

# 1. 大模型动态工具封装规范

工具配置页面支持选择工具类型，当选择“动态工具”时，需要输入工具执行的脚本，脚本目前仅支持python。

## 1.1 脚本说明

```
import xxx
def execute(tool_input: dict) -> dict:
    def sub_method():
        print('123')
    '''业务逻辑编写'''
    return dict(success=True, message='执行完成', data='执行结果')
```

1. 对外暴露的方法只能有一个，但方法内部支持定义子方法，如sub\_method。
2. 支持import导入整个模块，但不支持 from xxx import kkk的导入方式。
3. 方法名称不是固定的，可以随意更改，如execute可以更改为tool\_exe。
4. 输入参数只能是一个，类型dict不可以更改，内容是大模型基于参数配置提取的，业务属性从dict中获取即可，
5. 返回类型必须是dict，格式 dict(success=True, message='成功失败描述', data='执行结果')。
6. 返回结果中的data内容需要根据工具配置的输出类型相匹配，比如工具定义的输出类型是图表，data就要返回图表格式的内容。

## 1.2 脚本样例

### 1.2.1 rest接口调用

基于requests模块实现rest接口调用。

```
import requests
def execute(tool_input: dict) -> dict:
    params = dict(
        param1='param1',
        param2='param2'
    )
    response = requests.post(url, json=params, timeout=(10, 120))
    return dict(success=True, message='执行完成', data='执行结果')
```

### 1.2.2 from xxx import kkk方式替代

python开发过程中，可能会直接引用到模块下的类或者方法，`from shj.utils.cache import ShjCacheUtil`，这种方式目前脚本中是不支持的，但可以通过如下方式解决，先通过`importlib`引入模块，再通过`getattr`获取对应的类，然后再进行实例化调用。

```
import importlib
def execute(tool_input: dict) -> dict:
    module_name = 'shj.utils.cache'
    module = importlib.import_module(module_name)
    ShjCacheUtil = getattr(module, 'ShjCacheUtil')
    ShjCacheUtil().clear_args_confirm('xxx')
    return dict(success=True, message='执行完成', data='执行结果')
```

### 1.2.3 内部其他工具调用

动态工具脚本中支持调用工具管理中定义的其他工具，如下，通过`InterClient`类的`tool_exe`方法实现，3个参数，`tool_id`是要调用的工具的id，`tool_name`是要调用的工具的名称，`arguments`是该工具执行的入参（dict格式，包括集群名称和应用id）。

```
import importlib
def execute(tool_input: dict) -> dict:
    tool_id = 'current_cpu_query',
    tool_name = 'current_cpu_query',
    cluster = tool_input['cluster']
    app_id = tool_input['appId']
    arguments = dict(cluster=cluster, appId=app_id)
    module_name = 'shj.client.client'
    module = importlib.import_module(module_name)
    InterClient = getattr(module, 'InterClient')
    InterClient.tool_exe(tool_id, tool_name, arguments)
    return dict(success=True, message='执行完成', data='执行结果')
```