# **约旅平台API接口文档**

**版本**: v1.0  
**更新时间**: 2024年  
**文档说明**: 约旅平台前后端API接口对接技术文档

## **📋 目录**

• 1. 接口概述

• 1.1 基础信息

• 1.2 环境说明

• 1.3 参数类型说明

• 1.4 字段约定

• 2. 认证机制

• 2.1 JWT Token认证

• 2.2 Token获取

• 2.3 微信登录

• 2.4 刷新Token

• 3. 通用规范

• 3.1 响应格式

• 3.2 分页格式

• 4. 用户管理接口

• 5. 网红管理接口

• 6. 旅行服务接口

• 7. 订单管理接口

• 8. 支付接口

• 9. 评价系统接口

• 10. 消息通知接口

• 11. 数据统计接口

• 12. 错误码说明

• 12.1 通用错误码

• 12.2 业务错误码

• 12.3 错误处理示例

• 13. 接口调用示例

## **1. 接口概述**

### **1.1 基础信息**

• **基础URL**: https://api.yuelu.com

• **协议**: HTTPS

• **数据格式**: JSON

• **字符编码**: UTF-8

• **请求方式**: GET、POST、PUT、DELETE

### **1.2 环境说明**

**javascript**

// 开发环境

const DEV\_BASE\_URL = 'https://dev-api.yuelu.com';

// 测试环境

const TEST\_BASE\_URL = 'https://test-api.yuelu.com';

// 生产环境

const PROD\_BASE\_URL = 'https://api.yuelu.com';

### **1.3 参数类型说明**

|  |  |  |
| --- | --- | --- |
| 类型 | 说明 | 示例 |
| string | 字符串类型 | “北京市”、“13800138000” |
| integer | 整数类型 | 1、100、2001 |
| number | 数字类型（包含小数） | 299.50、4.8 |
| boolean | 布尔类型 | true、false |
| array | 数组类型 | [“美食”, “景点”]、[1, 2, 3] |
| object | 对象类型 | {“id”: 1, “name”: “张三”} |

### **1.4 字段约定**

|  |  |
| --- | --- |
| 约定 | 说明 |
| id | 主键ID，通常为integer类型 |
| \*\_id | 外键ID，如celebrity\_id、service\_id |
| \*\_at | 时间字段，格式为"YYYY-MM-DD HH:MM:SS" |
| \*\_url | URL地址字段，如avatar\_url、image\_url |
| is\_\* | 布尔字段，如is\_verified、is\_online |
| \*\_count | 计数字段，如review\_count、order\_count |

## **2. 认证机制**

### **2.1 JWT Token认证**

**请求头格式**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| Authorization | string | 是 | Bearer Token格式 | “Bearer {access\_token}” |
| Content-Type | string | 是 | 内容类型 | “application/json” |

**javascript**

// 请求头格式

{

  "Authorization": "Bearer {access\_token}",

  "Content-Type": "application/json"

}

### **2.2 Token获取**

**接口地址**: POST /api/v1/auth/login

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| phone | string | 是 | 手机号码，11位数字 | “13800138000” |
| password | string | 是 | 密码，6-20位字符 | “123456” |
| type | string | 是 | 用户类型：user-普通用户，celebrity-网红，admin-管理员 | “user” |

**请求示例**:

**javascript**

// 登录获取Token

POST /api/v1/auth/login

{

  "phone": "13800138000",

  "password": "123456",

  "type": "user"

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 登录结果对象 |
| data.access\_token | string | 访问令牌，有效期7天 |
| data.refresh\_token | string | 刷新令牌，用于获取新的访问令牌 |
| data.expires\_in | integer | 访问令牌有效期（秒） |
| data.user\_info | object | 用户信息对象 |
| data.user\_info.id | integer | 用户ID |
| data.user\_info.phone | string | 手机号码 |
| data.user\_info.nickname | string | 用户昵称 |
| data.user\_info.avatar | string | 用户头像URL |
| data.user\_info.type | string | 用户类型 |

**响应示例**:

**javascript**

// 响应

{

  "code": 200,

  "message": "登录成功",

  "data": {

    "access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

    "refresh\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",

    "expires\_in": 7200,

    "user\_info": {

      "id": 1001,

      "phone": "13800138000",

      "nickname": "旅行达人",

      "avatar": "https://cdn.yuelu.com/avatar/1001.jpg",

      "type": "user"

    }

  }

}

### **2.3 微信登录**

**接口地址**: POST /api/v1/auth/wx-login

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| code | string | 是 | 微信登录临时凭证 | “wx\_login\_code\_here” |

**请求示例**:

**javascript**

{

  "code": "wx\_login\_code\_here"

}

**响应参数**: 同普通登录响应格式

### **2.4 刷新Token**

**接口地址**: POST /api/v1/auth/refresh

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| refresh\_token | string | 是 | 刷新令牌 | “eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9…” |

**请求示例**:

**javascript**

{

  "refresh\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

**响应参数**: 同登录响应格式，返回新的access\_token

## **3. 通用规范**

### **3.1 响应格式**

**成功响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object/array | 具体数据，根据接口不同而不同 |
| timestamp | integer | 时间戳（毫秒） |

**错误响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 错误状态码 |
| message | string | 错误消息 |
| error | string | 详细错误信息 |
| timestamp | integer | 时间戳（毫秒） |

