

# 个人课表系统 - API 接口文档 (v3.0)

## 一、概述

本文档详细描述了个人课表管理系统的后端 API 接口，涵盖用户认证、学期管理、课程管理、管理员功能和导入功能。所有接口均基于 RESTful 风格设计，数据传输格式为 JSON。

### 项目最新特性：

- ✓ 课程标签功能（必修/选修）
- ✓ 用户权限管理（ADMIN/USER/TRIAL）
- ✓ JWT 认证与状态持久化
- ✓ 前端多视图支持（周视图/日视图/学期视图）
- ✓ Excel 批量导入

## 技术规范

- 根路径:** /api
- 认证:** 除用户注册和登录接口外，所有接口均需在请求头中携带 `Authorization: Bearer <token>` 进行 JWT 认证
- 时间格式:** 所有时间字段均为 ISO 8601 格式，如 `"2025-06-23T11:30:00"`
- 命名约定:** JSON 字段采用小驼峰命名法 (camelCase)

## 通用响应结构

```
{
  "code": 200,           // 响应码，200 代表成功，其他代表失败
  "message": "具体消息", // 提示信息
  "data": { ... }       // 返回的数据，成功时为具体数据，失败时为 null
}
```

## 开发测试建议

为了确保API开发的循序渐进，建议按以下顺序进行测试：

- 基础认证 → 用户注册/登录 → JWT验证

- 2. 学期管理 → 创建学期 → 获取学期列表
- 3. 课程管理 → 添加课程（含标签） → 获取课程列表
- 4. 高级功能 → 课程搜索 → Excel导入
- 5. 管理功能 → 用户管理 → 权限控制

## 二、用户认证接口 (Auth)

### 2.1 用户注册

#### 基本信息

- 请求路径： POST /api/auth/register
- 接口描述：创建新用户，默认为 USER 角色
- 前置条件：无

#### 请求参数

参数格式： application/json

参数名	类型	是否必须	验证规则	备注
username	string	必须	长度3-50，字母数字	用户名
password	string	必须	长度6-255	密码
email	string	必须	邮箱格式	电子邮箱

请求示例：

```
{
  "username": "newuser",
  "password": "Password123!",
  "email": "newuser@example.com"
}
```

#### 响应数据

成功响应 (201 Created):

```
{
  "code": 201,
  "message": "用户注册成功",
  "data": {
    "userId": 1,
    "username": "newuser"
  }
}
```

失败响应 (400 Bad Request):

```
{
  "code": 400,
  "message": "用户名或邮箱已存在",
  "data": null
}
```

测试步骤

- 1. 使用Postman发送注册请求
- 2. 验证用户名/邮箱唯一性
- 3. 检查密码加密存储
- 4. 确认默认角色为USER

2.2 用户登录

基本信息

- 请求路径： POST /api/auth/login
- 接口描述： 用户登录，返回 JWT token 及用户信息
- 前置条件： 用户已注册

请求参数

参数格式： application/json

参数名	类型	是否必须	备注
username	string	必须	用户名

参数名	类型	是否必须	备注
password	string	必须	密码

请求示例：

```
{
  "username": "testuser",
  "password": "Password123!"
}
```

## 响应数据

成功响应 (200 OK):

```
{
  "code": 200,
  "message": "登录成功",
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "expiresIn": 7200,
    "userInfo": {
      "id": 1,
      "username": "testuser",
      "email": "testuser@example.com",
      "role": "USER",
      "isEnabled": true
    }
  }
}
```

失败响应 (401 Unauthorized):

```
{
  "code": 401,
  "message": "用户名或密码错误，或账户已被禁用",
  "data": null
}
```

测试步骤

- 1. 使用注册的用户名密码登录
- 2. 验证返回的JWT token格式
- 3. 测试错误用户名/密码
- 4. 测试被禁用账户登录

三、学期管理接口 (Terms)

3.1 创建新学期

基本信息

- 请求路径： POST /api/terms
- 接口描述： 为当前用户创建一个新学期
- 前置条件： 用户已登录，获得有效token

请求参数

参数格式： application/json

参数名	类型	是否必须	验证规则	备注
name	string	必须	长度1-100	学期名称
startDate	string	非必须	YYYY-MM-DD格式	开始日期
endDate	string	非必须	YYYY-MM-DD格式	结束日期，需晚于开始日期

