

# 移动互联网业务平台及应用

## 目的与意义

利用基于SOAP的Web服务开发一个简单的餐位预订系统

## 需求分析

### 功能性需求

#### 1. 预订餐位功能

该系统使用户能够预订两天之内的晚餐餐位，晚上七点之后不能预订当天的餐位，但可以预订后天的餐位。

用户输入就餐日期，若不在两天之内，返回“不在预订日期内”信息，提示用户重新输入就餐日期。若在两天之内，提示用户输入就餐人数，如果没有空余的座位(为了简化，将所有的桌子都认为是8人桌)，返回“暂无座位”，若有空余的座位，提示用户输入手机号码。

若为11位有效号码并且此号码没有订过餐，则提示“订餐成功”，服务器端将此号码存起来作为预订信息，否则，提醒用户重新输入手机号码。

#### 2. 取消餐位功能

提示用户输入手机号，若为11位有效号码并且此号码订过餐，则返回“退餐成功”，并在服务器端将此号码的订餐信息删除，否则，返回“无效号码”或者“此号码尚未订餐”，再次提示用户输入手机号码。

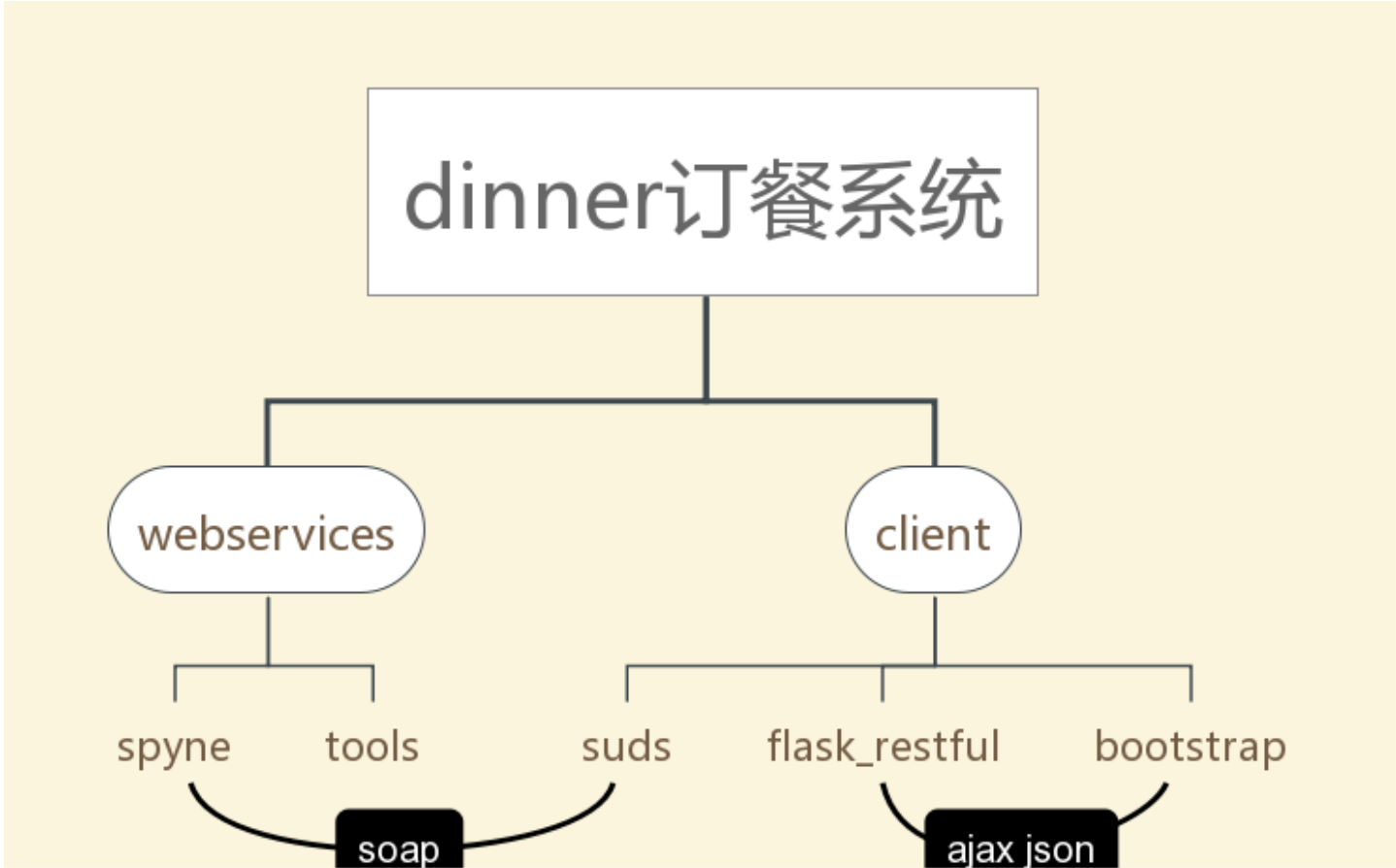
#### 3. 将该系统部署成web服务；同时编写一个客户端进行调用。

### 非功能性需求

基于SOAP的Web服务

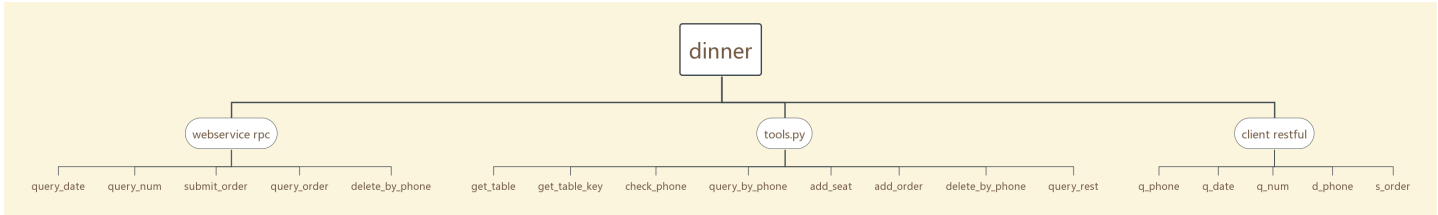
## 系统设计

### 系统体系结构设计



功能模块设计

服务端采用spyne框架构建webservice  
客户端采用suds与spyne互通，设置传输协议为soap  
客户端为基于flask-restful的web单页应用，前端采用bootstrap+jquery，前后端通过ajax沟通



系统实现

服务端实现步骤和服务的部署过程

webservice 类

```
class DinnerService(ServiceBase):
```

rpc方法

```

class DinnerService(ServiceBase):

    @rpc(DateTime,_returns=String)
    def query_date(self, query_time):
        rax = data.get_table(query_time)
        if rax != -1:
            return str(rax)
        else:
            return 'not the time'

    @rpc(DateTime,Integer,_returns=String)
    def query_num(self, book_date, book_num):
        rest = data.query_rest(book_date)
        if rest >= book_num:
            return 'ok:'+str(rest)
        else:
            return 'no'

```

调用tools类

```

def get_table(self, query_time):
    query_hour = query_time.hour
    now_time = datetime.now()
    chk1,chk2 = now_time.day,now_time.day+1
    if now_time.hour > 19:
        chk1 += 1
        chk2 += 1
    if query_time.day == chk1 or query_time.day == chk2:
        return self.get_table_key(query_time.strftime('%Y-%m-%d'))
    else:
        return -1

def get_table_key(self, key):
    if not self.seats.has_key(key):
        self.seats[key] = [[0] * 8) for i in range(6)]
    return self.seats[key]

```

## 客户端调用的构建过程

### html触发事件

```

$('#datetimepicker1').find("input").blur(function(){
    query_date($('#datetimepicker1').find("input").val())
})