**javascript**

// 成功响应

{

  "code": 200,

  "message": "操作成功",

  "data": {}, // 具体数据

  "timestamp": 1640995200000

}

// 错误响应

{

  "code": 400,

  "message": "参数错误",

  "error": "phone字段不能为空",

  "timestamp": 1640995200000

}

### **3.2 分页格式**

**分页请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| page | integer | 否 | 页码，从1开始，默认1 | 1 |
| page\_size | integer | 否 | 每页数量，默认20，最大100 | 20 |
| sort | string | 否 | 排序字段 | “created\_at” |
| order | string | 否 | 排序方向：asc-升序，desc-降序 | “desc” |

**分页响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| list | array | 数据列表 |
| pagination | object | 分页信息对象 |
| pagination.page | integer | 当前页码 |
| pagination.page\_size | integer | 每页数量 |
| pagination.total | integer | 总记录数 |
| pagination.total\_pages | integer | 总页数 |
| pagination.has\_next | boolean | 是否有下一页 |
| pagination.has\_prev | boolean | 是否有上一页 |

**javascript**

// 分页请求参数

{

  "page": 1,

  "page\_size": 20,

  "sort": "created\_at",

  "order": "desc"

}

// 分页响应格式

{

  "code": 200,

  "message": "获取成功",

  "data": {

    "list": [],

    "pagination": {

      "page": 1,

      "page\_size": 20,

      "total": 100,

      "total\_pages": 5,

      "has\_next": true,

      "has\_prev": false

    }

  }

}

## **4. 用户管理接口**

### **4.1 用户注册**

**接口地址**: POST /api/v1/users/register

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| phone | string | 是 | 手机号码，11位数字 | “13800138000” |
| password | string | 是 | 密码，6-20位字符 | “123456” |
| sms\_code | string | 是 | 短信验证码，6位数字 | “123456” |
| nickname | string | 是 | 用户昵称，2-20个字符 | “旅行达人” |
| avatar | string | 否 | 头像URL地址 | “https://cdn.yuelu.com/avatar/default.jpg” |
| gender | integer | 否 | 性别：0-未知，1-男，2-女 | 1 |
| birthday | string | 否 | 生日，格式：YYYY-MM-DD | “1990-01-01” |
| city | string | 否 | 所在城市 | “北京市” |

**请求示例**:

**javascript**

{

  "phone": "13800138000",

  "password": "123456",

  "sms\_code": "123456",

  "nickname": "旅行达人",

  "avatar": "https://cdn.yuelu.com/avatar/default.jpg",

  "gender": 1,

  "birthday": "1990-01-01",

  "city": "北京市"

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.user\_id | integer | 用户ID |
| data.access\_token | string | 访问令牌 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "注册成功",

  "data": {

    "user\_id": 1001,

    "access\_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

  }

}

### **4.2 获取用户信息**

**接口地址**: GET /api/v1/users/profile

**请求参数**: 无（需要Token认证）

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 用户信息对象 |
| data.id | integer | 用户ID |
| data.phone | string | 手机号码 |
| data.nickname | string | 用户昵称 |
| data.avatar | string | 头像URL |
| data.gender | integer | 性别：0-未知，1-男，2-女 |
| data.birthday | string | 生日，格式：YYYY-MM-DD |
| data.city | string | 所在城市 |
| data.level | integer | 用户等级 |
| data.points | integer | 积分余额 |
| data.balance | number | 账户余额（元） |
| data.created\_at | string | 注册时间 |
| data.updated\_at | string | 最后更新时间 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "获取成功",

  "data": {

    "id": 1001,

    "phone": "13800138000",

    "nickname": "旅行达人",

    "avatar": "https://cdn.yuelu.com/avatar/1001.jpg",

    "gender": 1,

    "birthday": "1990-01-01",

    "city": "北京市",

    "level": 2,

    "points": 1500,

    "balance": 299.50,

    "created\_at": "2024-01-01 10:00:00",

    "updated\_at": "2024-01-15 15:30:00"

  }

}

### **4.3 更新用户信息**

**接口地址**: PUT /api/v1/users/profile

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| nickname | string | 否 | 用户昵称，2-20个字符 | “新昵称” |
| avatar | string | 否 | 头像URL地址 | “https://cdn.yuelu.com/avatar/new.jpg” |
| gender | integer | 否 | 性别：0-未知，1-男，2-女 | 2 |
| birthday | string | 否 | 生日，格式：YYYY-MM-DD | “1992-05-15” |
| city | string | 否 | 所在城市 | “上海市” |

**请求示例**:

**javascript**

{

  "nickname": "新昵称",

  "avatar": "https://cdn.yuelu.com/avatar/new.jpg",

  "gender": 2,

  "birthday": "1992-05-15",

  "city": "上海市"

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.updated\_fields | array | 已更新的字段列表 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "更新成功",

  "data": {

    "updated\_fields": ["nickname", "avatar", "city"]

  }

}

### **4.4 发送短信验证码**

**接口地址**: POST /api/v1/users/send-sms

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| phone | string | 是 | 手机号码，11位数字 | “13800138000” |
| type | string | 是 | 验证码类型：register-注册，login-登录，reset-重置密码 | “register” |

**请求示例**:

**javascript**

{

  "phone": "13800138000",

  "type": "register"

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.expires\_in | integer | 验证码有效期（秒） |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "验证码发送成功",

