### HospitalClinicContrller

**1** CommonResult insertClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertClinic(@RequestBody HospitalClinicParam param) |
| Description 功能描述 | 添加诊室信息 |
| Calls 调用函数 | outpatientService.count()、clinicService.insert() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 诊室信息参数（门诊编号、地址） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、clinicService.insert()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

2. CommonResult updateClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateClinic(@PathVariable Long id, @RequestBody HospitalClinicParam param) |
| Description 功能描述 | 更新诊室信息 |
| Calls 调用函数 | outpatientService.count()、clinicService.insert() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 诊室信息参数（门诊编号、地址） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、clinicService.insert()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

3. CommonResult deleteClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult deleteClinic(@PathVariable Long id) |
| Description 功能描述 | 删除诊室信息 |
| Calls 调用函数 | clinicService.count(id)、clinicService.delete(id) |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 诊室编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、clinicService.insert()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

4. CommonResult<HospitalClinic> getClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<HospitalClinic> getClinic(@PathVariable Long id) |
| Description 功能描述 | 添加诊室信息 |
| Calls 调用函数 | clinicService.count(id) |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 诊室编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

### HospitalDoctorController

**1** insertDoctor ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertDoctor(@RequestBody HospitalDoctorParam param) |
| Description 功能描述 | 添加医生信息 |
| Calls 调用函数 | outpatientService.count()、specialService.count()、getGender()、doctorService.insert() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医生信息参数（姓名，性别，职称，专长，所属专科，所属门诊） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、specialService.count()、getGender()、doctorService.insert()对传入的数据进行判断,并再合理条件下向数据库插入信息和进行相应的处理，返回CommonResult传数据给前端

2.updateDoctor ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateDoctor(@PathVariable Long id, @RequestBody HospitalDoctorParam param) |
| Description 功能描述 | 更新医生信息 |
| Calls 调用函数 | outpatientService.count()、specialService.count()、getGender()、doctorService.update() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医生编号、医生信息参数（姓名，性别，职称，专长，所属专科，所属门诊） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、specialService.count()、getGender()、doctorService.update()对传入的数据进行判断，并再合理条件下向数据库修改信息和进行相应的处理，返回CommonResult传数据给前端

3.deleteClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult deleteDoctor(@PathVariable Long id) |
| Description 功能描述 | 删除医生信息 |
| Calls 调用函数 | doctorService.count()、doctorService.delete () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医生编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**doctorService.count()、doctorService.delete ()

对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

4. searchDoctor ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalDoctorDTO>> searchDoctor(@RequestParam(required = false) String name, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：通过姓名，搜索医生信息 |
| Calls 调用函数 | CommonPage.restPage()、doctorService.list() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医生姓名 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage()、doctorService.list()，返回CommonResult传数据给前端

5. listDoctorBySpecialAndOutpatient()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult listDoctorBySpecialAndOutpatient(@RequestParam(required = false) Long specialId, @RequestParam(required = false) Long outpatientId, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：通过专科、门诊，查找医生信息 |
| Calls 调用函数 | CommonPage.restPage()、doctorService.list() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号、门诊编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage()、doctorService.list()，返回CommonResult传数据给前端

6. listDoctorBySpecialAndOutpatientAndName()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult listDoctorBySpecialAndOutpatientAndName(@RequestParam(required = false) String name, @RequestParam Long specialId, @RequestParam Long outpatientId, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：通过姓名、专科、门诊，搜索医生信息 |
| Calls 调用函数 | CommonPage.restPage()、doctorService.list() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医生姓名、专科编号、门诊编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage()、doctorService.list()，返回CommonResult传数据给前端

7. getDoctor()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<HospitalDoctorDTO> getDoctor(@PathVariable Long id) |
| Description 功能描述 | 获取医生信息 |
| Calls 调用函数 | doctorService.count(id)、doctorService.getConvert(id) |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医生编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**doctorService.count(id)、doctorService.getConvert(id)，返回CommonResult传数据给前端

