# 学习伙伴系统「StudyPAL+」完整接口文档

## 基础信息

- \*\*Base URL\*\*: `https://api.studypal.example.com/v1`

- \*\*认证方式\*\*: Bearer Token (JWT)

- \*\*响应格式\*\*: JSON

## 1. 用户认证模块

### 1.1 用户注册

- \*\*Endpoint\*\*: `/auth/register`

- \*\*Method\*\*: POST

- \*\*Request\*\*:

```json

{

"student\_id": "20250001",

"password": "SecurePass123!",

"name": "张三",

"major": "计算机科学"

}

```

- \*\*Response\*\*:

```json

{

"token": "eyJhbGci...",

"user\_info": {

"student\_id": "20250001",

"name": "张三",

"major": "计算机科学"

}

}

```

### 1.2 用户登录

- \*\*Endpoint\*\*: `/auth/login`

- \*\*Method\*\*: POST

- \*\*Request\*\*:

```json

{

"student\_id": "20250001",

"password": "SecurePass123!"

}

```

- \*\*Response\*\*: 同注册接口

## 2. MCP数据服务模块

### 2.1 获取完整课表

- \*\*Endpoint\*\*: `/mcp/courses`

- \*\*Method\*\*: GET

- \*\*Headers\*\*:

```

Authorization: Bearer <token>

```

- \*\*Response\*\*:

```json

{

"current\_term": "2025-秋季",

"courses": [

{

"course\_id": "CS101",

"name": "机器学习",

"schedule": [

{

"weekday": 1,

"start\_time": "09:00",

"end\_time": "11:00",

"location": "开发区A101"

}

],

"instructor": "王教授"

}

]

}

```

### 2.2 获取近期课程

- \*\*Endpoint\*\*: `/mcp/courses/upcoming`

- \*\*Method\*\*: GET

- \*\*Query Params\*\*:

- `days=7` (可选，默认7天)

- \*\*Response\*\*: 同2.1，但只返回指定天数内的课程

## 3. 智能学习计划模块

### 3.1 生成学习计划

- \*\*Endpoint\*\*: `/plan/generate`

- \*\*Method\*\*: POST

- \*\*Request\*\*:

```json

{

"preferences": {

"study\_times": ["morning", "evening"],

"daily\_max\_hours": 6

}

}

```

- \*\*Response\*\*:

```json

{

"generated\_at": "2025-08-20T14:30:00Z",

"weekly\_plan": {

"Monday": [

{

"course\_id": "CS101",

"task": "预习第三章",

"duration": 90,

"priority": "high"

}

]

}

}

```

### 3.2 更新学习进度

- \*\*Endpoint\*\*: `/plan/progress`

- \*\*Method\*\*: PATCH

- \*\*Request\*\*:

```json

{

"course\_id": "CS101",

"task\_id": "task\_001",

"completed": true

}

```

## 4. AI辅导模块

### 4.1 智能问答

- \*\*Endpoint\*\*: `/ai/ask`

- \*\*Method\*\*: POST

- \*\*Request\*\*:

```json

{

"question": "反向传播算法的核心思想是什么？",

"context": {

"course\_id": "CS101"

}

}

```

- \*\*Response\*\*:

```json

{

"answer": "反向传播通过链式法则...",

"resources": [

{

"type": "video",

"title": "反向传播详解",

"url": "https://example.com/bp.mp4",

"duration": "15:30"

}

]

}

```

### 4.2 学习诊断报告

- \*\*Endpoint\*\*: `/ai/diagnosis`

- \*\*Method\*\*: GET

- \*\*Response\*\*:

```json

{

"knowledge\_map": {

"strong\_topics": ["线性代数", "Python基础"],

"weak\_topics": ["概率论", "矩阵运算"]

},

"suggestions": ["建议每天花30分钟练习矩阵运算"]

}

```

## 5. 成就系统模块

### 5.1 获取成就列表

- \*\*Endpoint\*\*: `/achievements`

- \*\*Method\*\*: GET

- \*\*Response\*\*:

```json

{

"unlocked": [

{

"id": "achv\_001",

"name": "七日学霸",

"icon": "🏆",

"description": "连续学习7天",

"unlocked\_at": "2025-08-15"

}

],

"in\_progress": [

{

"id": "achv\_002",

"name": "效率达人",

"progress": 3,

"target": 5,

"reward": "⚡"

}

]

}

```

### 5.2 成就详情

- \*\*Endpoint\*\*: `/achievements/{achv\_id}`

- \*\*Method\*\*: GET

- \*\*Response\*\*:

```json

{

"id": "achv\_001",

"name": "七日学霸",

"description": "连续学习7天",

"unlock\_condition": "连续7天每天学习≥1小时",

"unlocked\_users": 243

}

```

## 6. 实用工具接口

### 6.1 校历查询

- \*\*Endpoint\*\*: `/utils/academic-calendar`

- \*\*Method\*\*: GET

- \*\*Response\*\*:

```json

{

"term": "2025-秋季",

"events": [

{

"date": "2025-09-01",

"name": "开学典礼",

"type": "holiday"

}

]

}

```

### 6.2 校园地图

- \*\*Endpoint\*\*: `/utils/campus-map`

- \*\*Method\*\*: GET

- \*\*Response\*\*:

```json

{

"buildings": [

{

"id": "A101",

"name": "开发区教学楼A",

"location": {

"lat": 38.880,

"lng": 121.538

}

}

]

}

```

## 错误处理

所有错误返回格式：

```json

{

"error": {

"code": "ERR\_4001",

"message": "Invalid token",

"details": "Token expired"

}

}

```

## 实现建议

### 后端文件结构

```

study\_pal/

├── app.py # 主应用入口

├── auth.py # 认证相关

├── mcp.py # MCP服务对接

├── plan\_service.py # 学习计划逻辑

├── ai\_service.py # AI功能实现

└── achievement.py # 成就系统

```

### 示例实现代码（app.py）

```python

from flask import Flask, request, jsonify

from auth import login\_required

app = Flask(\_\_name\_\_)

@app.route('/mcp/courses')

@login\_required

def get\_courses():

# 从MCP获取数据

return jsonify({

"courses": [],

"term": "2025-秋季"

})

@app.route('/ai/ask', methods=['POST'])

@login\_required

def ask\_question():

data = request.json

# 调用AI服务

return jsonify({

"answer": "示例回答",

"resources": []

})

```

这个文档涵盖了从用户认证、数据获取到智能功能的所有必要接口，每个接口都包含：

1. 明确的Endpoint和Method

2. 请求参数格式

3. 响应数据结构

4. 必要的认证信息

您可以根据实际开发进度，分阶段实现这些接口。建议先实现核心的MCP数据获取和学习计划生成，再逐步添加AI功能和成就系统。