

Sheets 文件夹置于工程主目录下

setting 中 INSTALLED\_APPS 增加一项: 'Sheets'

工作目录下终端输入命令

```
python manage.py migrate
```

```
python manage.py makemigrations Sheets
```

```
python manage.py migrate Sheets
```

views 中调用数据库操作接口: from Sheets import db\_intfs

用户注册/登陆/注销:

```
def create_superuser(username, password, email):
    "创建超级用户, 所有参数均为字符串类型"
def create_new_doctor(username, password, name, sex, tel, email, department):
    "创建医生用户, 所有参数均为字符串类型, sex 请传 'f' 或 'm'"
def create_new_patient(username, password, name, sex, tel, email):
    "创建患者用户, 所有参数均为字符串类型, sex 请传 'f' 或 'm'"
def verify_and_login(username, password, login_request):
    "检验用户信息是否合法: 若合法则调用 auth.login 登陆且返回 True, 否则返回 False, 最后一个参数为 HttpRequest 类型"
def logout(logout_request):
    "注销用户登陆, 最后一个参数为 HttpRequest"
```

新增表项:

```
def depa_insert_value(name, desc='暂无描述'):
    "新建科室项, 参数均为字符串格式, desc 为对科室的描述"
def medi_insert_value(doctor_id, patient_id, submit_time, appo_time,
    symptom='患者暂未描述症状', message='无消息'):
    "患者进行预约, 前两个参数为 doctor 和 patient 的主键, 两个 time 参数为字符串, 格式为 'YYY-MM-DD HH:MM:SS', submit_time 为预约的提交时间, appo_time 为想预约的时间"
def doctor_insert_value(name, sex, tel, email, department, user_key):
    "新建医生项, 最后一个参数传 user_id, 其它参数均为字符串格式, sex 一定要传 'm' 或者 'f'"
def patient_insert_value(name, sex, tel, email, user_key):
    "新建患者项, 最后一个参数传 user_id, 参数均为字符串格式"
def drug_insert_value(name, price=100.00, desc='暂无描述'):
    "新建药品项, price 为 xxx.xx 的浮点数, 其它均为字符串"
def init_database():
    "初始化数据库, 服务器运行后只能执行一次该函数"
    "三人一人一个超级用户, 插入一些科室和药品表项"
```

数据库查询:

```
def doctor_of_depa(depa_name):
```

"通过科室名称获取相关科室的所有医生，返回字典列表：[{第一个医生信息}，{第二个医生信息}，...]"

医生信息的字典键值：

'user\_id' : 医生用户 id  
'doctor\_id' : 医生编号 id  
'doctor\_name' : 医生姓名  
'doctor\_sex' : 医生性别  
'doctor\_tel' : 医生电话  
'user\_department' : 医生所属科室

```
def get_user_info(user_id):
```

"通过 user\_id 获取用户信息，返回 user\_info 字典，若不存在此用户则返回空字典"

若为医生：

user\_info['group'] = 'doctor'  
user\_info['user\_id']  
user\_info['doctor\_id']  
user\_info['doctor\_name']  
user\_info['doctor\_sex']  
user\_info['doctor\_tel']  
user\_info['doctor\_email']  
user\_info['doctor\_department']

若为患者：

user\_info['group'] = 'patient'  
user\_info['user\_id']  
user\_info['patient\_id']  
user\_info['patient\_name']  
user\_info['patient\_sex']  
user\_info['patient\_tel']  
user\_info['patient\_email']

```
def look_up_medi_record(record_id):
```

"通过病历 ID 调取整条病历记录，返回字典 record"

病历状态说明：医生通过设置病历表中 confirm, cancel, complete 三个布尔字段来改变病历状态

...

confirm	cancel	complete	status
0	0	0	预约中 (newappo)
0	1	0	已拒绝 (refused)
1	0	0	待诊断 (waiting)
1	0	1	已诊断 (handled)
1	1	0	已作废 (calloff)
其它			有误 (error)

...