### HospitalInfoController

**1.** insertClinic ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertHospitalInfo(@RequestBody HospitalInfoParam param) |
| Description 功能描述 | 添加医院信息 |
| Calls 调用函数 | infoService.county()、infoService.insert () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院信息参数（名称，图片、电话，地址，简介） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**infoService.county()、infoService.insert ()对传入的数据进行判断，并在合理条件下添加信息到数据库和进行相应的处理，返回CommonResult传数据给前端

2. updateHospitalInfo ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateHospitalInfo(@PathVariable Long id, @RequestBody HospitalInfoParam param) |
| Description 功能描述 | 更新医院信息 |
| Calls 调用函数 | infoService.count()、infoService.update () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院编号、医院信息参数（名称，图片、电话，地址，简介） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**infoService.count()、infoService.update ()对传入的数据进行判断，并在合理条件下修改信息到数据库和进行相应的处理，返回CommonResult传数据给前端

3. deleteHospitalInfo ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult deleteHospitalInfo(@PathVariable Long id) |
| Description 功能描述 | 删除医院信息 |
| Calls 调用函数 | infoService.count()、infoService.delete () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**infoService.count()、infoService.delete ()

对传入的数据进行判断，并在合理条件下删除数据库信息和进行相应的处理，返回CommonResult传数据给前端

4. searchHospitalInfo ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalInfo>> searchHospitalInfo(@RequestParam(required = false) String name, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：搜索医院信息 |
| Calls 调用函数 | CommonPage.restPage()、infoService.list () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院名称 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage()、infoService.list (),进行相应的处理，返回CommonResult传数据给前端.

5. insertSpecialRelation()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertSpecialRelation(@RequestBody HospitalSpecialRelationParam param) |
| Description 功能描述 | 添加专科到医院中 |
| Calls 调用函数 | infoService.count()、specialService.count ()、infoService.countSpecialRelation ()、infoService.insertSpecialRelation () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院专科关系参数（医院编号、专科编号） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**infoService.count()、specialService.count ()、infoService.countSpecialRelation ()、infoService.insertSpecialRelation (),进行相应的处理，返回CommonResult传数据给前端.

6. deleteSpecialRelation()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult deleteSpecialRelation(@RequestParam Long hospitalId, @RequestParam Long specialId) |
| Description 功能描述 | 移除医院中的专科 |
| Calls 调用函数 | infoService.count()、specialService.count ()、infoService.deleteSpecialRelation () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院编号、专科编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**infoService.count()、specialService.count ()、infoService.deleteSpecialRelation (),进行相应的处理，返回CommonResult传数据给前端.

7. insertOutpatientRelation()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertOutpatientRelation(@RequestBody HospitalOutpatientRelationParam param) |
| Description 功能描述 | 添加门诊到医院中 |
| Calls 调用函数 | infoService.count()、specialService.count ()、infoService.deleteSpecialRelation ()、outpatientService.count() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院门诊关系参数（医院编号、门诊编号） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**infoService.count()、specialService.count ()、infoService.deleteSpecialRelation ()、outpatientService.count(),进行相应的处理，返回CommonResult传数据给前端.

8. listSpecialByHospital()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalSpecial>> listSpecialByHospital(@PathVariable Long hospitalId, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 获取医院，所属专科列表 |
| Calls 调用函数 | CommonPage.restPage()、specialService.list ()、infoService.count() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科名称 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage()、specialService.list ()、infoService.count(),进行相应的处理，返回CommonResult传数据给前端.

