



Android + AI, решаем задачи с умом

Анна Жаркова
Lead Mobile developer

Обо мне



- В мобильной разработке с 2013
- Ведущий мобильный разработчик в Usetech
- Нативная разработка под iOS и Android (Swift/Objective-C, Kotlin/Java) кросс-платформа (Xamarin, Kotlin multiplatform)
- Ментор, управляю командой направления
- Спикер на конференциях AppsConf, Mobius, TechTrain, DroidCon (2022)
- Преподаватель в Otus (iOS Pro и базовый)
- Автор статей по мобильной разработке (SwiftUI, iOS, KMM)
- Эксперт Skillbox

Обсудим:

- Для чего нам это все?
- LLM, анатомия и назначения
- Открытые и закрытые LLM
- Готовые сервисы: ChatGPT, Gemini, GigaCode, Yandex GPT
- Настраиваем свою LLM
- Челлендж: генерируем приложение Android

Что было в прошлый раз

ДОКЛАД

Практика

Под капотом

31.05 / 14:30 – 15:15 (UTC+7)

Тулкиты кодогенерации Android vs iOS

RU



Презентация pdf

Спикеры



Анна Жаркова
Usetech

Ведущие



Дмитрий Гайдукевич
VK / Дзен

Инструменты кодогенерации. Назначение

- Преобразование существующего кода
- Генерация нового кода
- Решение однотипных задач
- Упрощение и ускорение разработки
- Сокращение написания кода разработчиком

Инструменты кодогенерации. Назначение

- Преобразование существующего кода
- Генерация нового кода
- Решение однотипных задач
- Упрощение и ускорение разработки
- Сокращение написания кода разработчиком



Зачем мне знать и уметь это

- AI же умный и умеет делать все сам



AI – это инструмент

- Писать собственные решения под свои конкретные задачи и проекты
- Ускорить и упростить рутинные задачи
- Повысить свой уровень разработчика



AI – это инструмент

Чтобы использовать ИИ, надо познать LLM



Что такое LLM, и с чем это едят



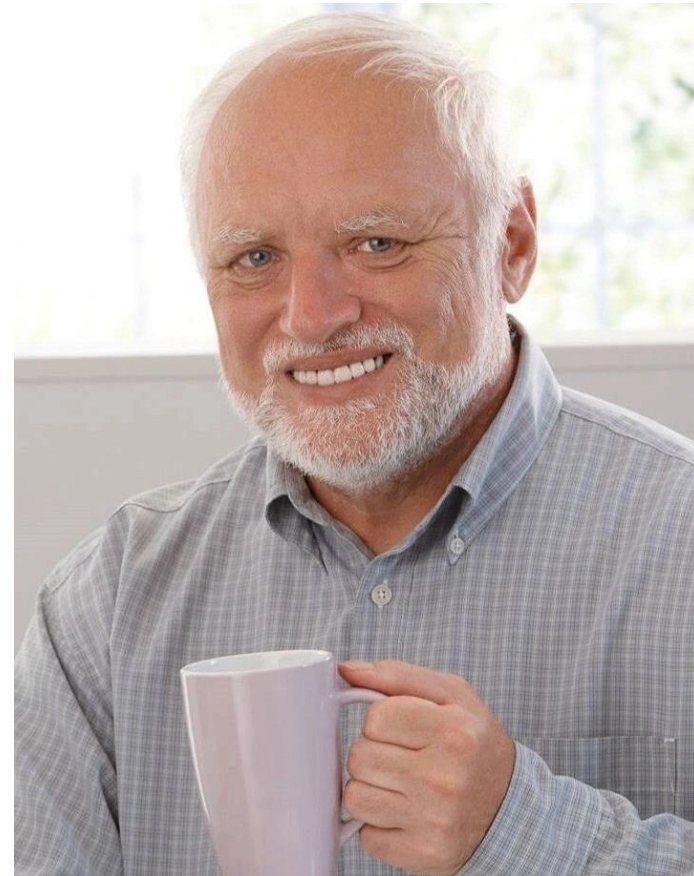
Что делает LLM

- Перевод на разные языки
- Генерация текста и контента для задачи
- Рекомендации и объяснения
- Анализ
- Ответы на вопросы....

