# 配置管理

两层服务器，第一层node鉴权反向代理，第二层spring api服务器。

暂定 middleware server localhost:8081/api/^ -----------proxy---------> localhost:8080/

api server localhost:8080

# 0.0个人中心

## (Build:ok, Test:?) 1.获取用户信息

提交uid获取基本的信息（常用api，配合redis缓存）

POST /user/info

REQ BODY {

\_uid:1

}

RES {

code:200,

msg:"",

data:{

username:"",

real\_name:"",

department\_id:1,

role\_id:1,

title:"",

department\_name:"",

role\_name:"",

}

}

注意:此后如果在req的json地一层中出现的\_uid，均表示请求者的id，用来标记操作者

## (Build:?,Test:?)用户修改自己的信息

注意密码要用md5 hash

POST /user/updateInfo

REQ BODY {

\_uid:1,

username:"",

password:"",

}

RES {

code:200,

msg:""

}

# 3.1科室管理

## (Build: ok,Test: ok)1.获得全部的科室信息

REQ POST /departmentManage/getAll

RES 返回所有的科室信息

{

"code": 200,

"data":{

department\_classification":[{"id": 8, "name": "中医科" }, {"id": 10, "name": "五官科"…],

"department":[

{"id": 1, "pinyin": "SJK", "name": "神经科", "type": "门诊科室",…}

]

}

}

## (Build: ok,Test: ok)2.通过科室名称查找科室信息

REQ POST /departmentManage/findByName

REQ BODY {"name":"神经科"}

RES 返回所查找到的数据

{

"code": 200,

"data":{

}

}

## (Build: ok,Test: ok)3.添加新的科室

REQ POST /departmentManage/add

REQ BODY

{

"id": 2,

"pinyin": "YK",

"name": "眼科",

"type": "门诊科室",

"classification\_id": 2

}

RES 返回处理过的数据

{

"code": 200

}

## (Build: Ok,Test: ok)4.修改科室

REQ POST /departmentManage/update

REQ BODY

{

"id": 2,

"pinyin": "YK",

"name": "眼科",

"type": "门诊科室",

"classification\_id": 2

}

RES

{

"code": 200

}

## (Build: ok,Test: ok)5.删除科室

REQ POST /departmentManage/delete

BODY 根据所选择的行进行删除

{

"data":[2]

}

RES

{

"code": 200

}

## (Build: Ok,Test: ?)6.批量导入数据

REQ POST /departmentManage/import

POST {file:binery}

RES

{code:200}

# 3.2用户管理

用户管理的Controller里面没有判断报错”编号冲突或该类别不存在！“的提示

## (Build: ok,Test: ok)1.获取全部用户

REQ POST /userManagement/all

RES 返回全部用户的信息

{

"code": 200,

"data":{

"roles":[ 角色

{"id": 6, "name": "医技工作者"},

{"id": 1, "name": "医院管理员"},

{"id": 4, "name": "药房操作员"},

{"id": 3, "name": "财务管理员"},

{"id": 5, "name": "门诊医生"},

{"id": 2, "name": "门诊收费员"}

],

"departments":[ 部门

{

"id": 1,

"pinyin": "SJK",

"name": "神经科",

"type": "临床科室",

"classification\_id": 1,

"classification\_name": "内科"

}

],

"users":[ 用户

{

"uid": 1,

"username": "admin",

"password": null,

"real\_name": "XUranus",

"department\_id": 1,

"role\_id": 1,

"participate\_in\_scheduling": true,

"title": "主任",

"role\_name": "医院管理员",

"department\_name": "神经科"

}

]

}

}

## (Build: ok,Test: ok)2.增加用户

POST /userManagement/add

REQ BODY

通过输入的信息增加用户，其中用户名，密码，真实姓名，角色，科室都是必填项

（如果为医生则要填写职称，以及选择是否参加排班）

{

"username": "admin2",

"password": "2e23e23e",

"real\_name": "XUranus",

"department\_id": 1,

"role\_id": 1,

"participate\_in\_scheduling": true,

"title": "主任",

"role\_name": "医院管理员",

"department\_name": "神经科"

}

RES

{

"code": 200

}

错误描述：

增加相同用户时会直接报错

## (Build: ok,Test: ok)3.修改用户信息

POST /userManagement/update

REQ BODY{

"uid":2,

"username": "admin3",

"password": "2e23e23e",

"real\_name": "XUranus",

"department\_id": 1,

"role\_id": 1,

"participate\_in\_scheduling": true,

"title": "主任",

"role\_name": "医院管理员",

"department\_name": "神经科"

}

RES

{

"code": 200

}

错误描述：测试数据如上的数据

在修改时没有uid=10000时也会正常执行，没有提示

## (Build: ok,Test: ok)4.删除用户信息

POST /userManagement/delete

REQ {

"data":[10000]

}

RES {

"code": 200

}

错误描述：

删除不存在用户时没有提示

## 5.批量导入

REQ POST /userManagement/import

POST {file:binery}

RES

{code:200}

# 3.3挂号等级管理

## (Build: ok,Test: ok)1.获取所有的挂号等级

POST /registrationLevelManagement/all

{

"code": 200,

"data":[

{

"id": 1,

"name": "普通号",

"is\_default": true,

"seq\_num": 1,

"fee": 10

}

]

}

## (Build: ok,Test: ok)2.添加挂号等级

POST /registrationLevelManagement/add

REQ BODY

{

"id": 2,

"name": "专家号",

"is\_default": true,

"seq\_num": 2,

"fee": 20

}

RES

{

"code": 200

}

## (Build: ok,Test: ok)3.更新挂号等级

POST /registrationLevelManagement/update

REQ BODY

{

"id": 2,

"name": "专家号",

"is\_default": true,

"seq\_num": 2,

"fee": 30

}

RES

{

"code": 200

}

## (Build: ok,Test: ok)4.删除挂号等级

POST /registrationLevelManagement/delete

REQ BODY

{

"data": [2]

}

RES

{

"code":200

}

# 3.4支付方式管理

整体没有对错误进行判断

## (Build: ok,Test: ok)1.获取全部的支付方式

POST /settlementCategoryManagement/all

RES

{

"code": 200,

"data":[

{

"id": 4,

"name": "中国交通银行"

},

{

"id": 2,

"name": "中国农业银行"

},

{

"id": 3,

"name": "中国建设银行"

},

{

"id": 6,

"name": "微信"

},

{

"id": 5,

"name": "支付宝"

},

{

"id": 1,

"name": "现金"

}

]

}

## (Build: ok,Test: ok)2.添加支付方式类别

POST /settlementCategoryManagement/add

REQ BODY

{

"id": 10,

"name": "中国人民银行"

}

RES

{

"code":200

}

错误描述：

添加相同id时直接报错

## (Build: ok,Test: ok)3.修改支付方式

POST /settlementCategoryManagement/update

REQ BODY

{

"id": 10,

"name": "中国人民银行"

}

RES

{

"code":200

}

错误描述：

修改不存在的id值时，不报错，不提示

## (Build: ok,Test: ok)4.删除支付方式

POST /settlementCategoryManagement/delete

REQ BODY

{

"data":[10]

