

今日学习目标

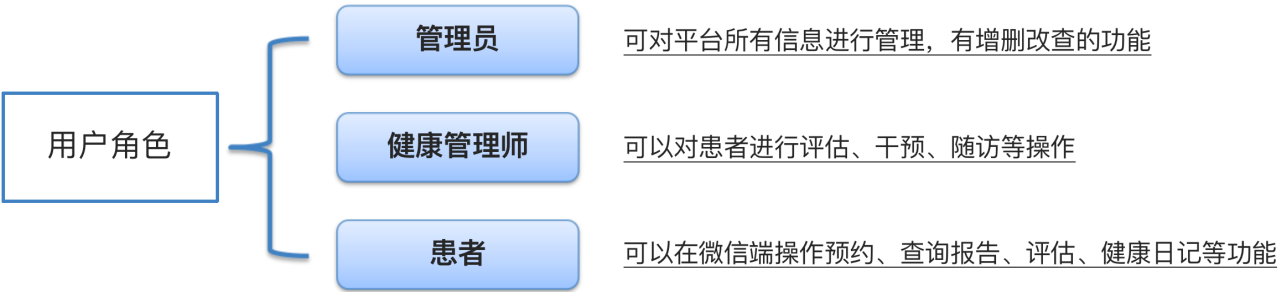
掌握telnet和python代码测试dubbo接口的方法

传智健康项目介绍

项目描述

- 传智健康管理系统，是一款应用于健康管理机构的业务系统。