record['record\_id']

record['appo\_confirm']: 医生是否确认了预约，布尔型

record['appo\_complete']: 医生是否完成了预约，布尔型

record['appo\_cancel']: 医生是否取消了预约，布尔型

```

record['doctor_id']
record['doctor_name']
record['doctor_sex']
record['doctor_tel']
record['doctor_email']
record['doctor_department']
record['patient_id']
record['patient_name']
record['patient_sex']
record['patient_tel']
record['patient_email']
record['submit_time'] : 预约提交时间, 格式'YYYY-MM-DD HH:MM:SS'
record['appo_time'] : 预约时间点, 格式'YYYY-MM-DD HH:MM:SS'
record['symptom']: 患给出的症状描述
record['appo_message']: 患者预约时可能提交的附带信息
record['cancel_message']: 医生取消/拒绝预约时的携带信息
record['disease'] : 医生给出的患病名称
record['detail'] : 医生给出的患病细节
record['diag_message']: 医嘱
record['prescription'] : 医生给的处方, 也是字典列表
    prescription: 列表类型[
        字典 1{
            'drug': 药品名称
            'usage': 用法
            'dosage': 用量
        },
        字典 2{
            'drug': ''
            'usage': ''
            'dosage': ''
        },
        ...
    ]
def doc_ask_newappo(doctor_id):
    "通过医生 ID 获取新的预约记录, 返回字典列表 tmp: [{患者 1 的预约情况},{患者 2 的},{患者 3 的},...]"
    tmp[i]['record_id']: 病历 id
    tmp[i]['patient_id']
    tmp[i]['patient_name']
    tmp[i]['patient_sex']
    tmp[i]['patient_tel']
    tmp[i]['patient_email']
    tmp[i]['submit_time'] : 预约提交时间, 格式'YYYY-MM-DD HH:MM:SS'
    tmp[i]['appo_time'] : 预约时间点, 格式'YYYY-MM-DD HH:MM:SS'

```

```

tmp[i]['symptom'] : 患者症状描述
tmp[i]['appo_message'] : 患者预约时可能提交的附带信息
def doc_ask_waiting(doctor_id):
    "通过医生 ID 获取待诊断的预约记录, 返回字典列表: [{患者 1 的预约情况},{患者 2 的},{患者 3 的},...]"
    tmp[i]['record_id']
    tmp[i]['patient_id']
    tmp[i]['patient_name']
    tmp[i]['patient_sex']
    tmp[i]['patient_tel']
    tmp[i]['patient_email']
    tmp[i]['submit_time']
    tmp[i]['appo_time']
    tmp[i]['symptom']
    tmp[i]['appo_message']
def pat_ask_history(patient_id):
    "通过患者 ID 获取历史病历, 返回字典 tmp: 包含三个键'refused','handled','calloff', 分别对应被拒绝的预约记录, 被处理的预约记录, 被取消的预约记录。键值为字典列表:
    {
    'refused': [{患者第一条被拒记录},{第二条},...],
    'handled': [{患者已完成的第一条看病记录},{第二}, ...],
    'calloff': [{被医生取消的第一条预约记录},{第二条},...]}"
    }
    举例: 若查询 1 号患者的历史病历, 返回 tmp 字典, 查看其第一条已经处理完成的病历记录
    tmp['handled'][0]['record_id']: 病历编号
    tmp['handled'][0]['doctor_id']
    tmp['handled'][0]['doctor_name']
    tmp['handled'][0]['doctor_sex']
    tmp['handled'][0]['doctor_tel']
    tmp['handled'][0]['doctor_email']
    tmp['handled'][0]['doctor_department']
    tmp['handled'][0]['submit_time']
    tmp['handled'][0]['appo_time']
    tmp['handled'][0]['symptom']
    tmp['handled'][0]['appo_message']
    tmp['handled'][0]['disease']: 医生给出的患病名称
    tmp['handled'][0]['detail']: 医生给出的患病细节
    tmp['handled'][0]['diag_message']: 医嘱
    tmp['handled'][0]['prescription']: 医生给的处方, 也是字典列表
        prescription: 列表类型[
            字典 1{
                'drug': 药品名称
                'usage': 用法
                'dosage': 用量

```

```

    },
    字典 2{
        'drug': ''
        'usage': ''
        'dosage': ''
    },
    ...
]

```

更新数据库:

```

def doctor_confirm(record_id):
    "医生确认预约"
def doctor_diagnose(record_id, diag_message, prescription=[{"drug":'*',"usage":'*',
"dosage":'*'}], disease='未定义', detail='略'):
    "医生记录病历信息，完成诊断，prescription 请传字典列表，若无就别传这个参数"
def doctor_cancel(record_id, cancel_message='更多信息请联系您预约的医生'):
    "医生取消或者拒绝一次预约，第一个参数为整形，第二个参数为字符串类型"

```

删除表项:

```

def delete_doctor(doctor_id):
    "通过医生 id 删除医生"
def delete_patient(patient_id):
    "通过患者 id 删除患者"
def delete_department(name):
    "通过部门字符串名称删除部门"
def delete_medi_record(record_id):
    "通过 id 删除一条病历记录"
def delete_drug(name):
    "通过药品名删除药品"

```