

陕西省预约挂号及支付平台号源

接入接口规范

v4.0.4

微医集团

2016-06

修改记录

日期	版本	描述	修改人
2016-06	4.0	根据之前版本升级，优化部分接口和字段	张景
2017-07-17	4.0.1	查询充值记录、查询挂号记录、查询缴费记录接口增加返回字段	张景
2017-08-07	4.0.2	查询缴费记录接口返回增加科室 id 和医生 id	张景
2017-08-08	4.0.3	新增对账单获取接口，对账单下载接口	张景
2017-09-13	4.0.4	查询医生接口返回新增排序字段	张景

目录

1.	文档说明.....	- 5 -
1.1.	接口说明.....	- 5 -
1.2.	交互方式.....	- 5 -
1.3.	接口规范.....	- 5 -
1.4.	基本参数.....	- 5 -
2.	接口定义.....	- 8 -
2.1.	医院提供给平台调用（SPI 部分）.....	- 8 -
2.1.1.	公共部分.....	- 8 -
2.1.2.	交易部分.....	错误!未定义书签。
2.1.3.	用户部分.....	- 8 -
2.1.4.	诊疗卡支付部分.....	错误!未定义书签。
2.1.5.	门诊部分.....	- 14 -
2.1.6.	住院部分.....	错误!未定义书签。
2.1.7.	商保部分.....	错误!未定义书签。
2.2.	平台提供给医院调用（API 部分）.....	- 31 -
2.2.1.	公共部分.....	错误!未定义书签。
2.2.2.	交易部分.....	错误!未定义书签。
2.2.3.	用户部分.....	错误!未定义书签。
2.2.4.	门诊部分.....	- 32 -
2.2.5.	住院部分.....	错误!未定义书签。
2.3.	接口统一返回码说明.....	- 34 -
3.	数据字典定义.....	- 36 -
4.	加签和加密校验.....	- 42 -
4.1.	签名机制.....	- 42 -
4.1.1.	需要参与签名的参数.....	- 42 -
4.1.2.	生成待签名字符串.....	- 42 -
4.2.	加解密机制.....	- 43 -
4.2.1.	加密顺序.....	- 43 -
4.2.2.	加密方式.....	- 43 -
4.2.3.	带加密的字符串.....	- 43 -
4.2.4.	待解密的字符串.....	- 45 -
4.2.5.	加密/解密代码示例.....	- 46 -
5.	接口调用流程.....	- 49 -
6.	附录.....	- 51 -

1. 文档说明

1.1. 接口说明

本接口文档是全流程医疗平台与医院(或第三方平台)之间数据接入及同步交互的开发文档。

1.2. 交互方式

◆ 调用方式:

WebServices: http://ip:port/ApiService.asmx

HTTP:

GET 示例: http://ip:port/wd/api/{service}

http://ip:port/wd/spi/testnet?partner=TEST_WX&logNo=Spi00001&version=V1.0.0&hospitalCode=43010001&inputCharset=utf-8&signType=md5&sign=2A3D153984E85ABBA3B3E0E3E1622AAB&requestEncrypted=WxzV3/6APXYdqsoZIVuANTSxfNRbE6H1PB17Bw4MTA4=

GET 方式调用参数值请先进行 UriEncode。

POST 示例:

url: <http://ip:port/wd/api>

◆ 数据: post 提交数据格式: 支持 xml、json

1.3. 接口规范

- 1、解析请求参数: 先验签后解密; 组装返回参数: 先加密后加签;
- 2、接口 xml/json 结构严格区分大小写, 请按接口规范进行 xml/json 请求和返回的组装;
- 3、接口 xml 结构字段内容都需要加上 CDATA 转义, 如<service><![CDATA[NetTest]]></service>;
- 4、接口请求和返回的时间使用 24 小时制;
- 5、接口 xml 必须包含 XmlDeclaration xml 声明节点, 例: <?xml version="1.0" encoding="utf-8"?>, encoding 属性和 xml 参数 inputCharset 定义的字符集编码一致;
- 6、所有返回的文本字段, 内容如果需要换行请加\r\n;

1.4. 基本参数

接口请求参数由两部分组成: 基本参数和业务参数。

基本请求参数 xml 串以<root>为根节点:

节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
service	root	String(32)	√	接口名称, 平台分配, 唯一

partner	root	String(32)	√	接入方接口用户，平台分配，唯一
hospitalCode	root	String(64)	√	接入医疗机构编码，平台分配，唯一
dataFormat	root	String(16)	√	数据格式：xml、json
inputCharset	root	String(16)	√	参数编码字符集，目前仅支持 utf-8 编码类型
signType	root	String(16)	√	签名方式，目前仅支持 md5(小写)签名（不参与加签）
sign	root	String(64)	√	签名串，详见 “签名机制” 除去 sign、signType 两个参数外，其他需要使用到的参数皆是要签名的参数。（不参与加签）
业务请求参数				
requestEncrypted	root	String	√	待解密的请求业务参数报文，报文中可包含多条业务数据，详见各个接口 requestEncrypted 报文参数说明

➤ 正常请求示例：

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <service><![CDATA[NetTest]]></service>
  <partner><![CDATA[TEST_WX]]></partner>
  <hospitalCode><![CDATA[0]]></hospitalCode>
  <dataFormat><![CDATA[xml]]></dataFormat>
  <inputCharset><![CDATA[utf-8]]></inputCharset>
  <signType><![CDATA[md5]]></signType>
  <sign><![CDATA[389F6997DA8530F147D4AD7AF41C5C30]]></sign>
  <requestEncrypted><![CDATA[oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAXm7Ia
os/Yw9ZCsw3CtC0Q6j6X9C+CQ==]]></requestEncrypted>
</root>
或
{
  "service": "NetTest",
  "partner": "TEST_WX",
  "hospitalCode": "0",
  "dataFormat": "json",
  "inputCharset": "utf-8",
  "signType": "md5",
  "sign": "722289C58286EC2B41EF764CD5B0259D",
  "requestEncrypted": "WXzV3/6APXYdqsoZIVuANePKKHbc47QODNPniynaD9s="
}
```

接口返回参数由两部分组成：基本参数和业务参数。

整个返回参数 xml 串以<root>为根节点：

节点名称	父节点名称	数据类型	必填	说明
基本返回参数				
resultCode	root	String(32)	√	返回结果，详见 “返回码说明” 和各接口业务返回码
resultMessage	root	String(128)	√	返回结果描述
signType	root	String(16)	√	签名方式，目前仅支持 md5(小写)签名（不参与加签）

sign	root	String(64)	√	签名串，详见 “签名机制” 除去 sign、signType 参数外，其他需要使用到的参数皆是要签名的参数。（不参与加签）
业务返回参数				
responseEncrypted	root	String		带加密的返回业务参数报文，报文中可包含多条业务数据，详见各个接口 responseEncrypted 报文参数说明

➤ 正常返回

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <resultCode><![CDATA[SUCCESS]]></resultCode>
  <resultMessage><![CDATA[成功]]></resultMessage>
  <sign><![CDATA[5A79CE775EC879E11B88EF71D69F131]]></sign>
  <signType><![CDATA[md5]]></signType>
  <responseEncrypted><![CDATA[dbvhhbM8Ckn+xUMWC8mmVO3bMEcsql4P4sUDAKjE6ZEQeQwz7o0vN/+xdsjsmNLo6PISHwoS9d+6LzWgJYeRMhHFs99uN84ajwMMqEeSLDvU=]]></responseEncrypted>
</root>
```

或

```
{
  "resultCode": "SUCCESS",
  "resultMessage": "成功",
  "signType": "md5",
  "sign": "4376E2DF2191EFDCA849153DFF4A11DC",
  "responseEncrypted": "vTmXQJ1INyhGdVsRQWHn0+INhPQt1/4YXp72AfY0mtR6O+GUTd5hTDtp0RNYAw+K"
}
```

➤ 发生错误时返回

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <resultCode><![CDATA[SYSTEM_ERROR]]></resultCode>
  <resultMessage><![CDATA[系统错误]]></resultMessage>
  <sign><![CDATA[0335185EA82D8257F7645307B7CF008E]]></sign>
  <signType><![CDATA[md5]]></signType>
</root>
```

或

```
{
  "resultCode": "SYSTEM_ERROR",
  "resultMessage": "系统错误",
  "signType": "md5",
  "sign": "0335185EA82D8257F7645307B7CF008E"
}
```

2. 接口定义

2.1. 医院提供给平台调用（SPI 部分）

2.1.1. 公共部分

2.1.1.1. 网络通讯测试（TestNet）

接口名称	TestNet			
接口说明	平台调用该测试医院接口服务网络是否通畅，服务是否处于工作状态、数据库是否处于连接状态。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
ip	request	String(16)	√	请求 IP
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
systemTime	response	String(32)	√	返回系统时间，格式：YYYY-MM-DD HH24:MI:SS

➤ 正常请求业务参数：