}

RES

{

"code":200

}

错误描述：

删除不存在的id值时，不报错，不提示

# 3.6非药品目录管理

## (Build: ok,Test: ok)1.获取全部的非药品目录

POST /nonDrugChargeItemManagement/all

RES

{

"code": 200,

"data":{

"expense\_classification":[

{"id": 1, "pinyin": "GHF", "fee\_name": "挂号费"},

{"id": 2, "pinyin": "ZLF", "fee\_name": "诊疗费"},

{"id": 3, "pinyin": "JYF", "fee\_name": "检验费"},

{"id": 4, "pinyin": "JYCLF", "fee\_name": "检验材料费"},

{"id": 5, "pinyin": "CSJCF", "fee\_name": "超声检查费"},

{"id": 6, "pinyin": "CSCLF", "fee\_name": "超声材料费"},

{"id": 7, "pinyin": "FSJCF", "fee\_name": "放射检查费"},

{"id": 8, "pinyin": "FSCLF", "fee\_name": "放射材料费"},

{"id": 9, "pinyin": "MRIJCF", "fee\_name": "MRI检查费"},

{"id": 10, "pinyin": "MRICLF", "fee\_name": "MRI材料费"},

{"id": 11, "pinyin": "CTJCF", "fee\_name": "CT检查费"},

{"id": 12, "pinyin": "CTCLF", "fee\_name": "CT材料费"},

{"id": 13, "pinyin": "XYF", "fee\_name": "西药费"},

{"id": 14, "pinyin": "ZCYF", "fee\_name": "中成药费"},

{"id": 15, "pinyin": "ZCYF", "fee\_name": "中草药费"},

{"id": 16, "pinyin": "CZF", "fee\_name": "处置费"},

{"id": 17, "pinyin": "CZCLF", "fee\_name": "处置材料费"},

{"id": 18, "pinyin": "MZF", "fee\_name": "麻醉费"},

{"id": 19, "pinyin": "MZYF", "fee\_name": "麻醉药费"},

{"id": 20, "pinyin": "MZSSF", "fee\_name": "门诊手术费"},

{"id": 21, "pinyin": "QT", "fee\_name": "其他"}

],

"department":[ 科室

{

"id": 1,

"pinyin": "SJK",

"name": "神经科",

"type": "临床科室",

"classification\_id": 1,

"classification\_name": "内科"

}

],

"non\_drug\_charge":[ 非药品收费目录

{

"id": "A1010.10.111",

"pinyin": "DQJ",

"format": "次",

"name": "大抢救",

"fee": 100,

"expense\_classification\_id": 20,

"department\_id": 1,

"expense\_classification\_name": "门诊手术费",

"department\_name": "神经科"

}

]

}

}

## (Build: ok,Test: ok)2.添加非药品收费目录

POST /nonDrugChargeItemManagement/add

REQ BODY

{

"id": "A1010.10.121",

"pinyin": "DQJ",

"format": "次",

"name": "小抢救",

"fee": 100.0,

"expense\_classification\_id": 20,

"department\_id": 1,

"expense\_classification\_name": "门诊手术费",

"department\_name": "神经科"

}

RES {"code":200}

## (Build: ok,Test: ok)3.修改非药品收费目录

POST /nonDrugChargeItemManagement/update

REQ BODY

{

"id": "A1010.10.121",

"pinyin": "DQJ",

"format": "次",

"name": "小抢救",

"fee": 100.0,

"expense\_classification\_id": 20,

"department\_id": 1,

"expense\_classification\_name": "门诊手术费",

"department\_name": "神经科"

}

RES {"code":200}

## (Build: ok,Test: ok)4.删除非药品收费目录

POST /nonDrugChargeItemManagement/delete

REQ BODY 根据被选中的统一编号

{

"data":["A1010.10.121"]

}

RES {"code":200}

错误描述：

删除不存在的id时不做错误提示，不报错

## (build:?, test:?)5.非药品收费目录导入

REQ POST /nonDrugChargeItemManagement/import

POST {file:binery}

RES

{code:200}

# 3.5诊断目录管理

## (Build: ok,Test: ok)1.获取全部的诊断目录

POST diagnoseDirectoryManagement/allClassification

REQ BODY {}

RES

{

"code": 200,

"data":[

{

"id": 1,

"name": "发烧"

}

]

}

## (Build: ok,Test: ok)2.通过疾病目录类别查询在此类别中的所有的疾病

POST /diagnoseDirectoryManagement/searchAllByClassificationId

REQ BOSY 通过疾病目录类别来查找诊断目录

{

"classification\_id":1

}

RES

{

"code": 200,

"data":{

"diseases":[

{

"id": "A028191",

"name": "猩红热",

"classification\_id": 1,

"pinyin": "XHR",

"custom\_name": "陈女士",

"custom\_pinyin": "chennvshi",

"classification\_name": "发烧"

}

]

}

}

## (Build: ok,Test: ok)3.通过疾病名称查询疾病信息

POST /diagnoseDirectoryManagement/findByName

REQ BODY {"name":"猩红热"}

RES {

"code": 200,

"data":{

"id": "A028191",

"name": "猩红热",

"classification\_id": 1,

"pinyin": "XHR",

"custom\_name": "猩红热",//自定义疾病名称

"custom\_pinyin": "XHR",

"classification\_name": "发烧"

}

}

## (Build: ok,Test: ok)4.修改疾病目录

Fix Log:

raw\_id是用于定位原来行的id,post的id是改后的新id，update ...set id=id，，where id=raw\_id

POST /diagnoseDirectoryManagement/update

REQ BODY

{

"raw\_id":1,

"id":1,

"code":"A1019"

"name":"传染性单核细胞增多症",

"classification\_id":1,"

pinyin":"CRXDHXBZDZ",

"custom\_name":"传单",

"custom\_pinyin":"ChuanDan"

}

RES

{

"code": 200

}

## (Build: ok,Test: ok)5.添加疾病

POST /diagnoseDirectoryManagement/add

REQ BODY

{

"id":"99999999",

"name":"传染性单核细胞增多2",

"classification\_id":1,

"code":"B0347238",

"pinyin":"CRXDHXBZDZ",

"custom\_name":"传单",

"custom\_pinyin":"ChuanDan"

}

RES {code:200}

错误描述：

添加疾病不成功，无法正确添加

## (Build: ok,Test: ok)6.删除疾病

POST /diagnoseDirectoryManagement/delete

REQ BODY

{

"data":["29395"]

}

RES {

"code":200

}

7.REQ POST http://localhost:8080/diagnoseDirectoryManagement/import

POST {file:binery}

RES

{code:200}

# 3.7排班

## (Build:ok,Test:ok)1.获取并显示全部可排班人员信息

REQ POST /doctorSchedulingManagement/all

RES

{

"code": 200,

"data":[

{

"id": 10001,

"name": "菜徐坤1",

"department\_name": "心血管内科",

"title": "专家",

"shift": "上午",

"expiry\_date": "2019-06-19",

"scheduling\_limit": 50,

"registration\_Level": "专家号"

},

{

"id": 10004,

"name": "菜徐坤4",

"department\_name": "心血管内科",

"title": "专家",

"shift": "全天",

"expiry\_date": "2019-06-13",

"scheduling\_limit": 60,

"registration\_Level": "专家号"

}]

}

## (Build:ok,Test:ok)2.将可排班人员移出可排班人员列表

POST /doctorSchedulingManagement/delete

REQ BODY

{

"data":[10003]

}

RES

{

code=200

}

## (Build:ok,Test:ok)3.选择一个用户加入可排班人员列表

POST /doctorSchedulingManagement/add

REQ BODY{

"id":"10004",

"name":"菜徐坤4",

"department\_name":"心血管内科",

"shift":"全天",

"expiry\_date":"2019-06-13 03:08:26",

"registration\_Level":"专家号",

"scheduling\_limit":50

}

RES

{

code=200

}

## (Build:ok,Test:ok)4.修改可排班人员信息

POST /doctorSchedulingManagement/update

REQ BODY{

"id":10004,

"shift":"全天",

"expiry\_date":"2019-06-13T03:08:26.608Z",

"registration\_Level":"专家号",

"scheduling\_limit":60

}

RES

{

code=200

}

## (Build:ok,Test:ok)5.根据id查找用户信息

POST /doctorSchedulingManagement/addPersonnelTable

REQ BODY{

"data":"10004"

}

RES

{

code=200,

data":[

{

"name": "菜徐坤4",

"department\_name": "心血管内科"

}]

}

## (Build:ok,Test:ok)6.根据name查找用户信息

POST /doctorSchedulingManagement/addNamePersonnelTable

REQ BODY{

"data":"菜徐坤5"

}

RES

{

"code": 200,

"data":[

{

"id": 10005,

"department\_name": "心血管内科",

}]