  "data": {

    "expires\_in": 300

  }

}

## **5. 网红管理接口**

### **5.1 网红列表**

**接口地址**: GET /api/v1/celebrities

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| page | integer | 否 | 页码，从1开始，默认1 | 1 |
| page\_size | integer | 否 | 每页数量，默认20，最大100 | 20 |
| city | string | 否 | 城市筛选 | “北京市” |
| category | string | 否 | 分类筛选：美食、景点、购物、文化 | “美食” |
| price\_min | integer | 否 | 最低价格（元/小时） | 100 |
| price\_max | integer | 否 | 最高价格（元/小时） | 1000 |
| sort | string | 否 | 排序字段：rating-评分，price-价格，popularity-人气 | “rating” |
| order | string | 否 | 排序方向：asc-升序，desc-降序 | “desc” |

**请求示例**:

**javascript**

GET /api/v1/celebrities?page=1&page\_size=10&city=北京市&category=美食&sort=rating&order=desc

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.list | array | 网红列表 |
| data.list[].id | integer | 网红ID |
| data.list[].nickname | string | 网红昵称 |
| data.list[].avatar | string | 头像URL |
| data.list[].cover\_image | string | 封面图片URL |
| data.list[].city | string | 所在城市 |
| data.list[].category | array | 服务分类列表 |
| data.list[].description | string | 个人简介 |
| data.list[].price\_per\_hour | number | 每小时价格（元） |
| data.list[].rating | number | 评分（1-5分） |
| data.list[].review\_count | integer | 评价数量 |
| data.list[].order\_count | integer | 订单数量 |
| data.list[].tags | array | 标签列表 |
| data.list[].is\_verified | boolean | 是否认证 |
| data.list[].is\_online | boolean | 是否在线 |
| data.list[].response\_rate | integer | 回复率（百分比） |
| data.list[].response\_time | string | 平均回复时间 |
| data.pagination | object | 分页信息 |
| data.pagination.page | integer | 当前页码 |
| data.pagination.page\_size | integer | 每页数量 |
| data.pagination.total | integer | 总记录数 |
| data.pagination.total\_pages | integer | 总页数 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "获取成功",

  "data": {

    "list": [

      {

        "id": 2001,

        "nickname": "北京美食达人",

        "avatar": "https://cdn.yuelu.com/celebrity/2001.jpg",

        "cover\_image": "https://cdn.yuelu.com/celebrity/cover/2001.jpg",

        "city": "北京市",

        "category": ["美食", "景点"],

        "description": "专业美食探店，带你发现北京隐藏美味",

        "price\_per\_hour": 299,

        "rating": 4.8,

        "review\_count": 156,

        "order\_count": 89,

        "tags": ["美食专家", "本地向导", "摄影达人"],

        "is\_verified": true,

        "is\_online": true,

        "response\_rate": 95,

        "response\_time": "5分钟内"

      }

    ],

    "pagination": {

      "page": 1,

      "page\_size": 20,

      "total": 50,

      "total\_pages": 3

    }

  }

}

### **5.2 网红详情**

**接口地址**: GET /api/v1/celebrities/{celebrity\_id}

**路径参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| celebrity\_id | integer | 是 | 网红ID | 2001 |

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 网红详情对象 |
| data.id | integer | 网红ID |
| data.nickname | string | 网红昵称 |
| data.avatar | string | 头像URL |
| data.cover\_image | string | 封面图片URL |
| data.city | string | 所在城市 |
| data.category | array | 服务分类列表 |
| data.description | string | 详细介绍 |
| data.price\_per\_hour | number | 每小时价格（元） |
| data.rating | number | 评分（1-5分） |
| data.review\_count | integer | 评价数量 |
| data.order\_count | integer | 订单数量 |
| data.tags | array | 标签列表 |
| data.is\_verified | boolean | 是否认证 |
| data.is\_online | boolean | 是否在线 |
| data.response\_rate | integer | 回复率（百分比） |
| data.response\_time | string | 平均回复时间 |
| data.gallery | array | 相册图片URL列表 |
| data.services | array | 服务项目列表 |
| data.services[].id | integer | 服务ID |
| data.services[].name | string | 服务名称 |
| data.services[].description | string | 服务描述 |
| data.services[].duration | integer | 服务时长（小时） |
| data.services[].price | number | 服务价格（元） |
| data.services[].includes | array | 包含项目列表 |
| data.schedule | object | 可预约时间，key为日期，value为时间段数组 |
| data.reviews | array | 最新评价列表 |
| data.reviews[].id | integer | 评价ID |
| data.reviews[].user\_nickname | string | 用户昵称 |
| data.reviews[].user\_avatar | string | 用户头像 |
| data.reviews[].rating | integer | 评分（1-5分） |
| data.reviews[].content | string | 评价内容 |
| data.reviews[].images | array | 评价图片URL列表 |
| data.reviews[].created\_at | string | 评价时间 |