### HospitalOutpatientController

**1** insertOutpatient ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertOutpatient(@RequestBody HospitalOutpatientParam param) |
| Description 功能描述 | 添加门诊信息 |
| Calls 调用函数 | specialService.count()、outpatinetService.insert() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 门诊信息参数（名称，描述） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**specialService.count()、outpatinetService.insert()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

2. public CommonResult updateClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateClinic(@PathVariable Long id, @RequestBody HospitalClinicParam param) |
| Description 功能描述 | 更新门诊信息 |
| Calls 调用函数 | outpatientService.count()、outpatientService.update()、specialService.count () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 门诊编号、门诊信息参数（名称，描述） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、outpatientService.update()、specialService.count ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

3. public CommonResult deleteClinic()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult deleteOutpatient(@PathVariable Long id) |
| Description 功能描述 | 删除门诊信息 |
| Calls 调用函数 | outpatientService.count()、outpatientService.delete() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 门诊编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()、outpatientService.delete()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

4. getOutpatient ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<HospitalOutpatient> getOutpatient(@PathVariable Long id) |
| Description 功能描述 | 获取门诊信息 |
| Calls 调用函数 | outpatientService.count() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 门诊编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**outpatientService.count()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

5 searchOutpatient()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalOutpatient>> searchOutpatient(@RequestParam(required = false) String name, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：搜索门诊信息 |
| Calls 调用函数 | CommonPage.restPage ()、outpatientService.list () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 门诊名称 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、outpatientService.list ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

6 getOutpatient()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<HospitalOutpatient> getOutpatient(@PathVariable Long id) |
| Description 功能描述 | 分页：查找指定医院、专科的门诊列表 |
| Calls 调用函数 | iutpatientService.count()、CommonPage.restPage ()、outpatientService.list ()、specialService.count() |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 医院编号、专科编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**iutpatientService.count()、CommonPage.restPage ()、outpatientService.list ()、specialService.count()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

7 listAloneOutpatient()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult listAloneOutpatient(@RequestParam(required = false) Long specialId, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：通过专科编号，获取未添加到医院的门诊列表 |
| Calls 调用函数 | CommonPage.restPage ()、outpatientService.list () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号，第几页，页大小 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、outpatientService.list ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

8 listClinic ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalClinic>> listClinic(@PathVariable Long id, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 获取门诊所属诊室信息 |
| Calls 调用函数 | CommonPage.restPage ()、outpatientService.list ()、outpatientService.count () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 门诊编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、outpatientService.list ()、outpatientService.count ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

### HospitalSpecialController

**1** insertSpecial ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertSpecial(@RequestBody HospitalSpecialParam param) |
| Description 功能描述 | 添加专科信息 |
| Calls 调用函数 | specialService.count ()、specialService.insert () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科信息参数（名称，描述） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**specialService.count ()、specialService.insert ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

2. updateSpecial ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateSpecial (@PathVariable Long id, @RequestBody HospitalClinicParam param) |
| Description 功能描述 | 更新专科信息 |
| Calls 调用函数 | specialService.count ()、specialService.update () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号、专科信息参数（名称，描述） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**specialService.count ()、specialService.update ()

对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

3. deleteSpecial ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult deleteSpecial(@PathVariable Long id) |
| Description 功能描述 | 删除专科信息 |
| Calls 调用函数 | specialService.count (id)、specialService.delete(id) |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**specialService.count (id)、specialService.delete(id)对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

4. getSpecial ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<HospitalSpecial> getSpecial(@PathVariable Long id) |
| Description 功能描述 | 获取专科信息 |
| Calls 调用函数 | specialService.count (id) |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**specialService.count (id) 对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

5 searchSpecial()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalSpecial>> searchSpecial(@RequestParam(required = false) String name, @RequestParam Integer pageNum @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：搜索专科信息 |
| Calls 调用函数 | CommonPage.restPage ()、specialService.list () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、specialService.list ()，对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

6 searchOutpatientBySpecial ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<HospitalOutpatient>> searchOutpatientBySpecial(@RequestParam(required = false) Long specialId, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：通过专科编号，获取所属门诊信息 |
| Calls 调用函数 | CommonPage.restPage ()、specialService.list () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 专科编号 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、specialService.list (),对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

