**OpenAI Agent SDK**

**Overview**

The **OpenAI Agent SDK** is a framework designed to simplify building intelligent agents powered by OpenAI's language models. It enables developers to create flexible, modular agents that can interpret natural language instructions, manage contextual state, and execute complex workflows autonomously or interactively.

The SDK focuses on making AI agents more maintainable, reusable, and easy to extend for a wide range of applications, including chatbots, automation workflows, intelligent assistants, and more.

**Key Concepts**

**Agent**

An **Agent** is the core entity that processes instructions, manages context, and produces outputs. Agents use OpenAI models internally to understand and generate responses or actions based on inputs.

* Agents can be simple or complex depending on your use case.
* They maintain a context or state which can be passed through steps or tasks.
* Designed to be reusable and composable.

**Context**

Context represents the current state or environment the agent operates in. This could include user input, previous dialogue history, external data, or any relevant information needed for decision-making.

* Context is often modeled as a generic type or data class.
* Helps keep agent interactions stateful and meaningful.

**Instructions**

Instructions are callable units or functions representing discrete tasks or actions the agent can perform. Instructions accept input, operate on context, and produce output.

* Instructions are composable.
* Can represent API calls, database queries, or natural language tasks.

**Runner**

The Runner orchestrates the execution of agents and their instructions, managing the flow of data and handling retries or error handling.

**Features**

* **Modular design**: Easily compose multiple instructions into complex workflows.
* **Generic context handling**: Define flexible, typed contexts tailored to your application.
* **Typed interfaces**: Strong typing support (TypeScript, Python typing) for better developer experience.
* **Easy integration**: Works seamlessly with OpenAI’s API for prompt generation and response parsing.
* **Extensible**: Add custom instruction types and context handlers.

**Typical Use Cases**

* Building chatbots with contextual awareness and memory.
* Automating multi-step tasks involving language understanding.
* Intelligent assistants capable of executing commands and fetching external data.
* Custom workflows powered by OpenAI models with structured input/output.