Model Context Protocol (MCP)

Model Context Protocol (MCP)

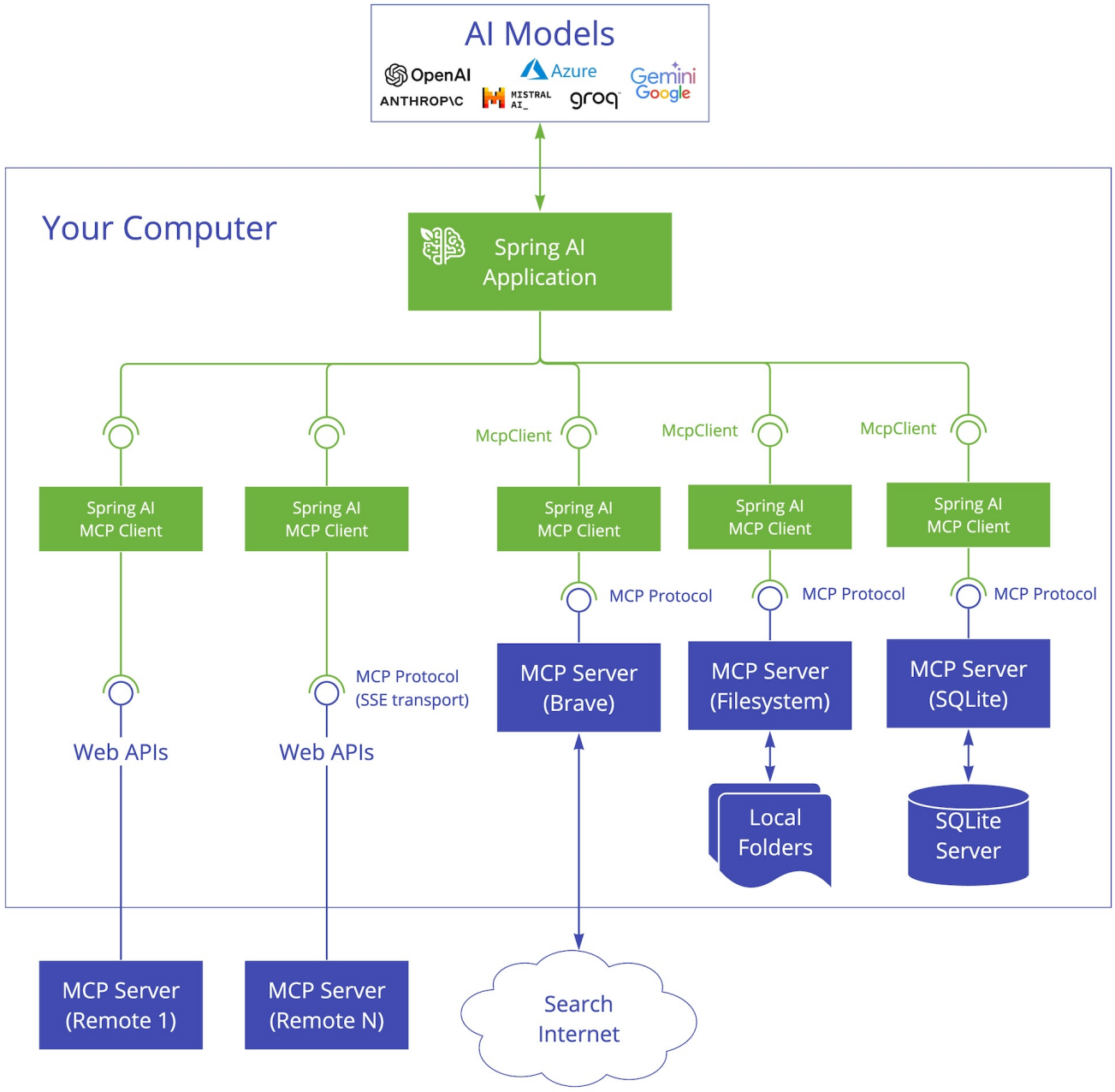
> Spring AI MCP
> Getting Started
> Example Demos

The Model Context Protocol (MCP) is an open protocol that standardizes how applications provide context to Large Language Models (LLMs). MCP provides an unified way to connect AI models to different data sources and tools, making integration seamless and consistent. It helps you build agents and complex workflows on top of LLMs. LLMs frequently need to integrate with data and tools, and MCP provides: - A growing list of pre-built integrations that your LLM can directly plug into - The flexibility to switch between LLM providers and vendors

Spring AI MCP

> Spring AI MCP is an experimental project and subject to change.

