

Научная статья

УДК 004.89

EDN FCAWTW

DOI 10.17150/2713-1734.2023.5(4).409-416



Д.В. Кокотов

ООО «Цифровые привычки»,
г. Томск, Российская Федерация

Использование платформы ChatGPT при разработке программных сервисов на языке Kotlin

Аннотация. Ученые и инженеры во всем мире все больше внимания уделяют теме разработки искусственного интеллекта. Серьезным шагом в этой области стал запуск AI-платформы ChatGPT компанией OpenAI. Сервис ChatGPT способен генерировать ответы на запросы пользователей в разнообразных направлениях. AI может помочь выбрать товар, преобразовать аудио в текст или сгенерировать изображения, а интеграция ChatGPT в программные сервисы может существенно оптимизировать многие бизнес процессы.

Целью данной работы является общий обзор платформы ChatGPT, а также исследование возможности взаимодействия с ChatGPT на программном уровне используя разработанное web-приложение на языке Kotlin. В ходе исследования были сформулированы основные принципы работы ChatGPT, а также продемонстрированы примеры программного кода для взаимодействия с API платформы.

В заключительной части сформулированы общие выводы относительно перспективных возможностей использования ChatGPT для автоматизации бизнес-процессов в современных информационных системах.

Ключевые слова. Платформа ChatGPT, архитектура Transformer, искусственный интеллект, язык программирования Kotlin, фреймворк Spring, реактивный клиент, разработка программных сервисов, информационные системы, качество клиентского сервиса, web-приложения.

Информация о статье. Дата поступления: 19 апреля 2023 г.; дата принятия к публикации: 20 ноября 2023 г.; дата онлайн-размещения: 12 декабря 2023 г.

Original article

D.V. Kokotov

Digital Habits LLC,
Tomsk, Russian Federation

Using the ChatGPT Platform in the Development of Software Services in the Kotlin Language

Abstract. Scientists and engineers around the world are paying more and more attention to the development of artificial intelligence. A major step in this area was the launch of the ChatGPT AI platform by OpenAI. The ChatGPT service is capable of generating responses to user requests in a variety of ways. AI can help you select a product, convert audio to text or generate images, and integrating ChatGPT into software services can significantly streamline many business processes.

The purpose of this work is a general overview of the ChatGPT platform, as well as a study of the possibility of interacting with ChatGPT at the program level using the developed web application in the Kotlin language. During the study, the basic principles

of ChatGPT were formulated, and examples of program code for interacting with the platform API were demonstrated.

In the final part, general conclusions are formulated regarding the promising possibilities of using ChatGPT to automate business processes in modern information systems.

Keywords. ChatGPT platform, Transformer architecture, artificial intelligence, Kotlin programming language, Spring framework, reactive client, software services development, information systems, customer service quality, web applications.

Article info. Received 19 April, 2023; Accepted 20 November, 2023; Available online 12 December, 2023.

Введение

В настоящее время, сфера информационных технологий стремительно развивается. Все больше людей покупает товары и услуги в интернете. Компаниям необходимо постоянно улучшать свои сервисы, для того чтобы выдерживать конкуренцию на этом быстро растущем рынке. В результате, существенно повышается сложность информационных систем, а также стоимость их сопровождения. Поэтому компании ищут альтернативные подходы для улучшения своих приложений. И одним из крайне эффективных методов развития программных сервисов является внедрение искусственного интеллекта.

Существуют различные модели и алгоритмы для реализации искусственного интеллекта. В данной статье будет рассмотрено использование AI-платформы ChatGPT разработанной компанией OpenAI. ChatGPT (Chat Generative Pre-trained Transformer) — это чатбот на основе искусственного интеллекта, который позволяет обрабатывать тексты на естественном языке (англ. — Natural Language Processing, NLP) [1]. Он принимает входные данные и генерирует грамматически правильные и семантически верные ответы, понятные для человека. ChatGPT основан на модели глубокого обучения под названием Transformer, которая была представлена компанией Google в 2017 году [2]. Данный механизм по-разному взвешивает значимость каждой части входных данных и позволяет ChatGPT генерировать естественный язык намного более качественно¹. ChatGPT способен оптимизировать многие бизнес-процессы в разных направлениях бизнеса [3–6].