采用可视化界面管理，提高健康管理师工作效率，加强与患者间的互动。

目标用户群体

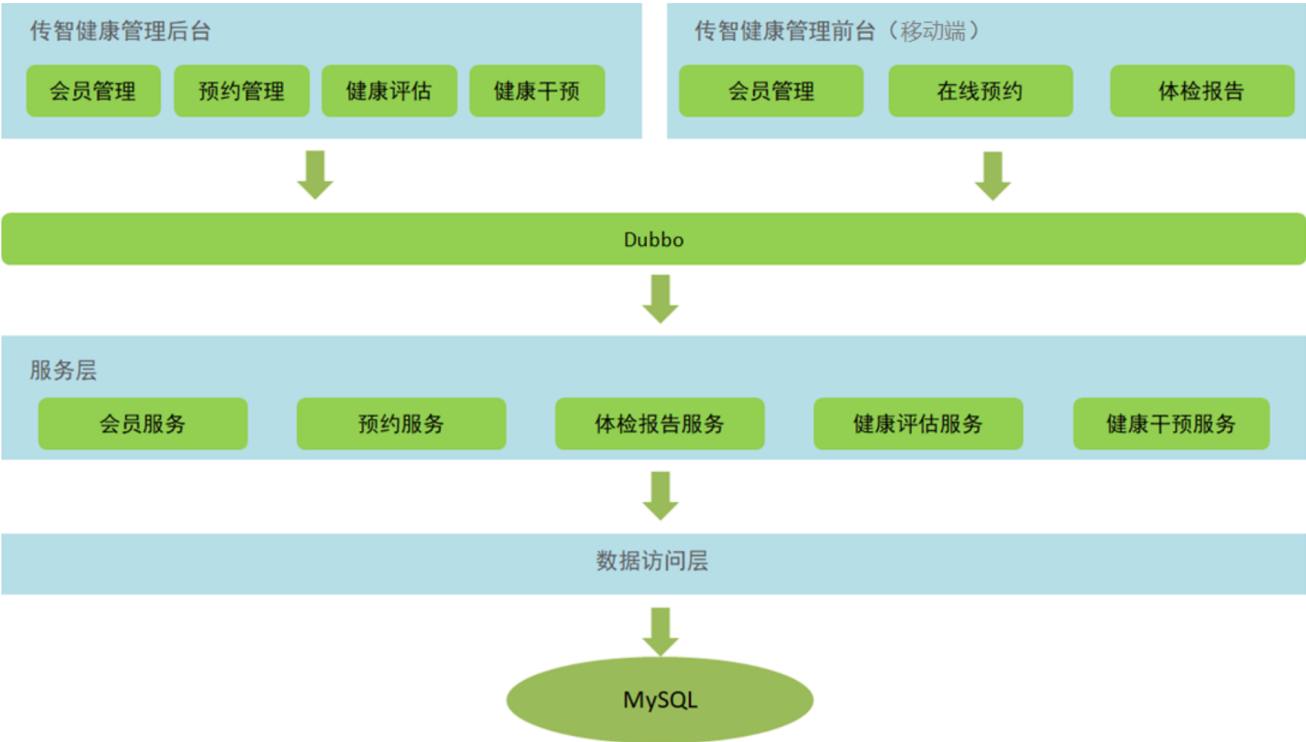
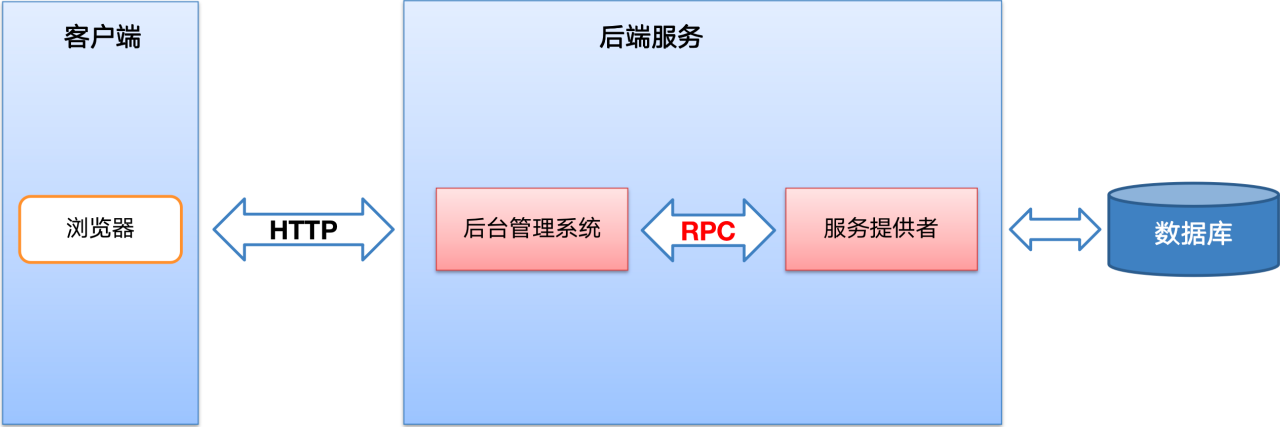