}

## (Build:ok,Test:ok)7.得到这一次的排班结果

REQ POST /doctorWorkforceManagement/all

RES:{

"code": 200,

"data":[]

}

## (Build:ok,Test:ok)8.在排班页面获取所有已排班信息

REQ POST /doctorWorkforceManagement/getAll

RES:{

"code": 200,

"data":[

{

"id": 326,

"schedule\_date": "2019-06-13",

"week": "4",

"name": "菜徐坤4",

"department\_name": "心血管内科",

"registration\_Level": "专家号",

"shift": "上午",

"reg\_limit": "50",

"residue": "50",

"valid": "有效"

},

{

"id": 327,

"schedule\_date": "2019-06-13",

"week": "4",

"name": "菜徐坤1",

"department\_name": "心血管内科",

"registration\_Level": "专家号",

"shift": "上午",

"reg\_limit": "90",

"residue": "90",

"valid": "有效"

}]

}

## (Build:ok,Test:ok)9.选择排班人员

POST /doctorWorkforceManagement/choose

REQ BODY{

"data":[

{"id":10001,

"name":"菜徐坤1",

"department\_name":"心血管内科",

"title":"专家",

"shift":"全天",

"expiry\_date":"2019-06-30",

"scheduling\_limit":90,

"registration\_Level":"专家号",

"key":10001}],

"dateRange":["2019-06-13", "2019-06-14"],

"num":1

}

RES

{

code=200

}

## (Build:ok,Test:ok)10.向数据库中添加排班信息

POST /doctorWorkforceManagement/add

REQ BODY:{

"data":[

{"name":"菜徐坤4",

"i":0,

"j":1,

"scheduleDate":"2019-06-13",

"shift":"上午",

"registration\_Level":"专家号",

"week":4,

"residue":50,

"valid":"有效",

"reg\_limit":50,

"key":"菜徐坤4"},

{"name":"菜徐坤1",

"i":0,

"j":2,

"scheduleDate":"2019-06-13",

"shift":"上午",

"registration\_Level":"专家号",

"week":4,

"residue":90,

"valid":"有效",

"reg\_limit":90,

"key":"菜徐坤1"},

{"name":"菜徐坤4",

"i":1,

"j":1,

"scheduleDate":"2019-06-13",

"shift":"下午",

"registration\_Level":"专家号",

"week":4,

"residue":50,

"valid":"有效",

"reg\_limit":50,

"key":"菜徐坤4"}]}

RES

{

code=200

}

## (Build:ok,Test:ok)11.添加排班信息

POST /doctorWorkforceManagement/addRow

REQ BODY:{

"name":"菜徐坤4",

"schedule\_date":"2019-06-13",

"shift":"上午",

"department\_name":"心血管内科",

"registration\_Level":"专家号",

"residue":"50",

"valid":"有效",

"reg\_limit":"50"

}

RES

{

code=200

}

## (Build:ok,Test:ok)12.添加排班信息时，查找是否已存在该排班信息。

POST /doctorWorkforceManagement/addRowConflict

REQ BODY:{

"name":"菜徐坤4",

"schedule\_date":"2019-06-13",

"shift":"上午",

"department\_name":"心血管内科",

"registration\_Level":"专家号",

"residue":"50",

"valid":"有效",

"reg\_limit":"50"

}

RES:{

"code": 200,

"data":[

{

"id": 394,

"schedule\_date": "2019-06-13",

"week": null,

"name": null,

"department\_name": null,

"registration\_Level": null,

"shift": null,

"reg\_limit": null,

"residue": null,

"valid": null

},

{

"id": 397,

"schedule\_date": "2019-06-13",

"week": null,

"name": null,

"department\_name": null,

"registration\_Level": null,

"shift": null,

"reg\_limit": null,

"residue": null,

"valid": null

}

]

}

## (Build:ok,Test:ok)13.删除一行排班信息

POST /doctorWorkforceManagement/delete

RES BODY:{

"data":[{

"id":397,

"schedule\_date":"2019-06-30",

"week":4,

"name":"菜徐坤4",

"department\_name":"心血管内科",

"registration\_Level":"专家号",

"shift":"上午",

"reg\_limit":50,

"residue":50,

"valid":"有效",

"key":"菜徐坤4"

}]

}

RES

{

code=200

}

## (Build:ok,Test:ok)14.获取排班时间,如果该时间段已排班，返回这段时间的排班信息

POST /doctorWorkforceManagement/time

REQ BODY:{

"data":["2019-06-13", "2019-06-14"]

}

RES:{

"code": 200,

"data":[

{

"id": 394,

"schedule\_date": "2019-06-13",

"week": null,

"name": null,

"department\_name": null,

"registration\_Level": null,

"shift": null,

"reg\_limit": null,

"residue": null,

"valid": null

}]

}

## (Build:ok,Test:ok)15.输入name后将其余数据填入表格

POST /doctorWorkforceManagement/addTable

REQ BODY{

"data":"菜徐坤4"

}

RES:{

"code": 200,

"data":[

{

"id": 383,

"schedule\_date": "2019-06-26",

"week": "3",

"name": null,

"department\_name": "心血管内科",

"registration\_Level": "专家号",

"shift": "上午",

"reg\_limit": null,

"residue": "60",

"valid": "有效"

}]

}

## (Build:ok,Test:ok)16.覆盖排班信息

POST /doctorWorkforceManagement/overwrite

REQ BODY:{

"data":[

{

"id":396,

"scheduleDate":"2019-06-13"}

]}

RES:{

code=200

}

# 4.1 4.3 挂号/退号

## (Build: ok, Test: ok)1.初始化面板信息

POST /outpatientRegistration/init

RES {

departments:[ //所有科室

{name:"骨科",id:1},

{name:"脑科",id:2}

],

defaultRegistrationLevel:{ //默认的挂号等级

name:"普通号",id:2,price:10

},

registrationLevel:[ //所有挂号等级

{name:"专家号",id:1,price:20},

{name:"普通号",id:2,price:10}

],

settlementCategory:[ //所有支付方式

{name:"现金",id:1},

{name:"农行卡",id:2},

{name:"支付宝",id:3}

],

}

## (Build: ok,Test: ok)2.根据选择的科室id和号别信息拉取符合条件的医生信息

POST /outpatientRegistration/syncDoctorList

REQ BODY

{

"department\_id":1,

"registration\_level\_id":2,

}

RES

{

code:200,

msg:"",

data:[{name:"孙医生",id:1}]

}

(上述是必须字段，数据字段可以多 但是不能少，但是用户密码一定不能发回来，发回来之前setPassword(null)) 挂号等级是和医生的职称挂钩，你写个hashMap映射函数 比如registration\_level\_id:1是专家科, 就映射到职称="主任" 自己先编 。然后返回符合条件的医生数据

后续可能还要根据当前时间判断排班是否满足

后续可能要修改一下RegistrationConfig文件中的对应关系

## (Build: ok,Test:ok)3.提交挂号信息，计算费用

POST /outpatientRegistration/calulateFee

REQ {

"registration\_level\_id": 2 //挂号级别

"registration\_source": "app" //App挂号, 窗口挂号

"settlement\_category\_id": 2 //结算类别

}

RES {

code:200,

data:{

fee:20

}

}

## (Build: ok, Test: ok)4.挂号确认/收费

iancha

POST /outpatientRegistration/confirm

REQ {

"address" : "沈阳市东北大学",

"age" : 17,

"birthday" : "1998.03.23",

"consultation\_date": "2019.10.2",

"department\_id": 3,

"gender": "男",

"has\_record\_book": 1,

"medical\_category": "不知道什么意思",

"medical\_certificate\_number": "371625223284621134",

"medical\_certificate\_number\_type": "id",

"medical\_insurance\_diagnosis": "不知道什么意思",

"patient\_name": "王蓓蕾",

"outpatient\_doctor\_id" : 3,

"settlement\_category\_id": 2,

"registration\_source": "app挂号",

"registration\_level\_id": 1,

"should\_pay" : 28.24,

"truely\_pay" : 100,

"retail\_fee" : 71.76

}

if medical\_certificate\_number\_type==id then

medical\_certificate\_number 是身份证号码

else

medical\_certificate\_number 是医保卡号

end

操作3张表:

**挂号记录**插入一条记录, 返回id

**票据记录**插入一条 type"缴费"

**操作记录**插入一条 type "挂号"

RES {

code:200,

data:{

medical\_record\_number: 10000002 //生成的病历号, 自增主键

bill:{

"medical\_record\_id" : 10000002, //病历号

"user\_id" : 2, //操作员,

"expense\_classification" : "挂号费",

"quantity": "1",

"cost" : 20.01,

"create\_time": "2019-6-4 14:10"

}

}

}

## (Build: ok,Test: ok)5.输入病历号，退号 注意判断状态

POST /outpatientRegistration/withdrawNumber

REQ

{

"medical\_record\_id" : 10000002 //病历号

}

操作3张表:

**挂号记录**(registration) **update** status -> 已退号

**票据记录**(bill\_record)插入一条 type"退费" cost = -cost 对冲

**操作记录**(operate\_log) 插入一条 type "挂号" fee = -fee 对冲

RES

{

code : 200

}

# 4.2 4.4 收费/退费

## (Build:ok, Test:ok)1.通过病历号, 获取收费项目列表（只获取未缴费的项目）

POST /outpatientCharge/getChargeItems(这个应该是功能模块名 不是角色名)

REQ

{

"medical\_record\_id" : 10000002 //病历号

}

查询:

**收费记录**表(**outpatient\_charges\_record**)

RES

{

code : 200,

data: {

[

{

id:1

medical\_record\_id : 12312,

item\_name : "大抢救",

format : "日",

price : "200",

quantity: "2",

cost : "400",

excute\_department : "急救科",

expense\_classification: "处置费",

status: "已缴费",

create\_time :"2019-09-08"

},

{

id:2

medical\_record\_id : 12312,

item\_name : "小抢救",

format : "日",

price : "100",

mount : "2",

fee : "200",

excute\_department : "急救科",

expense\_classification: "处置费",

status: "未缴费",

create\_time :"2019-09-08"

}

]

}

}

## (Build:ok, Test:ok)2. 患者挂号基本信息

POST /outpatientCharge/registrationByRecordId

REQ {

"medical\_record\_id" : "100002"

}

RES{

code:200

data:{

medical\_record\_id :,

patient\_name :,

age :,

gender:,

medical\_insurance\_diagnosis :,

registration\_department\_id:

registration\_category :"",

registration\_source:""

}

## (Build:ok, Test:ok)3. 收费

POST /outpatientCharge/charge

REQ

{

"medical\_record\_id" : 10000002, //病历号

"charges\_id\_list": [9, 10, 11, 12],

"should\_pay" : 28.24,

"truely\_pay" : 100,

"retail\_fee" : 71.76

}

查询:

if 所有id属于本病历号 else "没有该收费条目"

if 所有id都未缴费 else "同步错误, 某条目已缴费"

then

修改**收费记录**(**outpatient\_charges\_record**)表

update 未缴费 -> 缴费

添加**票据记录(bill\_record)**表

type=缴费

添加**操作记录(operate\_log)**表

type = 开药 | 检查 | 检验

RES

{

code : 200

}

RES

{

code:500,

data{

ids\_not\_have:[],

ids\_have\_charged\_or\_refunded:[]

}

}

## (Build:ok, Test:ok)4. 退费

POST /outpatientCharge/withdraw

REQ

{

"medical\_record\_id" : 10000002, //病历号

"charges\_id\_list": [9],

}

查询:

if 所有id属于本病历号 else "没有该收费条目"

if 所有id都未缴费 else "同步错误, 某条目已缴费"

then

修改**收费记录**(**outpatient\_charges\_record)**表

update 缴费 -> 已退费

对冲

添加**票据记录(bill\_record)**表

type=退费

cost= -cost

添加**操作记录(operate\_log)**表

type = 开药 | 检查 | 检验

fee = -fee

RES

{

code : 200

}

RES

{

code:500,

data{

ids\_not\_have:[],

ids\_not\_charged\_or\_have\_refunded:[]

}

}

# 4.5 患者历史费用查询

## (Build:ok, Test:ok)1. 费用查询

POST /outpatientCharge/getHistoryChargeItems

REQ

{

"medical\_record\_id" : 10000002 //病历号

"start\_time":"",

"end\_time":""

}

检查

end\_time <= 当前时间

end\_time > start\_time

查询:

**收费记录**表

RES

{

code : 200,

data: {

[

{

id:1,

medical\_record\_id : 12312,

item\_name : "大抢救",

format : "日",

price : "200",

mount : "2",

fee : "400",

excute\_department : "急救科",

expense\_classification: "处置费",

status: "已缴费",

creat\_time :"2009-12-23 08:09:09",

collect\_time : "2009-02-02 17:09:09"

return\_time : "2009-02-02 17:09:09"

},

{

id:2,

medical\_record\_id : 12312,

item\_name : "小抢救",

format : "日",

price : "100",

amount : "2",

fee : "200",

excute\_department : "急救科",

expense\_classification: "处置费",

status: "未缴费",

creat\_time :"2009-12-23 08:09:09",

collect\_time : "2009-02-02 17:09:09"

return\_time : "2009-02-02 17:09:09"

}

]

}

}

# 4.6 收费员日结

## (Build:ok, Test:ok)1.日结历史列表

POST /dailyCollect/list

REQ

{

//隐式\_uid

}

查询日结记录, user\_id = \_uid

RES

{

code : 200,

data : [

{

id:,#日结表id

start\_time : "2019-6-4 16:15",

end\_time : "2019-6-4 20:15"},

{

id:,

start\_time : "2019-7-4 16:15",

end\_time : "2019-8-4 20:15"}

]

}

## (Build:ok, Test:ok)2.日结详细信息

POST /dailyCollect/detail

REQ

{

"daily\_collect\_id" : 4

}

操作:

select \* from daily\_detail where daily\_collect\_id = #{daily\_collect\_id}

ForEach

返回bill\_record信息

RES

{

code : 200,

data:[

{ id #收费id

medical\_record\_id

created\_date

expense\_classification\_id

type

item\_id

quantity

cost

should\_pay

truely\_pay

retail\_fee

user\_id

excute\_department\_id

create\_time},

{id

medical\_record\_id

created\_date

expense\_classification\_id

type

item\_id

quantity

cost

should\_pay

truely\_pay

retail\_fee

user\_id

excute\_department\_id

create\_time},

]

}

## (Build:ok, Test:ok)3.日结请求

POST /dailyCollect/collect

REQ

{

????"start\_time": "2019-6-4 16:15"

"end\_time": "2019-6-4 16:15"

}

???? 查询最后一条日结记录, 比较时间合法性

查找所有该user票据, 在当前时间段内.

将所有票据id写入daily\_detail

创造日结记录daily\_collect

返回该日结记录

RES

{

code : 200,

data:{

id: 4,

total\_fee: 总价,//

start\_time : "2019-6-4 16:15",

end\_time : "2019-6-4 20:15"

}

}

# 5.1 门诊病历首页

只有自己的病人才能修改 ma ?

## (Build:ok,Test:？)获取全部中医诊断和西医诊断

POST /medicalRecord/allDiagnoseDiseases

chineseDiagnoseDiseases:[

{id:10,name:"脑残1",key:10,code:"A1029"},

{id:11,name:"脑残2",key:11,code:"A1039"},

],

westernDiagnoseDiseases:[

{id:12,name:"脑残1",key:12,code:"A1049"},

{id:13,name:"脑残2",key:13,code:"A1059"},

]

}

## (Build:ok,Test:ok)0.获得等待看诊和暂存的病人挂号列表

POST /medicalRecord/getPatientList

REQ {}

RES {

code:200,

data:{

waiting:{..挂号信息表}, 还没创建病例记录的

pending:{...挂号信息表} 已经创建但没有诊断完毕的

}

}

## (Build:ok, Test:ok)1. 患者挂号基本信息

POST /medicalRecord/registrationInfo

前端 身份证中加 \*\*

REQ {

"medical\_certificate\_number" : "37152659294836"

"type":"id" //"mid"

}

查找该患者所有未看诊状态的挂号信息.