### **5.3 网红服务项目**

**接口地址**: GET /api/v1/celebrities/{celebrity\_id}/services

**路径参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| celebrity\_id | integer | 是 | 网红ID | 2001 |

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.list | array | 服务项目列表 |
| data.list[].id | integer | 服务ID |
| data.list[].name | string | 服务名称 |
| data.list[].description | string | 服务描述 |
| data.list[].duration | integer | 服务时长（小时） |
| data.list[].price | number | 服务价格（元） |
| data.list[].original\_price | number | 原价（元） |
| data.list[].discount | number | 折扣（0-1之间） |
| data.list[].includes | array | 包含项目列表 |
| data.list[].excludes | array | 不包含项目列表 |
| data.list[].max\_people | integer | 最大人数 |
| data.list[].category | string | 服务分类 |
| data.list[].difficulty | string | 难度等级：简单、中等、困难 |
| data.list[].images | array | 服务图片URL列表 |
| data.list[].itinerary | array | 行程安排列表 |
| data.list[].itinerary[].time | string | 时间段 |
| data.list[].itinerary[].location | string | 地点 |
| data.list[].itinerary[].activity | string | 活动内容 |
| data.list[].itinerary[].description | string | 活动描述 |
| data.list[].is\_available | boolean | 是否可预订 |
| data.list[].booking\_notice | string | 预订须知 |

## **6. 旅行服务接口**

### **6.1 创建预订**

**接口地址**: POST /api/v1/bookings

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| celebrity\_id | integer | 是 | 网红ID | 2001 |
| service\_id | integer | 是 | 服务项目ID | 3001 |
| booking\_date | string | 是 | 预订日期，格式：YYYY-MM-DD | “2024-01-25” |
| booking\_time | string | 是 | 预订时间段，格式：HH:MM-HH:MM | “09:00-13:00” |
| people\_count | integer | 是 | 参与人数，最小1人 | 2 |
| contact\_name | string | 是 | 联系人姓名 | “张三” |
| contact\_phone | string | 是 | 联系人手机号 | “13800138000” |
| special\_requirements | string | 否 | 特殊要求说明 | “希望多介绍一些历史文化” |
| coupon\_id | integer | 否 | 优惠券ID | 5001 |

**请求示例**:

**javascript**

{

  "celebrity\_id": 2001,

  "service\_id": 3001,

  "booking\_date": "2024-01-25",

  "booking\_time": "09:00-13:00",

  "people\_count": 2,

  "contact\_name": "张三",

  "contact\_phone": "13800138000",

  "special\_requirements": "希望多介绍一些历史文化",

  "coupon\_id": 5001

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 预订信息对象 |
| data.booking\_id | string | 预订单号 |
| data.status | string | 预订状态：pending\_payment-待支付 |
| data.total\_amount | number | 总金额（元） |
| data.discount\_amount | number | 优惠金额（元） |
| data.final\_amount | number | 最终支付金额（元） |
| data.payment\_deadline | string | 支付截止时间 |
| data.booking\_details | object | 预订详情对象 |
| data.booking\_details.celebrity\_name | string | 网红姓名 |
| data.booking\_details.service\_name | string | 服务名称 |
| data.booking\_details.booking\_date | string | 预订日期 |
| data.booking\_details.booking\_time | string | 预订时间 |
| data.booking\_details.people\_count | integer | 参与人数 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "预订成功",

  "data": {

    "booking\_id": "YL202401250001",

    "status": "pending\_payment",

    "total\_amount": 1198.00,

    "discount\_amount": 100.00,

    "final\_amount": 1098.00,

    "payment\_deadline": "2024-01-20 18:00:00",

    "booking\_details": {

      "celebrity\_name": "北京美食达人",

      "service\_name": "美食探店半日游",

      "booking\_date": "2024-01-25",

      "booking\_time": "09:00-13:00",

      "people\_count": 2

    }

  }

}

### **6.2 预订列表**

**接口地址**: GET /api/v1/bookings

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| page | integer | 否 | 页码，从1开始，默认1 | 1 |
| page\_size | integer | 否 | 每页数量，默认20，最大100 | 20 |
| status | string | 否 | 状态筛选：all-全部，pending\_payment-待支付，confirmed-已确认，in\_progress-进行中，completed-已完成，cancelled-已取消 | “all” |
| date\_from | string | 否 | 开始日期，格式：YYYY-MM-DD | “2024-01-01” |
| date\_to | string | 否 | 结束日期，格式：YYYY-MM-DD | “2024-01-31” |

**请求示例**:

**javascript**

GET /api/v1/bookings?page=1&page\_size=20&status=all&date\_from=2024-01-01&date\_to=2024-01-31

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.list | array | 预订列表 |
| data.list[].booking\_id | string | 预订单号 |
| data.list[].status | string | 预订状态 |
| data.list[].status\_text | string | 状态文本描述 |
| data.list[].celebrity\_info | object | 网红信息对象 |
| data.list[].celebrity\_info.id | integer | 网红ID |
| data.list[].celebrity\_info.nickname | string | 网红昵称 |
| data.list[].celebrity\_info.avatar | string | 网红头像 |
| data.list[].service\_info | object | 服务信息对象 |
| data.list[].service\_info.id | integer | 服务ID |
| data.list[].service\_info.name | string | 服务名称 |
| data.list[].service\_info.image | string | 服务图片 |
| data.list[].booking\_date | string | 预订日期 |
| data.list[].booking\_time | string | 预订时间 |
| data.list[].people\_count | integer | 参与人数 |
| data.list[].total\_amount | number | 总金额（元） |
| data.list[].created\_at | string | 创建时间 |
| data.list[].can\_cancel | boolean | 是否可取消 |
| data.list[].can\_reschedule | boolean | 是否可改期 |
| data.list[].can\_review | boolean | 是否可评价 |
| data.pagination | object | 分页信息 |

