

# django\_knowledge\_tree

## 项目名：知识点树管理系统

### 项目摘要：

该项目是一个基于 Django 和 PostgreSQL 构建的知识点树管理系统。通过使用 JSON 存储知识点数据，支持知识点树的创建、编辑、删除等操作，并通过前端组件（如 jsTree）将 JSON 数据可视化为树形结构，方便用户管理和查询。

## 项目结构

```
django_knowledge_tree/
├── Dockerfile
├── manage.py
├── requirements.txt
├── knowledge_tree/           # 主 Django 工程目录
│   ├── __init__.py
│   ├── settings.py          # Django 配置文件
│   ├── urls.py              # 项目级 URL 配置
│   └── wsgi.py              # WSGI 入口
├── tree/                     # 应用目录，包含模型、视图、模板、静态文件等
│   ├── __init__.py
│   ├── models.py            # 模型定义（知识点树记录）
│   ├── forms.py             # Django 表单定义
│   ├── views.py             # 视图函数，包括知识点树的创建、编辑、展示、删除等
│   ├── urls.py              # 应用级 URL 配置
│   └── templates/
│       └── tree/
│           ├── base.html     # 基础模板，包含全局静态资源加载（jQuery、
│                               Bootstrap、sb-admin-2 等）
│           ├── index.html    # 知识点树列表页
│           ├── edit.html     # 编辑/创建知识点树页面
│           ├── tree_view.html # 知识点树展示页（同时显示 JSON 原文和图形化树）
│           └── confirm_delete.html # 删除确认页
├── static/
│   ├── css/
│   │   └── sb-admin-2.min.css # 引入 sb-admin-2 的 CSS 文件
│   └── js/
│       ├── jquery-3.7.1.min.js # jQuery 脚本
│       ├── sb-admin-2.js       # sb-admin-2 调试版脚本（可选）
│       └── sb-admin-2.min.js   # sb-admin-2 压缩版脚本
```

## 环境要求

- Python 3.8 及以上
- Django 3.2 及以上
- PostgreSQL 12及以上

## 安装步骤

### 1. 克隆代码

```
git clone https://github.com/inkcakesci/knowledgetree.git
cd django_knowledge_tree
```

### 2. 创建并激活虚拟环境

```
python -m venv .venv
# Windows:
.venv\Scripts\activate
# macOS/Linux:
source .venv/bin/activate
```

### 3. 安装依赖

```
pip install -r requirements.txt
```

### 4. 数据库迁移

在数据库启动前，需要先运行 PostgreSQL。确保 Docker 中的 PostgreSQL 容器正常运行，具体容器的启动命令可见在 Lark 上的备份说明。

启动 PostgreSQL 数据库：

```
docker run --name postgres -e POSTGRES_USER=klgtree -e
POSTGRES_PASSWORD=klgtree123 -e POSTGRES_DB=knowledge_tree -p
5432:5432 -d postgres
```

### 5. 静态文件收集

使用 `python manage.py collectstatic` 生成静态文件集合。

### 6. 运行项目

在项目根目录下运行：

```
python manage.py runserver
```

浏览器中打开 `http://127.0.0.1:8000/` 即可访问知识点树列表页面。

## 数据库备份

当前数据库会定期备份，具体 SQL 文件可以在 Lark 上查看，不再赘述如何安装 PostgreSQL 和 Docker。

其他

- 目前尚未完成 Docker Compose 配置文件的编写，因此 Docker 部署的服务需要手动启动。
- 本项目基于 sb-admin-2 前端模板进行二次开发，前端样式和布局可根据需要自行修改。
- 如果需要扩展其他功能，例如用户认证、权限管理等，请根据 Django 官方文档进行相应配置。

使用说明

录入格式说明

本项目中的知识点树数据采用符合 jsTree 组件要求的嵌套 JSON 格式。整个 JSON 数据由一个或多个节点组成，每个节点包含以下主要字段：

- **id**：字符串类型，用于唯一标识该节点。
- **text**：表示节点要显示的文本内容。
- **children**：数组类型，包含该节点的子节点。如果该节点没有子节点，可以设为一个空数组或省略此字段。
- **keywords**（可选）：一个数组，包含与该知识点相关的关键词。

