# All types of AgentTypes

Here's a comprehensive breakdown of the built-in agent types in LangChain's `AgentType` enum, including their intended use cases:

---

## 🧠 1. ZERO_SHOT_REACT_DESCRIPTION

- **What it does**: A ReAct-style (Reason + Act) agent that uses LLM reasoning followed by tool calls—without examples—based solely on tools' descriptions.
- **When to use**: General-purpose tasks like answering questions, web search, math via tools.

---

## 📚 2. REACT_DOCSTORE

- **What it does**: Similar to zero-shot ReAct, but with built-in access to a document store (e.g., a vector DB or knowledge base) for additional lookups.
- **When to use**: When your agent must reference structured knowledge or indexed documents in addition to tools.

---

## 🔍 3. SELF_ASK_WITH_SEARCH

- **What it does**: Breaks a complex question into sub-questions, uses search tools on each, then synthesizes the final answer.
- **When to use**: Complex queries requiring iterative research and fact-checking, like historical events or multi-step info retrieval.

---

## 💬 4. CONVERSATIONAL_REACT_DESCRIPTION

- **What it does**: A conversational version of ReAct that includes memory/context support for ongoing dialogues.
- **When to use**: Chatbots, helpdesk agents, or use cases where follow-up questions and context are key.

---

## 🗣️ 5. CHAT_ZERO_SHOT_REACT_DESCRIPTION

- **What it does**: Zero-shot ReAct agent optimized specifically for chat-based LLMs.
- **When to use**: Similar to `ZERO_SHOT_REACT_DESCRIPTION`, but with chat-model formatted prompts.

---

## 🔄 6. CHAT_CONVERSATIONAL_REACT_DESCRIPTION

- **What it does**: A chat-friendly conversational ReAct agent that retains chat history.
- **When to use**: When you need both follow-up context and multi-step reasoning in a conversational setting.

---

## 🛠️ 7. STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION

- **What it does**: A structured ReAct agent built for chat models that supports multi-input tools (i.e., tools requiring structured arguments).
- **When to use**: Use when you have chat-based LLMs and tools that expect structured inputs (e.g., functions or APIs.

---

## 🧩 8. OPENAI_FUNCTIONS

- **What it does**: Uses OpenAI's function-calling interface to let the LLM decide when to invoke functions via structured tool schemas.
- **When to use**: Best for models and ecosystems supporting OpenAI function-call mechanisms (ChatGPT GPT-4, etc.).

---

## 📦 9. OPENAI_MULTI_FUNCTIONS

- **What it does**: Similar to `OPENAI_FUNCTIONS` but supports multiple functions being called in a single response.
- **When to use**: When multiple tools or function calls are needed in a single interaction—like complex workflows.

---

## 🧾 Choosing the right agent

| AgentType | Best Use Case |
|---|---|
| `ZERO_SHOT_REACT_DESCRIPTION` | Simple tool chaining with zero-shot reasoning |
| `REACT_DOCSTORE` | Tool + document lookups (e.g., vector DB) |
| `SELF_ASK_WITH_SEARCH` | Multi-step research via search |
| `CONVERSATIONAL_REACT_DESCRIPTION` | Context-aware conversational agents |
| `CHAT_ZERO_SHOT_REACT_DESCRIPTION` | Chat-agent zero-shot tool use |
| `CHAT_CONVERSATIONAL_REACT_DESCRIPTION` | Chat with memory + multi-step reasoning |
| `STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION` | Chat with structured API/function calls |
| `OPENAI_FUNCTIONS` | Function-calling via OpenAI schema |
| `OPENAI_MULTI_FUNCTIONS` | Complex multi-function workflows |

---

- Use **ReAct agents** (`zero-shot`, `conversational`, structured) for reasoning + tools.
- Pick **Search** (`self-ask`) for multi-step research tasks.
- Choose **Conversational** variants when you need memory/context across exchanges.
- Opt for **Function-based agents** (`OPENAI_FUNCTIONS`, `OPENAI_MULTI_FUNCTIONS`) for structured tool/function invocation using OpenAI's function-calling.

| `AgentType` | **Supports Multi-Input?** | **Supports Single-Input (String)?** | **Description** |
|---|---|---|---|
| `ZERO_SHOT_REACT_DESCRIPTION` | ❌ No | ✅ Yes | Basic ReAct agent, single string input to tools |
| `REACT_DOCSTORE` | ❌ No | ✅ Yes | ReAct agent with document store lookup |
| `SELF_ASK_WITH_SEARCH` | ❌ No | ✅ Yes | Decomposes questions and uses search |
| `CONVERSATIONAL_REACT_DESCRIPTION` | ❌ No | ✅ Yes | ReAct with conversational memory |
| `CHAT_ZERO_SHOT_REACT_DESCRIPTION` | ❌ No | ✅ Yes | ReAct for chat models |
| `CHAT_CONVERSATIONAL_REACT_DESCRIPTION` | ❌ No | ✅ Yes | Chat ReAct with memory |
| **`STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION`** | ✅ Yes | ✅ Yes | ✅ Best for multi-input tools with chat models |

| | | | |
|---|---|---|---|
| `OPENAI_FUNCTIONS` | ✅ Yes | ✅ Yes | ✅ Uses OpenAI’s function-calling (structured tool schemas) |
| `OPENAI_MULTI_FUNCTIONS` | ✅ Yes | ✅ Yes | ✅ Supports calling multiple structured tools |