

Ollama 部署 Deepseek R1

官网: <https://www.deepseek.com/>

Github: <https://github.com/deepseek-ai>

Ollama: <https://ollama.com/>

Docker Compose

部署一个 Ollama 和 open-webui 服务:

```
services:

  ollama:
    volumes:
      - ./models:/root/.ollama # 将本地文件夹挂载到容器中的 /root/.ollama 目录 (模型下载位置)
    container_name: ollama
    pull_policy: always
    tty: true
    restart: unless-stopped
    image: ollama/ollama:latest
    ports:
      - 11434:11434 # Ollama API 端口

  open-webui:
    build:
      context: .
      args:
        OLLAMA_BASE_URL: '/ollama'
      dockerfile: Dockerfile
    image: ghcr.io/open-webui/open-webui:main
    container_name: open-webui
    volumes:
      - ./open-webui:/app/backend/data # 前端页面数据挂载位置
    depends_on:
      - ollama
    ports:
      - ${OPEN_WEBUI_PORT-3005}:8080
    environment:
      - 'OLLAMA_BASE_URL=http://ollama:11434'
      - 'WEBUI_SECRET_KEY='
    extra_hosts:
      - host.docker.internal:host-gateway
    restart: unless-stopped
```

安装 DeepSeek-R1

进入 docker ollama 容器下载 deepseek-r1 模型

```
# 进入容器
$ docker exec -it ollama bash

# 查看 ollama 已有的模型 (第一次下载没有正常)
$ root@c5e5ff20a533:/# ollama list
```

NAME	ID	SIZE	MODIFIED
llama3:latest	365c0bd3c000	4.7 GB	7 months ago
qwen:4b	d53d04290064	2.3 GB	7 months ago

选择下载 8b（可以根据机器环境选择不同的模型）

```
ollama run deepseek-r1:8b
```

下载成功如下所示

open-webui 使用

如果出现模型失败的错误，尝试更新先 ollama 版本之后重试！

在右上角选择模型：

使用演示：

看起来效果不错，歪瑞古德！

Spring AI Alibaba 调用

接下来演示如何 Spring AI Alibaba 完成一个简单的 Chat 应用。

pom.xml

因为我们使用 ollama 部署 deepseek r1，所以这里使用 ollama starter。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>indi.yuluo</groupId>
        <artifactId>examples</artifactId>
        <version>1.0-SNAPSHOT</version>
    </parent>

    <groupId>indi.yuluo.deepseek</groupId>
    <artifactId>deepseek-r1-saa</artifactId>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <version>3.3.4</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.ai</groupId>
            <artifactId>spring-ai-ollama-spring-boot-starter</artifactId>
            <version>1.0.0-M6</version>
```

```

        </dependency>
    </dependencies>

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

</project>

```

application.yml

```

server:
  port: 8080

spring:
  application:
    name: deepseek-r1-saa

ai:
  ollama:
    base-url: http://localhost:11434
  chat:
    model: deepseek-r1:8b

```

启动类

```

package indi.yuluo.deepseek;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

/**
 * @author yuluo
 * @author <a href="mailto:yuluo08290126@gmail.com">yuluo</a>
 */

@SpringBootApplication
public class DeepSeekChatApplication {

    public static void main(String[] args) {

        SpringApplication.run(DeepSeekChatApplication.class, args);
    }

}

```

controller

```
package indi.yuluo.deepseek.controller;

import jakarta.servlet.http.HttpServletResponse;
import reactor.core.publisher.Flux;

import org.springframework.ai.chat.client.ChatClient;
import org.springframework.ai.chat.prompt.Prompt;
import org.springframework.ai.ollama.OllamaChatModel;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * @author yuluo
 * @author <a href="mailto:yuluo08290126@gmail.com">yuluo</a>
 */

@RestController
@RequestMapping("/deepseek/chat")
public class DeepSeekController {

    private final ChatClient chatClient;

    public DeepSeekController (OllamaChatModel chatModel) {

        this.chatClient = ChatClient.builder(chatModel).build();
    }

    @GetMapping("/{prompt}")
    public Flux<String> chat(
        @PathVariable(value = "prompt") String prompt,
        HttpServletResponse response
    ) {

        response.setCharacterEncoding("UTF-8");
        return this.chatClient.prompt(new Prompt(prompt)).stream().content();
    }

}
```

浏览器请求测试

DeepSeek4j 调用

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>indi.yuluo</groupId>
        <artifactId>examples</artifactId>
        <version>1.0-SNAPSHOT</version>
    </parent>

    <groupId>indi.yuluo.deepseek</groupId>
    <artifactId>deepseek4j</artifactId>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <version>3.3.4</version>
        </dependency>

        <dependency>
            <groupId>io.github.pig-mesh.ai</groupId>
            <artifactId>deepseek-spring-boot-starter</artifactId>
            <version>1.4.1</version>
        </dependency>
    </dependencies>

</project>

```

application.yml 配置

```

deepseek:
  base-url: http://127.0.0.1:11434/v1
  model: deepseek-r1:8b
  api-key: deepseek
  default-system-prompt: false

server:
  port: 8080

```

controller

```

package indi.yuluo.deepseek.controller;

import io.github.pigmesh.ai.deepseek.core.DeepSeekClient;
import jakarta.servlet.http.HttpServletResponse;
import reactor.core.publisher.Flux;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * @author yuluo

```

```

* @author <a href="mailto:yuluo08290126@gmail.com">yuluo</a>
*/

@RestController
@RequestMapping("/deepseek4j")
public class DeepSeek4JController {

    @Autowired
    private DeepSeekClient deepSeekClient;

    @GetMapping(value = "/chat/{prompt}")
    public Flux<String> chat(
        @PathVariable(value = "prompt") String prompt,
        HttpServletResponse response
    ) {

        response.setCharacterEncoding("UTF-8");

        return deepSeekClient.chatFluxCompletion(prompt).map(
            chatCompletionResponse -> {

                System.out.println(chatCompletionResponse.choices().get(0).delta().toString());
                return
                chatCompletionResponse.choices().get(0).delta().content();
            }
        );
    }
}

```

DeepSeek4j 透出 reasoning content

当 `default-system-prompt` 为 false 时

```

deepseek:
  base-url: http://127.0.0.1:11434/v1
  model: deepseek-r1:8b
  api-key: deepseek
  default-system-prompt: false

```

输出为:

你好！很高兴见到你，有什么我可以帮忙的吗？无论是问题、建议还是闲聊，我都在这儿为你服务。😊

当 `default-system-prompt` 为 true 时

```

deepseek:
  base-url: http://127.0.0.1:11434/v1
  model: deepseek-r1:8b
  api-key: deepseek
  default-system-prompt: true

```

输出为:

用户说“你好”，这是个常见的问候，我应该用中文回答，保持亲切。 我是DeepSeek-R1，由中国公司DeepSeek开发的AI助手，可以处理中文和英文查询。 接下来，我会详细介绍一下DeepSeek-R1的功能和特点，让用户有更全面的了解。 你好！我是由中国公司深度求索（DeepSeek）开发的智能助手DeepSeek-R1。我擅长通过文本对话方式为您提供信息，解答问题并进行交流。如有任何需要，我会尽力帮助您，同时确保回答准确、有条理地呈现给您。如果你有任何具体的需求或疑问，请随时告诉我！

由此可见，似乎是由 prompt 控制的？