### **6.3 预订详情**

**接口地址**: GET /api/v1/bookings/{booking\_id}

**路径参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| booking\_id | string | 是 | 预订单号 | “YL202401250001” |

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 预订详情对象 |
| data.booking\_id | string | 预订单号 |
| data.status | string | 预订状态 |
| data.status\_text | string | 状态文本描述 |
| data.celebrity\_info | object | 网红信息（确认后显示联系方式） |
| data.celebrity\_info.id | integer | 网红ID |
| data.celebrity\_info.nickname | string | 网红昵称 |
| data.celebrity\_info.avatar | string | 网红头像 |
| data.celebrity\_info.phone | string | 网红手机号（仅确认后显示） |
| data.celebrity\_info.wechat | string | 网红微信号（仅确认后显示） |
| data.service\_info | object | 服务信息对象 |
| data.booking\_details | object | 预订详情对象 |
| data.booking\_details.booking\_date | string | 预订日期 |
| data.booking\_details.booking\_time | string | 预订时间 |
| data.booking\_details.people\_count | integer | 参与人数 |
| data.booking\_details.contact\_name | string | 联系人姓名 |
| data.booking\_details.contact\_phone | string | 联系人手机 |
| data.booking\_details.special\_requirements | string | 特殊要求 |
| data.booking\_details.meeting\_point | string | 集合地点 |
| data.booking\_details.meeting\_time | string | 集合时间 |
| data.payment\_info | object | 支付信息对象 |
| data.payment\_info.total\_amount | number | 总金额（元） |
| data.payment\_info.discount\_amount | number | 优惠金额（元） |
| data.payment\_info.final\_amount | number | 最终金额（元） |
| data.payment\_info.payment\_method | string | 支付方式 |
| data.payment\_info.payment\_time | string | 支付时间 |
| data.payment\_info.transaction\_id | string | 交易流水号 |
| data.timeline | array | 订单时间线 |
| data.timeline[].status | string | 状态 |
| data.timeline[].status\_text | string | 状态描述 |
| data.timeline[].time | string | 时间 |
| data.can\_cancel | boolean | 是否可取消 |
| data.can\_reschedule | boolean | 是否可改期 |
| data.can\_review | boolean | 是否可评价 |
| data.cancellation\_policy | string | 取消政策说明 |

## **7. 订单管理接口**

### **7.1 取消预订**

**javascript**

POST /api/v1/bookings/{booking\_id}/cancel

// 请求参数

{

  "reason": "行程变更", // 取消原因

  "description": "临时有事无法出行" // 详细说明

}

// 响应

{

  "code": 200,

  "message": "取消成功",

  "data": {

    "refund\_amount": 1098.00, // 退款金额

    "refund\_method": "original", // 退款方式：original(原路退回)、balance(余额)

    "refund\_time": "3-5个工作日",

    "cancellation\_fee": 0.00 // 取消手续费

  }

}

### **7.2 改期预订**

**javascript**

POST /api/v1/bookings/{booking\_id}/reschedule

// 请求参数

{

  "new\_date": "2024-01-28",

  "new\_time": "14:00-18:00",

  "reason": "时间冲突"

}

// 响应

{

  "code": 200,

  "message": "改期申请已提交",

  "data": {

    "reschedule\_id": "RS202401200001",

    "status": "pending", // 待网红确认

    "new\_booking\_details": {

      "booking\_date": "2024-01-28",

      "booking\_time": "14:00-18:00"

    },

    "additional\_fee": 0.00 // 额外费用

  }

}

## **8. 支付接口**

### **8.1 创建支付订单**

**接口地址**: POST /api/v1/payments/create

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| booking\_id | string | 是 | 预订单号 | “YL202401250001” |
| payment\_method | string | 是 | 支付方式：wechat\_pay-微信支付，alipay-支付宝，balance-余额支付 | “wechat\_pay” |
| return\_url | string | 否 | 支付成功回调URL | “https://app.yuelu.com/payment/success” |
| client\_type | string | 是 | 客户端类型：miniprogram-小程序，h5-网页，app-APP | “miniprogram” |

**请求示例**:

**javascript**

{

  "booking\_id": "YL202401250001",

  "payment\_method": "wechat\_pay",

  "return\_url": "https://app.yuelu.com/payment/success",

  "client\_type": "miniprogram"

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 支付信息对象 |
| data.payment\_id | string | 支付订单ID |
| data.payment\_params | object | 支付参数对象（根据支付方式不同而不同） |
| data.payment\_params.timeStamp | string | 时间戳（微信支付） |
| data.payment\_params.nonceStr | string | 随机字符串（微信支付） |
| data.payment\_params.package | string | 预支付交易会话标识（微信支付） |
| data.payment\_params.signType | string | 签名类型（微信支付） |
| data.payment\_params.paySign | string | 支付签名（微信支付） |
| data.expires\_at | string | 支付过期时间 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "支付订单创建成功",

  "data": {

    "payment\_id": "PAY202401200001",

    "payment\_params": {

      "timeStamp": "1640995200",

      "nonceStr": "abc123",

      "package": "prepay\_id=wx20240120164500001",

      "signType": "RSA",

      "paySign": "signature"

    },

    "expires\_at": "2024-01-20 18:00:00"

