**LangChain 核心組成架構**

LangChain 主要由以下幾個核心組件構成：

**1. Models 模型**

* **功能**：負責與各種 LLM 和嵌入模型進行互動
* **主要組件**：
  + LLMs (大型語言模型)
  + Chat Models (聊天模型)
  + Text Embedding Models (文本嵌入模型)
* **目的**：提供統一的介面來使用不同的模型，如 OpenAI、Anthropic、Hugging Face 等

**2. Prompts 提示詞**

* **功能**：管理和優化與 LLM 的輸入
* **主要組件**：
  + Prompt Templates (提示詞模板)
  + Output Parsers (輸出解析器)
  + Example Selectors (範例選擇器)
* **目的**：幫助開發者建構結構化和動態的提示，提高模型回應的質量

**3. Memory 記憶**

* **功能**：管理對話狀態和歷史
* **主要組件**：
  + Conversation Buffer
  + Summary Memory
  + Vector Store Memory
* **目的**：使應用能夠維護上下文並進行有狀態的對話

**4. Chains 鏈條**

* **功能**：將多個組件組合成可重用的工作流
* **主要組件**：
  + LLM Chain
  + Sequential Chain
  + Transform Chain
* **目的**：實現複雜的多步驟操作，將不同的組件串聯起來

**5. Indexes & Retrievers 索引與檢索器**

* **功能**：管理和檢索外部數據
* **主要組件**：
  + Document Loaders (文檔加載器)
  + Text Splitters (文本分割器)
  + Vector Stores (向量存儲)
* **目的**：實現高效的文檔處理和檢索功能

**6. Tools & Agents 工具與代理**

* **功能**：實現自動化決策和執行
* **主要組件**：
  + Tools (工具集)
  + Agents (代理)
  + Agent Executors (代理執行器)
* **目的**：使 AI 能夠根據需求選擇和使用適當的工具完成任務

**7. Callbacks & Logging 回調與日誌**

* **功能**：提供監控和調試功能
* **主要組件**：
  + Callbacks Handlers
  + Loggers
  + Tracers
* **目的**：幫助開發者追蹤和優化應用性能

**架構特點**

1. **模組化設計**
   * 各組件可以獨立使用
   * 支持靈活組合和擴展
2. **抽象層設計**
   * 提供統一的介面
   * 降低切換不同提供商的成本
3. **可擴展性**
   * 支持自定義組件
   * 易於整合新的功能和服務
4. **開發友好**
   * 提供豐富的文檔和範例
   * 具有完整的類型提示支持