RES{

code:200

data:[{

medical\_record\_id :,

**outpatient\_doctor\_id**:我觉得要加上医生的id，不是自己的病人病例不能修改，只读

patient\_name :,

age :,

gender:,

medical\_insurance\_diagnosis :,

registration\_department:"神经科",

registration\_category :"",

registration\_source:""},

{

medical\_record\_id :,

patient\_name :,

age :,

gender:,

medical\_insurance\_diagnosis :,

registration\_department:"神经科",

registration\_category :"",

registration\_source:""},

]}

获得等待 和暂存的病人挂号信息列表

## (Build:ok, Test:ok)2.病历获取

POST medicalRecord/getMedicalRecord

REQ {

"medical\_record\_id" : 100000002

}

验证是否已经有病历,

创建病历, 状态为"暂存"

<http://www.mocky.io/v2/5d037b2530000024001f4d42>

## (Build:ok，Test: ok)3.查看全部历史病历

POST /medicalRecord/allHistoryMedicalRecord

REQ {

type:"mid" /"id",

"medical\_certificate\_number":"123123132"

}

<http://www.mocky.io/v2/5d03185330000051001f4a44>

## (Build: ok, Test: ok)4.病历提交保存 (修改状态为"已提交")

POST medicalRecord/saveMedicalRecord（这里改了）

REQ {

和上述的返回的data一样，太大了，不写了

}

检查数据库是否有该病历

RES

{

code:200

}

## (Build:ok, Test: ok)5.病历修改暂存 (维持状态为"暂存")

新加的

POST medicalRecord/updateMedicalRecord

REQ {

和上述的返回的data一样，太大了，不写了

}

检查数据库是否有该病历

RES

{

code:200

}

## (Build:ok, Test:ok)6.病历提交确诊 (维持状态为"诊毕")

新加的

POST medicalRecord/confirmMedicalRecord

REQ {

和上述的返回的data一样，太大了，不写了

}

检查数据库是否有该病历

RES

{

code:200

}

下面outpatientDoctor/medicalRecordTemplate/XXX可以作为一个功能模块（病例模板功能）

## (Build:ok, Test:ok)7. 病历模板列表

POST /medicalRecordTemplate/list

REQ {

}

<http://www.mocky.io/v2/5d030d5c30000055001f4a1f>

## (Build:ok, Test:ok)8.病历模板详情

POST /medicalRecordTemplate/detail

REQ {

id : 2,

}

查询病历模板 找到id

<http://www.mocky.io/v2/5d030bc630000060001f4a1e>

## (Build:ok, Test:ok)9. 病历模板创建

我觉得没有创建病例模板，只有存为病例模板，REQ和病例的保存类似

POST /medicalRecordTemplate/create

REQ {

"title":"神经病模板",

"type":1,

"chief\_complaint":"主诉",

"current\_medical\_history":"无现病史",

"current\_treatment\_situation":"现病治疗情况",

"past\_history":"无既往史",

"allergy\_history":"无过敏史",

"physical\_examination":"正常",

}

1.创建病历模板

创建病历模板对象

名称是否重复, 重复后自动重命名(1)

2. 选择type = 0 user\_id == \_uid

选择type=1 deparment\_id = \_uid,department\_id

选择type=2 所有

**返回病历模板列表, 用以更新列表**

RES

{

code:200

}

## (Build:ok, Test:ok)10. 病历模板更新

POST /medicalRecordTemplate/update

REQ {

"id":2,

"title":"神经病模板",

"type":1,

"chief\_complaint":"主诉",

"current\_medical\_history":"无现病史",

"current\_treatment\_situation":"现病治疗情况",

"past\_history":"无既往史",

"allergy\_history":"无过敏史",

"physical\_examination":"正常",

}

返回病历模板列表, 用以更新列表

RES

{

code:200

}

## (Build:ok, Test:ok)11. 病历模板删除

POST /medicalRecordTemplate/delete

REQ {

idArr:[1，2,3]

}

返回病历模板列表, 用以更新前端列表

RES

{

code:200,

}

# 5.4 门诊断模板(常用诊断)

## (Build: ok,Test: ok)1.常用诊断模板列表

POST /diagnoseTemplate/list

REQ{}

<http://www.mocky.io/v2/5d05088d3200004b00d78c1b>

## (Build: ok,Test: ok)2.创建诊断模板

POST /diagnoseTemplate/create

RES

{

"title":"XX模板",

"type":1,

"diagnose":{

"western\_diagnose":[

{

"disease\_id":1,

"disease\_name":"神经病",

"disease\_code":"XY82873",

"diagnose\_type":"西医",

"main\_symptom":true,

"suspect":true

}

],

"chinese\_diagnose":[

{

"disease\_id":3,

"disease\_name":"气虚",

"disease\_code":"ZY23873",

"diagnose\_type":"中医",

"syndrome\_differentiation":"气血逆行，印堂发黑",

}

]

}

}

RES{

code:200,

}

## (Build: ok,Test: ok)3.更新诊断模板

POST /diagnoseTemplate/update

{

"id":1,

"title":"XX模板",

"type":1,

"diagnose":{

"western\_diagnose":[

{

"disease\_id":1,

"disease\_name":"神经病",

"disease\_code":"XY82873",

"diagnose\_type":"西医",

"main\_symptom":true,

"suspect":true

}

],

"chinese\_diagnose":[

{

"disease\_id":3,

"disease\_name":"气虚",

"disease\_code":"ZY23873",

"diagnose\_type":"中医",

"syndrome\_differentiation":"气血逆行，印堂发黑",

}

]

}

}

{code:200}

## (Build: ok,Test: ok)4.删除诊断模板

POST /diagnoseTemplate/delete

REQ {

idArr:[1,2,3]

}

注意级联删除子目

RES {code:200}

## (Build: ok,Test: ok)5.诊断模板详情

POST /diagnoseTemplate/detail

REQ {id:1}

RES {

"id":1,

"title":"XX模板",

"type":1,

"diagnose":{

"western\_diagnose":[

{

"disease\_id":1,

"disease\_name":"神经病",

"disease\_code":"XY82873",

"diagnose\_type":"西医",

"main\_symptom":true,

"suspect":true

}

],

"chinese\_diagnose":[

{

"disease\_id":3,

"disease\_name":"气虚",

"disease\_code":"ZY23873",

"diagnose\_type":"中医",

"syndrome\_differentiation":"气血逆行，印堂发黑",

}

]

}

}

# 5.2 5.3 5.5 0检查/1检验/2处置

inspection检查 analysis检验 disposal处置

**0 检查 1 检验 2 处置**

@edit

## (Build: ok,ok,Test: ok)1.根据类型 获取全部的项目（非药品）目录

POST /exam/allItems

REQ {type:0}

RES {

code:200,

msg:"",

data:[{

id,code,pinyin,format,name,fee,**expense\_classification\_id,department\_id,**

}]

}

@Edit

## (Build:ok,ok, Test:ok)3.创建(检查/检验/处置)

POST /exam/create

REQ {

type:0,

medical\_record\_id : 1000002

non\_drug\_id\_list:[]

}

检查病历表, 是否已经提交

插入检查检验处置表, 状态为 "暂存"

foreach non\_drug\_id\_list:[] 插入检查检验处置详情表 状态为未缴费

RES

{

code:200,

所有的创建 删除 更新 返回都是code 200,不需要数据

}

@Add

## (Build:ok,ok, Test:ok)3.删除(检查/检验/处置)

POST /exam/delete

REQ {

id:[1]

}

RES

{code:200}

@Add

## (Build:ok,ok, Test:ok)3.更新(检查/检验/处置)

POST /exam/update

REQ {

type:0,

id : 1000002

non\_drug\_id\_list:[]