```

### ajax

```

function query_date(time){
    alert(time)
    var a = {}
    a['datetime'] = time
    $.ajax({
        type: "POST",
        contentType: "application/json",
        url: "/query_date",
        data: JSON.stringify(a),
        dataType: 'json',
        cache: false,
        timeout: 600000,
        success: function (data) {
            // alert("success")
            if(data['ret'] == 'not the time'){
                // window.alert("not")
                $('#cc').html('不在预订日期内,请重新输入日期\n注意: 用户能够预订两天之内的晚餐餐位')
            }
            else{
                // alert("good")
                var t = data['ret']
                t = JSON.parse(t)
                res = ""
                for (var i = 0;i < 6;i++){
                    res += "第" + i + "桌空闲座位: "
                    var ans = 0
                    for(var j = 0;j < 8;j++){
                        if(t[i][j] ==0){
                            ans += 1
                        }
                    }
                    res += ans
                    res += "\n"
                }
                res += '输入就餐人数'
                $('#cc').html(res)
            }
        },
        error: function (e) {
            alert(XMLHttpRequest.readyState + XMLHttpRequest.status + XMLHttpRequest.re
        }
    });

    return "moren"
}

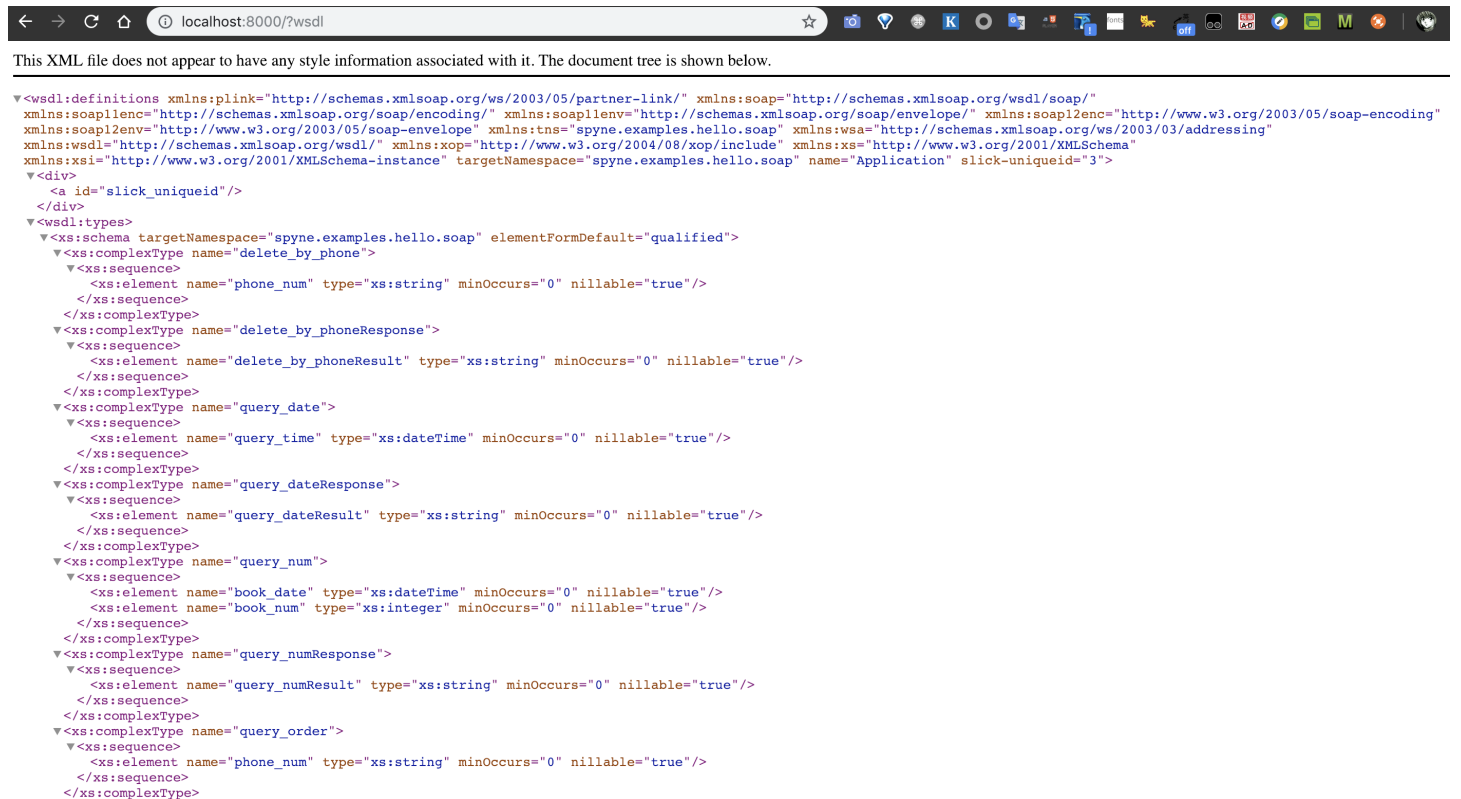
```