```
<request>
  <ip><![CDATA[127.0.0.1]]></ip>
</request>
```

➤ 正常返回业务参数：

```
<response>
  <systemTime>2014-10-01 20:22:10</systemTime>
</response>
```

2.1.2. 用户部分

2.1.2.1. 用户注册（CreatePatient）

接口名称	CreatePatient
------	----------------------

接口说明	平台调用该接口向医院系统发起用户注册。 如果医院已注册该用户，那医院系统返回已注册的用户 ID； 如果医院没有注册该用户，那医院系统需要注册用户后返回注册的用户 ID，并返回相应的用户信息。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
patientType	request	Int	√	用户类型：0-无证件（小孩） 1-有证件
certifcateType	request	Int	√	证件类型，详见“证件类型” patientType=0 时为监护人证件
certifcateNo	request	String(32)	√	证件号码 patientType=0 时为监护人证件
issueDate	request	String(32)		用户证件发证日期，格式 YYYY-MM-DD
validDate	request	String(32)		用户证件有效日期，格式 YYYY-MM-DD
name	request	String(32)	√	用户姓名
sex	request	Int	√	用户性别，详见“性别”
birthday	request	String(32)	√	用户出生日期，格式：YYYY-MM-DD
address	request	String(128)		用户地址
mobile	request	String(16)	√	用户手机号码
country	request	String(16)		国家和地区，默认中国，详见“国家和地区”
password	request	String(32)		用户院内诊疗卡密码
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
createTime	response	String(32)		用户注册时间，格式：YYYY-MM-DD HH24:MI:SS
patientId	response	String(32)	√	院内用户 ID，医院系统中用户唯一 ID
medicalNo	response	String(32)		病历号/门诊号/就诊号

➤ 正常请求业务参数：

```

<request>
  <patientType><![CDATA[1]]></patientType>
  <certifcateType><![CDATA[1]]></certifcateType>
  <certifcateNo><![CDATA[420102030400404]]></certifcateNo>
  <issueDate><![CDATA[]]></issueDate>
  <validDate><![CDATA[]]></validDate>
  <name><![CDATA[测试]]></name>
  <sex><![CDATA[1]]></sex>
  <birthday><![CDATA[2010-01-01]]></birthday>

```

```

<address><![CDATA[]]></address>
<mobile><![CDATA[1234322123]]></mobile>
<password><![CDATA[123456]]></password>
</request>

```

➤ 正常返回业务参数:

```

<response>
  <createTime>2017-01-19 16:17:28</createTime>
  <patientId>123456</patientId>
  <medicalNo></medicalNo>
</response>

```

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
S03000101	用户注册失败

2.1.2.2. 用户信息查询 (GetPatientInfo)

接口名称	GetPatientInfo			
接口说明	根据查询规则（院内用户 id，证件、卡等）查询在院内的用户信息。 强制用卡医院返回的卡以使用时间倒序，取最近的不超过 10 张卡，可以包括院内卡、医保卡等。 如果存在一个证件对应多个用户档案的情况，医院根据实际请求筛选一个档案返回给平台。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
user	request	list 节点	√	用户规则集合，根据不同医院需求传入
id	user	String(16)	√	用户规则，详见 “用户规则”
type	user	String(16)		用户规则对应的类型 例：id=card,type=1，根据健康卡操作
value	user	String(32)	√	用户规则对应的值
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
createTime	response	String(32)	√	用户注册时间，格式：YYYY-MM-DD HH24:MI:SS
patientId	response	String(32)	√	院内用户 ID，医院系统中用户唯一 ID
patientType	response	Int	√	用户类型：0-无证件（小孩） 1-有证件
patientFlag	response	Int	√	用户类别：0-成人 1-儿童

certificateType	response	Int	√	证件类型，详见“证件类型” patientType=0 时为监护人证件
certificateNo	response	String(32)	√	证件号码 patientType=0 时为监护人证件
name	response	String(16)	√	用户姓名
sex	response	Int	√	用户性别，详见“性别”
birthday	response	String(16)	√	用户出生日期，格式：YYYY-MM-DD
age	response	Decimal(10,1)	√	年龄
address	response	String(128)		用户地址
mobile	response	String(16)		用户手机号码
patient	response	list 节点		门诊号信息
medicalNo	patient	String(32)	√	病历号/门诊号/就诊号
status	patient	Int	√	状态：0-正常 1-已挂失 2-已注销
card	patient	list 节点		用户卡信息
cardType	card	Int	√	用户卡类型，详见“卡类型”
cardNo	card	String(32)	√	用户卡号
cardStatus	card	Int	√	卡状态：0-正常 1-已挂失 2-已注销
balance	card	Decimal(10,2)	√	余额，单位：分（针对院内卡返回）
cardTime	card	String(32)	√	发卡/绑卡时间，格式：YYYY-MM-DD HH24:MI:SS
lastTime	card	String(32)		最近一次用卡时间，格式：YYYY-MM-DD HH24:MI:SS

➤ 正常请求业务参数：

```
<request>
  <user>
    <type><![CDATA[1]]></type>
    <id><![CDATA[certificate]]></id>
    <value><![CDATA[210102199001012010]]></value>
  </user>
  <user>
    <type><![CDATA[]]></type>
    <id><![CDATA[name]]></id>
    <value><![CDATA[测试]]></value>
  </user>
</request>
```

➤ 正常返回业务参数：

```
<response>
  <createTime>2017-01-19 16:17:28</createTime>
  <patientId>123456</patientId>
  <patientType>1</patientType>
  <certificateType>1</certificateType>
  <certificateNo>431022199205200418</certificateNo>
```

```

<name>吕俊</name>
<sex>1</sex>
<birthday>1992-05-29</birthday>
<address>湖南省郴州市</address>
<mobile>13107499464</mobile>
<patient>
  <medicalNo>hzh01</medicalNo>
  <status>0</status>
  <card>
    <cardType>1</cardType>
    <cardNo>742871174</cardNo>
    <cardStatue>0</cardStatue>
    <balance>100000.00</balance>
    <cardTime>2017-01-10 16:21:58</cardTime>
    <lastTime>2017-01-19 16:22:16</lastTime>
  </card>
</patient>
</response>

```

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
S03000201	未查询到用户信息
S03000202	证件信息和手机号不匹配

2.1.2.3. 卡验证 (CheckCard)

接口名称	CheckCard			
接口说明	平台调用该接口向医院系统发起用户卡信息验证。 门诊号和卡信息必填一项。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
patientType	request	Int	√	用户类型: 0-无证件 (小孩) 1-有证件
certficateType	request	Int	√	证件类型, 详见 “证件类型” patientType=0 时为监护人证件
certificateNo	request	String(32)	√	证件号码 patientType=0 时为监护人证件
cardType	request	Int		用户卡类型, 详见 “卡类型”
cardNo	request	String(32)		用户卡号
medicalNo	patient	String(32)		病历号/门诊号/就诊号, 门诊唯一

name	request	String(16)	√	用户姓名
sex	request	Int	√	用户性别，详见“性别”
birthday	request	String(16)	√	用户出生日期，格式：YYYY-MM-DD
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
medicalNo	response	String(32)		病历号/门诊号/就诊号
cardStatus	response	Int	√	卡状态：0-正常 1-已挂失 2-已注销
cardTime	response	String(32)	√	发卡时间，格式：YYYY-MM-DD HH24:MI:SS
lastTime	response	String(32)		最近一次用卡时间，格式：YYYY-MM-DD HH24:MI:SS

➤ 正常请求业务参数：

```
<request>
  <patientType><![CDATA[1]]></patientType>
  <certificateType><![CDATA[1]]></certificateType>
  <certificateNo><![CDATA[1]]></certificateNo>
  <cardType><![CDATA[1]]></cardType>
  <cardNo><![CDATA[1]]></cardNo>
  <medicalNo><![CDATA[1]]></medicalNo>
  <name><![CDATA[1]]></name>
  <sex><![CDATA[1]]></sex>
  <birthday><![CDATA[2010-01-01]]></birthday>
</request>
```

➤ 正常返回业务参数：

```
<response>
  <medicalNo>hzh01</medicalNo>
  <cardStatue>0</cardStatue>
  <cardTime>2017-01-10 16:21:58</cardTime>
  <lastTime>2017-01-19 16:22:16</lastTime>
</response>
```

➤ 业务返回码：

编码(resultCode)	说明(resultMessage)
S03000301	卡验证失败

2.1.3. 门诊部分

2.1.3.1. 挂号部分

2.1.3.1.1. 科室信息查询（GetDepartmentInfo）

接口名称	GetDepartmentInfo			
接口说明	平台调用该接口，获取有排班的科室信息。 当科室 ID 为空时，查询所有科室列表。 建议医院返回两层或三层科室结构。 医院医生出诊科室，非行政科室。 获取频率：原则上一天一次。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
departmentId	request	String(32)		科室 ID，医院系统中科室唯一 ID，为空时查询所有科室信息，不为空查本科室信息
currentPage	request	Int	√	当前页数（从 1 开始）
pageSize	request	Int	√	每页查询数量
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
count	response	Int	√	科室总数量，用于分页
department	response	list 节点	√	科室信息集合
id	department	String(32)	√	科室 ID，医院系统中科室唯一 ID
name	department	String(128)	√	科室名称
parentId	department	String(32)	√	上级科室 ID，顶级时为-1
introduction	department	String(2560)		介绍
expertise	department	String(256)	√	科室主治，例：颈椎病、腰腿痛等
level	department	Int	√	科室等级：1-一级 2-二级 3-三级
location	department	String(256)		科室所在位置
status	department	Int	√	状态：0-无效 1-有效
linkMan	department	String(16)		科室联系人
telephone	department	String(16)		科室联系电话
rule	department	String(258)		科室预约规则说明

➤ 正常请求业务参数：