}

根据medical\_record\_id，删掉所有子目，然后重新插入

RES {code:200}

@Add

## (Build:ok,ok, Test:ok)3.作废(检查/检验/处置)

POST /exam/cancel

REQ {

id:[1]

}

根据medical\_record\_id，删掉所有子目，然后重新插入,注意状态

RES {code:200}

@Add

## (Build:ok,ok, Test:ok)3.发送(检查/检验/处置)

确认，改变状态，发送到医技科

POST /exam/send

REQ {

id:[1]

}

RES {code:200}

@Add

## (Build:ok, Test:?)6.获取全部,显示用(检查/检验/处置)

POST outpatientDoctor/exam/list

REQ {

type:0,

medical\_record\_id:1,

}

RES

{

"code":200,

"data":[

{

"id":1,

"type":0,

"medical\_record\_id":1,

"user\_id":1,

"create\_time":"2019-09-08",

"status":"暂存",

"items":

[

{

"id":1,

"exam\_id":1,

"non\_drug\_item\_id":1,

"status":"已等级",

"non\_drug\_item":{

"id":1,

"code":"UIW82",

"pinyin":"SY",

"format":"XX规格",

"name":"输液",

"fee":"10",

"expense\_classification\_id":1,

"department\_id":1

}

}

]

}

]

}

------- 以下是组套

@Edit

## (Build:ok,ok Test:ok)7.获取全部组套列表

POST outpatientDoctor/examTemplate/all

REQ {type:0}

RES

{

code:200,

data : [

{

id :0,

name:"神经病检查",

display\_type:1,

},

{

id:1,

name:"赚钱检查"

display\_type:2

}

}

@Edit

## (Build:ok,ok, Test:ok)8.存为组套

POST outpatientDoctor/examTemplate/create

REQ {

type:0,

display\_type:0,

template\_name :"神经病检查"

non\_drug\_id\_list: []

}

检查name是否重复(前后端)

检查list是否非空(前后端)

插入检查检验处置组套表, 获取id

non\_drug\_id\_list foreach 插入

检查/检验/处置 组套详情

exam\_template\_id = id

返回 type:0 组套列表, 前端更新列表

RES

{

code:200,

}

## (Build:ok, ok,Test:failed)9.组套详细信息

POST outpatientDoctor/examTemplate/detail

REQ {

exam\_template\_id : 2

}

获取组套id相关信息

RES

{

"code": 200,

"data": {

"template\_name":"wxwxw",

"display\_type":1,

items:

[{

"id": 0,

"exam\_template\_id": 2,

"non\_drug\_item": {

"id": "A1010.10.111",

"pinyin": "DQJ",

"format": "次",

"name": "大抢救",

"fee": 100,

"expense\_classification\_id": 20,

"expense\_classification\_name": "门诊手术费",

"department\_name": "神经科"

}

}, {

"id": 1,

"exam\_template\_id": 2,

"non\_drug\_item": {

"id": "A1010.10.111",

"pinyin": "DQJ",

"format": "次",

"name": "大抢救",

"fee": 100,

"expense\_classification\_id": 20,

"expense\_classification\_name": "门诊手术费",

"department\_name": "神经科"

}

}]

}

}

## (Build:ok,ok, Test:ok)1. 组套删除

POST outpatientDoctor/examTemplate/delete

REQ {

id :[]

}

检查是否存在

删除组套表

删除组套详情表

返回组套列表, 用以更新前端列表

RES

{

code:200,

}

@Add

## (Build:ok,ok,Test:ok)更新组套

POST outpatientDoctor/examTemplate/update

REQ {

id:1,

type:0,

display\_type:0,

template\_name :"神经病检查"

non\_drug\_id\_list: []

}

# 5.6 5.7 0成药处方/1草药处方

@Edit

## (Build: ok,ok,Test: ok)1.根据分类获取全部的药品目录

POST outpatientDoctor/prescription/allDrugs

REQ {type:0} 选择成药 0 草药 1

RES {

code:200,

data:[{...drug}]

}

<http://www.mocky.io/v2/5d089c6d34000059005d989d>

@Add

## (Build:ok,Test:ok)获取全部处方

POST outpatientDoctor/prescription/allPrescription

REQ {type:1，medical\_record\_id} 选择成药 0 草药 1

{

"code":200,

"data":[

{

"id":1,

"medical\_record\_id":1,

"type":1,

"status":"已作废",

"create\_time":"1998-19-09",

"user\_id":1,

"prescription\_item\_list":[

{

"id":1,

"note":"多吃",

"usage":"口服",

"frequency":"一天一次",

"day\_count":1,

"times":1,

"amount":3,

"prescription\_id":1,

"status":"未取药",

"drug":{

"id":1,

"code":"IXIIX",

"name":"含笑半步癫",

"format":"format",

"unit":"unit",

"manufacturer":"",

"dosage\_form":"",

"type":"中草药",

"price":99999999,

"stock":"2"

}

}

]

}

]

}

@Edit

## (Build:ok, Test:ok)2.创建处方

POST outpatientDoctor/prescription/create

REQ {

type:0, **# 0 成药 1 草药 2 医技补录**

medical\_record\_id : 1000002

**prescription\_item\_list**:[

{drug\_id:9, amount:2 note usage dosage frequency day\_count times }

{drug\_id:3, amount:3}

],

}

检查病历表, 是否已经提交

插入处方表, 状态为 "暂存"

foreach drug\_id\_list:[] 插入处方详情表

RES

{

code:200

}

@Edit

## (Build:ok,ok, Test:ok)5.更新

和创建一样

POST outpatientDoctor/prescription/update

REQ {

id:1,

**prescription\_item\_list**:[

{drug\_id:9, amount:2 note usage dosage frequency day\_count times

],

}

RES {code:200}

@Edit

## (Build:ok, Test:ok)5.提交/发送

POST outpatientDoctor/prescription/submit

REQ {

id:[1]

}

修改处方表, 状态为 "已提交"

添加收费记录表, 状态为 "未缴费"

RES

{

code:200

}

## (Build:ok, Test:ok)5.删除

POST outpatientDoctor/prescription/delete

REQ {

id:[1]}

RES

{

code:200

}

## (Build:ok, Test:ok)3.作废

POST outpatientDoctor/prescription/cancel

REQ {

id:[1]

}

RES {code:200}

---------以下为组套

@Edit

## (Build:ok,ok Test:ok)6.根据类型获得 组套列表

POST outpatientDoctor/prescriptionTemplate/list

REQ {

type **# 0 成药 1 草药 2 医技补录**

}

RES

{

code:200,

data : {

personal:[

{

id :0,

name:"神经病检查"

}

],

department:[],

hospital:[]

}

}

@Edit

## (Build:ok,ok Test:ok)7.存为组套

POST outpatientDoctor/prescriptionTemplate/create

REQ {

type:0,

display\_type:1,

template\_name :"神经病特效药"

**prescription\_item\_list**:[

{drug\_id:9, amount:2 note usage dosage frequency day\_count times

],

}

RES {code:200}

检查name是否重复(前后端)

检查list是否非空(前后端)

插入检查检验处置组套表, 获取id

drug\_id\_list foreach 插入

检查/检验/处置 组套详情

exam\_template\_id = id

返回 type:0 组套列表, 前端更新列表

RES

{

code:200,

data:[

{id: 0,

name : "神经病检查"}

]

}

## (Build:ok, Test:ok)7.更新组套

POST outpatientDoctor/prescriptionTemplate/update

REQ {

id:1,

type:0,

display\_type:1,

template\_name :"特效药"

**prescription\_item\_list**:[

{drug\_id:9, amount:2 note usage dosage frequency day\_count times

],

}

RES {code:200}

检查name是否重复(前后端)

检查list是否非空(前后端)