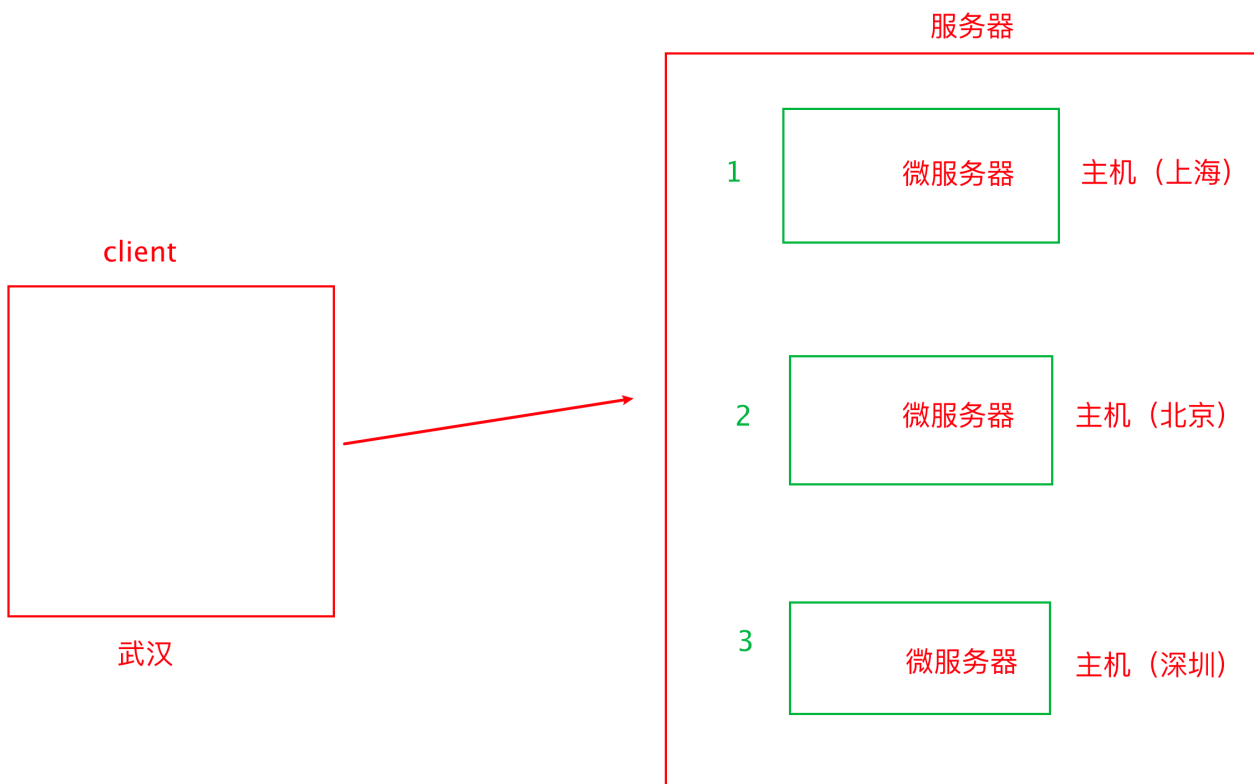
项目模块

- 会员服务、预约服务、体检报告服务、健康评估服务、健康干预服务

系统框架

- 前端：<http://mobile-health-test.itheima.net>
- 后端：<http://manager-health-test.itheima.net>





Dubbo接口测试

RPC

- 远程过程调用 (Remote Procedure Call) : 像调用本地方法一样, 调用远程方法。
- 常见的RPC框架有 Dubbo、Thrift、grpc

Dubbo

- Dubbo是一款高性能、轻量级、基于Java的开源RPC框架 (最早由阿里开源, 2018年贡献给了Apache组织)
- Dubbo接口的作用: 远程调用 java 写的方法。需要传参、获取返回值。

==查阅API文档==

RPC接口

后台管理系统:

- 会员服务
 - 根据手机号查询会员信息
 - 添加会员
 - 根据日期统计会员数
- 预约设置服务
 - 添加预约设置
 - 根据日期修改预约设置数量
- 用户服务
 - 根据用户名查询用户信息

从中获取哪些信息?

- 服务名
- 方法名
- 参数类型、返回值类型

java中 方法定义语法结构:

返回值类型 方法名 (数据类型 形参1, 数据类型 形参2,)