```
<request>
  <departmentId><![CDATA[-1]]></departmentId>
  <currentPage><![CDATA[1]]></currentPage>
  <pageSize><![CDATA[10]]></pageSize>
</request>
```

➤ 正常返回业务参数：

```
<response>
  <count>2</count>
  <department>
    <id>001</id>
    <name>精神病科</name>
    <parentId>-1</parentId>
    <introduction></introduction>
    <expertise>精神病</expertise>
    <level>1</level>
    <location> </location>
    <status>1</status>
    <linkMan></linkMan>
    <telephone></telephone>
    <rule></rule>
  </department>
  <department>
    .....
  </department>
</response>
```

➤ 业务返回码：

编码(resultCode)	说明(resultMessage)
S05010101	未查询到科室记录

2.1.3.1.2. 医生信息查询（GetDoctorInfo）

接口名称	GetDoctorInfo			
接口说明	平台调用该接口获取科室下有排班的医生信息。 当科室 ID 为空时查询所有科室下的医生列表； 当科室 ID 不为空，医生 ID 为空时查询该科室下所有医生列表； 当科室 ID、医生 ID 都不为空时查询该医生信息。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点

departmentId	request	String(32)		科室 ID, 医院系统中科室唯一 ID, 为空时查所有科室医生
doctorId	request	String(32)		医生 ID, 医院系统中医生唯一 ID, 为空时查询该科室下所有医生, 否则查指定某个医生信息
currentPage	request	Int	√	当前页数 (从 1 开始)
pageSize	request	Int	√	每页查询数量
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
count	response	Int	√	医生总数量, 用于分页
doctor	response	list 节点	√	医生信息
departmentId	doctor	String(32)	√	科室 ID
id	doctor	String(32)	√	医生 ID, 医院系统中医生唯一 ID
name	doctor	String(16)	√	医生名称
idCard	doctor	String(16)		医生身份证号
professionTitle	doctor	Int	√	专业职称, 详见 “专业技术职称”
office	doctor	Int	√	医生职级, 详见 “专业技术职务”
introduction	doctor	String(2560)		介绍
expertise	doctor	String(256)		医生擅长
status	doctor	Int	√	状态: 0-无效 1-有效
sex	doctor	Int	√	性别, 详见 “性别”
birthday	doctor	String(16)		出生日期, 格式: YYYY-MM-DD
mobile	doctor	String(16)		手机号码
telephone	doctor	String(16)		办公室号码
picture	doctor	String(128)		医生照片绝对路径
sort	doctor	Int	√	医生排序

➤ 正常请求业务参数:

```
<request>
  <departmentId><![CDATA[001]]></departmentId>
  <doctorId><![CDATA[-1]]></doctorId>
  <currentPage><![CDATA[1]]></currentPage>
  <pageSize><![CDATA[10]]></pageSize>
</request>
```

➤ 正常返回业务参数:

```
<response>
  <count>4</count>
  <doctor>
```



```

<departmentId>001</departmentId>
<id>1</id>
<name>精神病医生 1 号</name>
<idCard></idCard>
<professionTitle>1</professionTitle>
<office>1</office>
<introduction></introduction>
<expertise></expertise>
<status>1</status>
<sex>1</sex>
<birthday></birthday>
<mobile></mobile>
<telephone></telephone>
<picture></picture>
</doctor>
<doctor>
    .....
</doctor>
</response>

```

➤ 业务返回码：

编码(resultCode)	说明(resultMessage)
S05010201	未查询到医生记录

2.1.3.1.3. 排班信息查询（GetScheduleInfo）

接口名称	GetScheduleInfo			
接口说明	平台调用该接口获取医生排班信息。 医生 ID 为空时查询科室 ID 下所有医生排班。 如果医院挂号是挂到科室，需要虚拟一个医生返回给平台。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
departmentId	request	String(32)	√	科室 ID，医院系统中科室唯一 ID
doctorId	request	String(32)		医生 ID，医院系统中医生唯一 ID，为空时查询科室 ID 下所有医生排班
startDate	request	String(16)	√	排班开始日期，格式：YYYY-MM-DD
endDate	request	String(16)	√	排班结束日期，格式：YYYY-MM-DD
currentPage	request	Int	√	当前页数（从 1 开始）
pageSize	request	Int	√	每页查询数量

基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
count	response	Int	√	排班总数量，用于分页
schedule	response	list 节点	√	有排班的医生信息
doctorId	schedule	String(32)	√	医生 ID
doctorName	schedule	String(16)	√	医生名称
date	schedule	list 节点	√	出诊日期信息
registerDate	date	String(16)	√	出诊日期，格式：YYYY-MM-DD
weekday	date	String(16)	√	出诊日期对应星期，如：星期五
noon	date	list 节点	√	时段信息
id	noon	String(32)	√	排班 ID
noon	noon	Int	√	时段，详见 “时段”
startTime	noon	String(8)	√	时段开始时间，格式：HH24:MI
endTime	noon	String(8)	√	时段结束时间，格式：HH24:MI
status	noon	Int	√	出诊状态：0-停诊 1-正常 2-未开放
totalCount	noon	Int	√	该时段可挂号源总数，不限号源数量默认传 99
remainCount	noon	Int	√	该时段剩余号源数，不限号源数量默认传 99
registerFee	noon	Decimal(10,2)	√	挂号费用，单位：分
treatFee	noon	Decimal(10,2)	√	诊疗费用，单位：分
serviceFee	noon	Decimal(10,2)	√	服务费用，单位：分
otherFee	noon	Decimal(10,2)	√	其他费用，单位：分
level	noon	Int	√	号源类别：1-普通 2-专家 3-急诊 4-专病 5-特需 6-膏方
sexLimit	noon	Int		班次性别限制：0-全部 1-男 2-女
ageLimitMax	noon	Int		班次年龄上限，最大值 150
ageLimitMin	noon	Int		班次年龄下限，最小值 0
visitAddress	noon	String(64)		就诊地点
takeAddress	noon	String(64)		取号地点
takeTime	noon	String(20)		取号时间，格式：HH24MI-HH24MI
clinicNo	noon	String(32)		就诊序号，多个序号用中文逗号“，”隔开
updateTime	noon	String(32)		排班更新时间，格式：YYYY-MM-DD HH24:MI:SS
isTime	noon	Int	√	是否有分时：0-否 1-是

➤ 正常请求业务参数：

<request>

```

<departmentId><![CDATA[001]]></departmentId>
<doctorId><![CDATA[3]]></doctorId>
<startDate><![CDATA[2017-02-08]]></startDate>
<endDate><![CDATA[2017-02-16]]></endDate>
<currentPage><![CDATA[1]]></currentPage>
<pageSize><![CDATA[10]]></pageSize>
</request>

```

➤ 正常返回业务参数：

```

<response>
  <count>4</count>
  <schedule>
    <doctorId>1</doctorId>
    <doctorName>医生 1 号</doctorName>
    <date>
      <registerDate>2017-01-19</registerDate>
      <weekday>星期四</weekday>
      <noon>
        <id>0001</id>
        <noon>1</noon>
        <startTime>09:00</startTime>
        <endTime>17:30</endTime>
        <status>1</status>
        <totalCount>99</totalCount>
        <remainCount>99</remainCount>
        <registerFee>100</registerFee>
        <treatFee>0</treatFee>
        <serviceFee>0</serviceFee>
        <otherFee>0</otherFee>
        <level>2</level>
        <sexLimit></sexLimit>
        <ageLimitMax>0</ageLimitMax>
        <ageLinitMin>150</ageLinitMin>
        <visitAddress></visitAddress>
        <takeAddress></takeAddress>
        <takeTime></takeTime>
        <clinicNo></clinicNo>
        <updateTime></updateTime>
        <isTime>0</isTime>
      </noon>
    </date>
  </schedule>
  <schedule>
    .....
  </schedule>

```

</response>

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
S05010301	未查询到排班记录

2.1.3.1.4. 排班分时信息查询 (GetScheduleTimeInfo)

接口名称	GetScheduleTimeInfo			
接口说明	平台调用该接口，获取医生分时排班列表。 如果医生排班有分时信息，则需要调用该接口。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
departmentId	request	String(32)	√	科室 ID，医院系统中科室唯一 ID
doctorId	request	String(32)	√	医生 ID，医院系统中医生唯一 ID
registerDate	request	String(16)	√	出诊日期，格式：YYYY-MM-DD
noon	request	Int	√	时段，详见 “时段” ，为-1 时查询全天时段
scheduleId	request	String(256)		排班 ID，支持传多个，用“,”分隔（英文逗号）
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
time	response	list 节点	√	医生分时排班信息
noon	time	Int	√	时段，详见 “时段”
scheduleId	time	String(32)		排班 ID
id	time	String(32)	√	分时 ID
startTime	time	String(8)	√	分时开始时间，格式：HH24:MI
endTime	time	String(8)	√	分时结束时间，格式：HH24:MI
totalCount	time	Int	√	该分时可挂号源总数，不限号源数量默认传 99
remainCount	time	Int	√	该分时剩余号源数，不限号源数量默认传 99
clinicNo	time	String(32)		就诊序号，多个序号用中文逗号“，”隔开
status	time	Int	√	出诊状态：0-停诊 1-正常 2-未开放

➤ 正常请求业务参数:

<request>

<departmentId><![CDATA[001]]></departmentId>