Компания OpenAI представила публичный API для доступа к платформе ChatGPT². А также разработала библиотеку на языке Python для взаимодействия с данным API. Но в настоящее время, на рынке очень много приложений написано на других языках программирования, в том числе на смежных Java и Kotlin. Поэтому разработка интерфейса для доступа к платформе ChatGPT на одном из этих языков видится перспективной задачей.

¹ Hugging Face. Transformers. URL: <https://huggingface.co/docs/transformers/index>.

² OpenAI platform API reference. URL: <https://platform.openai.com/docs/api-reference>.

Методы

Целью данной статьи является обзор функциональных возможностей платформы ChatGPT, а также исследование возможности взаимодействия с ChatGPT на программном уровне используя разработанное web-приложение на языке Kotlin³.

В процессе разработки web-приложения использовался фреймворк Spring Framework⁴, который в настоящее время является популярным фреймворком для создания корпоративных приложений, и предоставляет для этого мощную инфраструктуру. Программный код сервисов написан на современном языке программирования Kotlin, который является основным языком для разработки мобильных приложений на платформе Android. Для выполнения HTTP запросов использовался реактивный web-клиент фреймворка Spring WebFlux.

Результаты

Прежде всего, разработаем web-сервис для взаимодействия с API платформы ChatGPT, который сможет выполнять произвольные HTTP запросы (рис. 1).

```

1 import org.springframework.beans.factory.annotation.Value
2 import org.springframework.context.annotation.Bean
3 import org.springframework.context.annotation.Configuration
4 import org.springframework.http.MediaType
5 import org.springframework.http.client.reactive.ClientHttpRequest
6 import org.springframework.stereotype.Service
7 import org.springframework.web.reactive.function.BodyInserter
8 import org.springframework.web.reactive.function.client.WebClient
9
10 @Configuration
11 class WebClientConfiguration {
12     @Value("${openai-service.apikey}")
13     val apiKey: String? = null
14
15     @Bean
16     fun openAIWebClient(): WebClient {
17         return WebClient.builder()
18             .baseUrl("https://api.openai.com")
19             .defaultHeader("Authorization", "Bearer $apiKey")
20             .build()
21     }
22 }
23
24 @Service
25 class OpenAIService(val openAIWebClient: WebClient) {
26     final inline fun <reified T: Any> makePostRequest(
27         uri: String,
28         bodyInserter: BodyInserter<*, in ClientHttpRequest>,
29         contentType: MediaType
30     ): T? {
31         return openAIWebClient
32             .post()
33             .uri(uri)
34             .contentType(contentType)
35             .body(bodyInserter)
36             .retrieve()
37             .bodyToMono(T::class.java)
38             .block()
39     }
40 }

```

Рис. 1. Программный код сервиса для выполнения HTTP запросов к API платформы ChatGPT на языке Kotlin

³ Kotlin docs. URL: <https://kotlinlang.org/docs/home.html>.

⁴ Spring Framework Documentation. URL: <https://docs.spring.io/spring-framework/reference>.

Класс `WebClientConfiguration` описывает конфигурацию реактивного web-клиента из фрейворка Spring WebFlux и его основные параметры. В том числе приватный ключ `"OPENAI_API_KEY"`, необходимый для доступа к API. Данный ключ можно получить в личном кабинете на сайте платформы OpenAI (<https://platform.openai.com/>).

Сервис `OpenAIService` определяет универсальный метод для выполнения различных http запросов к API платформы ChatGPT. Сигнатура метода `makePostRequest()` содержит возвращаемое значение типа `T`, который определяется на этапе компиляции в зависимости от контекста вызова метода. Это позволяет привести ответ от платформы ChatGPT (приходящий как строка в формате JSON) к любому необходимому типу и использовать его на программном уровне.