void: 代表没有返回值、没有参数。

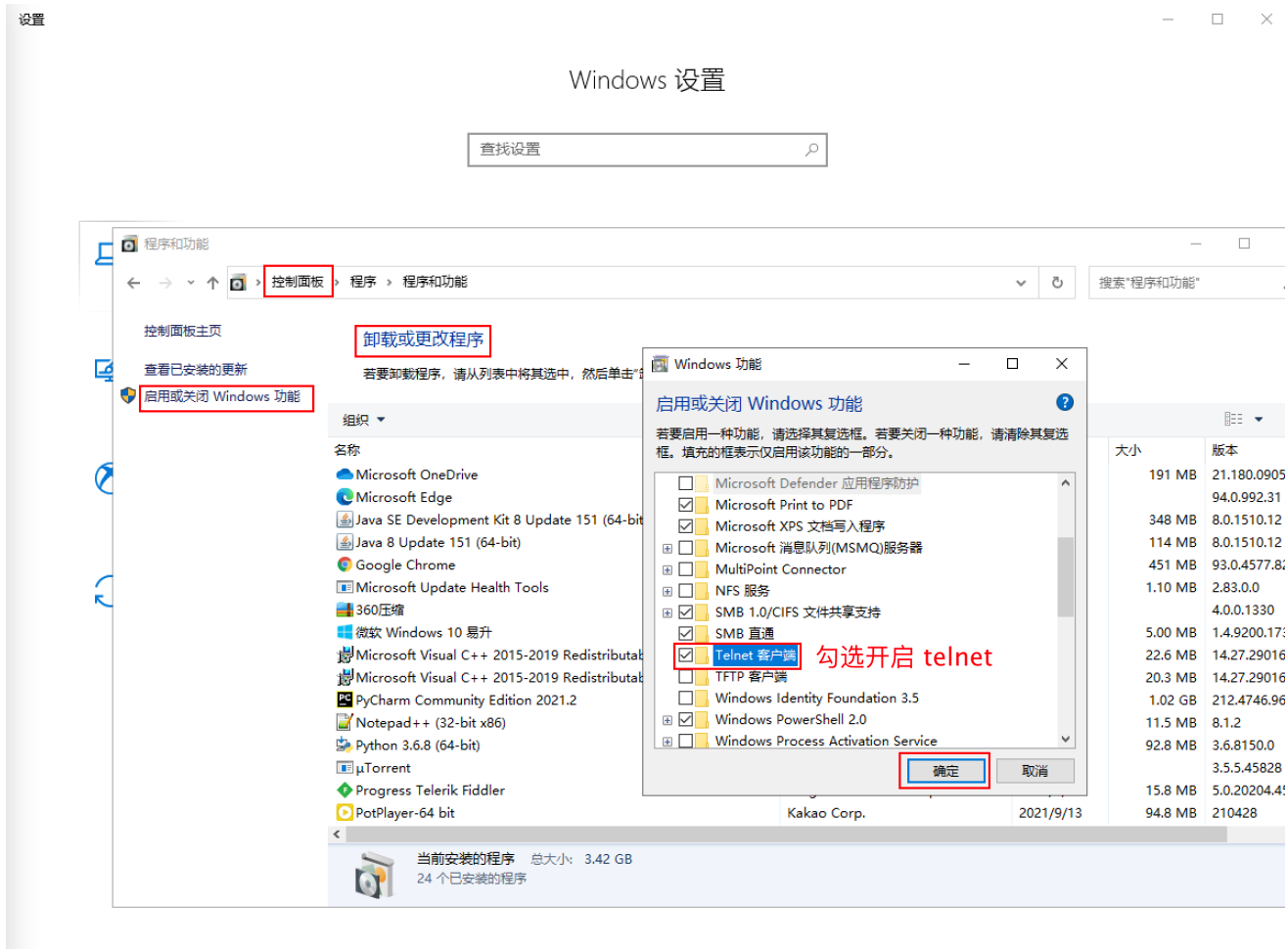
Member findByTelephone(String telephone)

返回值类型 方法名 类型 形参

ameters:

Telnet工具远程调用

启用telnet



telnet远程连接服务

连接语法: telnet IP 端口号

```
wangfei@afei_Mac ~ % telnet 211.103.136.244 6502
Trying 211.103.136.244...
Connected to 211.103.136.244.
Escape character is '^]'.
dubbo>
```

敲两次回车

```
dubbo>ls
com.itheima.service.OrderSettingService
com.itheima.service.CheckItemService
com.itheima.service.OrderService
com.itheima.service.UserService
com.itheima.service.CheckGroupService
com.itheima.service.MemberService
com.itheima.service.ReportService
com.itheima.service.SetmealService
dubbo>ls -l MemberService
java.util.List findMemberCountByMonths(java.util.List)
com.itheima.pojo.Member findByTelephone(java.lang.String)
void add(com.itheima.pojo.Member)
dubbo>???
```

