路径参数还是url参数中更好？

* + 1. GET /elder/{eid}/duty\_carer 获取老人的负责护工

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

* 1. 医生列表

4.4.1 GET /gero/{gid}/doctors 获取指定养老院医生列表

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}, //医生id

name:{string}, //医生姓名

photo:{string}, //医生照片

contact:{string}, //联系方式

}

* 1. 老人护工负责的老人

（其实这部分api存储的是一种关系）

* + 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

leave\_date:{date} //离院日期

}

* + 1. GET /eldercarer/{sid}/elder/{eid}/duty 获取老人护工负责的某个老人的相关信息

路径参数：

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

leave\_date:{date} //离院日期

area:{

building:{string}

floor:{string}

room:{string}

bed:{string}

}

area\_list:[{

id

parent\_id

parent\_ids,

type

level

name

},…

]

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

{

start\_date:{date}

end\_date:{date}

},

……

]

}

* + 1. 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

* + 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post方法的区别，其实可以统一用post方法…另外写后端时post方法要兼容put方法（不存在旧关系的情况也是有可能的！）

* + 1. 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，旧记录中填入的end\_date为url参数中start\_date的值减一天

* 1. 房间护工的房间列表
     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

}

* + 1. 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增序排列

{

start\_date:{date}

end\_date:{date}

},

……

]

}

* + 1. 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，只是在end\_date填入日期，表明关系的终止

* + 1. 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post方法的区别，其实可以统一用post方法…另外写后端时post方法要兼容put方法（不存在旧关系的情况也是有可能的！）

* + 1. 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，旧记录中填入的end\_date为url参数中start\_date的值减一天

* 1. 护工的老人记录
     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:{ //老人专属项目,并未返回全部嵌套对象，因为层次太深

id:{int}

name:{int}

level:{int}

start\_time:{time}

end\_time:{time}

}

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

* + 1. 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\_id:[{int}, //老人专属项目id

……],

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

}

* + 1. GET /carer/{cid}/elder/records 获取老人护工的项目记录列表（所有老人）

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

}，

……..

]

}

* 1. 房间护工的房间记录
     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:{ //养老院房间项目信息,并未返回全部嵌套对象，因为层次太深

id:{int} //养老院房间项目id

name:{string}

}

finish\_time:{datetime} //记录时间

}

* + 1. 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

{

id: {int}, //房间项目记录id

area\_item\_id:[{int}, //房间项目id列表

…

]

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：

* + 1. GET /carer/{cid}/area/records 获取房间护工的项目记录列表（所有房间）

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:{ //房间信息

id:{int} //房间id

name:{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

{ //并未返回全部嵌套对象,仅返回需要的

id:{int} //养老院房间项目id

name:{string}

finish\_time:{datetime}, //记录时间

},

….

]

}

1. 排班相关
   1. 从养老院维度出发的排班（所有员工）
      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** |
| **type** | **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

carer\_id:{int} //员工id

work\_date: { object[date]}, //记录时间

}

* + 1. PUT /gero/{gid}/schedule 修改养老院老人护工排班信息

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** |
| **date** | **是** | **工作日期** |

提交数据：json

{

staffs: [{

id: int //员工id

},

…

]

}

说明： 一次请求排一天的班,7天请求7次？？

* 1. 从单个员工维度出发的排班（包括医生和护工）
     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

}

* + 1. POST /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

* + 1. 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

work\_date:{date}, //工作日期

}

说明：若不上班，在错误信息里进行标注

* + 1. PUT /staff/{sid}/schedule/{date} 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

* + 1. 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

1. 医疗相关
   1. 老人体温数据
      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

tpList：[{

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

}

}…]

}

* + 1. 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

{

id: {int}, //老人id

doctor\_id:{int} //医生id

temperature:{string}, //测得体温

time:{string} //请注意时间格式

}

* 1. 老人血压数据
     1. 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

}

}…]

}

* + 1. 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

{

id: {int}, //老人id

doctor\_id:{int} //医生id

diastolic\_pressure:{float}, //舒张压

systolic\_pressure:{float},//收缩压

time:{string} //测量时间

}

* 1. 老人心率数据
     1. 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

}

}…]