```

<doctorId><![CDATA[3]]></doctorId>
<registerDate><![CDATA[2017-01-19]]></registerDate>
<noon><![CDATA[2]]></noon>
<scheduleId><![CDATA[0003]]></scheduleId>
</request>

```

➤ 正常返回业务参数：

```

<response>
  <time>
    <noon>2</noon>
    <id>0003</id>
    <startTime>14:30</startTime>
    <endTime>17:30</endTime>
    <totalCount>99</totalCount>
    <remainCount>99</remainCount>
    <clinicNo></clinicNo>
    <status>1</status>
  </time>
  <time>
    .....
  </time>
</response>

```

➤ 业务返回码：

编码(resultCode)	说明(resultMessage)
S05010401	未查询到排班分时记录

2.1.3.1.5. 预约挂号（OrderRegister）

接口名称	OrderRegister			
接口说明	平台调用该接口进行预约挂号操作，挂号下单，如果用户在医院系统不存在则登记用户信息。用户通过平台进行挂号，生成相应的订单，同时平台调用医院接口将订单信息同步到医院系统，医院系统根据传入的用户参数匹配相应的用户挂号成功。 1、对于给本人和他人挂号的，必须要填写患者证件信息，挂号规则 rule 节点 id= certificate 必传，取号的时候需要出示患者的证件 2、对于给子女挂号的，患者证件信息允许为空。当子女证件不为空时，挂号规则 rule 节点 id = certificate 必传，取号时出示子女证件取号；当子女证件为空时，挂号规则 rule 节点 id = parentCertificate 必传，取号时出示监护人有效证件取号 3、对于锁定号源类型的类型，挂号接口提交时，HIS 需要将原来锁定的号源分配给挂号的患者			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点

orderId	request	String(64)	√	平台订单号，如果是通过锁号挂号，那此订单号和锁号 ID 一致
hospitalLockId	request	String(64)		院内号源锁定 ID
channelId	request	Int	√	挂号渠道 ID，详见 “渠道类型”
scheduleId	request	String(32)	√	排班 ID
timeId	request	String(32)		分时 ID
departmentId	request	String(32)	√	科室 ID
doctorId	request	String(32)	√	医生 ID
registerDate	request	String(16)	√	挂号时间，格式 YYYY-MM-DD
noon	request	Int	√	时段，详见 “时段”
startTime	request	String(8)		分时开始时间，格式：HH24:MI
endTime	request	String(8)		分时结束时间，格式：HH24:MI
registerFee	request	Decimal(10,2)	√	挂号费用，单位：分
treatFee	request	Decimal(10,2)	√	诊疗费用，单位：分
serviceFee	request	Decimal(10,2)	√	服务费用，单位：分
otherFee	request	Decimal(10,2)	√	其他费用，单位：分
clinicNo	request	String(16)		就诊序号
registerType	request	Int	√	挂号类型：1-当天挂号 2-预约挂号（锁号挂号） 3-预约挂号（直接挂号）
patientName	request	String(16)	√	就诊人姓名
patientMobile	request	String(16)	√	就诊人手机号码
patientBirthday	request	String(16)	√	就诊人出生日期，格式：YYYY-MM-DD
patientSex	request	Int	√	就诊人性别，详见 “性别”
patientFlag	request	Int	√	用户类别：0-成人 1-儿童
agentId	request	String(128)		座席工号/操作员编码
orderTime	request	String(32)	√	下单时间，格式：YYYY-MM-DD HH24:MI:SS
group	request	list 节点	√	挂号规则组
flag	group	Int	√	规则组类型：0-初诊 1-复诊 2-出院复诊
rule	group	list 节点	√	用户规则集合，根据不同医院需求传入
id	rule	String(16)	√	用户规则，详见 “用户规则”
type	rule	String(32)		用户规则对应的类型 例如 id=card, type=1, 代表根据健康卡挂号
value	rule	String(32)	√	用户规则对应的值
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				

response			√	根节点
hospitalOrderId	response	String(64)	√	医院订单号，唯一
clinicNo	request	String(32)		就诊序号
takeTime	response	String(32)		医院取号时间段，格式：HH24:MI-HH24:MI
visitTime	response	String(32)		就诊时间段，格式：HH24:MI-HH24:MI
visitAddress	response	String(256)		医院就诊地址，例：门诊大楼四楼外科一诊室
patientId	response	String(32)	√	就诊人在院内唯一 ID
visitNo	response	String(32)		就诊号，当次就诊唯一
cardType	response	Int		用户卡类型，详见 “卡类型”
cardNo	response	String(32)		用户卡号
remark	response	String(1280)		备注
isConcessions	response	Int	√	是否享受院内优惠：0-不享受 1-享受
concessions	response	list 节点	√	优惠信息，当 isConcessions =1 必传
concessionsFee	concessions	Decimal(10,2)	√	本次挂号优惠的总金额，无优惠返回 0，单位：分
realFee	concessions	Decimal(10,2)	√	在院内优惠后实际需要支付的总费用，无优惠返回原挂号费用，单位：分
concessionsType	concessions	String(32)		优惠类型，例：老人优惠，军人优惠等

➤ 正常请求业务参数：

```

<request>
  <orderId><![CDATA[201702081513554853]]></orderId>
  <hospitalLockId><![CDATA[]]></hospitalLockId>
  <channelId><![CDATA[1]]></channelId>
  <scheduleId><![CDATA[0003]]></scheduleId>
  <timeId><![CDATA[]]></timeId>
  <departmentId><![CDATA[002]]></departmentId>
  <doctorId><![CDATA[3]]></doctorId>
  <registerDate><![CDATA[2017-01-19]]></registerDate>
  <noon><![CDATA[2]]></noon>
  <startDate><![CDATA[14:30]]></startDate>
  <endDate><![CDATA[17:30]]></endDate>
  <registerFee><![CDATA[0]]></registerFee>
  <treatFee><![CDATA[0]]></treatFee>
  <serviceFee><![CDATA[0]]></serviceFee>
  <otherFee><![CDATA[0]]></otherFee>
  <clinicNo><![CDATA[]]></clinicNo>
  <registerType><![CDATA[2]]></registerType>
  <patientName><![CDATA[患者]]></patientName>
  <patientMobile><![CDATA[123456677777]]></patientMobile>
  <patientBirthday><![CDATA[1992-05-29]]></patientBirthday>
  <patientSex><![CDATA[1]]></patientSex>

```

```

<agentId><![CDATA[]]></agentId>
<orderTime><![CDATA[2017-02-08 15:13:55]]></orderTime>
<group>
  <flag><![CDATA[1]]></flag>
  <rule>
    <type><![CDATA[1]]></type>
    <id><![CDATA[certificate]]></id>
    <value><![CDATA[431022199205293415]]></value>
  </rule>
</group>
</request>

```

➤ 正常返回业务参数：

```

<response>
  <hospitalOrderId>hosp12345677</hospitalOrderId>
  <clinicNo></clinicNo>
  <takeTime></takeTime>
  <visitTime></visitTime>
  <visitAddress></visitAddress>
  <patientId>123456</patientId>
  <visitNo></visitNo>
  <medicalNo></medicalNo>
  <cardType></cardType>
  <cardNo></cardNo>
  <remark></remark>
  <isConcessions>0</isConcessions>
</response>

```

➤ 业务返回码：

编码(resultCode)	说明(resultMessage)
S05010701	医院挂号规则限制
S05010702	不符合科室挂号规则
S05010703	用户建档失败
S05010704	院内多个用户档案，请先联系医院处理
S05010705	无效的排班
S05010706	平台订单号已存在
S05010707	该排班挂号已满（无剩余号源）
S05010708	该排班下的当前号源已被占用
S05010709	卡信息和院内用户档案不匹配
S05010710	证件信息和院内用户档案不匹配

2.1.3.1.6. 挂号变更（NotifyRegister）

接口名称	NotifyRegister
------	-----------------------

接口说明	平台调用该接口进行挂号订单的取消、取号操作。 1、取消： 已挂号成功但未支付订单，用户可主动发起取消操作，通知医院释放号源。另外对于预挂号成功且30分钟内没有支付的情况，平台自动进行取消操作。 如果医院订单已经取消成功的情况下，医院应该直接返回成功。 2、取号：适用于非 his 端的取号，例如自助终端的取号。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
notifyType	request	Int	√	变更类型：1-取消 2-取号
orderId	request	String(64)	√	平台挂号订单号
hospitalOrderId	request	String(64)	√	医院订单号，预挂号下单成功后产生
hospitalTradeId	request	String(64)		医院支付单号，预挂号订单支付成功后产生
notifyTime	request	String(32)	√	变更时间，格式：YYYY-MM-DD HH24:MI:SS
reason	request	String(128)		变更原因（notifyType=1 必填）
user	request	list 节点		用户规则集合，根据不同医院需求传入
id	user	String(16)	√	用户规则，详见 “用户规则”
type	user	String(16)		用户规则对应的类型 例：id=card,type=1，根据健康卡操作
value	user	String(32)	√	用户规则对应的值
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
无				

➤ 正常请求业务参数：