  }

}

### **8.2 支付状态查询**

**接口地址**: GET /api/v1/payments/{payment\_id}/status

**路径参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| payment\_id | string | 是 | 支付订单ID | “PAY202401200001” |

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 支付状态对象 |
| data.payment\_id | string | 支付订单ID |
| data.booking\_id | string | 预订单号 |
| data.status | string | 支付状态：pending-待支付，paid-已支付，failed-支付失败，cancelled-已取消 |
| data.amount | number | 支付金额（元） |
| data.payment\_method | string | 支付方式 |
| data.transaction\_id | string | 第三方交易流水号 |
| data.paid\_at | string | 支付完成时间 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "查询成功",

  "data": {

    "payment\_id": "PAY202401200001",

    "booking\_id": "YL202401250001",

    "status": "paid",

    "amount": 1098.00,

    "payment\_method": "wechat\_pay",

    "transaction\_id": "wx20240120164500001",

    "paid\_at": "2024-01-20 16:45:00"

  }

}

## **9. 评价系统接口**

### **9.1 提交评价**

**接口地址**: POST /api/v1/reviews

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| booking\_id | string | 是 | 预订单号 | “YL202401250001” |
| celebrity\_id | integer | 是 | 网红ID | 2001 |
| service\_id | integer | 是 | 服务项目ID | 3001 |
| rating | integer | 是 | 评分，1-5星 | 5 |
| content | string | 是 | 评价内容，10-500字符 | “非常棒的体验，网红很专业，推荐的美食都很好吃！” |
| images | array | 否 | 评价图片URL列表，最多9张 | [“https://cdn.yuelu.com/review/upload/1.jpg”] |
| tags | array | 否 | 评价标签列表 | [“专业”, “热情”, “准时”] |

**请求示例**:

**javascript**

{

  "booking\_id": "YL202401250001",

  "celebrity\_id": 2001,

  "service\_id": 3001,

  "rating": 5,

  "content": "非常棒的体验，网红很专业，推荐的美食都很好吃！",

  "images": [

    "https://cdn.yuelu.com/review/upload/1.jpg",

    "https://cdn.yuelu.com/review/upload/2.jpg"

  ],

  "tags": ["专业", "热情", "准时"]

}

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 评价结果对象 |
| data.review\_id | integer | 评价ID |
| data.points\_earned | integer | 获得的积分 |

**响应示例**:

**javascript**

{

  "code": 200,

  "message": "评价提交成功",

  "data": {

    "review\_id": 4001,

    "points\_earned": 50

  }

}

### **9.2 评价列表**

**接口地址**: GET /api/v1/reviews

**请求参数**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 | 示例值 |
| celebrity\_id | integer | 否 | 网红ID，按网红筛选 | 2001 |
| service\_id | integer | 否 | 服务项目ID，按服务筛选 | 3001 |
| rating | integer | 否 | 评分筛选，1-5星 | 5 |
| page | integer | 否 | 页码，从1开始，默认1 | 1 |
| page\_size | integer | 否 | 每页数量，默认20，最大100 | 20 |

**请求示例**:

**javascript**

GET /api/v1/reviews?celebrity\_id=2001&service\_id=3001&rating=5&page=1&page\_size=20

**响应参数**:

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| code | integer | 状态码，200表示成功 |
| message | string | 响应消息 |
| data | object | 响应数据 |
| data.list | array | 评价列表 |
| data.list[].id | integer | 评价ID |
| data.list[].user\_info | object | 用户信息对象 |
| data.list[].user\_info.nickname | string | 用户昵称 |
| data.list[].user\_info.avatar | string | 用户头像 |
| data.list[].user\_info.level | integer | 用户等级 |
| data.list[].celebrity\_info | object | 网红信息对象 |
| data.list[].celebrity\_info.id | integer | 网红ID |
| data.list[].celebrity\_info.nickname | string | 网红昵称 |
| data.list[].service\_info | object | 服务信息对象 |
| data.list[].service\_info.id | integer | 服务ID |
| data.list[].service\_info.name | string | 服务名称 |
| data.list[].rating | integer | 评分（1-5星） |
| data.list[].content | string | 评价内容 |
| data.list[].images | array | 评价图片URL列表 |
| data.list[].tags | array | 评价标签列表 |
| data.list[].created\_at | string | 评价时间 |
| data.list[].helpful\_count | integer | 有用数 |
| data.pagination | object | 分页信息 |
| data.statistics | object | 评价统计信息 |
| data.statistics.average\_rating | number | 平均评分 |
| data.statistics.rating\_distribution | object | 评分分布，key为评分，value为数量 |

## **10. 消息通知接口**

### **10.1 消息列表**

**javascript**

GET /api/v1/notifications

// 请求参数

{

  "type": "all", // all、booking、payment、system

  "status": "all", // all、unread、read

  "page": 1,

  "page\_size": 20

}

// 响应

{

  "code": 200,

  "message": "获取成功",

  "data": {

    "list": [

      {

        "id": 6001,

        "type": "booking",

        "title": "预订确认通知",

        "content": "您的预订"美食探店半日游"已被网红确认",

        "data": { // 附加数据

          "booking\_id": "YL202401250001",

          "action": "view\_booking"

        },

        "is\_read": false,

        "created\_at": "2024-01-20 17:20:00"