请求示例：

```
{
  "name": "2025-2026 第二学期",
  "startDate": "2026-02-16",
  "endDate": "2026-07-05"
}
```

# 响应数据

成功响应 (201 Created):

```
{
  "code": 201,
  "message": "学期创建成功",
  "data": {
    "id": 3,
    "name": "2025-2026 第二学期",
    "startDate": "2026-02-16",
    "endDate": "2026-07-05",
    "createdAt": "2025-06-23T11:30:00",
    "updatedAt": "2025-06-23T11:30:00"
  }
}
```

# 测试步骤

- 1. 先完成用户注册登录获取token
- 2. 使用token创建学期
- 3. 验证学期名称唯一性（同一用户下）
- 4. 测试日期格式验证

## 3.2 获取所有学期 (分页)

### 基本信息

- 请求路径： GET /api/terms?page=0&size=10
- 接口描述： 获取当前用户的所有学期列表
- 前置条件： 用户已登录

### 请求参数

查询参数：

参数名	类型	是否必须	默认值	备注
page	number	非必须	0	页码

参数名	类型	是否必须	默认值	备注
size	number	非必须	10	每页数量

## 响应数据

成功响应 (200 OK):

```
{
  "code": 200,
  "message": "查询成功",
  "data": {
    "content": [
      {
        "id": 2,
        "name": "2025-2026 第一学期",
        "startDate": "2025-09-01",
        "endDate": "2026-01-18",
        "createdAt": "2025-01-01T00:00:00",
        "updatedAt": "2025-01-01T00:00:00"
      }
    ],
    "pageable": {
      "pageNumber": 0,
      "pageSize": 10
    },
    "totalPages": 1,
    "totalElements": 2,
    "last": true,
    "first": true
  }
}
```

## 测试步骤

1. 创建多个学期后测试获取
2. 验证分页功能
3. 确认只返回当前用户的学期

## 3.3 更新学期信息

### 基本信息

- 请求路径：PUT /api/terms/{termId}
- 接口描述：更新指定学期的信息
- 前置条件：学期属于当前用户

### 请求参数

路径参数：termId (学期ID)

请求体参数同创建学期，所有字段均为可选。

### 响应数据

响应格式同创建学期响应。

### 测试步骤

1. 更新已创建的学期信息
2. 测试权限验证（不能更新他人学期）
3. 验证部分字段更新

## 3.4 删除学期

### 基本信息

- 请求路径：DELETE /api/terms/{termId}
- 接口描述：删除指定学期及其下的所有课程
- 前置条件：学期属于当前用户

### 响应数据

成功响应 (200 OK):



```
{
  "code": 200,
  "message": "学期删除成功",
  "data": null
}
```

## 测试步骤

- 1. 创建学期并添加课程
- 2. 删除学期，验证课程是否级联删除
- 3. 测试权限验证

# 四、课程管理接口 (Courses)

## 4.1 手动添加新课程

### 基本信息

- 请求路径： POST /api/courses
- 接口描述：为指定学期添加新课程及其排程
- 前置条件：学期存在且属于当前用户

### 请求参数

参数格式： application/json

参数名	类型	是否必须	验证规则	备注
termId	number	必须	存在的学期ID	学期 ID
name	string	必须	长度1-100	课程名称
teachers	string	非必须	长度0-255	教师姓名（逗号分隔）
mainTeacherEmail	string	非必须	邮箱格式	主讲教师邮箱
courseGroupChatId	string	非必须	长度0-50	课程群号
tag	number	非必须	0或1	课程标签：0=选修，1=必修

参数名	类型	是否必须	验证规则	备注
note	string	非必须	长度0-1000	备注
scheduleEntries	object[]	非必须	-	排程列表
└─ location	string	非必须	长度0-255	上课地点
└─ dayOfWeek	number	必须	1-7	星期几 (1=周一)
└─ startPeriod	number	必须	1-12	开始节次
└─ endPeriod	number	必须	≥startPeriod	结束节次
└─ startWeek	number	必须	1-20	开始周
└─ endWeek	number	必须	≥startWeek	结束周

