

一、上传医院接口

参考《尚医通API接口文档.doc》业务接口4.1上传医院

医院编号是平台分配的，全局唯一，上传医院接口可以多次调用，如果存在相同编号的为更新操作

1、数据分析

```
1 {
2   "hoscode": "1000_0",
3   "hosname": "北京协和医院",
4   "hostype": "1",
5   "provinceCode": "110000",
6   "cityCode": "110100",
7   "districtCode": "110102",
8   "address": "大望路",
9   "intro": "北京协和医院是集医疗、教学、科研于一体的大型三级甲等综合医院，是国家卫生计生委
10  "route": "东院区乘车路线：106、...更多乘车路线详见须知。",
11  "logoData": "iVBORw0KGgoAAAA...NSUhEUg==",
12  "bookingRule": {
13    "cycle": "1",
14    "releaseTime": "08:30",
15    "stopTime": "11:30",
16    "quitDay": "-1",
17    "quitTime": "15:30",
18    "rule": [
19      "西院区预约号取号地点：西院区门诊楼一层大厅挂号窗口取号",
20      "东院区预约号取号地点：东院区老门诊楼一层大厅挂号窗口或新门诊楼各楼层挂号/收费窗口取
21    ]
22  }
23 }
```

说明：

1，数据分为医院基本信息与预约规则信息

2, 医院logo转换为base64字符串

3, 预约规则信息属于医院基本信息的一个属性

4, 预约规则rule, 以数组形式传递

5, 数据传递过来我们还要验证签名, 只允许平台开通的医院可以上传数据, 保证数据安全性

2、添加service接口

在HospService添加

```
1 /**
2  * 上传医院信息
3  * @param paramMap
4  */
5 void save(Map<String, Object> paramMap);
```

在HospitalServiceImpl类添加实现

```
1 @Override
2 public void save(Map<String, Object> paramMap) {
3     Hospital hospital = JSONObject.parseObject(JSONObject.toJSONString(paramMap), Hospi
4     //判断是否存在
5     Hospital targetHospital = hospitalRepository.getHospitalByHoscode(hospital.getHos
6     if(null != targetHospital) {
7         hospital.setStatus(targetHospital.getStatus());
8         hospital.setCreateTime(targetHospital.getCreateTime());
9         hospital.setUpdateTime(new Date());
10        hospital.setIsDeleted(0);
11        hospitalRepository.save(hospital);
12    } else {
13        //0: 未上线 1: 已上线
14        hospital.setStatus(0);
15        hospital.setCreateTime(new Date());
16        hospital.setUpdateTime(new Date());
17        hospital.setIsDeleted(0);
```

```
18         hospitalRepository.save(hospital);
19     }
20 }
```

3、添加repository接口

在HospitalRepository类添加接口

```
1 Hospital getHospitalByHoscode(String hoscode);
```

4、添加controller接口

```
1 @ApiOperation(value = "上传医院")
2 @PostMapping("saveHospital")
3 public Result saveHospital(HttpServletRequest request) {
4     Map<String, Object> paramMap = HttpRequestHelper.switchMap(request.getParameterMa
5
6     hospitalService.save(paramMap);
7     return Result.ok();
8 }
```

5、添加帮助类

```
1 public class HttpRequestHelper {
2     public static Map<String, Object> switchMap(Map<String, String[]> paramMap) {
3         Map<String, Object> resultMap = new HashMap<>();
4         for (Map.Entry<String, String[]> param : paramMap.entrySet()) {
5             resultMap.put(param.getKey(), param.getValue()[0]);
6         }
7         return resultMap;
8     }
9 }
```