项目中有哪些服务

指定的服务下，有哪些接口

内置数据类型

自定义数据类型

==telnet调用服务接口==

命令格式: `invoke 服务名.方法名 (实参)`

示例: `invoke MemberService.findByTelephone("13020210001")`

```
dubbo>invoke MemberService.findByTelephone("13020210001")
{"fileNumber":"D0003","id":3,"idCard":"210102199003073415","name":"李白",
"phoneNumber":"13020210001","regTime":1623427200000,"sex":"1"}
elapsed: 3 ms.
dubbo>
```

调用接口后的响应结果

==python借助dubbo远程调用==

Dubboclient, 封装了 telnetlib 库。telnetlib 是 python 内置模块, 可实现远程调用 Dubbo 接口。

安装dubboclient

```
pip install dubboclient
```

查验:

- 在 pip 中: `pip list` 或 `pip show dubboclient`

- 在 pycharm中: file - settings - 项目名下的 python 解释器列表。

实现步骤

1. 导包 `from dubboclient import DubboClient`
2. 创建 `DubboClient`类实例, 指定 IP 和 port
3. 使用 实例调用 `invoke()` 方法。传入: 服务名、方法名、实参(方法使用)。获取响应结果
4. 打印响应结果

会员服务 (入门)

案例1

根据手机号, 查询会员信息 (传递 普通参数)

```
dubbo> ls -l MemberService
com.itheima.pojo.Member findByTelephone(java.lang.String)
```

接口定义: `Member findByTelephone(String telephone)`

参数:

字符串格式手机号。唯一

返回值:

成功: 返回 会员的信息内容。string类型 包裹的 字典数据。

失败: 返回 `null`。string类型

实现代码:

```
# 1. 导包 from dubboclient import DubboClient
from dubboclient import DubboClient

# 2. 创建 DubboClient类实例, 指定 IP 和 port
dubboclt = DubboClient("211.103.136.244", 6502)

# 3. 使用 实例调用 invoke() 方法。传入: 服务名、方法名、实参(方法使用)。获取响应结果
resp = dubboclt.invoke("MemberService", "findByTelephone", "13020210001")

# 4. 打印响应结果
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

案例2

添加会员 (传递 对象参数)

```
dubbo> ls -l MemberService

void add(com.itheima.pojo.Member)
```

接口定义: `void add(Member member)`

参数:

1. 自定义类 做 参数, 根据接口文档, 组织 “字典” 格式数据传参
2. 给字典增加 键k: “class”, 值v: 指明 类 对应的 完整 包名和类名
如: “class”: “com.itheima.pojo.Member”
`ls -l MemberService` 可以查看完整包名和类名。
区分自定义类: 包名不以“java.”开头。一般采用: com.公司名.项目名.类名

返回值:

成功: 返回 `null`
失败: 返回 `Failed`

实现代码:

```
# 1. 导包
from dubboclient import DubboClient

# 2. 创建 dubboclient 实例
dubboclt = DubboClient("211.103.136.244", 6502)

# 准备 add 方法, 所需要的数据
info = {"id": 911, "name": "杜甫", "phoneNumber": "13048379884"}
# 如果 class 已经存在, 覆盖原有class值; 如果不存在 class, 新增一组 元素到 字典中。
info["class"] = "com.itheima.pojo.Member"

# 3. 调用 invoke 传 服务名、方法名、实参。得响应结果
resp = dubboclt.invoke("MemberService", "add", info)

# 4. 打印
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

案例3

根据日期统计会员数 (传递 字符串列表)

```
dubbo> ls -l MemberService
java.util.List findMemberCountByMonths(java.util.List)

接口定义: List<Integer> findMemberCountByMonths(List<String> months)

参数:
1. 字符串列表。用字符串表示年、月, 用“.”衔接
   如: ["2021.3", "2021.9"]

返回值:
成功: 返回列表, 对应参数设置的月份的会员数。
失败: Failed
```