请求示例：

```
{
  "termId": 1,
  "name": "高等数学",
  "teachers": "张三,李四",
  "mainTeacherEmail": "zhangsan@example.com",
  "courseGroupChatId": "123456789",
  "tag": 1,
  "note": "期末考试占比70%",
  "scheduleEntries": [
    {
      "location": "教学楼A-101",
      "dayOfWeek": 1,
      "startPeriod": 1,
      "endPeriod": 2,
      "startWeek": 1,
      "endWeek": 16
    },
    {
      "location": "教学楼B-203",
      "dayOfWeek": 3,
      "startPeriod": 3,
      "endPeriod": 4,
      "startWeek": 1,
      "endWeek": 8
    }
  ]
}
```

## 响应数据

成功响应 (201 Created):

```
{
  "code": 201,
  "message": "课程创建成功",
  "data": {
    "id": 1,
    "name": "高等数学",
    "teachers": "张三,李四",
    "mainTeacherEmail": "zhangsan@example.com",
    "courseGroupChatId": "123456789",
    "tag": 1,
    "note": "期末考试占比70%",
    "scheduleEntries": [
      {
        "id": 1,
        "location": "教学楼A-101",
        "dayOfWeek": 1,
        "startPeriod": 1,
        "endPeriod": 2,
        "weeks": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
      },
      {
        "id": 2,
        "location": "教学楼B-203",
        "dayOfWeek": 3,
        "startPeriod": 3,
        "endPeriod": 4,
        "weeks": [1, 2, 3, 4, 5, 6, 7, 8]
      }
    ],
    "createdAt": "2025-06-23T11:40:00",
    "updatedAt": "2025-06-23T11:40:00"
  }
}
```

## 测试步骤

1. 先创建学期
2. 为学期添加课程（包含tag字段）
3. 验证排程时间冲突检测
4. 测试必修/选修标签功能

## 4.2 获取指定学期的所有课程 (分页)

### 基本信息

- 请求路径：GET /api/terms/{termId}/courses?page=0&size=20
- 接口描述：获取指定学期下的所有课程
- 前置条件：学期属于当前用户

### 请求参数

路径参数：termId (学期ID)

参数名	类型	是否必须	默认值	备注
page	number	非必须	0	页码
size	number	非必须	20	每页数量

### 响应数据

响应格式同创建课程，但为分页结构。课程列表中每个课程都包含 tag 字段。

### 测试步骤

- 创建多门课程（包含必修和选修）
- 获取课程列表，验证tag字段正确返回
- 测试分页功能

## 4.3 高级课程搜索 (筛选与分页)

### 基本信息

- 请求路径：GET /api/courses/search
- 接口描述：多条件课程搜索功能

### 请求参数

查询参数：

参数名	类型	是否必须	备注
termId	number	非必须	学期 ID
name	string	非必须	课程名 (模糊匹配)
teacher	string	非必须	教师名 (模糊匹配)
tag	number	非必须	课程标签筛选：0=选修，1=必修
dayOfWeek	number	非必须	星期几 (1-7)
page	number	非必须	页码 (默认 0)
size	number	非必须	每页数量 (默认 10)

