# （一）管理员端接口

## 用户登录

### 登录

项目统一才用JSON格式进行请求和数据返回，示例如下：

**url：/identity/login.do （前缀：项目部署地址+项目别名）**

**Request：post**

**Params：{**

**userName:用户名**

**password:密码**

**}**

**Return: {**

**code: 登录状态（0,1）**

**msg: 返回消息**

**}**

### 退出

**url：/identity/logout.do**

**Request：post**

**Params：{**

**userName:用户名**

**}**

**Return: {**

**code: 登录状态（0,1）**

**msg: 返回消息**

**}**

## 网站管理

### Banner上传

### 明星学员管理

#### 查询明星学员列表

#### 编辑、录入明星学员

#### 查看明星学员

#### 删除明星学员

## 师资管理（注： 导入教师信息同时需在user表添加记录）

### 教员信息

#### 查询教员信息列表

#### 编辑教员信息

#### 查看教员信息

#### 删除教员信息

### 批量导入

#### 模板下载

#### 数据校验

#### 导入进度查询

#### 数据导入

### 教员课时管理

#### 课时列表查询

#### 冻结课时

#### 解冻课时

## 预约管理

### 增加预约记录

url：/schedule/addSchedule.do

Request：post

Params：{

openId: 预约微信号（家长）ID

telNum: 联系方式（家长）

weChatNum: 微信号（家长）

studentName: 学生姓名

lessonType: 预约课时类型（小学、初中、高中）

}

Return: {

code: 插入状态（0: 成功,1: 失败）

msg: 返回消息

}

### 查询预约列表

url：/schedule/getScheduleList.do

Request：post

Params：{

telNum: 联系方式（家长-optional）

stuName: 学生姓名（optional）

}

Return: {

code: 返回状态（0: 成功,1: 失败）

msg: 返回消息

data: [{

"scheduleId":"15578901234",

"createTime":1515416388000,

"status":0,

"lessonType":1,

"weChatNum":"15578901234",

"telNum":"18711019945",

"notes":"未处理",

"openId":"aaaaaaa"

}]

}

### 更新预约条目

url：/schedule/updateSchedule.do

Request：post

Params：{

scheduleId: 预约代码

status: 预约状态

notes: 备注

}

Return: {

code: 更新状态（0: 成功,1: 失败）

msg: 返回消息

}

### 删除预约条目

url：/schedule/deleteSchedule.do

Request：post

Params：{

scheduleId: 预约代码

}

Return: {

code: 删除状态（0: 成功,1: 失败）

msg: 返回消息

}

## 订单管理

### 订单查询

url：/order/queryOrders.do

Request：post

Params：{

telNum: 联系方式（家长）

parentName: 家长姓名

studentName: 学生姓名

lessonLeftNum: 剩余课时数（需求：小于6个课时）

}

Return: {

data: [{

openId: 预约微信号（家长）ID

purchaseTime: 下单日期

lessonType: 年级

telNum: 联系方式（家长）

parentName: 家长姓名

studentName: 学生姓名

lessonLeftNum: 剩余课时数

totalLessonNum: 总购买课时数

bindTeachers: [{ //绑定的任课教师（以逗号分隔）

teacherId: 教师Id

teacherName: 教师姓名

lessonType: 绑定科目

courseName: 科目名称

}]

}]

code: 订单状态（0: 下单成功,1: 下单失败）

msg: 返回消息

}

### 录入订单

url：/order/addOrder.do

Request：post

Params：{

openId: 预约微信号（家长）ID

telNum: 联系方式（家长）

parentName: 家长姓名

studentName: 学生姓名

lessonType: 预约课时类型（小学、初中、高中）

purchaseNum: 购买课时数

}

Return: {

code: 订单状态（0: 下单成功,1: 下单失败）

msg: 返回消息

}

### 绑定任教关系

#### 老师查询（模糊查找）

url：/order/teaching/getTeacher.do

Request：post

Params：{

~~orderId : 订单Id~~

lessonType: 科目类型（年级、科目）

teacherName: 老师姓名（为空时，查询所有满足条件的老师）

}

Return: {

data: [{

teachingId: 教学任务Id

teacherId: 教师Id

teacherName: 教师姓名

lessonType: 任教科目

}]

code: 获取状态（0: 获取成功,1: 获取失败）

msg: 返回消息

}

#### 科目查询

url：/order/teaching/getCourses.do

Request：post

Params：{

lessonType: 科目类型（年级）

}

Return: {

data: [{

courseId: 科目Id

courseName: 科目名称

}]

code: 获取状态（0: 获取成功,1: 获取失败）

msg: 返回消息

}

#### 保存任教关系

url：/order/teaching/saveTeaching.do

Request：post

Params：{

orderId : 订单Id

teachingIds : 任教关系（多个任教关系以逗号分隔）

}

Return: {

code: 保存状态（0: 保存成功,1: 保存失败）

msg: 返回消息

}

### 查询老师授课详情(按月份统计)

url：/order/getMTeaching.do

Request：post

Params：{

orderId : 订单Id

teacherId: 操作类型

}

Return: {

data: [{

months: 月份（年月）

monthOfLessons：当月授课课时数

}]

