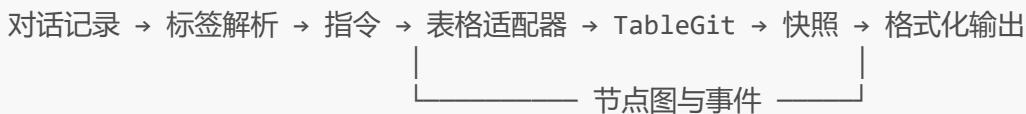


# Table Memory Engine 概览

Table Memory Engine 聚焦于“表格即长期记忆”的业务场景，提供一套独立于视图层的工作流引擎。通过节点图的方式，将对话记录中的指令解析为表格操作，再将最新的表格快照格式化注入后续对话上下文，形成闭环记忆能力。

## 架构全景



- **核心层 (core/)**：定义对话、标签、表格快照等领域模型，并实现组合式标签解析器、格式化器注册表、事件总线。
- **运行时 (runtime/)**：提供 `FlowBuilder` 和 `NodeRuntime`，支撑节点图的构建、执行与状态管理。
- **节点层 (nodes/)**：实现内置节点（如 `LoadTable`、`ParseTags`、`ApplyChanges`、`FormatPrompt`），并冒泡注册接口便于扩展。
- **适配层 (adapters/)**：目前包含 `TableGitAdapter`，将抽象命令映射到 `table-git` 的版本化表格操作，可扩展为其他存储后端。

## 关键概念

- **Flow Graph**：由节点、连线和配置组成的工作流定义，可通过 DSL 构建或序列化为 JSON。
- **Node**：执行单个职责的组件，拥有 schema、配置验证与执行逻辑。节点之间通过变量和事件共享数据。
- **Tag Instruction**：对话中标签解析器输出的结构化指令，描述目标表格、操作类型及元数据。
- **Table Adapter**：连接不同表格存储的适配层，负责载入、应用变更、生成快照。
- **Formatter**：将最新快照与执行上下文组合，输出给下游模型所需的 Prompt 或结构化数据。
- **Event Bus**：贯穿生命周期的事件系统，用于外部订阅或自定义扩展（如日志、埋点、实时同步）。

## 典型使用流程

1. **初始化 TableGit 与适配器**：准备基础表格结构与版本化能力。
2. **构建节点图**：通过 `FlowBuilder` 串联加载、解析、应用、格式化等步骤。
3. **注册内置节点**：调用 `registerBuiltinNodes` 注入标准节点与格式化器。
4. **执行运行时**：传入对话记录、上下文变量与辅助服务，执行节点图。
5. **消费结果**：从 `NodeRuntime` 返回的状态中读取快照、格式化结果或自定义事件。

## 适用场景

- 多轮对话中持续维护“记忆表格”的 AI 助手。
- 根据用户对话中的标签指令自动更新知识库或 CRM。
- 需要将表格状态注入 Prompt 的生成式应用。
- 期望通过节点方式搭建数据操作流程的低代码平台。

更多细节请参阅：

- [getting-started.md](#)：从零构建记忆工作流
- [api-reference.md](#)：核心 API 与类型说明
- [node-library.md](#)：内置节点和扩展机制