请求示例：

```
GET /api/courses/search?termId=1&tag=1&teacher=张三&page=0&size=10
```

测试步骤

- 1. 创建多样化的课程数据
- 2. 测试各种筛选条件组合
- 3. 重点测试tag字段筛选功能

4.4 更新课程信息

基本信息

- 请求路径： PUT /api/courses/{courseId}
- 接口描述： 完全替换课程信息及其所有排程

请求参数

请求体格式同创建课程，但不包含 termId ，且所有字段均为可选。

特别注意：更新时 tag 字段可以在必修和选修之间切换。

测试步骤

- 1. 更新课程基本信息

2. 重点测试tag字段的修改
3. 测试排程信息更新

## 4.5 删除课程

### 基本信息

- 请求路径：DELETE /api/courses/{courseId}
- 接口描述：删除指定课程及其所有排程

### 响应数据

成功响应 (200 OK):

```
{
  "code": 200,
  "message": "课程删除成功",
  "data": null
}
```

### 测试步骤

1. 删除课程，验证排程级联删除
2. 测试权限验证

## 五、管理员接口 (Admin)

访问控制：以下接口仅限 ADMIN 角色的用户访问。

### 5.1 获取用户列表 (分页与筛选)

#### 基本信息

- 请求路径：GET /api/admin/users
- 接口描述：获取用户列表，支持分页和多条件筛选
- 前置条件：当前用户角色为ADMIN

## 请求参数

查询参数：

参数名	类型	是否必须	备注
username	string	非必须	用户名 (模糊匹配)
email	string	非必须	邮箱 (模糊匹配)
role	string	非必须	角色 (ADMIN, USER, TRIAL)
isEnabled	boolean	非必须	账户状态 (true/false)
page	number	非必须	页码 (默认 0)
size	number	非必须	每页数量 (默认 20)

## 响应数据

成功响应 (200 OK)：



```
{
  "code": 200,
  "message": "查询成功",
  "data": {
    "content": [
      {
        "id": 2,
        "username": "user_one",
        "email": "user_one@example.com",
        "role": "USER",
        "isEnabled": true,
        "createdAt": "2025-01-01T00:00:00",
        "updatedAt": "2025-01-01T00:00:00"
      }
    ],
    "pageable": {
      "pageNumber": 0,
      "pageSize": 20
    },
    "totalPages": 5,
    "totalElements": 48
  }
}
```

## 测试步骤

1. 使用ADMIN角色用户登录
2. 测试各种筛选条件
3. 验证非ADMIN用户无法访问（403错误）

## 5.2 更新用户状态 (启用/禁用)

### 基本信息

- **请求路径:** PUT /api/admin/users/{userId}/status
- **接口描述:** 更新指定用户的启用状态
- **前置条件:** 当前用户角色为ADMIN

请求参数

路径参数： userId (用户ID)

请求体参数：

参数名	类型	是否必须	备注
isEnabled	boolean	必须	账户状态

响应数据

成功响应 (200 OK):

```
{
  "code": 200,
  "message": "用户状态更新成功",
  "data": null
}
```

测试步骤

- 1. 禁用用户账户
- 2. 验证被禁用用户无法登录
- 3. 重新启用用户

六、导入接口 (Import)

6.1 通过 Excel 导入课程

基本信息

- 请求路径： POST /api/import/excel?termId=1
- 接口描述： 将 Excel 文件中的课程数据导入到指定学期
- 前置条件： 学期存在且属于当前用户

请求参数

请求头： Content-Type: multipart/form-data

参数名	类型	是否必须	备注
termId	number	必须	学期 ID
file	file	必须	Excel 文件

## Excel 格式约定

Excel文件第一行必须包含以下表头：

字段名称	是否必须	备注
课程名称	必须	-
教师	非必须	多个教师用逗号分隔
标签	非必须	<b>必修/选修</b> ，默认为必修
上课地点	非必须	-
星期	必须	1-7的数字
开始节次	必须	1-12的数字
结束节次	必须	≥开始节次的数字
开始周	必须	1-20的数字
结束周	必须	≥开始周的数字
备注	非必须	-

Excel示例数据：

课程名称	教师	标签	上课地点	星期	开始节次	结束节次	开始周	结束周	备注
高等数学	张三	必修	教学楼A-101	1	1	2	1	16	重要
大学英语	李四	选修	教学楼B-201	3	3	4	1	8	无

## 响应数据

成功响应 (200 OK)：

```
{
  "code": 200,
  "message": "导入成功",
  "data": {
    "totalRows": 10,
    "importedCourses": 5,
    "failedRows": 0,
    "errors": []
  }
}
```