Spring AI MCP is an experimental project that provides Java and Spring Framework integration for the Model Context Protocol. It enables Spring AI applications to interact with different data sources and tools, through a standardized interface, supporting both synchronous and asynchronous communication patterns.



The Spring AI MCP implements a modular architecture with the following components:

- Spring AI Application: Uses Spring AI framework to build Generative AI applications that want to access data through MCP
- Spring MCP Clients: Spring AI implementation of the MCP protocol that maintain 1:1 connections with servers
- MCP Servers: Lightweight programs that each expose specific capabilities through the standardized Model Context Protocol
- Local Data Sources: Your computer's files, databases, and services that MCP servers can securely access
- Remote Services: External systems available over the internet (e.g., through APIs) that MCP

servers can connect to

The architecture supports a wide range of use cases, from simple file system access to complex multi-model AI interactions with database and internet connectivity.

Getting Started

Add the SDK to your Maven project:

Maven | Gradle

```xml
<dependency>
    <groupId>org.springframework.experimental</groupId>
    <artifactId>spring-ai-mcp</artifactId>
    <version>0.2.0</version>
</dependency>
```

The Spring AI MCP milestones are not available in the Maven Central Repository yet. Please add tehe Spring Milestone Repository to your build file to access the Spring AI MCP artifacts:

```
<repositories>
  <repository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>https://repo.spring.io/libs-milestone-local</url>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

The latter builds on top of mcp-core to provide some useful Spring AI abstractions, such as `McpFunctionCallback`.

Now create an `McpClient` to regester the MCP Brave server tools with your ChatClient and let the LLM call them:

```java
// https://github.com/modelcontextprotocol/servers/tree/main/src/brave-search
var stdioParams = ServerParameters.builder("npx")
        .args("-y", "@modelcontextprotocol/server-brave-search")
        .addEnvVar("BRAVE_API_KEY", System.getenv("BRAVE_API_KEY"))
        .build();

var mcpClient = McpClient.using(new StdioClientTransport(stdioParams)).sync();
```

```
var init = mcpClient.initialize();

var chatClient = chatClientBuilder
        .defaultFunctions(mcpClient.listTools(null)
                .tools()
                .stream()
                .map(tool -> new McpFunctionCallback(mcpClient, tool))
                .toArray(McpFunctionCallback[]::new))
        .build();

String response = chatClient
        .prompt("Does Spring AI supports the Model Context Protocol? Please
provide some references.")
        .call().content();
```

Example Demos

There is a growing list of MCP Servers that you can use with Spring AI MCP. Explore these MCP examples in the spring-ai-examples/model-context-protocol repository:

- SQLite Simple - Demonstrates LLM integration with a database
- SQLite Chatbot - Interactive chatbot with SQLite database interaction
- Filesystem - Enables LLM interaction with local filesystem folders and files
- Brave - Enables natural language interactions with Brave Search, allowing you to perform internet searches.

Copyright © 2005 - 2025 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries.
Terms of Use • Privacy • Trademark Guidelines • Thank you • Your California Privacy Rights • Cookie Settings

Apache®, Apache Tomcat®, Apache Kafka®, Apache Cassandra™, and Apache Geode™ are trademarks or registered trademarks of the Apache Software Foundation in the United States and/or other countries. Java™, Java™ SE, Java™ EE, and OpenJDK™ are trademarks of Oracle and/or its affiliates. Kubernetes® is a registered trademark of the Linux Foundation in the United States and other countries. Linux® is the registered trademark of Linus Torvalds in the United States and other countries. Windows® and Microsoft® Azure are registered trademarks of Microsoft Corporation. “AWS” and “Amazon Web Services” are trademarks or

registered trademarks of Amazon.com Inc. or its affiliates. All other trademarks and copyrights are property of their respective owners and are only mentioned for informative purposes. Other names may be trademarks of their respective owners.