      }

    ],

    "pagination": {

      "page": 1,

      "page\_size": 20,

      "total": 25,

      "total\_pages": 2

    },

    "unread\_count": 5 // 未读数量

  }

}

### **10.2 标记已读**

**javascript**

POST /api/v1/notifications/mark-read

// 请求参数

{

  "notification\_ids": [6001, 6002], // 消息ID数组，空数组表示全部标记已读

}

// 响应

{

  "code": 200,

  "message": "标记成功",

  "data": {

    "marked\_count": 2

  }

}

## **11. 数据统计接口**

### **11.1 首页统计数据**

**javascript**

GET /api/v1/statistics/dashboard

// 响应

{

  "code": 200,

  "message": "获取成功",

  "data": {

    "user\_stats": {

      "total\_bookings": 15, // 总预订数

      "completed\_bookings": 12, // 已完成预订

      "total\_spent": 8500.00, // 总消费金额

      "points\_balance": 1500, // 积分余额

      "level": 2, // 用户等级

      "next\_level\_points": 500 // 升级所需积分

    },

    "recent\_bookings": [ // 最近预订

      {

        "booking\_id": "YL202401250001",

        "celebrity\_name": "北京美食达人",

        "service\_name": "美食探店半日游",

        "booking\_date": "2024-01-25",

        "status": "confirmed"

      }

    ],

    "recommended\_celebrities": [ // 推荐网红

      {

        "id": 2002,

        "nickname": "上海购物达人",

        "avatar": "https://cdn.yuelu.com/celebrity/2002.jpg",

        "rating": 4.9,

        "price\_per\_hour": 399

      }

    ]

  }

}

## **12. 错误码说明**

### **12.1 通用错误码**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 200 | 操作成功 | - |
| 400 | 请求参数错误 | 检查请求参数格式和必填字段 |
| 401 | 未授权，请先登录 | 检查Token是否正确或已过期 |
| 403 | 权限不足 | 确认用户权限或联系管理员 |
| 404 | 资源不存在 | 检查请求的资源ID是否正确 |
| 409 | 资源冲突 | 检查是否存在重复操作 |
| 422 | 请求参数验证失败 | 检查参数类型和格式 |
| 429 | 请求过于频繁 | 降低请求频率，稍后重试 |
| 500 | 服务器内部错误 | 联系技术支持 |
| 502 | 网关错误 | 稍后重试或联系技术支持 |
| 503 | 服务暂不可用 | 稍后重试 |
| 504 | 网关超时 | 稍后重试 |

### **12.2 业务错误码**

#### **用户相关错误码 (1xxx)**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 1001 | 手机号已注册 | 使用其他手机号或直接登录 |
| 1002 | 验证码错误 | 重新获取验证码 |
| 1003 | 密码错误 | 检查密码是否正确 |
| 1004 | 账号已被禁用 | 联系客服处理 |

#### **网红相关错误码 (2xxx)**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 2001 | 网红不存在 | 检查网红ID是否正确 |
| 2002 | 网红已下线 | 选择其他在线网红 |
| 2003 | 网红服务不可用 | 选择其他可用服务 |

#### **预订相关错误码 (3xxx)**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 3001 | 预订时间不可用 | 选择其他可用时间 |
| 3002 | 预订人数超限 | 减少预订人数或分批预订 |
| 3003 | 预订已取消 | 重新创建预订 |
| 3004 | 预订不可修改 | 联系客服或网红协商 |

#### **支付相关错误码 (4xxx)**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 4001 | 支付失败 | 检查支付方式或重新支付 |
| 4002 | 余额不足 | 充值后重新支付 |
| 4003 | 支付已过期 | 重新创建支付订单 |
| 4004 | 重复支付 | 检查订单状态，避免重复操作 |

#### **评价相关错误码 (5xxx)**

|  |  |  |
| --- | --- | --- |
| 错误码 | 说明 | 解决方案 |
| 5001 | 已评价过 | 每个订单只能评价一次 |
| 5002 | 评价内容违规 | 修改评价内容，避免违规词汇 |
| 5003 | 评价时间已过期 | 评价时间已超过限制 |

### **12.3 错误处理示例**

**javascript**

// 错误处理示例

try {

  const result = await api.createBooking(bookingData);

  console.log('预订成功:', result);

} catch (error) {

  switch (error.code) {

    case 1003:

      alert('密码错误，请重新输入');

      break;

    case 3001:

      alert('该时间段不可预订，请选择其他时间');

      break;

    case 4002:

      alert('余额不足，请先充值');

      break;

    default:

      alert(error.message || '操作失败，请稍后重试');

  }

}

## **13. 接口调用示例**

### **13.1 JavaScript示例**

**javascript**

// 封装API请求类

class YueluAPI {

  constructor(baseURL = 'https://api.yuelu.com') {

    this.baseURL = baseURL;

    this.token = localStorage.getItem('access\_token');

  }

  // 设置Token

  setToken(token) {

    this.token = token;

    localStorage.setItem('access\_token', token);

  }

  // 通用请求方法

  async request(method, url, data = null) {

    const config = {

      method,

      headers: {

        'Content-Type': 'application/json',

      }

    };

    // 添加认证头

    if (this.token) {

      config.headers.Authorization = `Bearer ${this.token}`;

    }

    // 添加请求体

    if (data && (method === 'POST' || method === 'PUT')) {

      config.body = JSON.stringify(data);

    }

    try {

      const response = await fetch(`${this.baseURL}${url}`, config);

      const result = await response.json();

      if (result.code === 200) {

        return result.data;