В методе `makePostRequest()` есть следующие параметры:

- `uri` — универсальный идентификатор ресурса, к которому производится запрос;
- `bodyInserter` — тело отправляемого запроса;
- `contentType` — тип содержимого запроса.

Далее рассмотрим некоторые возможности использования разработанного сервиса.

AI-платформа ChatGPT способна генерировать тексты на многие темы. Это может быть полезно, например, для формирования рекомендаций по выбору товаров или услуг. Использование ChatGPT может существенно сэкономить время и деньги на маркетинге и рекламе, а также улучшить клиентский сервис. На рис. 2 представлен программный код метода `createCompletion()`, который генерирует текст на основе запроса пользователя.

```

1 import org.springframework.http.MediaType
2 import org.springframework.web.bind.annotation.PostMapping
3 import org.springframework.web.bind.annotation.RequestBody
4 import org.springframework.web.bind.annotation.RestController
5 import org.springframework.web.reactive.function.BodyInserters
6
7 data class CreateCompletionRequest {
8     val model: String,
9     val prompt: String?,
10    val max_tokens: Long?,
11    val temperature: Double?
12 }
13 data class CreateCompletionResponse(val choices: List<CompletionChoice>)
14 data class CompletionChoice(val text: String)
15
16 @RestController
17 class OpenAICompletionController(val openAIService: OpenAIService) {
18     @PostMapping("/completions")
19     fun createCompletion(@RequestBody request: CreateCompletionRequest)
20         : String? {
21         val body = BodyInserters.fromValue(request)
22         val response = openAIService.makePostRequest<CreateCompletionResponse>(
23             "/v1/completions", body, MediaType.APPLICATION_JSON
24         )
25         return response?.choices?.joinToString { it.text }?.trim()
26     }
27 }

```

Рис. 2. Программный код метода для генерации текста

Класс *CreateCompletionRequest* содержит параметры запроса к API платформы ChatGPT для генерации текста. Параметр *model* описывает модель, которая будет применяться при генерации ответа. Модели различаются по своим возможностям и стоимостью использования. Параметр *prompt* содержит входящий запрос в виде строки или массива строк. Параметр *max_tokens* определяет максимальное количество слов в сгенерированном ответе. Параметр *temperature* задает «креативность» генерируемого ответа, чем больше значение, тем выше дисперсия в ответах. Вызовем метод *createCompletion()* с телом запроса в формате JSON и продемонстрируем результат генерации текста (рис. 3).

Request:
<pre>{ "model": "text-davinci-003", "prompt": "Generate product recommendations with prices and explain selection based on user request: winter car tires", "max_tokens": 1000, "temperature": 0.5 }</pre>
Response:
<p>Based on the user request for winter car tires, we recommend the following products:</p> <ol style="list-style-type: none"> 1. Michelin X-Ice Snow Tires (Set of 4): \$499.99. These tires provide superior traction and grip on snowy and icy surfaces, making them ideal for winter driving conditions. They also provide a comfortable and quiet ride. 2. Bridgestone Blizzak Snow Tires (Set of 4): \$419.99. These tires are designed for winter driving and provide excellent traction and grip on snowy and icy surfaces. They also feature an advanced tread pattern for improved braking and cornering. 3. Goodyear UltraGrip Ice WRT Snow Tires (Set of 4): \$399.99. These tires are designed to provide superior traction and grip on snow and ice, making them perfect for winter driving conditions. <p>We selected these products based on their performance in winter driving conditions, as well as their price. All of these tires provide excellent traction and grip on snowy and icy surfaces, making them ideal for winter driving.</p>

Рис. 3. Пример параметров запроса и результата генерации текста

Сервис ChatGPT также способен генерировать изображения. Данный функционал можно использовать, например, для наполнения информационных разделов web-сайтов или социальных сетей. Разработаем методы для формирования соответствующих запросов (рис. 4 и 5).

```
1 data class CreateImageRequest(val prompt: String)
2 data class CreateImageResponse(val data: List<ImageInfo>)
3 data class ImageInfo(val url: String)
4
5 @RestController
6 class OpenAICreateImageController(val openAIService: OpenAIService) {
7     @PostMapping("/images/generations")
8     fun createImage(@RequestBody request: CreateImageRequest)
9         : CreateImageResponse? {
10         val body = BodyInserters.fromValue(request)
11         return openAIService.makePostRequest(
12             "/v1/images/generations", body, MediaType.APPLICATION_JSON
13         )
14     }
15 }
```