```

<request>
  <notifyType><![CDATA[0]]></notifyType>
  <orderId><![CDATA[1]]></orderId>
  <hospitalOrderId><![CDATA[1]]></hospitalOrderId>
  <hospitalTradeId><![CDATA[1]]></hospitalTradeId>
  <notifyTime><![CDATA[2017-02-08 15:47:11]]></notifyTime>
  <reason><![CDATA[]]></reason>
  <user>
    <type><![CDATA[1]]></type>
    <id><![CDATA[card]]></id>
    <value><![CDATA[13123]]></value>
  </user>
</request>

```

```

<type><![CDATA[1]]></type>
<id><![CDATA[certificate]]></id>
<value><![CDATA[210102199001012010]]></value>

</user>

</request>

```

➤ 正常返回业务参数:

无

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
S05010801	挂号订单不存在
S05010802	挂号订单已支付
S05010803	挂号订单已取消
S05010804	挂号订单已取号

2.1.3.1.7. 挂号订单记录查询 (QueryRegister)

接口名称	QueryRegister			
接口说明	平台调用该接口查询挂号订单在院内信息，订单信息和用户信息必传一项。 1、订单信息节点（orderId、hospitalOrderId、hospitalTradeId）不为空，根据订单号查询； 2、订单信息节点为空，根据用户+挂号单起止日期查询。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
orderId	request	String(64)		平台挂号订单号（可不传，根据日期、用户查）
hospitalOrderId	request	String(64)		医院订单号，预挂号下单成功后产生（可不传，根据日期、用户查）
hospitalTradeId	request	String(64)		医院支付单号，预挂号订单支付成功后产生（可不传，根据日期、用户查）
hospitalRefundId	request	String(64)		医院退款单号，预挂号订单退款成功后产生（可不传，根据日期、用户查）
user	request	list 节点		用户规则集合，根据不同医院需求传入
id	user	String(16)	√	用户规则，详见 “用户规则”
type	user	String(16)		用户规则对应的类型 例：id=card,type=1，根据健康卡操作
value	user	String(32)	√	用户规则对应的值
startDate	request	String(16)	√	起始日期，格式：YYYY-MM-DD
endDate	request	String(16)	√	结束日期，格式：YYYY-MM-DD
queryType	request	Int	√	查询状态类型：0-所有 1-失败订单 2-未支付订单 3-已支付订单 4-停诊 5-未取号订单

currentPage	request	Int	√	当前页数（从 1 开始）
pageSize	request	Int	√	每页查询数量
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
count	response	Int	√	订单总数量，用于分页
order	response	list 节点	√	挂号订单信息
orderId	order	String(64)	√	平台订单号
hospitalOrderId	order	String(64)	√	医院订单号
hospitalTradeId	order	String(64)		医院支付单号
receiptId	order	String(64)		收据号/发票号
visitNo	order	String(64)		就诊号，当次就诊唯一
channelId	order	Int	√	挂号渠道，详见 “渠道类型”
scheduleId	order	String(32)	√	排班 ID
timeId	order	String(32)		分时 ID
departmentId	order	String(32)	√	科室 ID
departmenName	order	String(16)	√	科室名称
doctorId	order	String(32)	√	医生 ID
doctorName	order	String(16)	√	医生名称
registerDate	order	String(16)	√	挂号就诊日期，格式 YYYY-MM-DD
noon	order	Int	√	时段，详见 “时段”
startTime	order	String(8)		分时开始时间，格式：HH24:MI
endTime	order	String(8)		分时结束时间，格式：HH24:MI
clinicNo	order	String(16)		就诊序号
visitAddress	order	String(256)		医院就诊地址，例：门诊大楼四楼外科一诊室
getTime	order	String(32)		要求取号时间段，格式：HH24:MI-HH24:MI
registerFee	order	Decimal(10,2)	√	挂号费用，单位：分
treatFee	order	Decimal(10,2)	√	诊疗费用，单位：分
serviceFee	order	Decimal(10,2)	√	服务费用，单位：分
otherFee	order	Decimal(10,2)	√	其他费用，单位：分
totalFee	order	Decimal(10,2)	√	订单总金额，单位：分
realFee	order	Decimal(10,2)	√	实际支付金额，单位：分
orderTime	order	String(32)	√	下单时间，格式：YYYY-MM-DD HH24:MI:SS

isPay	order	Int	√	是否支付：0-未支付 1-平台支付 2-非平台支付
payTime	order	String(32)		支付时间，格式：YYYY-MM-DD HH24:MI:SS
isTake	order	Int	√	是否取号：0-未取号 1-已取号
takeTime	order	String(32)		取号时间，格式：YYYY-MM-DD HH24:MI:SS
isCancel	order	Int	√	是否取消：0-未取消 1-已取消
isRefund	order	Int	√	是否退款：0-未退款 1-平台退款 2-非平台退款
refundlFee	order	Decimal(10,2)	√	平台退款金额，单位：分
isStop	order	Int	√	是否停诊：0-未停诊 1-已停诊
cancelTime	order	String(32)		取消/退款时间，格式：YYYY-MM-DD HH24:MI:SS
noshow	order	Int	√	是否爽约：0-未爽约 1-已爽约
patientId	order	String(32)	√	院内用户 ID，医院系统中用户唯一 ID
medicalNo	order	String(32)		病历号/门诊号
certfcateType	order	Int		用户证件类型，详见 “证件类型”
certfcateNo	order	String(32)		用户证件号码
cardType	order	Int		用户卡类型，详见 “卡类型”
cardNo	order	String(32)		用户卡号
patientName	order	String(16)	√	用户姓名
patientSex	order	Int	√	用户性别，详见 “性别”
patientBirthday	order	String(16)	√	用户出生日期，格式：YYYY-MM-DD
patientAddress	order	String(128)		用户地址
patientMobile	order	String(20)		用户手机号码
remark	order	String(128)		备注

➤ 正常请求业务参数：

```

<request>
  <orderId><![CDATA[]]></orderId>
  <hospitalOrderId><![CDATA[]]></hospitalOrderId>
  <hospitalTradeId><![CDATA[]]></hospitalTradeId>
  <hospitalRefundId><![CDATA[]]></hospitalRefundId>
  <user>
    <type><![CDATA[1]]></type>
    <id><![CDATA[certfcate]]></id>
    <value><![CDATA[431022199205293415]]></value>
  </user>
  <startDate><![CDATA[2017-01-19]]></startDate>
  <endDate><![CDATA[2017-01-22]]></endDate>
  <currentPage><![CDATA[1]]></currentPage>
  <pageSize><![CDATA[10]]></pageSize>
</request>

```

➤ 正常返回业务参数：

```

<response>
  <count>1</count>
  <order>
    <orderId>123132131231</orderId>
    <hospitalOrderId>123455555</hospitalOrderId>
    <visitNo></visitNo>
    <channelId>1</channelId>
    <scheduleId>003</scheduleId>
    <timeId></timeId>
    <departmentId>001</departmentId>
    <departmenName>精神病科</departmenName>
    <doctorId>1</doctorId>
    <doctorName>精神病医生 1 号</doctorName>
    <registerDate>2017-01-19</registerDate>
    <noon>1</noon>
    <startTime></startTime>
    <endTime></endTime>
    <clinicNo></clinicNo>
    <visitAddress></visitAddress>
    <registerFee>100.00</registerFee>
    <treatFee>0.00</treatFee>
    <otherFee>0.00</otherFee>
    <totalFee>100.00</totalFee>
    <realFee>100.00</realFee>
    <orderTime>2017-01-19 17:20:56</orderTime>
    <isPay>1</isPay>
    <payTime>2017-01-19 17:21:14</payTime>
    <isTake>1</isTake>
    <takeTime>2017-01-19 17:22:01</takeTime>
    <isCancel>0</isCancel>
    <isRefund>0</isRefund>
    <isStop>0</isStop>
    <cancelTime></cancelTime>
    <patientId>123456</patientId>
    <medicalNo></medicalNo>
    <certificateType> </certificateType>
    <certificateNo></certificateNo>
    <cardType></cardType>
    <cardNo></cardNo>
    <patientName>吕俊</patientName>
    <patientSex>1</patientSex>
    <patientBirthday>1992-05-29</patientBirthday>
    <patientAddress></patientAddress>

```

```

    <patientMobile> </patientMobile>
    <remark></remark>
  </order>
</response>

```

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
S05010901	未查询到挂号订单记录
S05010902	用户不存在

2.1.3.1.8. 停诊查询 (QueryStopRegister)

接口名称	QueryStopRegister			
接口说明	平台调用该接口主动查询院内医生停诊或排班调整信息。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
departmentId	request	String(32)	√	科室 ID
doctorId	request	String(32)		医生 ID
startDate	request	String(32)	√	查询开始日期, 格式: YYYY-MM-DD
endDate	request	String(32)	√	查询截止日期, 格式: YYYY-MM-DD
currentPage	request	Int	√	当前页数 (从 1 开始)
pageSize	request	Int	√	每页查询数量
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
response			√	根节点
count	response	Int	√	总数量, 用于分页
callback	response	Int	√	是否回调医院接口处理退款/取消: 0-否 1-是
isRefund	response	Int	√	是否需要平台进行退款: 0-是 1-否
stop	response	list 节点	√	停诊信息
scheduleId	stop	String(32)	√	排班 ID
timeId	stop	String(32)		分时 ID
departmentId	stop	String(32)	√	科室 ID
doctorId	stop	String(32)	√	医生 ID

date	stop	String(16)	√	停诊日期，格式：YYYY-MM-DD
noon	stop	Int	√	时段，详见 “时段”
startTime	stop	String(8)		停诊开始时间，格式：HH24:MI 没有分时排班的，此项必须为空
endTime	stop	String(8)		停诊结束时间，格式：HH24:MI 没有分时排班的，此项必须为空
reason	stop	String(512)	√	停诊原因

➤ 正常请求业务参数：