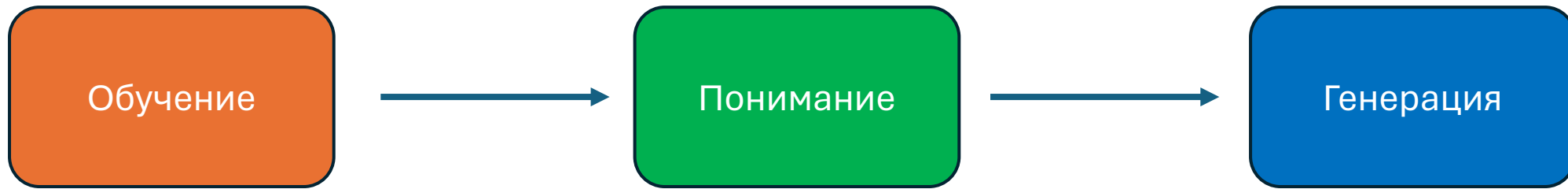
Что делает LLM

- Перевод на разные языки
- Генерация текста и контента для задачи
- Рекомендации и объяснения
- Анализ
- Ответы на вопросы...

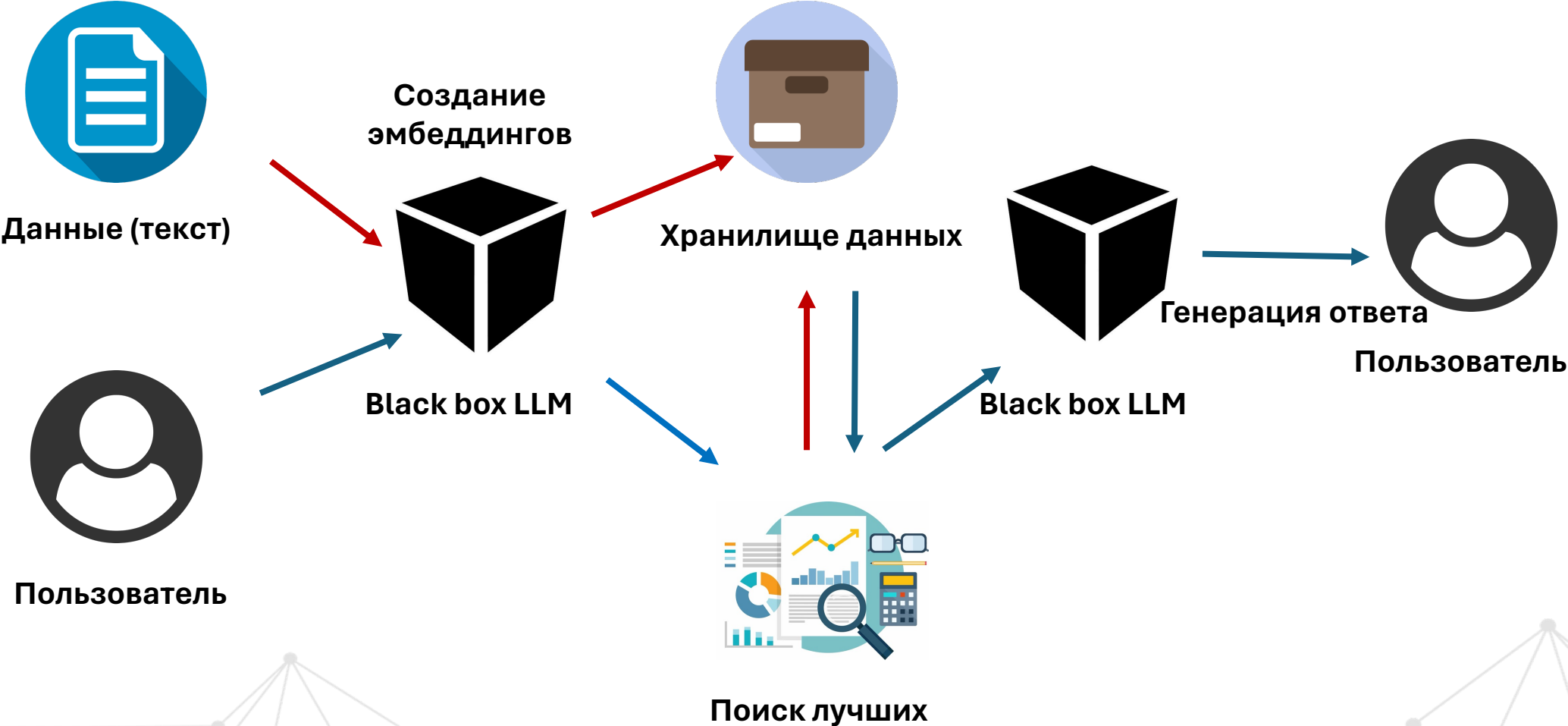
С переменным успехом



Как работает LLM



Добавление новых данных



Тренировка



Gigantic Training Set

pre-training



Base LLM

fine-tuning



Specific Knowledge Base



Fine-Tuned LLM



Модели открытые и закрытые. Open Source

- LLaMa (Meta)
- Falcon
- Mistral ...



