

🔌 AI 用电助手 API 接口文档

基础路径 (Base URL): <http://localhost:8000> **认证方式:** 所有非 Auth 接口均需在 Request Header 中携带 `Authorization: Bearer <token>`

1. 🔒 身份认证 (Authentication)

1.1 用户注册

创建新的家庭账户。

- **URL:** </api/auth/register>
- **Method:** POST
- **Request Body:**

```
{  
    "username": "user1",  
    "password": "securepassword123",  
    "address": "杭州市西湖区" // 用于AI获取当地天气  
}
```

- **Response (201 Created):**

```
{  
    "message": "User registered successfully",  
    "user_id": 1  
}
```

1.2 用户登录 (获取 Token)

获取访问令牌 (Access Token)。

- **URL:** </api/auth/token>
- **Method:** POST
- **Request Body (JSON):**

```
{  
    "username": "user1",  
    "password": "securepassword123"  
}
```

- **Response (200 OK):**

```
{  
    "access_token": "eyJhbGciOiJIUzI1NiIsIn...".  
    "token_type": "bearer",  
    "expires_in": 3600  
}
```

1.3 获取当前用户信息

用于前端刷新页面后获取用户状态。

- **URL:** /api/auth/me
- **Method:** GET
- **Response (200 OK):**

```
{  
    "id": 1,  
    "username": "user1",  
    "address": "杭州市西湖区",  
    "created_at": "2023-10-27T10:00:00"  
}
```

1.4 用户登出

后端将当前 Token 加入黑名单（可选）或前端仅需丢弃 Token。

- **URL:** /api/auth/logout
- **Method:** POST
- **Response (200 OK):**

```
{  
    "message": "Successfully logged out"  
}
```

2. 仪表盘数据 (Dashboard Data)

此模块将数据拆分为多个接口，以便前端组件独立加载。

2.1 获取今日概览 (KPI Cards)

用于顶部卡片显示关键指标。

- **URL:** /api/data/summary
- **Method:** GET
- **Response (200 OK):**

```
{
  "total_power_now": 4.5,           // 当前总功率 (kW)
  "daily_cost_estimate": 12.8,      // 今日预计电费 (元)
  "month_usage_kwh": 320.5,         // 本月累计用电 (kWh)
  "active_appliances_count": 3     // 开启的电器数量
}
```

2.2 获取用电趋势 (Line Chart)

用于渲染“家庭实时用电”折线图。

- **URL:** /api/data/consumption/trend
- **Method:** GET
- **Query Params:**
 - range: 24h (默认) | week | month
- **Response (200 OK):**

```
{
  "range": "24h",
  "x_axis": ["10:00", "10:30", "11:00", "11:30", "12:00"], // 时间轴
  "y_axis": [1.2, 1.5, 2.1, 0.8, 3.5] // 对应时间点的耗电量 (kWh)
}
```

2.3 获得用影响因素 (AI Analysis)

用于渲染雷达图或饼图。此数据由 Agent 后台分析生成。

- **URL:** /api/data/consumption/factors
- **Method:** GET
- **Response (200 OK):**

```
{
  "updated_at": "2023-10-27T12:00:00",
  "factors": [
    { "name": "天气因素 (制冷/制热)", "value": 40 },
    { "name": "基础待机", "value": 15 },
    { "name": "大功率电器使用", "value": 35 },
    { "name": "峰时用电", "value": 10 }
  ],
  "suggestion": "今日空调用电占比过高，建议调高1度。" // 简短的AI摘要
}
```

3. 智能设备管理 (Smart Control)

3.1 获得电器列表与状态

用于显示电器网格。

- **URL:** /api/appliances
- **Method:** GET
- **Response (200 OK):**

```
[  
  {  
    "id": 101,  
    "name": "客厅空调",  
    "type": "ac",  
    "is_on": true,  
    "current_power_kw": 2.5  
  },  
  {  
    "id": 102,  
    "name": "主卧灯",  
    "type": "light",  
    "is_on": false,  
    "current_power_kw": 0.0  
  }  
]
```

3.2 控制电器 (AI 介入)

核心接口。 用户点击开关后，后端会调用 Agent 进行判断并返回建议。

- **URL:** /api/appliances/{id}/control
- **Method:** POST
- **Request Body:**

```
{  
  "action": "ON" // 或 "OFF"  
}
```

- **Response (200 OK):**

```
{  
  "success": true,  
  "appliance_id": 101,  
  "new_status": "ON",  
  "ai_feedback": {  
    "level": "info", // info | warning | success  
    "message": "空调已开启。检测到当前室外温度较高，建议设置在26度以平衡舒适度与电  
费。",  
    "cost_projection": "预计每小时花费 1.5 元"  
  }  
}
```

4. ⚑ AI 用电顾问 (Chat)

4.1 发送对话消息

与 AI Agent 进行自然语言交互。

- **URL:** /api/chat/completions
- **Method:** POST
- **Request Body:**

```
{  
  "message": "为什么我昨天的电费这么贵？",  
  "history": [] // 可选，附带上下文  
}
```

- **Response (Stream):**
 - 建议使用 Server-Sent Events (SSE) 或流式响应。
 - 如果是普通 JSON 响应：

```
{  
  "reply": "通过分析数据，我发现昨天下午 14:00 到 16:00 之间，您的中央空调处于全功率运行状态，而当时正处于峰值电价时段（1.2元/度）。这导致了电费激增。建议您下次尝试在该时段前前提前预冷房间。"  
}
```

5. 数据字典与枚举

5.1 电器类型 (Appliance Type)

- ac: 空调
- fridge: 冰箱
- light: 照明
- tv: 电视
- heater: 热水器/暖气
- other: 其他

5.2 API 错误码

- 400 Bad Request: 参数错误
- 401 Unauthorized: Token 无效或过期
- 403 Forbidden: 权限不足
- 404 Not Found: 资源不存在
- 500 Internal Server Error: 服务器内部错误 (如 Agent 服务超时)

