DeepAgents（create_deep_agent）

---组成：核心部分（模型、sys提示词、工具），特征（**存储后端**、子代理、中断）

---特色：todo_list、文件系统工具、子代理委派

---Agent harness：

文件系统访问

large tool result eviction（工具结果淘汰到文件系统中）

pluggable storage backends（状态、fileSys、store、路由、自定义）

task delegation（主代理为隔离的多步任务task创建临时的子代理）

conversation history summarization（pre_model的中级件）

Dangling tool call repair（会给被中断或取消的AI.toolcall创建已取消的ToolMessage响应）

to-do list tracking（write_todos）

human-in-the-loop（**todo**）

prompt caching（？？？必需）

---Backends（存储后端）

StateBackends

FileSystemBackends

StoreBackends

CompositeBacked（组合state和store：短长期...）

---todo：

1.user a virtual filesystem（**todo？？？**）custom backend to project a remote or database filesystem、

2.企业路径限制：子类化的存储后端、通用包装器

---子代理：

1.两种申明方式：字典、create_agent+CompliedSubAgent

2.通用子代理：防止过长的中间调用

**---最佳实践：**

**1.clear description**

**2.detailed system prompts**

**3.Minimize tool sets**

**4.Choose models by task**

**5.Return concise results**

---Common patterns 常见模式

Multiple specialized subagents

---问题解决：

1.子代理未被调用：描述更具体、指示deepAgent去委派任务

2.上下文膨胀：系统提示词提示返回结果、使用文件系统保存数据

3.选择错误的代理：代理的描述要有区分

---Human-in-the-loop

1.配置interrupt_on（字典：{"方法名":"True/False/自定义"}）

2.自定义：决策类型：approve、reject、edit（allowed_decisions）

3.处理中断：中断信息打印（！！！）+Command

4.多工具调用：所有中断会组合到一个中断中，须顺序决策

5.编辑工具参数

6.子代理中断：在子代理里声明，会覆盖主代理设置

---**最佳实践：**

**1.检查点**

**2.恢复时使用相同的线程id**

**3.决策列表必须与 action_requests 的顺序匹配**

**4.根据风险等级配置不同的决策属性**

---长期记忆：

1.如何运行：短期的文件系统是存储在状态里的，长期的存储后端可以用于长期记忆的存储。（跨线程持久化）

2.使用场景：用户偏好，自我改进指令、知识库、研究项目（/memories/research/sources.txt - List of sources found - /memories/research/notes.txt - Key findings and notes）

3.存储实现：InMemoryStore、PostgresStore

---**最佳实践：**

**1.使用描述性的路径**

**2.记录记忆结构**

Document the memory structure

**记录记忆结构**

Tell the agent what's stored where in your system prompt:

告诉代理在您的系统提示中存储了什么：

```
Your persistent memory structure:
- /memories/preferences.txt: User preferences and settings
- /memories/context/: Long-term context about the user
- /memories/knowledge/: Facts and information learned over time
```

**3.修剪旧信息**: 怎么修剪？人工?

**4.选择合适的存储**：开发、生产、多租户

- **Multi-tenant**: Consider using assistant_id-based namespacing in your store

  多租户：考虑在您的存储中使用基于 assistant_id 的命名空间

---Middleware

1.深度代理自带的中间件（每一个都可以被create_agent所使用）

Each feature is implemented as separate middleware. When you create a deep agent with `create_deep_agent`, we automatically attach `TodoListMiddleware`, `FilesystemMiddleware`, and `SubAgentMiddleware` to your agent.

2.todoList（create_agent使用该middleware）

```
# Custom planning instructions can be added via middleware
middleware=[
    TodoListMiddleware(
        system_prompt="Use the write_todos tool to..."  # Optional: Custom add
    ),
]
```

3.FilesystemMiddleware和SubAgentMiddleware类似（**SubAM使用额外的中间件？**）

A subagent is defined with a **name**, **description**, **system prompt**, and **tools**. You can also provide a subagent with a custom **model**, or with additional **middleware**. This can be particularly useful when you want to give the subagent an additional state key to share with the main agent.

子代理由**名称** 、 **描述** 、 **系统提示**和**工具**定义。您还可以为子代理提供自定义**模型**或额外的**中间件** 。当您希望为子代理提供一个与主代理共享的额外状态键时，这将特别有用。

---使用cli：

1.DeepAgentCli的内置功能：文件操作、shell命令执行、网络搜索（配置Tavily秘钥）、http请求、任务规划和跟踪、记忆存储与检索（跨会话）、人机交互（敏感工具操作）

2.

哔哩哔哩 (゜-゜)つロ 干杯~-bilibili

| aa | aa | aa |
| --- | --- | --- |
| aa | aa | aa |
| aa | aa | aa |
| aa | aa | aa |