}

* + 1. 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

{

id: {int}, //老人id

doctor\_id:{int} //医生id

heart\_rate:{string}, //测得心率值

time:{string} //测量时间

}

七、账户管理

1.用户登陆

1.1 登陆

~~1.1.1 GET /user/login 用户登录~~

~~提交数据：~~

~~{~~

~~Username:{string}~~

~~}~~

~~返回数据：~~

~~{~~

~~Password:{string} \\加密过的~~

~~}~~

2.用户管理

2.1 用户列表

2.1.1 GET /users/ 获取用户列表

返回数据:

{

Id:{int},

Username:{string}

Password:{string}

userType:{int} \\3种用户：员工、老人、管理员

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

}

2.1.2 PUT /users 新增账户信息

提交数据：

{

Username:{string}

Password:{string}

userType:{int} \\3种用户：员工、老人、管理员

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

}

2.2 用户信息

2.2.1 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：

{

username:{string}

password:{string}

userType:{int} \\3种用户：员工、老人、管理员

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

}

2.2.2 POST /user/{uid} 修改账户信息

路径参数：

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

返回数据：

{

Username:{string}

Password:{string}

userType:{int} \\3种用户：员工、老人、管理员

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

}

2.2.4 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** |

提交数据：

{

}

八、职工管理

1.员工列表

1.1 GET /gero/{gid}/staff 获取员工列表

URL参数：

{

id:{int},

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

idno:{string}, \\身份证号码

nssf:{string}, \\社保卡账号

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

sortBy:{string} \\排序

}

返回数据：

{

Staff\_list:[{

id:{int},

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

idno:{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

}]

}

1.2 员工信息

1.2.1 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

idno:{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

photo\_url:{string}, \\照片url

}

1.2.2 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

idno:{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

}

1.2.3 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

idno:{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

}

1.2.4 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** |

九、院属互动信息

9.1 GET \gero\{gid}\exchanges 获取院属互动列表

路径参数：

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

返回数据：

{

id:{int},

participants:{string}, \\事件相关人员，用json字符串

mediators:{string}, \\调解人，用json字符串

result:{string}, \\调解结果

recorder:{int}, \\记录人staff\_id

time:{string},

}

9.2院属信息

9.2.1 GET /gero/{gid}/exchange/{eid} 查询交流信息

路径参数：

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

返回数据：

{

participants:{string}, \\事件相关人员，用json字符串

mediators:{string}, \\调解人，用json字符串

description:{string}, \\描述

result:{string}, \\调解结果

recorder:{int}, \\记录人staff\_id

time:{string},

}

9.2.2 POST /gero/{gid}/exchange/{eid} 修改交流信息

路径参数：

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

提交数据：

{

participants:{string}, \\事件相关人员，用json字符串

mediators:{string}, \\调解人，用json字符串

description:{string}, \\描述

result:{string}, \\调解结果

recorder:{int}, \\记录人staff\_id

time:{string},

}

9.2.3 POST /gero/{gid}/exchanges 新增交流信息

路径参数：

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

提交数据：

{

participants:{string}, \\事件相关人员，用json字符串

mediators:{string}, \\调解人，用json字符串

description:{string}, \\描述

result:{string}, \\调解结果

recorder:{int}, \\记录人staff\_id

time:{string},

}

9.2.4 DELETE /gero/{gid}/exchange/{eid} 删除交流信息

路径参数：

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

十、权限管理

1.角色管理

1.1 GET /gero/{gid}/role 获取角色列表

路径参数：

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

返回数据：

{

role\_list:[{

id:{int},

name:{string},

type:{int}

privilege\_list:[

{

id:{int}

name:{string}

}]

}]

}

1.2 角色信息

1.2.1 GET /gero/{gid}/role/{rid}

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},

Privileges:[

{

Id:{int}

Name:{string}

Remarks:{string}

}]

}

1.2.2 POST /gero/{gid}/role/{rid}

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},

type: {int},

privileges:[

{

Id:{int}

}]

}

1.2.3 PUT /gero/{gid}/role/

路径参数：

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

提交数据:

{

name:{string},

type: {int},

privilege\_list:[

{

Id:{int}

}]

}

1.2.4 DELETE /gero/{gid}/role/{rid}

路径参数：

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

1.3 职工角色列表

1.3.2 POST /gero/{gid}/staffs/role/{rid}

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：

{

staffList:[{

id:{int},

name:{string}, \\员工名称

phone:{string}, \\员工电话

email:{string}, \\员工email地址

idno:{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

}]

}