```
<request>
  <departmentId><![CDATA[1]]></departmentId>
  <doctorId><![CDATA[1]]></doctorId>
  <startDate><![CDATA[2017-02-08]]></startDate>
  <endDate><![CDATA[2017-02-08]]></endDate>
  <currentPage><![CDATA[1]]></currentPage>
  <pageSize><![CDATA[9999]]></pageSize>
</request>
```

➤ 正常返回业务参数：

```
<response>
  <count>1</count>
  <stop>
    <scheduleId>1</scheduleId>
    <timeId></timeId>
    <departmentId>1</departmentId>
    <doctorId>1</doctorId>
    <date>2017-02-06</date>
    <noon>1</noon>
    <startTime></startTime>
    <endTime></endTime>
    <reason>停诊</reason>
  </stop>
  <stop>
    .....
  </stop>
</response>
```

➤ 业务返回码：

编码(resultCode)	说明(resultMessage)
S05011101	未查询医生停诊记录

2.2. 平台提供给医院调用（API 部分）

2.2.1. 门诊部分

2.2.1.1. 挂号部分

2.2.1.1.1. 停诊通知（StopRegister）

接口名称	StopRegister			
接口说明	医院系统调用平台的该接口通知医生出诊信息变更（停诊或者排班调整）。平台在收到变更通知后，对受影响的订单进行相应的操作。 平台对挂号成功但未支付成功状态的订单自动进行取消操作处理，取消平台订单并调用医院挂号取消接口通知医院。 平台对挂号成功且支付成功状态的订单自动进行退款操作处理，退款平台订单并调用医院挂号退款接口通知医院。 如果平台状态为取号的，则不做自动退款，由患者收到通知后主动到窗口进行退款或咨询。 对于停诊时间比较短的情况，可以输入停诊起止时间。 如果医院因为医院停诊通知平台，请医院保证所发停诊的排班状态修正为停诊状态。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
departmentId	request	String(32)	√	科室 ID
doctorId	request	String(32)	√	医生 ID
date	request	String(16)	√	停诊日期，格式：YYYY-MM-DD
noon	request	Int	√	时段，详见 “时段”
startTime	request	String(8)		停诊开始时间，格式：HH24:MI 没有分时排班的，此项必须为空
endTime	request	String(8)		停诊结束时间，格式：HH24:MI 没有分时排班的，此项必须为空
reason	request	String(512)	√	停诊原因
callback	request	Int	√	是否回调医院接口处理退款/取消：0-否 1-是
isRefund	request	Int	√	是否需要平台进行退款：0-是 1-否
基本返回参数				
基本返回参数				
responseEncrypted 报文参数				
无				

➤ 正常请求业务参数：

```
<request>
  <departmentId><![CDATA[a]]></departmentId>
  <doctorId><![CDATA[1]]></doctorId>
  <date><![CDATA[2016-01-01]]></date>
```



```

<noon><![CDATA[1]]></noon>
<startTime><![CDATA[]]></startTime>
<endTime><![CDATA[]]></endTime>
<reason><![CDATA[aa]]></reason>
</request>

```

➤ 正常返回业务参数:

无

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
A05010101	没有需要处理停诊的订单
A05010102	没有查询到可停诊的排班信息

2.2.1.1.2. 挂号变更通知 (NotifyRegisterByHis)

接口名称	NotifyRegisterByHis			
接口说明	医院系统调用该接口通知平台订单状态变更（在窗口/护士站等）。 1、取消：医院主动通知平台，未支付挂号订单在院内的取消操作； 2、停诊：医院主动通知平台，挂号订单的停诊操作，平台根据订单是否支付进行相应的操作；未支付订单，回调医院挂号变更通知接口，通知医院取消订单；已支付订单，平台处理订单退款操作，同时回调医院退款接口，通知医院退号退费。如果医院都已处理，直接返回成功即可。 3、取号：医院主动通知平台，已支付挂号订单在取号操作，用户有在医院窗口取号、自助终端取号、护士站/医生站报到等都应该通知平台该用户已接诊。 4、医院支付：医院主动通知平台，未支付订单在院内的支付操作。			
节点名称	父节点名称	数据类型	必填	说明
基本请求参数				
基本请求参数				
requestEncrypted 报文参数				
request			√	根节点
notifyType	request	Int	√	变更类型：1-取消 2-停诊 3-取号 4-医院支付
orderId	request	String(64)	√	平台订单号
hospitalOrderId	request	String(64)	√	医院订单号
hospitalTradeId	request	String(64)		医院支付单号
notifyTime	request	String(32)	√	变更时间，格式：YYYY-MM-DD HH24:MI:SS
payFee	request	Decimal(10,2)		支付金额，单位：分（notifyType=4 必填）
clinicNo	request	String(16)		就诊序号
reason	request	String(512)		变更原因（notifyType=1, 2 必填）
callback	request	Int	√	是否回调医院接口处理退款/取消：0-否 1-是
isRefund	request	Int	√	是否需要平台进行退款：0-是 1-否
基本返回参数				
基本返回参数				

responseEncrypted 报文参数

无

➤ 正常请求业务参数:

```

<request>
  <notifyType><![CDATA[1]]></notifyType>
  <orderId><![CDATA[1]]></orderId>
  <hospitalTradeId><![CDATA[1]]></hospitalTradeId>
  <notifyTime><![CDATA[2017-02-07 14:54:41]]></notifyTime>
  <payFee><![CDATA[0]]></payFee>
  <clinicNo><![CDATA[1]]></clinicNo>
  <reason><![CDATA[1]]></reason>
  <extend><![CDATA[1]]></extend>
</request>

```

➤ 正常返回业务参数:

无

➤ 业务返回码:

编码(resultCode)	说明(resultMessage)
A05010201	挂号订单不存在

2.3. 接口统一返回码说明

编码(resultCode)	说明(resultMessage)
SUCCESS	成功
SYSTEM_ERROR	系统错误
SESSION_TIMEOUT	连接超时错误
ILLEGAL_SECURITY_KEY	密钥不匹配
ILLEGAL_SIGN	签名不正确
ILLEGAL_ENCRYPT	加密不正确
NULL_REQUEST	请求数据为空
ILLEGAL_REQUEST	请求数据无效
NULL_RESPONSE	返回数据为空
ILLEGAL_RESPONSE	返回数据无效
ILLEGAL_ARGUMENT	参数不正确
HAS_NO_PRIVILEGE	未开通接口访问权限
ILLEGAL_SERVICE	传入非法接口名
ILLEGAL_PARTNER	接入方用户不正确
ILLEGAL_EXTERFACE	接口配置不正确
REPEAT_TRANSACTION	重复提交
ILLEGAL_SIGN_TYPE	签名类型不正确
ILLEGAL_CHARSET	字符集不合法

ILLEGAL_ENCODING	不支持该编码类型
ILLEGAL_CLIENT_IP	客户端 IP 地址无权访问服务
PAGE_LIMIT	分页查询超限

3. 数据字典定义

字典类别	字典值	字典说明
性别	0	未知
	1	男
	2	女
	9	其他
卡类型	1	健康卡
	2	市民卡/社保卡
	3	医保卡
	4	银行卡
	5	公费医疗证
	6	农合证
	7	院内诊疗卡
	8	条形码
	99	其它卡
证件类型	1	身份证
	2	港澳居民来往内地通行证
	3	台湾居民来往大陆通行证/台胞证
	4	护照
	5	军官证
	6	出生证
	7	驾驶证
	8	残疾证
	9	其它
医院等级	0	其他
	1	一级
	2	二级
	3	三级
	4	特级
	31	三甲
	32	三乙
	33	三丙
	21	二甲
	22	二乙
	23	二丙
	11	一甲
	12	一乙

	13	一丙
时段	4	全天
	1	上午
	2	下午
	3	晚上
渠道类型	1	微信公众号
	2	支付宝服务窗
	3	手机 APP
	4	网站
	5	微医
	6	医院窗口
	9	自助终端
	99	其他
支付类型	101	微信 JSAPI 支付（公众号支付）
	102	微信 APP 支付
	103	微信扫码支付
	104	微信刷卡支付
	201	支付宝手机网站支付
	202	支付宝及时到账支付
	203	支付宝 APP 支付
	204	支付宝扫码支付
	205	支付宝当面付
	301	银联 h5 支付
	302	银联手机控件支付
	401	诊疗卡支付
	999	其他
医生专业技术职称	23	卫生技术人员（医疗）
	231	主任医师
	232	副主任医师
	233	主治医师
	234	医师
	235	医士
	24	卫生技术人员（药剂）
	241	主任药师
	242	副主任药师
	243	主管药师
	244	药师
	245	药士
	25	卫生技术人员（护理）

	251	主任护师
	252	副主任护师
	253	主管护师
	254	护师
	255	护士
	26	卫生技术人员（技师）
	261	主任技师
	262	副主任技师
	263	主管技师
	264	技师
	265	技士
	999	其他
费用类别	1	西药费
	2	中成药费
	3	中草药费
	4	放射费
	5	CT 费
	6	手术费
	7	检验费
	8	治疗费
	9	检查费
	10	材料费
	11	护理费
	12	挂号费
	13	诊疗费
	14	留观费
	99	其它费
排队类别	1	挂号候诊
	2	取药排队
	3	输液排队
	4	抽血排队
	5	医技检查排队
	6	检验排队
	99	其他排队
用户规则（根据需要动态传入，不是所有规则都要传，同一个规则只允许传入一次）	id	院内用户 id
	medicalNo	门诊号/病历号/就诊号
	admissionNo	住院号
	card	卡信息
	certificate	证件信息

	password	卡密码
	name	姓名
	sex	性别
	birthday	出生日期
	nation	民族
	mobile	手机号码
	parentName	监护人姓名
	parentCertificate	监护人证件信息
	address	地址
	disease	疾病
	allergicHistory	过敏史
	pastHistory	既往史
	chiefComplaint	主诉
支付方式	1	现金
	2	银行卡
剂型	11	口服常释剂型
	12	缓释控释剂型
	13	口服液体剂
	14	丸剂
	15	颗粒剂
	16	口服散剂
	17	外用散剂
	18	软膏剂
	19	贴剂
	20	外用液体剂
	21	硬膏剂
	22	凝胶剂
	23	涂剂
	24	栓剂
	25	滴眼剂
	26	滴耳剂
	27	滴鼻剂
	28	吸入剂
	29	注射剂
专业技术职务	1	正高/教授
	2	副高/副教授
	3	中级/讲师
	4	师级/助理
	5	士级

医疗类别	11	普通门诊
	12	急诊
	13	门诊特病
	14	门诊慢病
	15	特需门诊
	21	普通住院
	22	特殊病种住院
	23	转外诊治住院
	24	特需住院
	31	家庭病床
	41	药店购药
民族	1	汉族
	2	蒙古族
	3	回族
	4	藏族
	5	维吾尔族
	6	苗族
	7	彝族
	8	壮族
	9	布依族
	10	朝鲜族
	11	满族
	12	侗族
	13	瑶族
	14	白族
	15	土家族
	16	哈尼族
	17	哈萨克族
	18	傣族
	19	黎族
	20	傈僳族
	21	佤族
	22	畲族
	23	高山族
	24	拉祜族
	25	水族
	26	东乡族
	27	纳西族
	28	景颇族

	29	柯尔克孜族
	30	土族
	31	达翰尔族
	32	佤族
	33	羌族
	34	布朗族
	35	撒拉族
	36	毛南族
	37	仡佬族
	38	锡伯族
	39	阿昌族
	40	普米族
	41	塔吉克族
	42	怒族
	43	乌孜别克族
	44	俄罗斯族
	45	鄂温克族
	46	德昂族
	47	保安族
	48	裕固族
	49	京族
	50	塔塔尔族
	51	独龙族
	52	鄂伦春族
	53	赫哲族
	54	门巴族
	55	珞巴族
	56	基诺族
	90	外籍人士
	99	其他
国家和地区	156	中国
	826	英国
	392	日本
	410	韩国
	840	美国

4. 加签和加密校验

4.1. 签名机制

4.1.1. 需要参与签名的参数

在请求和返回参数列表中，除去 **sign**、**signType** 两个参数外，其他需要使用到的参数皆是要签名的参数。

4.1.2. 生成待签名字符串

对于请求参数，如下格式：

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <service><![CDATA[NetTest]]></service>
  <partner><![CDATA[TEST_WX]]></partner>
  <hospitalCode><![CDATA[0]]></hospitalCode>
  <dataFormat><![CDATA[xml]]></dataFormat>
  <inputCharset><![CDATA[utf-8]]></inputCharset>
  <signType><![CDATA[md5]]></signType>
  <sign><![CDATA[389F6997DA8530F147D4AD7AF41C5C30]]></sign>
  <requestEncrypted><![CDATA[oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAXm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==]]></requestEncrypted>
</root>
```