6、参数签名

添加工具类MD5

```
1 public final class MD5 {
2
3     public static String encrypt(String strSrc) {
4         try {
5             char hexChars[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8',
6                                 '9', 'a', 'b', 'c', 'd', 'e', 'f' };
7             byte[] bytes = strSrc.getBytes();
8             MessageDigest md = MessageDigest.getInstance("MD5");
9             md.update(bytes);
10            bytes = md.digest();
11            int j = bytes.length;
12            char[] chars = new char[j * 2];
13            int k = 0;
14            for (int i = 0; i < bytes.length; i++) {
15                byte b = bytes[i];
16                chars[k++] = hexChars[b >>> 4 & 0xf];
17                chars[k++] = hexChars[b & 0xf];
18            }
19            return new String(chars);
20        } catch (NoSuchAlgorithmException e) {
21            e.printStackTrace();
22            throw new RuntimeException("MD5加密出错!!" + e);
23        }
24    }
25
26
27 }
```

7、在HospitalSetService类添加方法

```
1 /**
2  * 获取签名key
3  * @param hoscode
4  * @return
```

```

5      */
6      String getSignKey(String hoscode);
7
8      @Override
9      public String getSignKey(String hoscode) {
10         HospitalSet hospitalSet = this.getByHoscode(hoscode);
11         if(null == hospitalSet) {
12             throw new YyghException(20001, "失败");
13         }
14         return hospitalSet.getSignKey();
15     }
16
17     /**
18      * 根据hoscode获取医院设置
19      * @param hoscode
20      * @return
21      */
22     private HospitalSet getByHoscode(String hoscode) {
23         return baseMapper.selectOne(new QueryWrapper<HospitalSet>().eq("hoscode", hos
24     }

```

8、修改ApiController代码

```

1 package com.atguigu.yygh.hosp.controller.api;
2
3 import com.atguigu.common.exception.YyghException;
4 import com.atguigu.common.utils.R;
5 import com.atguigu.common.utils.Result;
6 import com.atguigu.yygh.hosp.service.HospitalService;
7 import com.atguigu.yygh.hosp.service.HospitalSetService;
8 import com.atguigu.yygh.hosp.utils.HttpRequestHelper;
9 import com.atguigu.yygh.hosp.utils.MD5;
10 import com.atguigu.yygh.model.hosp.Hospital;
11 import io.swagger.annotations.Api;
12 import io.swagger.annotations.ApiOperation;
13 import org.springframework.beans.factory.annotation.Autowired;
14 import org.springframework.util.StringUtils;
15 import org.springframework.web.bind.annotation.PostMapping;
16 import org.springframework.web.bind.annotation.RequestMapping;
17 import org.springframework.web.bind.annotation.RestController;

```

```
18
19 import javax.servlet.http.HttpServletRequest;
20 import java.util.Map;
21
22 @Api(tags = "医院管理API接口")
23 @RestController
24 @RequestMapping("/api/hosp")
25 public class ApiController {
26
27     @Autowired
28     private HospitalService hospitalService;
29
30     @Autowired
31     private HospitalSetService hospitalSetService;
32
33     @ApiOperation(value = "上传医院")
34     @PostMapping("saveHospital")
35     public Result saveHospital(HttpServletRequest request) {
36         Map<String, Object> paramMap = HttpRequestHelper.switchMap(request.getParameterMap());
37
38         //必须参数校验 略
39         String hoscode = (String)paramMap.get("hoscode");
40         if(StringUtils.isEmpty(hoscode)) {
41             throw new YyghException(20001, "失败");
42         }
43         //签名校验
44         //1 获取医院系统传递过来的签名,签名进行MD5加密
45         String hospSign = (String)paramMap.get("sign");
46         //2 根据传递过来医院编码, 查询数据库, 查询签名
47         String signKey = hospitalSetService.getSignKey(hoscode);
48         //3 把数据库查询签名进行MD5加密
49         String signKeyMd5 = MD5.encrypt(signKey);
50         //4 判断签名是否一致
51         if(!hospSign.equals(signKeyMd5)) {
52             throw new YyghException(20001, "校验失败");
53         }
54
55         //传输过程中“+”转换为了“ ”, 因此我们要转换回来
56         String logoData = (String)paramMap.get("logoData");
57         logoData = logoData.replaceAll(" ", "+");
58         paramMap.put("logoData", logoData);
59
60         hospitalService.save(paramMap);
61     }
62 }
```

```
61         return Result.ok();
62     }
63 }
```