Рис. 4. Программный код метода для генерации изображений

```

1 data class EditImageRequest(val prompt: String, val sourceImageUrl: String)
2 data class EditImageResponse(val data: List<ImageInfo>)
3
4 @RestController
5 class OpenAIImageController(val openAIService: OpenAIService) {
6     @PostMapping("/images/edits")
7     fun editImage(@RequestBody request: EditImageRequest)
8         : EditImageResponse? {
9         val image = UrlResource(request.sourceImageUrl)
10        val builder = MultipartBodyBuilder()
11        builder.part("image", image)
12        .header("Content-Type", "application/octet-stream")
13        builder.part("prompt", request.prompt)
14        val body = BodyInserters.fromMultipartData(builder.build())
15        return openAIService.makePostRequest<EditImageResponse>(
16            "/v1/images/edits", body, MediaType.MULTIPART_FORM_DATA
17        )
18    }
19 }

```

Рис. 5. Программный код метода для редактирования изображений

В результате вызова данных методов ChatGPT возвращает ссылку на сгенерированные изображения, которые можно посмотреть или загрузить. Стоит отметить, что метод *editImage()* получает файл изображения источника по ссылке из интернета используя для этого специальный объект *UrlResource* фреймворка Spring.

ChatGPT также может обработать аудио файл и сформировать транскрипцию в виде текста. Это может быть очень полезно, например, для формирования протоколов длительных конференций или совещаний. Разработаем метод для реализации данного функционала (рисунок 6). В алгоритме метода *transcriptAudio()* прежде всего считывается файл из файловой системы, используя специальный объект *FileSystemResource* фреймворка Spring. А затем формируется тело запроса для обработки аудио в формате *"multipart/form-data"* и выполняется обращение к соответствующему API. Платформа ChatGPT способна обрабатывать файлы в различных форматах, в том числе mp3 и mp4.

```

1 data class TranscriptAudioRequest(val filePath: String, val model: String)
2 data class ProcessAudioResponse(val text: String)
3
4 @RestController
5 class OpenAIAudioController(val openAIService: OpenAIService) {
6     @PostMapping("/audio/transcriptions")
7     fun transcriptAudio(@RequestBody request: TranscriptAudioRequest)
8         : ProcessAudioResponse? {
9         val audio = FileSystemResource(request.filePath)
10        val builder = MultipartBodyBuilder()
11        builder.part("file", audio)
12        .header("Content-Type", "application/octet-stream")
13        builder.part("model", request.model)
14        val body = BodyInserters.fromMultipartData(builder.build())
15        return openAIService.makePostRequest<ProcessAudioResponse>(
16            "/v1/audio/transcriptions", body, MediaType.MULTIPART_FORM_DATA
17        )
18    }
19 }

```

Рис. 6. Программный код метода для преобразования аудио файла в текст

Вызовем разработанный метод *transcriptAudio()* используя тестовый аудио файл и продемонстрируем результат его обработки (рис. 7).

Request:
<pre>{ "filePath": "/audio.mp4", "model": "whisper-1" }</pre>
Response:
<pre>This is an audio example. This part is in English. А это предложение мы говорим на русском языке.</pre>

Рис. 7. Пример вызова сервиса для преобразования аудио в текст

Заключение

Создание AI-платформы ChatGPT безусловно открывает людям широкие возможности для оптимизации и развития информационных систем. Уже сейчас, многие задачи можно делегировать искусственному интеллекту, которые он эффективно решит. В некоторых случаях, интеграция ChatGPT непосредственно в клиентский сервис, позволит существенно повысить качество предоставляемых услуг. Комбинируя работу AI и человека, компании могут увеличить доходы от продаж, оптимизировать затраты на рекламу и создание нового контента.