插入检查检验处置组套表, 获取id

drug\_id\_list foreach 插入

检查/检验/处置 组套详情

exam\_template\_id = id

返回 type:0 组套列表, 前端更新列表

RES

{

code:200

}

## (Build:ok, Test:ok)8.组套详细信息

POST outpatientDoctor/prescriptionTemplate/detail

REQ {

id : 2

}

获取组套id相关信息

RES

{

"code": 200,

"data": {

"template\_name":"wxwxw",

"display\_type":1,

items:

[{

"id": 0,

"drug\_item": {

"id": 1,

"code": "NMSL",

"name": "含笑半步癫",

"format": "一瓶10个",

"unit": "例",

"manufacturer": "NMSLyaochang",

"dosage\_form": "hehe",

"type": "zhongyao",

"price": 10,

"pinyin": "HXBBD",

"stock": 10

}, {

"id": 1,

"drug\_item": {

"id": 2,

"code": "NMSL",

"name": "一日丧命散",

"format": "一瓶10个",

"unit": "例",

"manufacturer": "NMSLyaochang",

"dosage\_form": "hehe",

"type": "zhongyao",

"price": 10,

"pinyin": "HXBBD",

"stock": 10

}

}]

}

}

## (Build:ok, Test:ok)9. 组套删除

POST outpatientDoctor/prescriptionTemplate/delete

REQ {

id :[1,2,3]

}

检查是否存在

删除组套表

删除组套详情表

返回组套列表, 用以更新前端列表

RES

{

code:200,

}

## 5.9 (Build:ok,Test:ok)患者费用查询（统计）

POST /outpatientCharge/historyChargeItems

REQ

{

"medical\_record\_id" : 10000002 //病历号

不要了

"name":"" //二选一

}

RES

{

code : 200,

data: {

[

{

id:1,

medical\_record\_id : 12312,

item\_name : "大抢救",

format : "日",

price : "200",

amount : "2",

fee : "400",

excute\_department : "急救科",

expense\_classification: "处置费",

status: "已缴费",

create\_time :"2009-12-23 08:09:09",

collect\_time : "2009-02-02 17:09:09"

return\_time : "2009-02-02 17:09:09"

},

{

id:2,

medical\_record\_id : 12312,

item\_name : "小抢救",

format : "日",

price : "100",

amount : "2",

fee : "200",

excute\_department : "急救科",

expense\_classification: "处置费",

status: "未缴费",

create\_time :"2009-12-23 08:09:09",

collect\_time : "2009-02-02 17:09:09"

return\_time : "2009-02-02 17:09:09"

}

]

}

}

## 5.15 (Build?,Test?)个人工作量统计

# 6.1 检查 检验登记

## (Build:ok, Test:ok)2. 根据姓名/身份证/医保/病例号 获取挂号列表

姓名要模糊搜索 %like&

POST medicalTechDoctor/examExcute/searchRegistration

REQ {

type:0,1,2,3 对应 姓名/身份证/医保/病例号

input:''"

}

RES {

code :200 ，

data:[

registration的表

]

}

## (Build:ok, Test:ok)2. 登记(做检查)

POST medicalTechDoctor/examExcute/register

REQ {

exam\_item\_id : [ 2 ]

}

exam\_info.id = exam\_item\_id[] 确定状态未登记

then 状态改为"已登记"

else 报错

RES {

code :200

}

## (其他见上文)

## (Build:OK, Test:ok)6.获取全部检查/检验

POST medicalTechDoctor/examExcute/allExam

REQ {

medical\_record\_id:1,

}

RES

{

"code":200,

"data":[

{

"id":1,

"medical\_record\_id":1,

"user\_id":1,

"create\_time":"2019-09-08",

"status":"暂存",

"items":

[

{

"id":1,

"exam\_id":1,

"non\_drug\_item\_id":1,

"status":"已等级",

"non\_drug\_item":{

"id":1,

"code":"UIW82",

"pinyin":"SY",

"format":"XX规格",

"name":"输液",

"fee":"10",

"expense\_classification\_id":1,

"department\_id":1

}

}

]

}

]

}

# 6.2 检查/检验结果录入

## (Build:ok, Test:ok)1.提交查验结果

POST medicalTechDoctor/examExcute/submitResult

REQ {

exam\_item\_id: 12,

result:"检查结果所见",

advice:"诊断意见"

images:[

{name:"hehe.png",url:"http://23.3.3.3./23323.png"}

]

检查详情状态, 是否"已登记"

更新检查检验处置详情"已登记"改为"已完成"

RES

{

code:200

}

## (Build:ok, Test:ok)2.查询查验结果

POST medicalTechDoctor/examExcute/getResult

REQ {

exam\_item\_id: 12,

}

RES

{

"code": 200,

"data": {

"result": "检查结果所见",

"images": [

{

"name": "1560673176199.jpg",

"url": "http://232.23.23.23/IMG\_20190601\_200404.jpg"

},

{

"name": "1560673176239.png",

"url": "http://232.23.23.23/clone\_jiri.png"

}

],

"exam\_item\_id": 2,

"user\_id": 10002,

"advice": "诊断意见",

"id": 6

}

}

# 6.3 个人工作量统计

# 7.1 门诊发药

## (build:ok,test:ok)可发药清单

POST drugDispense/list

POST {

medical\_record\_id :10000002,

}

查找处方表, 查找处方详情

!!! 只有已缴费, 未取药状态才行

RES{

code :200

data: [

{ //prescription表的list

id:

medical\_record\_id :"",

type:"成药",  **# 0 成药 1 草药 2 医技补录**

**status:"", # 暂存, 已提交,**

**create\_time:"",**

**user\_id:,**

**prescription\_item\_list**:[ //**prescription\_item的list**

{

drug\_item:{

**id**

**code**

**name**

**format**

**unit**

**manufacturer**

**dosage\_form**

**type**

**price**

**pinyin**

**stock**

},

**id**

**note #医嘱**

**usage**

**dosage**

**frequency**

**day\_count**

**times**

**amount**

**prescription\_id**

**drug\_id**

**status #未取药, 已取药, 已退药**

},

,,,,,

]

},

.....

]

}

## (build:ok, test:ok)2.发药请求

POST drugDispense/submit

POST {

prescription\_item\_id :[2,3,4]

}

处理

处方详情

RES{

code :200

}

## (build:?, test:?)打印药品清单（先不做）

# 7.2 门诊退药

## (build:ok, test:ok)1.可退药药品清单

POST drugWithdrawal/list

POST {

medical\_record\_id :10000002

}

查找处方表, 查找处方详情

!!! 只有已缴费, 未取药/已取药状态才行

RES{

code :200

data: [

{ //prescription表的list

id:

medical\_record\_id :"",

type:"成药",  **# 0 成药 1 草药 2 医技补录**

**status:"", # 暂存, 已提交,**

**create\_time:"",**

**user\_id:,**

**prescription\_item\_list**:[ //**prescription\_item的list**

{

id:1,

drug\_item:{

//drug表

},

status;"已缴费"

},

,,,,,

]

},

.....