      } else {

        throw new Error(result.message);

      }

    } catch (error) {

      console.error('API请求失败:', error);

      throw error;

    }

  }

  // 用户登录

  async login(phone, password, type = 'user') {

    const data = await this.request('POST', '/api/v1/auth/login', {

      phone, password, type

    });

    this.setToken(data.access\_token);

    return data;

  }

  // 获取网红列表

  async getCelebrities(params = {}) {

    const query = new URLSearchParams(params).toString();

    return this.request('GET', `/api/v1/celebrities?${query}`);

  }

  // 创建预订

  async createBooking(bookingData) {

    return this.request('POST', '/api/v1/bookings', bookingData);

  }

  // 获取预订列表

  async getBookings(params = {}) {

    const query = new URLSearchParams(params).toString();

    return this.request('GET', `/api/v1/bookings?${query}`);

  }

}

// 使用示例

const api = new YueluAPI();

// 登录

try {

  const loginResult = await api.login('13800138000', '123456');

  console.log('登录成功:', loginResult);

} catch (error) {

  console.error('登录失败:', error.message);

}

// 获取网红列表

try {

  const celebrities = await api.getCelebrities({

    city: '北京市',

    category: '美食',

    page: 1,

    page\_size: 10

  });

  console.log('网红列表:', celebrities);

} catch (error) {

  console.error('获取失败:', error.message);

}

### **13.2 微信小程序示例**

**javascript**

// 微信小程序API封装

class WxYueluAPI {

  constructor() {

    this.baseURL = 'https://api.yuelu.com';

    this.token = wx.getStorageSync('access\_token');

  }

  // 通用请求方法

  request(options) {

    return new Promise((resolve, reject) => {

      wx.request({

        url: `${this.baseURL}${options.url}`,

        method: options.method || 'GET',

        data: options.data || {},

        header: {

          'Content-Type': 'application/json',

          'Authorization': this.token ? `Bearer ${this.token}` : ''

        },

        success: (res) => {

          if (res.data.code === 200) {

            resolve(res.data.data);

          } else {

            wx.showToast({

              title: res.data.message,

              icon: 'none'

            });

            reject(new Error(res.data.message));

          }

        },

        fail: (error) => {

          wx.showToast({

            title: '网络请求失败',

            icon: 'none'

          });

          reject(error);

        }

      });

    });

  }

  // 微信登录

  async wxLogin() {

    try {

      // 获取微信登录code

      const loginRes = await new Promise((resolve, reject) => {

        wx.login({

          success: resolve,

          fail: reject

        });

      });

      // 发送code到后端

      const data = await this.request({

        url: '/api/v1/auth/wx-login',

        method: 'POST',

        data: {

          code: loginRes.code

        }

      });

      // 保存token

      this.token = data.access\_token;

      wx.setStorageSync('access\_token', this.token);

      return data;

    } catch (error) {

      console.error('微信登录失败:', error);

      throw error;

    }

  }

  // 微信支付

  async wxPay(bookingId) {

    try {

      // 创建支付订单

      const paymentData = await this.request({

        url: '/api/v1/payments/create',

        method: 'POST',

        data: {

          booking\_id: bookingId,

          payment\_method: 'wechat\_pay',

          client\_type: 'miniprogram'

        }

      });

      // 调起微信支付

      await new Promise((resolve, reject) => {

        wx.requestPayment({

          ...paymentData.payment\_params,

          success: resolve,

          fail: reject

        });

      });

      return { success: true };

    } catch (error) {

      console.error('支付失败:', error);

      throw error;

    }

  }

}

// 使用示例

const wxApi = new WxYueluAPI();

// 页面中使用

Page({

  data: {

    celebrities: []

  },

  async onLoad() {

    try {

      // 微信登录

      await wxApi.wxLogin();

      // 获取网红列表

      const celebrities = await wxApi.request({

        url: '/api/v1/celebrities',

        data: {

          city: '北京市',

          page: 1,

          page\_size: 10

        }

      });

      this.setData({

        celebrities: celebrities.list

      });

    } catch (error) {

      console.error('初始化失败:', error);

    }

  },

  // 预订服务

  async bookService(e) {

    const { celebrityId, serviceId } = e.currentTarget.dataset;

    try {

      const booking = await wxApi.request({

        url: '/api/v1/bookings',

        method: 'POST',

        data: {

          celebrity\_id: celebrityId,

          service\_id: serviceId,

          booking\_date: '2024-01-25',

          booking\_time: '09:00-13:00',

          people\_count: 2,

          contact\_name: '张三',

          contact\_phone: '13800138000'

        }

      });

      // 跳转到支付页面

      wx.navigateTo({

        url: `/pages/payment/payment?bookingId=${booking.booking\_id}`

      });

    } catch (error) {

      console.error('预订失败:', error);

    }

  }

});

## **📝 更新日志**

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 更新内容 |
| v1.0 | 2024-01-01 | 初始版本，包含核心API接口 |

## **📞 技术支持**

如有接口对接问题，请联系：

• **技术支持邮箱**: tech@yuelu.com

• **开发者QQ群**: 123456789

• **技术文档**: https://docs.yuelu.com

**重要提醒**:

1. 所有接口都需要HTTPS协议访问

2. 请妥善保管access\_token，避免泄露

3. 建议实现token自动刷新机制

4. 生产环境请使用正式域名和证书