В рамках данной статьи был произведен обзор некоторых функциональных возможностей ChatGPT, а также разработано web-приложение на языке Kotlin для получения программного доступа к API платформы ChatGPT. В будущих работах планируется продолжить исследования возможностей интеграции ChatGPT в различные бизнес-процессы.

Список использованной литературы

1. Hapke H. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python / H. Hapke, C. Howard, H. Lane. — Shelter Island : Manning Publications, 2019. — 544 p.
2. Attention Is All You Need / A. Vaswani, N. Shazeer, N. Parmar [at al.] // Advances in Neural Information Processing Systems. — 2017. — Vol. 30. — P. 5998–6008.
3. Natheem A. ChatGPT book for beginners / A .Natheem. — North Charleston : Independently published, 2023. — 165 p.
4. Бейненсон В.А. Применение генеративных нейросетей в журналистике: проблемы и перспективы / В.А. Бейненсон. — EDN THTQPB // Динамика медиасистем. — 2023. — Т. 3, № 1. — С. 352–359.
5. Testa R. Digital Marketing Mastery with ChatGPT: The Ultimate AI-Driven Guide / R. Testa. — North Charleston : Independently published, 2023. — 135 p.
6. Гаркуша Н.С. Педагогические возможности ChatGPT для развития когнитивной активности студентов / Н.С. Гаркуша, Ю.С. Городова. — DOI 10.52944/PORT.2023.52.1.001. — EDN NBBIRG // Профессиональное образование и рынок труда. — 2023. — Т. 11, № 1. — С. 6–23.

References

1. Hapke H., Howard C., Lane H. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python*. Shelter Island, Manning Publications, 2019, 544 p.
2. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomes A.N., Kaiser L., Polosukhin I. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 2017, vol. 30, pp. 5998–6008.

3. Natheem A. *ChatGPT Book for Beginners*. North Charleston, Independently published, 2023. 165 p.

4. Beynenson V.A. The Use of Generative Neural Networks in Journalism: Problems and Prospects. *Dinamika mediasistem = Dynamics of Media Systems*, 2023, vol. 3, no. 1, pp. 352–359. (In Russian). EDN: THTQPB.

5. Testa R. *Digital Marketing Mastery with ChatGPT: The Ultimate AI-Driven Guide*. North Charleston, Independently published, 2023. 135 p.

6. Garkusha N.S., Gorodova J.S. Pedagogical Opportunities of Chatgpt for Developing Cognitive Activity of Students. *Professional'noe obrazovanie i rynek truda = Vocational Education and Labour Market*, 2023, vol. 11, no. 1, pp. 6–23. (In Russian). EDN: NBBIRG. DOI: 10.52944/PORT.2023.52.1.001.

Информация об авторе

Кокотов Дмитрий Валерьевич — главный программист, ООО «Цифровые привычки», г. Томск, Российская Федерация, e-mail: kokotovdv@gmail.com, <https://orcid.org/0009-0004-1058-7594>, SPIN-код: 7899-1903, AuthorID РИНЦ: 1209755.

Information about the Author

Dmitrii V. Kokotov — Chief Programmer, Digital Habits LLC, Tomsk, Russian Federation, e-mail: kokotovdv@gmail.com, <https://orcid.org/0009-0004-1058-7594>, SPIN-Code: 7899-1903, AuthorID RSCI: 1209755.

Для цитирования

Кокотов Д.В. Использование платформы ChatGPT при разработке программных сервисов на языке Kotlin / Д.В. Кокотов. — DOI 10.17150/2713-1734.2023.5(4).409-416. — EDN FCAWTW // *System Analysis & Mathematical Modeling*. — 2023. — Т. 5, № 4. — С. 409–416.

For Citation

Kokotov D.V. Using the ChatGPT Platform in the Development of Software Services in the Kotlin Language. *System Analysis & Mathematical Modeling*, 2023, vol. 5, no. 4, pp. 409–416. (In Russian). EDN: FCAWTW. DOI: 10.17150/2713-1734.2023.5(4).409-416.