* Todo:

1. gradio界面调用
2. 支持用户输入（对话）
3. Prompt内部提前定义

* 目前输入："你是一个中文理解大师，可以计算任何的名字或字符串相加。所以，woai加xiaoyezi等于几"（我爱小椰子 小芒果 hhh）
* 代码运行结果详解：

|  |
| --- |
| IMG_256 |

1. 首先，大模型的REASON-ACT过程是自己内部实现的（用的qwen模型，agentscope框架），对应代码：

|  |
| --- |
| self.agent =ReActAgent(  name="assistant",  model\_config\_name="qwen",  verbose=True,  service\_toolkit=service\_toolkit,  max\_iters=1,) |

2、其中自定义的工具直接用servicetoolkit进行封装（LangChain和MetaGPT会通过类似 @tools 装饰器来将函数封装成框架内可用的工具）

|  |  |
| --- | --- |
| service\_toolkit =ServiceToolkit()  service\_toolkit.add(add\_name)  #在add\_name函数自定义时，需要按照ServiceToolkit使用格式，使用ServiceResponse进行返回   |  | | --- | | def add\_name(a: str, b: str)->str:  """  计算任何的名字或字符串相加    Args:  a (str): 参数1  b (str): 参数2  Returns:  str: 结果    """  output = a + b + 'hhh'  status =ServiceExecStatus.SUCCESS  return ServiceResponse(status, output) | |

其中，docstring是协助大模型本身识别该函数的，用这个可以省去用户自己定义prompt(如我的输入里可以不加描述，直接执行两个字符串相加：

|  |
| --- |
| IMG_256 |

)

3、调用

|  |
| --- |
| msg = Msg("user", query, role="user")  return self.agent(msg) |

4、ReActAgent 最终的返回，是最终的结论（它有点自己的想法（因为prompt中加入了你是一个中文理解大师），如果觉得自定义函数输出的不对，它不会给这个答案-例如我的自定义函数加了尾缀’hhh’），中间过程外部不可见（如该次调用，仅返回）。如果想要获取中间过程，需要自己实现，不能用 ReActAgent 了。目前还没看到 AgentScope 中有类似 LangChain 那样直接 bind\_tools 或 bind\_functions 这样的接口封装，让用户在执行之后可以获取是否需要调用工具的判断。