Spring Ai

Open AI is a organization, and chat GPT LLM (large language model) is a product of them

|  |  |
| --- | --- |
|  | <https://docs.spring.io/spring-ai/reference/index.html> |
| udemy sir git hub link e | <https://github.com/eazybytes/spring-ai> |
| open ai api link – here we should add 5 $ into our account and get the API key | <https://platform.openai.com/docs/overview> |
|  |  |

With spring AI we can integrate with AI model API’s (Open AI, ollama, AWS BedRock) in familiar spring interfaces

Gives you ChatClient, ImageModel, SpeechModel, and even Memory, all with Spring idioms.  
Makes AI integration feel like configuring a @Bean, not battling a black-box.

We can do below

* Prompt templates, streaming, chat options, prompt stuffing
* Built in advisors, custom advisors
* Converting AI output to java pojo beans

Typical Use Cases

Al-powered customer support: Build intelligent assistants with conversation memory and API integration.

Natural language search over documents: Summarize or query proprietary data using RAG and vector stores.

Personalized recommendations: Power intent-based search and recommendations with embeddings.

Data Processing: Add Al to pipelines for content moderation, transcription, or text generation.

ad

Invoking a LLM model

|  |  |
| --- | --- |
|  | classes for reference  Class OpenAIChatModel implements ChatModel { } |

We need a bean called Chatclient which we can build using chatClientBuilder

Sample code

Developers need to talk with ChatClient class, same like RESTClient

|  |
| --- |
| <http://localhost:8080/openai/query?search=is%20god%20exists>  @RequestMapping("/openai") @RestController public class ChatGPTController {   *ChatClient* chatClient;   @Autowired  public ChatGPTController(*ChatClient*.*Builder builder*) {  System.*out*.println("invoked cons of controller");  chatClient = *builder*.build();  }   @GetMapping("/query")  public String hitChatGPTLLM(@RequestParam("search") String *request*) {  System.*out*.println("received query is " + *request*);  *ChatClient*.*CallResponseSpec* call = chatClient.prompt(*request*).call();  return call.content();  } |