

快速开始

Spring AI Alibaba 实现了与阿里云通义模型的完整适配，接下来，我们将学习如何使用 `spring ai alibaba` 开发一个基于通义模型服务的智能聊天应用。

快速体验示例

注意：因为 Spring AI Alibaba 基于 Spring Boot 3.x 开发，因此本地 JDK 版本要求为 17 及以上。

1. 下载项目 运行以下命令下载源码，进入 helloworld 示例目录：



专家答疑

```
git clone --depth=1 https://github.com/alibaba/spring-ai-alibaba.git
cd spring-ai-alibaba/spring-ai-alibaba-examples/helloworld-example
```

2. 运行项目 首先，需要获取一个合法的 API-KEY 并设置 `AI_DASHSCOPE_API_KEY` 环境变量，可跳转 [阿里云百炼平台](#) 了解如何获取 API-KEY。

```
export AI_DASHSCOPE_API_KEY=${REPLACE-WITH-VALID-API-KEY}
```

启动示例应用：

```
./mvnw compile exec:java -Dexec.mainClass="com.alibaba.cloud.ai.example.helloworld.HelloWorldEx
```

访问 `http://localhost:8080/ai/chat?input=给我讲一个笑话吧`，向通义模型提问并得到回答。

1. 添加依赖

首先，需要在项目中添加 `spring-ai-alibaba-starter` 依赖，它将通过 Spring Boot 自动装配机制初始化与阿里云通义大模型通信的 `ChatClient`、`ChatModel` 相关实例。

```
<dependency>
  <groupId>com.alibaba.cloud.ai</groupId>
  <artifactId>spring-ai-alibaba-starter</artifactId>
  <version>1.0.0-M2.1</version>
</dependency>
```

注意：由于 `spring-ai` 相关依赖包还没有发布到中央仓库，如出现 `spring-ai-core` 等相关依赖解析问题，请在您项目的 `pom.xml` 依赖中加入如下仓库配置。

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



专家答疑

2. 注入 ChatClient

接下来，在普通 Controller Bean 中注入 `ChatClient` 实例，这样你的 Bean 就具备与 AI 大模型智能对话的能力了。

```
@RestController
@RequestMapping("/ai")
public class ChatController {

    private final ChatClient chatClient;
```

```
@GetMapping("/chat")
public String chat(String input) {
    return this.chatClient.prompt()
        .user(input)
        .call()
        .content();
}
```

以上示例中，ChatClient 调用大模型使用的是默认参数，Spring AI Alibaba 还支持通过 DashScopeChatOptions 调整与模型对话时的参数，DashScopeChatOptions 支持两种不同维度的配置方式：

1. 全局默认值，即 ChatClient 实例初始化参数

可以在 application.yaml 文件中指定 spring.ai.dashscope.chat.options.* 或调用 ChatClient.Builder.defaultOptions(options)、DashScopeChatModel(api, options) 完成配置初始化。



专家答疑

2. 每次 Prompt 调用前动态指定

```
ChatResponse response = chatModel.call(
    new Prompt(
        "Generate the names of 5 famous pirates.",
        DashScopeChatOptions.builder()
            .withModel("qwen-plus")
            .withTemperature(0.4F)
            .build()
    ));
```

关于 DashScopeChatOptions 配置项的详细说明，请查看参考手册。

更多资料

基础示例与API使用

本页内容 >

概述

高级示例

- [使用 RAG 开发 Q&A 答疑助手](#)
- [具备连续对话能力的聊天机器人](#)

 编辑此页

← 上一页
核心概念

下一页 →
Chat Client


专家答疑