]

}

## (Build:ok test:ok)已退列表

POST drugDispense/withdrawHistory

## (build:ok, test:ok)2.退药请求

POST drugWithdrawal/submit

POST {

prescription\_items:[

{id:1, amount:3}

{id:2, amount:2}

]

}

处理

处方详情

RES{

code :200

}

# 7.3 药品目录管理

## (Build:ok,Test:ok)添加药品

POST /drugInfoManagement/add

REQ {

"id":1,

"code":"AG8383",

"name":"脑残片",

"format":"10mg×28片/盒",

"unit":"盒",

"**manufacturer**":"ikun制药",

"dosage\_form":"片剂",

"type":"中成药",

"price":100,

"pinyin":"NCP",

"stock":100

}

RES {

code:200

}

<http://www.mocky.io/v2/5cff0e4b3200004d0045f2d5>

## (Build:ok,Test:ok)删除药品

POST /drugInfoManagement/delete

REQ {

"data":[1,2,3]

}

RES {

code:200

}

<http://www.mocky.io/v2/5cff0e4b3200004d0045f2d5>

## (Build:ok,Test:ok)药品信息更新

POST /drugInfoManagement/update

REQ {

"id":1,

"code":"AG8383",

"name":"脑残片",

"format":"10mg×28片/盒",

"unit":"盒",

"**manufacturer**":"ikun制药",

"dosage\_form":"片剂",

"type":"中成药",

"price":100,

"pinyin":"NCP",

"stock":100

}

RES {

code:200

}

<http://www.mocky.io/v2/5cff0e4b3200004d0045f2d5>

## (Build:ok,Test:ok)获得全部药品

POST /drugInfoManagement/all

REQ {}

RES {

code:200,

msg:"",

data:[{

"id":1,

"code":"AG8383",

"name":"脑残片",

"format":"10mg×28片/盒",

"unit":"盒",

"**manufacturer**":"ikun制药",

"dosage\_form":"片剂",

"type":"中成药",

"price":100,

"pinyin":"NCP",

"stock":100

}]

}

<http://www.mocky.io/v2/5cff0f0a3200004e0045f2d9>

## (Build:ok,Test:ok)搜索药品（根据名称模糊搜索）用like '%'

POST /drugInfoManagement/getDrugInfoByName

REQ {"input":"脑"}

RES {

code:200,

msg:"",

data:[{

"id":1,

"code":"AG8383",

"name":"脑残片",

"format":"10mg×28片/盒",

"unit":"盒",

"**manufacturer**":"ikun制药",

"dosage\_form":"片剂",

"type":"中成药",

"price":100,

"pinyin":"NCP",

"stock":100

}]

}

# 8.1 费用科目管理

## (Build: ok,Test: ?)1.用于财务管理员维护系统中的费用科目，获得全部列表

POST /expenseClassificationManage/all

RES BODY {

"code":200,

"data":[

{"id":1,"pinyin":"GHF","fee\_name":"挂号费"},

{"id":2,"pinyin":"ZYF","fee\_name":"中药费"},

]

}

## (Build: ok,Test: ?)2.增加

POST /expenseClassificationManage/add

REQ BODY {

"id":1,

"pinyin":"XYF",

"fee\_name":"西药费"

}

RES {

"code":200,

"msg":""

}

## (Build: ok,Test: ?)3.更新

POST /expenseClassificationManage/update

REQ BODY {

"id":1,

"pinyin":"XYF",

"fee\_name":"西药费"

}

RES {

"code":200,

"msg":""

}

## (Build: ok,Test: ?)4.删除

POST /expenseClassificationManage/delete

REQ BODY {

"idArr":[1,2,3]

}

RES {

"code":200,

"msg":""

}

# 8.2 门诊日结核对

## (Build: ok,Test: ?)1.初始化，加载基本数据(收费员名单)

POST /outpatientDailyReportCheck/init

RES {

"code":200,

"msg":"",

data:[

{"id":1,"name":"收费员1"}，

{"id":2,"name":"收费员2"}

]

}

## (Build: ok,Test: ?)2.提交起始日期，收费员id，获取报告

POST /outpatientDailyReportCheck/getReport

REQ BODY {

"start\_date":"2019-06-12",

"end\_date":"2019-06-14",

"toll\_collector\_id":10014

}

RES {

"code":200,

"msg":"",

data:{

获取表中数据 能给什么给什么

(收款人，制表人，应该报账金额，报账发票数量,收费记录列表（收费记录中药，西药，中草药的费用汇总）详细见需求文档，你会写几个就写几个，这部分比较杂，具体数据内容不做硬性规定，给的数据尽可能丰富，写完自己把这部分api补全)

}

"data":{

"toll\_collector\_id": 10014,

"classificationTotal": 155.23,//发票汇总总金额

"registrationTotal": 30,//挂号费总金额

"classifitationFees":[

{

"name": "西药费",

"fee": 5.23

},

{

"name": "处置费",

"fee": 120

}

],

"init\_bill\_record\_id":[

8,

9

],

"invalid\_bill\_record\_id":[

7

],

"reprint\_bill\_record\_id":[

]

}

}

## (Build: ok,Test: ?)3.提交起始日期，检查者和收费员id，完成财务入库

POST /outpatientDailyReportCheck/confirmCheck

REQ BODY{

start\_date:"2019-06-12",

end\_date:"2019-06-14",

toll\_collector\_id:10014

cheker\_id就是\_uid，不从request获取

}

RES {

"code":200,

"msg":""，

}

这个部分就是吧**daily\_collect的checked变成true**

## (Build:ok,Test:?)4.获取日结核对的记录列

POST /outpatientDailyReportCheck/history

REQ BODY{

start\_date:"2019-06-12",

end\_date:"2019-06-14",

}

RES {

"code":200,

"msg":""，

data:[

{

"id": 1,

"medical\_record\_id": 10000001,

"type": "缴费",

"print\_status": 0,

"cost": 10,

"should\_pay": 10,

"truely\_pay": 10,

"retail\_fee": 0,

"user\_id": 10002,

"create\_time": "2019-6-12 18:33"

},]

}

# 8.3门诊科室工作量统计

## (Build: ok,Test: ?)1.提交日期，获得各个科室各个费用项目的统计（两次group by）

POST /outpatientDailyReportCheck/departmentCheck

REQ BODY{

start\_date:"2019-06-12",

end\_date:"2019-06-14",

}

RES {

"code":200,

"msg":"",

"data":{

"columns":[

{"title":"科室名称/费用名称","dataIndex":"col\_1"},

{"title":"看诊人次","dataIndex":"col\_2"},

{"title":"发票数量","dataIndex":"col\_3"} 红色为固定参数

//下面的费用的数量 取决于费用类型管理

{"title":"中药费","dataIndex":"col1"},

{"title":"西药费","dataIndex":"col2"},

{"title":"门诊费","dataIndex":"col3"},

],

"tableData":[

{"col0":"眼科","col\_1":100,"col\_2":100,"col\_3":100,"col1":100,"col2":100,"col3":100,"key":1},

{"col0":"骨科","col\_1":100,"col\_2":100,"col\_3":100,"col1":100,"col2":100,"col3":100,"key":2},

{"col0":"牙科","col\_1":100,"col\_2":100,"col\_3":100,"col1":100,"col2":100,"col3":100,"key":3},

],

chartsData:[

{"name":"眼科",sum:"300"},

{"name":"骨科",sum:"300"},

{"name":"牙科",sum:"300"},

]

}

# 8.4门诊医生工作量统计

## (Build: ok,Test: ?)1.提交日期，获得各个医生各个费用项目的统计（两次group by同上）

POST /outpatientDailyReportCheck/userCheck

REQ BODY{

start\_date:"2019-06-12",

end\_date:"2019-06-14",

}

RES {

"code":200,

"msg":"",

"data":{

"columns":[

{"title":"员工/费用类别","dataIndex":"col\_1"},

{"title":"看诊人次","dataIndex":"col\_2"},

{"title":"发票数量","dataIndex":"col\_3"} 红色为固定参数

{"title":"中药费","dataIndex":"col1"},

{"title":"西药费","dataIndex":"col2"},

{"title":"门诊费","dataIndex":"col3"},

],

"tableData":[

{"col\_1":"路人甲","col\_2:100,"col\_2":100,"col\_3":100,"col1":100,"col2":100,"col3":100,"key":1},

{"col0":"炮兵乙","col\_1":100,"col\_2":100,"col\_3":100,"col1":100,"col2":100,"col3":100,"key":2},

{"col\_1":"土匪丁","col\_1":100,"col\_2":100,"col\_3":100,"col1":100,"col2":100,"col3":100,"key":3},

],

chartsData:[

{"name":"路人甲",sum:"300"},

{"name":"炮兵乙",sum:"300"},

{"name":"土匪丁",sum:"300"},

]

}

# WebSocket

所有的通知推送

挂号

有一个人挂号, 医生待诊患者列表

开药/开检查

医生开药, 患者付款后, 对应科室待检查/待取药列表增加一个

退药/退检查

药房退药后, 收费处待退款增加一个

日结核对

收费日结后, 门诊日结核对, 多一条信息