### 失败响应 (400 Bad Request):

```
{
  "code": 400,
  "message": "导入失败，文件格式错误或数据不合法",
  "data": {
    "totalRows": 10,
    "importedCourses": 0,
    "failedRows": 10,
    "errors": [
      "第 3 行: [星期] 格式错误，必须为1-7的数字",
      "第 5 行: [标签] 格式错误，必须为'必修'或'选修'",
      "第 7 行: [开始节次] 必须为数字"
    ]
  }
}
```

## 测试步骤

1. 准备Excel测试文件（包含标签列）
2. 测试正确格式的文件导入
3. 测试错误格式文件的错误处理
4. 验证导入后的课程标签正确性

# 七、错误码说明

错误码	说明	常见场景
200	请求成功	正常操作
201	资源创建成功	用户注册、学期/课程创建
400	请求参数错误	缺少必要参数、格式错误
401	未认证或认证失败	token过期、用户名密码错误
403	权限不足	非管理员访问管理接口
404	资源不存在	学期/课程不存在
409	资源冲突	用户名重复、课程时间冲突
500	服务器内部错误	数据库连接失败、未知异常

# 八、新功能特色说明

## 8.1 课程标签功能

**功能描述：**为每门课程添加标签属性，支持必修/选修分类。

**技术实现：**

- 数据库： `courses.tag` 字段，类型 `TINYINT(1)` ， 1=必修， 0=选修
- 前端：支持筛选、展示、编辑标签
- Excel导入：支持"标签"列导入

**使用场景：**

- 学生区分必修选修课程
- 按课程类型统计学分
- 课程表视觉化区分

## 8.2 多视图支持

周视图：传统课程表格网格视图

日视图：单日课程详细展示

学期视图：课程列表形式，支持展开详情

## 8.3 权限管理体系

角色类型：

- ADMIN：系统管理员，可管理所有用户
- USER：普通用户，管理个人课表
- TRIAL：试用用户，功能受限

# 九、开发与测试建议

## 9.1 开发环境搭建

### 1. 数据库初始化

```
-- 运行Data.py生成的SQL脚本
-- 包含admin用户（用户名：admin，密码：123456）
```

### 2. 测试数据

- 使用Data.py生成的示例数据
- 包含多种标签的课程样例

## 9.2 测试流程

### 阶段一：基础功能

1. 用户注册/登录 ✓
2. JWT认证验证 ✓
3. 学期CRUD操作 ✓

### 阶段二：核心功能

1. 课程CRUD操作（含标签） ✓

2. 课程搜索筛选 ✓
3. 前端多视图展示 ✓

## 阶段三：高级功能

1. Excel导入（含标签） ✓
2. 管理员用户管理 ✓
3. 权限控制验证 ✓

## 9.3 重点测试用例

### 1. 课程标签功能

- 创建必修课程（tag=1）
- 创建选修课程（tag=0）
- 必修选修之间切换
- 标签筛选功能

### 2. Excel导入功能

- 标准格式文件导入
- 标签列识别
- 错误格式处理

### 3. 权限验证

- 普通用户访问管理接口
- 禁用用户登录
- 跨用户数据访问

# 十、附录

## 10.1 快速开始

```
# 1. 管理员登录
POST /api/auth/login
{
  "username": "admin",
  "password": "123456"
}

# 2. 创建学期
POST /api/terms
Authorization: Bearer <token>
{
  "name": "2025年春季学期",
  "startDate": "2025-02-17",
  "endDate": "2025-06-20"
}

# 3. 添加课程（含标签）
POST /api/courses
{
  "termId": 1,
  "name": "数据结构",
  "teachers": "王教授",
  "tag": 1,
  "scheduleEntries": [...]
}
```

## 10.2 常用测试账户

用户名	密码	角色	状态	备注
admin	123456	ADMIN	启用	管理员
user1	123456	USER	启用	普通用户
trial	123456	TRIAL	启用	试用用户
user2	123456	USER	禁用	测试禁用



## 10.3 前端配置建议

- API Base URL: `http://localhost:8080/api`
- Token存储: `localStorage`
- 请求拦截器: 自动添加Authorization头
- 响应拦截器: 统一错误处理

**文档版本:** v3.0

**更新时间:** 2025年6月23日

**主要更新:** 新增课程标签功能、完善测试指南、更新Excel导入格式