Модели открытые и закрытые. Closed

- ChatGPT (Open AI)
- Gemini (Google)
- Claude (Anthropic)
- GigaChat (Сбер)
- Copilot (Microsoft)
- Yalm/YandexGPT (Yandex)

Модели открытые и закрытые. Closed

- [ChatGPT \(Open AI\)](#)
- [Gemini \(Google\)](#)
- Claude (Anthropic)
- [GigaChat \(Сбер\)](#)
- Copilot (Microsoft)
- [Yalm/YandexGPT \(Yandex\)](#)



Готовые сервисы

Готовые сервисы и плагины

Готовые сервисы на закрытых моделях

- ChatGPT (Open AI)
- Gemini (Google)
- GigaChat (Сбер)
- Yalm/YandexGPT (Yandex)

Почему не Copilot



GitHub Copilot Individual

The world's most widely adopted AI developer tool.



You can't proceed with your payment

We are sorry, but after reviewing we have determined that the billing information supplied is ineligible for transactions with GitHub due to government restrictions.

[Learn more](#) about GitHub and trade control regulations.

Почему не Copilot



GitHub Copilot Individual

The world's most widely adopted AI developer tool.



You can't proceed with your payment

We are sorry, but after reviewing we have determined that the billing information supplied is ineligible for transactions with GitHub due to government restrictions.

[Learn more](#) about GitHub and trade control regulations.



ЗЛЫЕ ВЫ...

Уйду я от вас...