### LogController

**1** searchAccountLogin ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<LogAccountLogin>> searchAccountLogin(@RequestParam(required = false) String accountName,  @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：搜索账号登录日志 |
| Calls 调用函数 | CommonPage.restPage ()、accountLoginService.list () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号名称 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、accountLoginService.list ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

2. searchOperation ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<CommonPage<LogOperation>> searchOperation(@RequestParam(required = false) String accountName, @RequestParam(required = false) String method, @RequestParam Integer pageNum, @RequestParam Integer pageSize) |
| Description 功能描述 | 分页：搜索账号操作日志 |
| Calls 调用函数 | CommonPage.restPage ()、operationService.search () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号名称，请求方法（get、post、delete、put） |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonPage.restPage ()、operationService.search ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

### PictureController

**1** insertClinic ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult insertClinic(@RequestBody HospitalClinicParam param) |
| Description 功能描述 | 上传图片，返回图片url |
| Calls 调用函数 | StringUtils.isEmpty () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 图片文件 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**StringUtils.isEmpty ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

### PowerAccountController

**1** verifyAccountName ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<Boolean> verifyAccountName(@RequestParam String name) |
| Description 功能描述 | 验证账号名称 |
| Calls 调用函数 | accountService.count () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号名称, 返回是否存在 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**accountService.count ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

2. loginAccount ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<String> loginAccount(@RequestParam String name, @RequestParam String password) |
| Description 功能描述 | 账号登录 |
| Calls 调用函数 | accountService.count ()、accountService.login () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号名称、账号密码 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**accountService.count ()、accountService.login ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

3. getCurrentAccount ()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<Map<String, Object>> getCurrentAccount(Principal principal) |
| Description 功能描述 | 退出登录账号 |
| Calls 调用函数 | CommonResult.unauthorized (id)、accountService.getByName (id)、accountOptional.isPresent ()、account.setPassword () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 无需参数，需要前端清除 header中的 jwt字符串 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**CommonResult.unauthorized (id)、accountService.getByName (id)、accountOptional.isPresent ()、account.setPassword ()

对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

4. refreshToken()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<String> refreshToken(@RequestParam String token) |
| Description 功能描述 | 刷新 token |
| Calls 调用函数 | StringUtils.isEmpty()、accountService.refreshToken () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | token |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**StringUtils.isEmpty()、accountService.refreshToken ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

**5** updateAccountPassword()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateAccountPassword(@RequestBody PowerAccountUpdatePasswordParam param) |
| Description 功能描述 | 更新账号密码 |
| Calls 调用函数 | accountService.count ()、accountService.checkPassword ()、accountService.updatePassword () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号名称、旧密码、新密码 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**accountService.count ()、accountService.checkPassword ()、accountService.updatePassword ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

**6** updateAccountStatus()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult updateAccountStatus(@RequestBody PowerAccountStatusParam param) |
| Description 功能描述 | 更新账号状态 |
| Calls 调用函数 | accountService.count ()、param.getStatus ()、accountService.updateStatus () |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号编号、账号状态 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**accountService.count ()、param.getStatus ()、accountService.updateStatus ()对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端

**7** updateAccountRoleRelation()

**（1）方法描述**

|  |  |
| --- | --- |
| Prototype 函数原型 | public CommonResult<Integer> updateAccountRoleRelation(@RequestParam Long accountId, @RequestBody List<Long> roleIdList) |
| Description 功能描述 | 更新账号分配角色 |
| Calls 调用函数 | accountService.count ()、accountService.updateRole |
| Called By 被调用函数 | 无 |
| Input 输入参数 | 账号编号，角色编号列表 |
| Output 输出参数 | 无 |
| Return 返回值 | CommonResult |
| Exception 抛出异常 | 无 |

**（2）实现描述**

**调用**accountService.count ()、accountService.updateRole对传入的数据进行判断，并进行相应的处理，返回CommonResult传数据给前端