或

```
{
  "service": "NetTest",
  "partner": "TEST_WX",
  "hospitalCode": "0",
  "dataFormat": "json",
  "inputCharset": "utf-8",
  "signType": "md5",
  "sign": "722289C58286EC2B41EF764CD5B0259D",
  "requestEncrypted": "WXzV3/6APXYdqsoZIVuANePKKHbc47QODNPniynaD9s="
}
```

对 xml 串里的每一个值从 **a** 到 **z** 的顺序排序，若遇到相同首字母，则看第二个字母，以此类推（**sign**、**signType** 字段不参与签名）。排序完成之后，再把所有数组值以 “&” 字符连接起来，并将 **key** 值加在最后面一起加签（**xml 结构的待加签的字符串要去掉 CDATA 转义符**）。

排序完成之后，如：

```
dataFormat=json&hospitalCode=0&inputCharset=utf-8&partner=TEST_WX&requestEncrypted=oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAXm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==&service=NetTest&key
```

```
=2098D32C4D1399EC
```

这串就是待签名的字符串，加签完成后放到 **sign** 字段中。

```
sign=md5("dataFormat=json&hospitalCode=0&inputCharset=utf-8&partner=TEST_WX&requestEncrypted=oJ5JYA4Q33sBLcRr98DeEyB83BVT5lec5O7okvxPm111P3hBM/g1/eLAXm7laos/Yw9ZCsw3CtC0Q6j6X9C+CQ==&service=NetTest&key=2098D32C4D1399EC")
```

```
sign=389F6997DA8530F147D4AD7AF41C5C30
```

4.2. 加解密机制

4.2.1. 加密顺序

发送数据时先加密再签名，接受数据后先验签在解密。

4.2.2. 加密方式

对称加密，使用的算法：AES

本接口请求包采用整体加密方式，即 AES(业务内容)，采用 AES 加密方式采用 AES-128-ECB 加密模式，PKCS5Padding 补码，key 需要为 16 位，并对加密的内容使用 BASE64 做转码功能，同时能起到 2 次加密的作用。

4.2.3. 带加密的字符串

对于如下请求参数，如下格式：

```
<root>
  <service><![CDATA[NetTest]]></service>
  <partner><![CDATA[TEST_WX]]></partner>
  <hospitalCode><![CDATA[0]]></hospitalCode>
  <dataFormat><![CDATA[xml]]></dataFormat>
  <inputCharset><![CDATA[utf-8]]></inputCharset>
  <request>
    <ip><![CDATA[127.0.0.1]]></ip>
  </request>
</root>
```

或

```
{
  "service": "NetTest",
  "partner": "TEST_WX",
  "hospitalCode": "0",
  "dataFormat": "json",
  "request"[
    {
      "ip": "127.0.0.1"
    }
  ]
}
```

```
]
}
```

将业务参数 **request** 节点的内容采用 AES 加密方式进行加密，和“带加密的字符串” **requestEncrypted** 相对应。

对请求业务参数：

```
<request>
  <ip><![CDATA[127.0.0.1]]></ip>
</request>
```

或

```
{
  "ip": "127.0.0.1"
}
```

加密后：

```
requestEncrypted=AES("<request><ip><![CDATA[127.0.0.1]]></ip></request>",key
)
```

```
requestEncrypted=oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hB
M/g1/eLAxm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==
```

```
<requestEncrypted><![CDATA[oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAxm7Iaos/Yw
9ZCsw3CtC0Q6j6X9C+CQ==]]></requestEncrypted>
```

或

```
requestEncrypted=AES("{\"ip\":\"127.0.0.1\"}",key)
```

```
requestEncrypted=WXzV3/6APXYdqsoZIVuANePKKHbc47QODNPniynaD9s=
```

```
"requestEncrypted":"WXzV3/6APXYdqsoZIVuANePKKHbc47QODNPniynaD9s="
```

加密结果如下：

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <service><![CDATA[NetTest]]></service>
  <partner><![CDATA[TEST_WX]]></partner>
  <hospitalCode><![CDATA[0]]></hospitalCode>
  <dataFormat><![CDATA[xml]]></dataFormat>
  <inputCharset><![CDATA[utf-8]]></inputCharset>
  <signType><![CDATA[md5]]></signType>
  <sign><![CDATA[389F6997DA8530F147D4AD7AF41C5C30]]></sign>
  <requestEncrypted><![CDATA[oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAxm7Ia
os/Yw9ZCsw3CtC0Q6j6X9C+CQ==]]></requestEncrypted>
</root>
```

或

```
{
  "service": "NetTest",
```

```

    "partner": "TEST_WX",
    "hospitalCode": "0",
    "dataFormat": "json",
    "inputCharset": "utf-8",
    "signType": "md5",
    "sign": "722289C58286EC2B41EF764CD5B0259D",
    "requestEncrypted": "WXzV3/6APXYdqsoZIVuANePKKHbc47QODNPniynaD9s="
  }

```

4.2.4. 待解密的字符串

正确返回参数，如下格式：

