MGS-104 Python SDK 及 API 使用说明

脚本编写规范 · 控制接口 · 数据处理工具

版本 V1.0

发布日期：2026年5月

⚠️ **免责声明**

本平台仅提供脚本运行接口、样例与开发支持说明。用户基于本平台进行的脚本编写、参数配置、控制策略设计、第三方库安装与运行操作，均由用户自行评估风险并承担后果。对于因用户操作不当、参数设置错误、脚本逻辑缺陷、环境变更或第三方依赖引发的直接或间接问题，本平台不承担责任。

📖 目录

0. 强烈建议加载 template.py 的结构，再改局部

为了保证脚本结构规范、便于排查问题，建议用户统一从 `template.py` 模板起步，而不是从空白文件自由发挥。

模板路径： `/extension/case/scriptcase/template.py`

建议只修改模板里你圈定的"可修改区"，其他结构尽量保留：

- Info 区（Name/Description/Author 等说明信息）
- Basic Parameters（运行时长、发送频率等参数）
- Your Parameter（你的策略参数）
- 主循环中的策略计算与发送逻辑

💡 这样做的好处：结构统一，团队内可维护性更高；运行行为更可预期，减少"脚本能跑但不可控"的问题；出现问题时更容易定位是"参数问题"还是"框架问题"。

1. 最快上手（30 秒）

建议先点击 Load Template，再按下面这种规范骨架填写参数与策略：

```
from algorithm_load.script_runtime import api
```

```
api.log("script start")
api.sleep(2.0)

while not api.should_stop():
    api.node[1].p = 10.0
    api.sleep(0.1)

api.log("script end")
```

✅ 关键点：控制接口统一走 api；主循环必须检查 api.should_stop()；主循环必须有 api.sleep(...)，避免占满 CPU 和高频误发命令。

2. 可调用对象与全局变量

运行器会给脚本注入以下可用对象：

- `api`：主控制对象（核心接口）。
- `ctx`：api 的别名（与 api 是同一个对象）。
- `math`、`time`：标准库模块。
- `pd` / `pandas`：pandas 模块（如果环境未安装，则为 None）。
- `run_sine_case(...)`：内置参考案例函数。
- `print(...)`：已重定向到日志面板，效果等价于 api.log(...)。

3. API 函数库速查

3.1 控制与流程

接口	说明
`api.log(text)`	输出日志到界面日志区
`api.sleep(seconds)`	可中断睡眠（停止脚本时会尽快返回）
`api.should_stop() -> bool`	查询是否收到停止请求

3.2 单节点控制

两种写法都支持：

```
api.set_node_p(1, 12.0)
api.node[1].p = 12.0
```

接口	说明
`api.set_node_p(node, value)`	设置单节点功率
`api.set_node_v(node, value)`	设置单节点电压

接口	说明
`api.set_node_i(node, value)`	设置单节点电流
`api.node[node].p = value`	属性写法，等价于 set_node_p
`api.node[node].v = value`	属性写法，等价于 set_node_v
`api.node[node].i = value`	属性写法，等价于 set_node_i
`api.set_node_shutdown(node)`	下发单节点关机
`api.node[node].shutdown()`	等价于 set_node_shutdown

3.3 广播控制

接口	说明
`api.set_broadcast_p(value)`	广播功率设定
`api.set_broadcast_v(value)`	广播电压设定
`api.set_broadcast_i(value)`	广播电流设定

3.4 梯度/步进变化（软启动、斜坡）

接口	说明
`api.step_node(node, field, current, target, steps, step_seconds)`	field 为 p/v/i 的通用分步
`api.step_node_p(node, current, target, steps, step_seconds)`	功率分步
`api.step_node_v(node, current, target, steps, step_seconds)`	电压分步
`api.step_node_i(node, current, target, steps, step_seconds)`	电流分步
`api.node[node].step(...)` / `api.node[node].step_p / step_v / step_i`	同步封装写法

3.5 节点数据读取

接口	说明
`api.node[node].get_data(timeout=0.0) -> dict or None`	读取节点最新数据
`api.get_node_data(node, timeout_s=0.0) -> dict or None`	与上面等价的函数式接口

接口	说明
`api.get_latest_node(node, timeout_s=0.3) -> dict or None`	兼容别名

> 📡 返回 dict 典型字段：`node`、`online`、`timestamp`、`v`、`i`、`p`、`raw`
> **读取语义**：timeout ≤ 0 立即返回缓存值（无缓存则 None）；timeout > 0 最多等待 timeout 秒取新数据；超时后可能返回缓存或 None。

3.6 时间转换与轨迹回放工具（DataFrame 友好）

接口	说明
`api.to_seconds_series(df, time_col)`	将时间列解析为秒（支持数值、datetime 字符串、时长字符串）
`api.to_relative_time(df, time_col, speedup=1.0, out_col=None)`	把时间列转为相对时间并可加速
`api.replay_relative_schedule(schedule, send_event, ...)`	按相对时间高精度回放事件序列

> 📂 这组接口可直接参考示例：
> `src/algorithm_load/scriptcase/case3_load_trajectory_loadcsv.py`

4. 模板怎么引用

4.1 UI 一键模板

在 Algorithm Load 面板点击 Load Template，会加载默认模板文件：
`extension/case/scriptcase/template.py`

4.2 推荐方式

1. 点击 Load Template。
2. 只改"可修改区"（参数区和波形/策略区）。
3. 另存为新脚本（不要直接覆盖模板）。
4. Run Script 验证。

4.3 哪些是"可修改区"

建议修改：文件头 Info 信息、Basic Parameters、Your Parameter、主循环内策略计算和发送内容。
建议保留：`from algorithm_load.script_runtime import api`、`while not api.should_stop()` 退出结构、`api.sleep(...)` 节流结构、启动日志和结束日志的基本框架。

4.4 案例参考

可直接复用这些案例结构：

- `src/algorithm_load/scriptcase/case1_programmable_source.py`
- `src/algorithm_load/scriptcase/case2_pv_battery_load_balance.py`
- `src/algorithm_load/scriptcase/case3_load_trajectory_loadcsv.py`
- `src/algorithm_load/scriptcase/case4_automatic_testing.py`
- `src/algorithm_load/scriptcase/case5_pv_battery_load_pcc.py`

5. 如何给提供的环境加库

使用 extension 目录下虚拟环境：

```
extension/.venv/bin/pip install 包名
```

或者按 requirements 安装：

```
extension/.venv/bin/pip install -r extension/requirements.txt
```

> ⚠️ 你脚本里 import 的第三方库，必须装在"脚本实际使用的 Python 环境"里。
> pandas 在运行器中是可选依赖；未安装时相关 DataFrame 功能不可用。

6. 脚本规范（建议强制遵守）

- 循环必须可停：所有主循环都写 `while not api.should_stop()`。
- 控制命令要限频：每轮循环至少一次 `api.sleep(...)`。
- 参数前置：把幅值、频率、目标节点、模式等参数集中在脚本头部。
- 日志可追溯：关键动作前后用 `api.log(...)` 留痕。
- 发送前缓冲：建议启动后先延时几秒，避免误触发。
- 数据读取判空：`get_data` 可能返回 None，必须判空再用。
- 数值类型明确：node 传整型，p/v/i 传可转 float 的数值。
- 单位自描述：变量命名建议带单位后缀（如 `power_w`、`voltage_v`）。

> 🔒 **当硬安全保护锁存后，运行时会丢弃控制发送命令并要求先复位。脚本层不应尝试绕过。**