Prompt样例

Prompt示例1：

```
请严格按照 JSON 格式生成符合 jsTree 规范的知识点树，要求如下：

1. **根节点**：节点名称为 "[学科]"（如：[数学] / [物理]）。
2. **第二层**：节点名称为指定教材名称 "[课本名称]"。
3. **教材节点**下按实际章节划分单元。
4. 每个单元包含 **3-5个核心知识点**，每个知识点应包括：
   - **唯一id**：如 `unit1_1`（用于标识层级结构）。
   - **text字段**：包括中文编号和标题（如：第一单元 单元名称）。
   - **keywords字段**：包含 **2-4个知识点关键词**，关键词应与知识点密切相关。
   - 叶子节点使用 `"icon": "jstree-file"`。

# 生成结构示例：
```json
{
  "id": "root",
  "text": "[学科]",
  "keywords": ["学科特征"],
  "children": [{
    "id": "book",
    "text": "[课本名称]",
    "keywords": ["版本特征"],
    "children": [{
      "id": "unit1",
      "text": "第一单元 单元名称",
      "keywords": ["模块特征"],
      "children": [
```

```
{
  "id": "unit1_1",
  "icon": "jstree-file",
  "text": "知识点标题 ( 核心内容说明 )",
  "keywords": ["知识点关键词"]
}
]
}]
}]
}
```

## Prompt示例2 :

请生成一份符合 jsTree 格式的 JSON 知识点树，要求如下：

- 根节点名称为 "[科目]"。
- 根节点下有一个子节点 "{年级+上下册}"。
- 每个单元下包含详细的知识点内容。
- 请确保生成的 JSON 数据结构符合 jsTree 的嵌套格式，并为每个节点指定唯一的 id。

## changelog

2025年2月4日：

修正了一次创建新知识点树会生成2个的逻辑错误。

2025年2月10日：

数据库改为使用PostgreSQL

```
docker run --name postgres -e POSTGRES_USER=klgtree -e
POSTGRES_PASSWORD=klgtree123 -e POSTGRES_DB=knowledge_tree -p 5432:5432 -d
postgres
```

备份方法：

### 1. 使用容器内存在的目录：

```
docker exec -t postgres pg_dump -U klgtree knowledge_tree -f
/tmp/backup.sql
```

### 2. 将备份文件复制到宿主机：

```
docker cp postgres:/tmp/backup.sql /path/to/host/backup.sql
```

### 3. 清理容器内的备份文件：

```
docker exec postgres rm /tmp/backup.sql
```

---

## 产品使用说明书

### 概述

该系统提供一个基于 JSON 存储的知识点树数据结构，用户可以通过图形化界面管理知识点树，并进行编辑、删除等操作。数据以层级结构保存，适用于教育管理和知识点梳理。

### 功能

1. **知识点树展示**：系统支持通过 jsTree 展示数据的树形结构，用户可以点击每个节点查看详情。
2. **创建/编辑功能**：支持创建新的知识点树，编辑现有树形数据，保存时自动格式化 JSON 数据。
3. **删除功能**：用户可以删除指定的知识点树。
4. **关键词查看**：每个节点可设置相关关键词，用户点击节点时，相关的关键词将在页面右侧显示。

### 用户流程

1. **首页**：用户可查看现有知识点树，点击树名称进入查看详情页面。
2. **编辑页面**：点击“编辑”按钮，进入编辑页面，可以修改 JSON 数据并保存。编辑器使用 Ace 或 CodeMirror 提供格式化和高亮显示功能。
3. **查看页面**：点击节点，相关关键词会在页面右侧展示，帮助用户理解节点内容。

### 使用要求

- 用户需要登录才能编辑知识点树。
- 知识点数据存储于 PostgreSQL 数据库，系统支持 Docker 部署，便于本地快速搭建。