## web-api相应，并调用webservice方法

```
class q_date(Resource):
    def post(self):
        ret = {}
        args = parser.parse_args()
        print ("post:" + str(args['datetime']))
        date = datetime.strptime(args['datetime'], '%Y-%m-%d')
        ret['ret'] = s.service.query_date(date)
        return ret
```

## 测试与验证

打开 <http://localhost:8000/?wsdl> 可以看到描述接口的文档wsdl



访问客户端网页 <http://127.0.0.1:5000/>

# Dinner

By --- Lometsj

订餐

退订

点餐

订餐



2019-01-15



第0桌空闲座位：8  
第1桌空闲座位：8  
第2桌空闲座位：8  
第3桌空闲座位：8  
第4桌空闲座位：8  
第5桌空闲座位：8  
输入就餐人数

2

输入手机号码

13700000000

手机尚未订餐

关闭

提交订餐

订餐

2019-01-15

第0桌空闲座位：8

第1桌空闲座位：8

重新查询，座位数相应减少

2019-01-15

第0桌空闲座位：6

第1桌空闲座位：8

第2桌空闲座位：8

第3桌空闲座位：8

第4桌空闲座位：8

第5桌空闲座位：8

输入就餐人数

订餐成功

OK

通过电话查询订单

- 若为有效订单

退订

13700000000

查询到订单：

电话号码：13700000000

日期：2019-01-15

人数：2

×

- 若为无效号码

退订



10000000001

无效号码

关闭

退订

- 若为未订餐手机号

退订



13700000001

手机尚未订餐

关闭

退订

退订

退订

13700000000

查询到订单:

电话号码: 13700000000  
日期: 2019-01-15  
人数: 2

127.0.0.1:5000 Says  
退订成功

OK

关闭

退订


Name	
<input type="checkbox"/>	query_c
<input type="checkbox"/>	query_r
<input type="checkbox"/>	query_c
<input type="checkbox"/>	submit_
<input type="checkbox"/>	query_c
<input type="checkbox"/>	query_c

重新查询，座位数相应增加



# 订餐

2019-01-15



第0桌空闲座位：8

第1桌空闲座位：8

第2桌空闲座位：8

第3桌空闲座位：8

第4桌空闲座位：8

第5桌空闲座位：8

输入就餐人数

## 总结

### WSDL

WSDL 是基于 XML 的语言，用于描述 Web services 以及如何访问它们。  
WSDL 可描述某个 web service，连同用于此 web service 的消息格式和协议细节。

### Webservice

WebService是一种跨编程语言和跨操作系统平台的远程调用技术

### soap

SOAP 是基于 XML 的简易协议，可使应用程序在 HTTP 之上进行信息交换。  
或者更简单地说：SOAP 是用于访问网络服务的协议。

参考文献：  
无