```

<root>
  <resultCode><![CDATA[SUCCESS]]></resultCode>
  <resultMessage><![CDATA[成功]]></resultMessage>
  <sign><![CDATA[5A79CE775EC879E11B88EF71D69F131E]]></sign>
  <signType><![CDATA[md5]]></signType>
  <responseEncrypted><![CDATA[dbvhhbM8Ckn+xUMWC8mmVO3bMEcsql4P4sUDAKjE6ZEQeQwz7o0vN/+xd
sjsmNL06PISHwoS9d+6LzWgJYeRMhHFs99uN84ajwMMqEeSLDvU=]]></responseEncrypted>
</root>
或
{
  "resultCode": "SUCCESS",
  "resultMessage": "成功",
  "signType": "md5",
  "sign": "4376E2DF2191EFDCA849153DFF4A11DC",
  "responseEncrypted": "vTmXQJ1INyhGdVsRQWHn0+INhPQt1/4YXp72AfY0mtR6O+GUTd5hTDtp0RNYAw+
K"
}

```

将“待解密的字符串” **responseEncrypted** 节点的内容采用 AES 解密方式进行解密，得到返回的业务参数。

```

<response>
  <systemTime><![CDATA[2017-01-19 15:54:42]]></systemTime>
</response>
或
{
  "systemTime": "2017-01-19 15:54:42"
}

```

解密结果如下：

```

<root>
  <resultCode><![CDATA[SUCCESS]]></resultCode>
  <resultMessage><![CDATA[成功]]></resultMessage>
  <response>
    <systemTime><![CDATA[2017-01-19 15:54:42]]></systemTime>
  </response>
</root>

```

```

    </response>
</root>
或
{
  "resultCode": "SUCCESS",
  "resultMessage": "成功",
  "request"[
    {
      "systemTime": "2017-01-19 15:54:42"
    }
  ]
}

```

4.2.5. 加密/解密代码示例

- .NET 示例:

- 加密:

```

String encryptKey ="2fMVbe6Nvtyy6CsVjaZKS6Gc4zC0";
String encryptString="<request><hospitalCode><![CDATA[1001]]></hospitalCode></request>";
byte[] keyArray = UTF8Encoding.UTF8.GetBytes(encryptKey);
byte[] toEncryptArray = UTF8Encoding.UTF8.GetBytes(encryptString);
RijndaelManaged rDel = new RijndaelManaged();
rDel.Key = keyArray;
rDel.Mode = CipherMode.ECB;
//rDel.Padding = PaddingMode.PKCS7;
ICryptoTransform cTransform = rDel.CreateEncryptor();
byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);
return Convert.ToBase64String(resultArray, 0, resultArray.Length);

```

- 解密:

```

String decryptKey="2fMVbe6Nvtyy6CsVjaZKS6Gc4zC0";
String decryptString="Eg7iJ8OS8UElWVf/UdJOkl6fsnV3wVS";
byte[] keyArray = UTF8Encoding.UTF8.GetBytes(decryptKey);
byte[] toEncryptArray = Convert.FromBase64String(decryptString);
RijndaelManaged rDel = new RijndaelManaged();
rDel.Key = keyArray;
rDel.Mode = CipherMode.ECB;
//rDel.Padding = PaddingMode.PKCS7;
ICryptoTransform cTransform = rDel.CreateDecryptor();
byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);
return UTF8Encoding.UTF8.GetString(resultArray);

```

JAVA 示例:

加密:

```
String encryptKey == "2fMVbe6Nvtyy6CsVjaZKS6Gc4zC0";
String encryptString=" <request><hospitalCode><![CDATA[1001]]></hospitalCode></request>";
if (encryptKey == null) {
    System.out.print("encryptKey 为空 null");
    return null;
}
// 判断 Key 是否为 16 位
if (encryptKey.length() != 16) {
    System.out.print("Key 长度不是 16 位");
    return null;
}
byte[] raw = encryptKey.getBytes("utf-8");
SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");//"算法/模式/补码方式"
cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
byte[] encrypted = cipher.doFinal(encryptString.getBytes("utf-8"));
return new Base64().encodeToString(encrypted);
```

解密：

```
String decryptKey=" 2fMVbe6Nvtty6CsVjaZKS6Gc4zC0";
String decryptString=" Eg7iJ8OS8UElWVf/UdJOkI6fsnV3wVS";
try {
    // 判断 Key 是否正确
    if (decryptKey == null) {
        System.out.print("Key 为空 null");
        return null;
    }
    // 判断 Key 是否为 16 位
    if (decryptKey.length() != 16) {
        System.out.print("Key 长度不是 16 位");
        return null;
    }
    byte[] raw = decryptKey.getBytes("utf-8");
    SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
    Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
    cipher.init(Cipher.DECRYPT_MODE, skeySpec);
    byte[] encrypted1 = new Base64().decode(decryptString);//先用 base64 解密

    byte[] original = cipher.doFinal(encrypted1);
    String originalString = new String(original,"utf-8");
    return originalString;
} catch (Exception ex) {
    System.out.println(ex.toString());
    return null;
}
```


5. 接口调用流程

✓ **SPI 调用流程**（平台调用医院提供的接口服务）：

以“**TestNet**”接口为例。

1、医院接收平台传入的完整请求基本参数

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <service><![CDATA[NetTest]]></service>
  <partner><![CDATA[TEST_WX]]></partner>
  <hospitalCode><![CDATA[0]]></hospitalCode>
  <dataFormat><![CDATA[xml]]></dataFormat>
  <inputCharset><![CDATA[utf-8]]></inputCharset>
  <signType><![CDATA[md5]]></signType>
  <sign><![CDATA[389F6997DA8530F147D4AD7AF41C5C30]]></sign>
  <requestEncrypted><![CDATA[oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAxm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==]]></requestEncrypted>
</root>
```

2、医院对接入的平台基本参数进行 MD5 验签

拼接后的 md5 参数串（按首字母 a-z 顺序拼接）：

```
dataFormat=json&hospitalCode=0&inputCharset=utf-8&partner=TEST_WX&requestEncrypted=oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAxm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==&service=NetTest&key=2098D32C4D1399EC
```

sign=md5("dataFormat=json&hospitalCode=0&inputCharset=utf-8&partner=TEST_WX&requestEncrypted=oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAxm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==&service=NetTest&key=2098D32C4D1399EC")

sign=389F6997DA8530F147D4AD7AF41C5C30

医院将得到的 **sign** 串和平台传入的 **sign** 串进行比对，如果相同，则传入请求基本参数串未被篡改。

3、医院对 **requestEncrypted** 串进行 AES 解密，得到请求业务参数串

request=AES("oJ5JYA4Q33sBLcRr98DeEyB83BVT5Iec5O7okvxPm111P3hBM/g1/eLAxm7Iaos/Yw9ZCsw3CtC0Q6j6X9C+CQ==",key)

request=<request><ip><![CDATA[127.0.0.1]]></ip></request>

```
<request>
  <ip><![CDATA[127.0.0.1]]></ip>
</request>
```

4、医院将得到请求业务参数串进行业务逻辑操作后返回

```
<response>
  <systemTime><![CDATA[2017-01-19 15:54:42]]></systemTime>
</response>
```

5、医院将请求业务参数串进行 AES 加密，和 **responseEncrypted** 相对应

responseEncrypted=AES("<response><systemTime><![CDATA[2017-01-19 15:54:42]]></systemTime></response>",key)

responseEncrypted=dbvhhbM8Ckn+xUMWC8mmVO3bMEcsqI4P4sUDAKjE6ZEqE6Qwz7o0vN/+xdsjsmNLo6PISHwoS9d+6LzWgJYeRMhHFs99uN84ajwMMqEeSLDvU=

6、医院根据配置将返回基本参数的值拼接进行 md5 签名，放到 **sign** 字段中

sign=md5("responseEncrypted=dbvhhbM8Ckn+xUMWC8mmVO3bMEcsqI4P4sUDAKjE6ZEqE6Qwz7o0vN/+xdsjsmNLo6PISHwoS9d+6LzWgJYeRMhHFs99uN84ajwMMqEeSLDvU=&resultCode=SUCCESS&resultMessage=成功&key=2098D32C4D1399EC",key)

sign=5A79CE775EC879E11B88EF71D69F131

7、医院根据文档定义整理返回基本参数串返回给平台

```
<root>
  <resultCode><![CDATA[SUCCESS]]></resultCode>
  <resultMessage><![CDATA[成功]]></resultMessage>
  <sign><![CDATA[5A79CE775EC879E11B88EF71D69F131]]></sign>
  <signType><![CDATA[md5]]></signType>
  <responseEncrypted><![CDATA[dbvhhbM8Ckn+xUMWC8mmVO3bMEcsqI4P4sUDAKjE6ZEqE6Qwz7o0vN/+xdsjsmNLo6PISHwoS9d+6LzWgJYeRMhHFs99uN84ajwMMqEeSLDvU=]]></responseEncrypted>
</root>
```

✓ **API 调用流程**（医院调用平台提供的接口服务）：

和 SPI 调用流程类似。

- 1、医院将请求业务参数串 **<request>...</request>** 进行 **ase** 加密，放在 **requestEncrypted** 字段中
- 2、医院将请求基本参数的值拼接进行 **md5** 签名，放到 **sign** 字段中
- 3、医院根据文档定义整理请求基本参数串传给平台 **<root>...</root>**
- 4、平台接收医院报文后，先进行 **md5** 验签，再进行 **aes** 解密，处理完业务逻辑后将返回基本参数返回给医院
- 5、医院接收平台返回基本参数串 **<root>...</root>**
- 6、医院对返回基本参数串进行 **md5** 验签，比对 **sign** 字段的值
- 7、医院对 **responseEncrypted** 进行 **aes** 解密，得到平台的返回业务参数串 **<response>...</response>**

6. 附录

C#Demo 示例:



WD.DemoFor.Net
.rar