code: 操作状态（0: 操作成功,1: 操作失败）

msg: 返回消息

}

### 更新订单（结算、充值）

url：/order/updateOrder.do

Request：post

Params：{

orderId : 订单Id

operateType: 操作类型（0：结算、1：充值）

operateNum: 结算/充值数量

}

Return: {

code: 操作状态（0: 操作成功,1: 操作失败）

msg: 返回消息

}

## 用户管理

### 查询用户列表

### 一键催费

### 发送补缴链接（微信）

## 投诉管理

# （二）老师端接口

## PC网站

### 老师登录管理

#### 登录

url：/teacher/ identity /login.do

**Request：post**

**Params：{**

**userName:用户名**

**password:密码**

**}**

**Return: {**

**code: 登录状态（0,1）**

**msg: 返回消息**

**}**

#### 签约状态查询

url：/teacher/domain/getSignStatus.do

Request：post

Params：{

teacherId : 老师Id

}

Return: {

signed：签约状态（0：未签约, 1： 已签约）

code: 操作状态（0: 操作成功,1: 操作失败）

msg: 返回消息

}

#### 同意签约

url：/teacher/domain/agree.do

Request：post

Params：{

teacherId : 老师Id

}

Return: {

code: 操作状态（0: 操作成功,1: 操作失败）

msg: 返回消息

}

#### 老师信息查询（个人空间）

url：/teacher/domain/getPrivateMsg.do

Request：post

Params：{

teacherId : 老师Id

}

Return: {

data: [{

}]

code: 操作状态（0: 操作成功,1: 操作失败）

msg: 返回消息

}

#### 账号（支付宝）绑定

#### 退出

**url：/teacher/identity/logout.do** （退出）

**Request：post**

**Params：{**

**userName:用户名**

**}**

**Return: {**

**code: 登录状态（0,1）**

**msg: 返回消息**

**}**

### 老师课时提现

#### 查询累计提现及课时情况

#### 查询带课记录

url：/teacher/trade/ showTRecords.do

Request：post

Params：{

teacherId : 老师Id

}

Return: {

data: [{

"parentName":"王家长",

"teacherId":"666teacher666",

memberId: 学生Id,

"orderId":"3caf39e6-add0-44d8-807f-739f43b69023",

"totalLessonNum":63,

"telNum":"18711018800",

"studentName":"333333",

"purchaseTime":1515672066000,

"lessonType":212,

"lessonLeftNum":57,

"parentId":"f39682f3-e4f2-4e4a-ab74-7558f08e4768"

}]

code: 获取状态（0: 获取成功,1: 获取失败）

msg: 返回消息

}

#### 查询反馈表

#### 确认提交入库（课时提现）

url：/teacher/trade/addTeachingDetail.do

Request：post

Params：{

orderId : 订单Id

teacherId: 老师Id

teachingId: 任教关系Id

lessonType: 提现课时类型

parentId: 家长Id

memberId: 学生Id

teachingDetails: [{

startTime: 开始时间

endTime: 结束时间

teachingDate： 上课日期

checkNum: 结算课时数（单条对应teachingNum）

}]

situations: [{ //进退步情况

}]

suggestions: [{ //建议及规划

}]

}

Return: {

code: 操作状态（0: 操作成功,1: 操作失败）

msg: 返回消息

}

### 老师提现记录查询

#### 查询提现记录列表

url：/teacher/trade/ showTRecords.do

Request：post

Params：{

teacherId : 老师Id

}

Return: {

data: [{

"parentName":"王家长",

" status": 提现状态,

"telNum":"18711018800",

"studentName":"333333",

" notes": 评价,

"lessonType": 212,

" feedback": 家长反馈,

" applyLessons": 申请提现课时数,

“actualIncome”: 实际到账课时

}]

code: 获取状态（0: 获取成功,1: 获取失败）

msg: 返回消息

}

## H5端（微信-公众号）

### 入口绑定

url：/teacher/h5/bindWechat.do (需先调用微信授权接口)

Request：post

Params：{

teacherId : 老师Id

openId: 微信openId

nickname：别名

headimgurl：头像地址

}

Return: {

code: 操作状态（0: 绑定成功,1: 绑定失败）

msg: 返回消息

}

### 微信推送提醒

#### 月初课时提现

### 其他

#### 提现课时进度

#### 微信授权

Get请求

<http://xiaoyikeji.w230.mc-test.com/xiaoyi/WechatCallbackApi>

Response:

{

openid: 微信openId

nickname: 别名

sex：性别

language：语言

city：城市

province：省份

country：国家

headimgurl：头像地址

privilege：权限

}

# 用户端接口