实现代码：

```
# 1. 导包
from dubboclient import DubboClient

# 2. 创建 dubboclient 实例
dubboctl = DubboClient("211.103.136.244", 6502)

# 3. 用实例 调用invoke()，传入 服务名、方法名、实参。 得响应结果
months = ["2021-7"]
resp = dubboctl.invoke("MemberService", "findMemberCountByMonths", months)

# 4. 查看响应结果
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

其他模块

添加预约设置

```
dubbo> ls -l OrderSettingService
```

```
void add(java.util.List)
```

接口定义: `void add(List<OrderSetting> list)`

参数:

1. 字典列表。字典有 `orderDate` 和 `number` 两个字段。

如: `[{"orderDate": "2021-09-20 16:45:12", "number": 20}]`

2. 日期格式: "2021-09-20 16:45:12", 必须包含时分秒, 否则失败。

返回值:

成功: `null`

失败: `Failed`

实现代码：

```
# 1. 导包
from dubboclient import DubboClient

# 2. 创建 dubboclient 实例
dubboctl = DubboClient("211.103.136.244", 6502)

# 准备 add 方法, 所需要的数据
info = [{"orderDate": "2021-05-18 18:89:02", "number": 346}]
```

```
# 3. 调用 invoke 传 服务名、方法名、实参。得响应结果
resp = dubboctl.invoke("OrderSettingService", "add", info)

# 4. 打印
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

按月统计预约设置信息

```
dubbo> ls -l OrderSettingService

java.util.List getOrderSettingByMonth(java.lang.String)

接口定义: List getOrderSettingByMonth(String date)

参数:
    字符串, 如: "2021-09"

返回值:
    成功: 返回字符串类型数据, 字符串内容为列表
    失败: Failed
```

实现代码:

```
# 1. 导包
from dubboclient import DubboClient

# 2. 创建 dubboclient 实例
dubboctl = DubboClient("211.103.136.244", 6502)

# 月份
moths = "2021.02"

# 3. 调用 invoke 传 服务名、方法名、实参。得响应结果
resp = dubboctl.invoke("OrderSettingService", "getOrderSettingByMonth", moths)

# 4. 打印
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

根据日期修改预约设置数量

```
dubbo> ls -l OrderSettingService

void editNumberByDate(com.itheima.pojo.OrderSetting)
```

接口定义: `void editNumberByDate(OrderSetting orderSetting)`

参数:

1. 自定义类, 用 字典 根据接口文档组织数据
2. 需要使用 `class` 指定参数对象的类型

如: `{"orderDate": "2021-10-13 21:04:33", "number": 15, "class": "com.itheima.pojo.OrderSetting"}`

3. 日期格式为: "2021-10-13 21:04:33", 必须包含时分秒

返回值:

成功: `null`

失败: `Failed`

实现代码:

```
# 1. 导包
from dubboclient import DubboClient

# 2. 创建 dubboclient 实例
dubboctl = DubboClient("211.103.136.244", 6502)

# 日期 和 设置数据
date = {"orderDate": "2021-06-15 16:99:77", "number": 120}
date["class"] = "com.itheima.pojo.OrderSetting"

# 3. 调用 invoke 传 服务名、方法名、实参。得响应结果
resp = dubboctl.invoke("OrderSettingService", "editNumberByDate", date)

# 4. 打印
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

根据用户名查询用户信息

```
dubbo> ls -l UserService
```

```
com.itheima.pojo.User findByUsername(java.lang.String)
```

接口定义: `User findByUsername(String username)`

参数: 字符串类型, 如: `'admin'`

返回值:

用户存在: 返回用户信息

用户不存在: 返回 `null`

实现代码：

```
# 1. 导包
from dubboclient import DubboClient

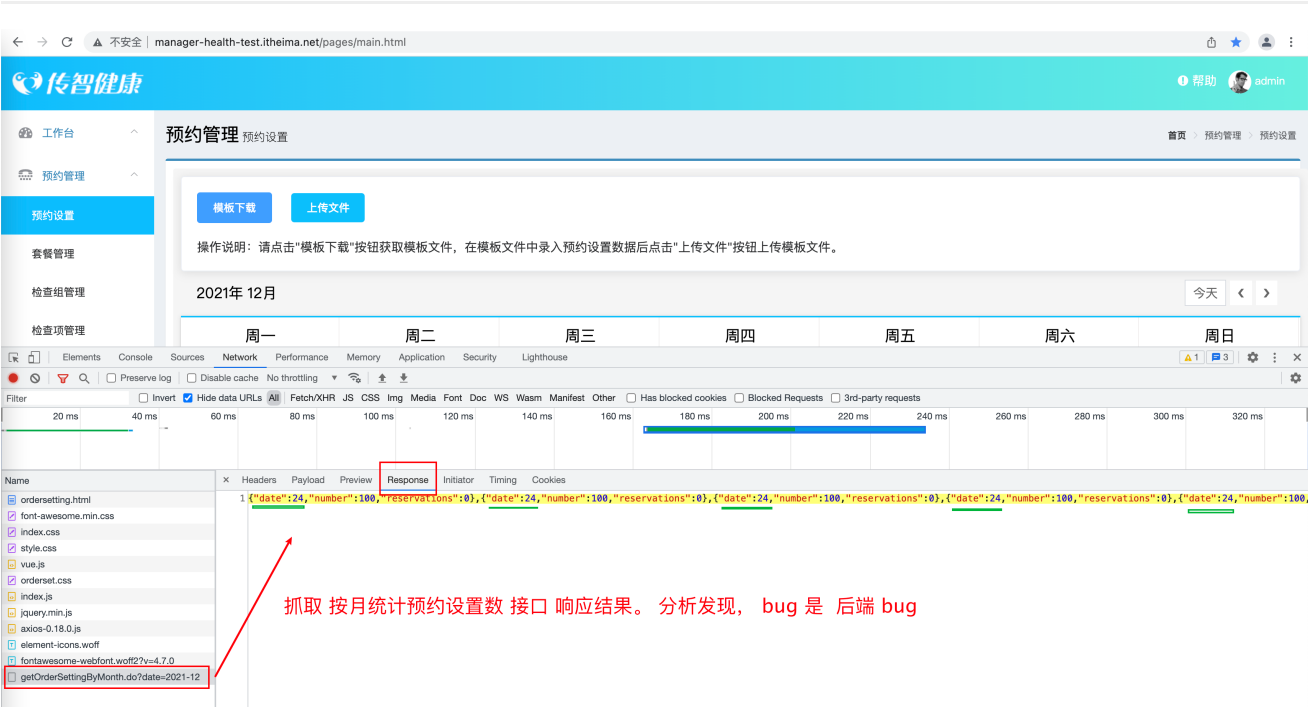
# 2. 创建 dubboclient 实例
dubboclt = DubboClient("211.103.136.244", 6502)

# 管理用户名
name = "admin"

# 3. 调用 invoke 传 服务名、方法名、实参。得响应结果
resp = dubboclt.invoke("UserService", "findByUsername", name)

# 4. 打印
print("响应结果 =", resp)
print("type(resp) =", type(resp))
```

分析bug来源



现有问题

远程调用的 7个dubbo接口 存在的问题：

- 1. 代码有 大量冗余
- 2. 测试接口时，除了要给 测试数据之外， 还需要 指定 服务名、方法名
- 3. 传参时，除了要考虑测试数据外，还要分析是否要添加 class 字段 及 对应数据。
- 4. 返回的数据类型统一为 string（不具体）

封装目标

1. 只关心：测试数据、响应结果
2. 返回的结果 分别为 不同的 具体类型。

作业

1. 完成每日反馈!!!
2. 复习本天课程内容，完成《接口测试-第11天-作业.md》中习题。