demotivation.ru

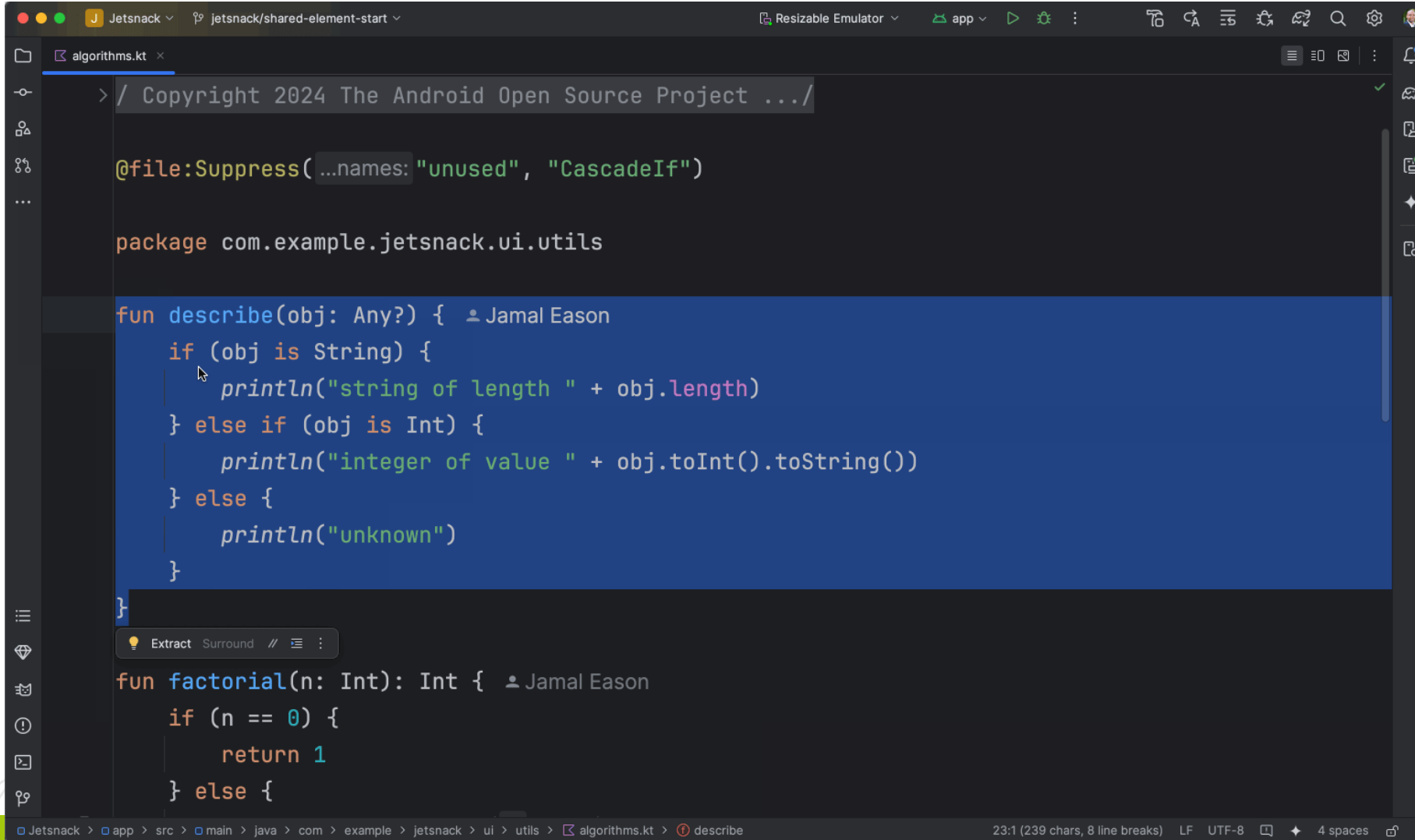
Google I/O 2024. Gemini

Встроен в Android Studio Koala

Есть версия для IOS (API)



Google I/O 2024. Gemini



The screenshot shows an IDE window with the following content:

```
> / Copyright 2024 The Android Open Source Project .../

@file:Suppress(...names: "unused", "CascadeIf")

package com.example.jetsnack.ui.utils

fun describe(obj: Any?) {
    // Jamal Eason
    if (obj is String) {
        println("string of length " + obj.length)
    } else if (obj is Int) {
        println("integer of value " + obj.toInt().toString())
    } else {
        println("unknown")
    }
}

fun factorial(n: Int): Int {
    // Jamal Eason
    if (n == 0) {
        return 1
    } else {
```

The IDE interface includes a top toolbar with icons for file operations, a left sidebar with navigation icons, and a bottom status bar showing the file path: `com.example.jetsnack.ui.utils> algorithms.kt> describe`. The status bar also displays `23:1 (239 chars, 8 line breaks) LF UTF-8 4 spaces`.

Google I/O 2024. Gemini

- Автодополнение
- Анализ краш-репортов
- Помощь в дебаге
- Оптимизация кода
- Генерация кода...

Gemini. Обучающий контекст

- Использовать контекст из проектов для улучшения ответов
- Для каждого проекта отдельно
- Не использовать контекст проектов

Thanks for trying Gemini!

To finish, please confirm what privacy setting you'd like to start with. These can be changed at any time in **Settings > Tools > Gemini**.

Use context from your projects to improve responses

Gemini will use context from your project along with your query, to offer better suggestions and responses, subject to our [privacy policy](#)

Note: You can use an `.aiexclude` file in your project to exclude specific files and directories. [Learn more](#)

Ask to decide per project

Use context from this project to improve responses

Do not use context from any project

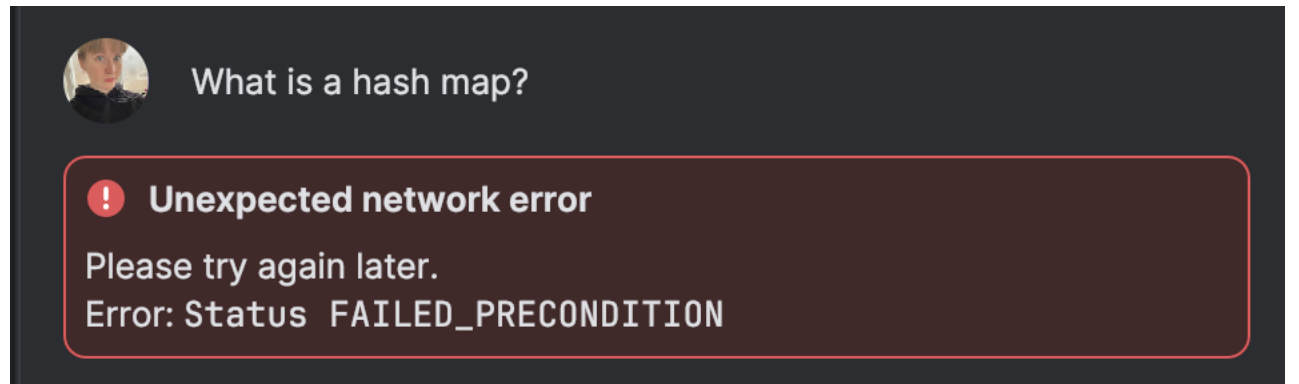
Gemini will only use your prompts and past conversation history to respond. Some features such as AI code completion will not be available in this mode.

Back

Finish

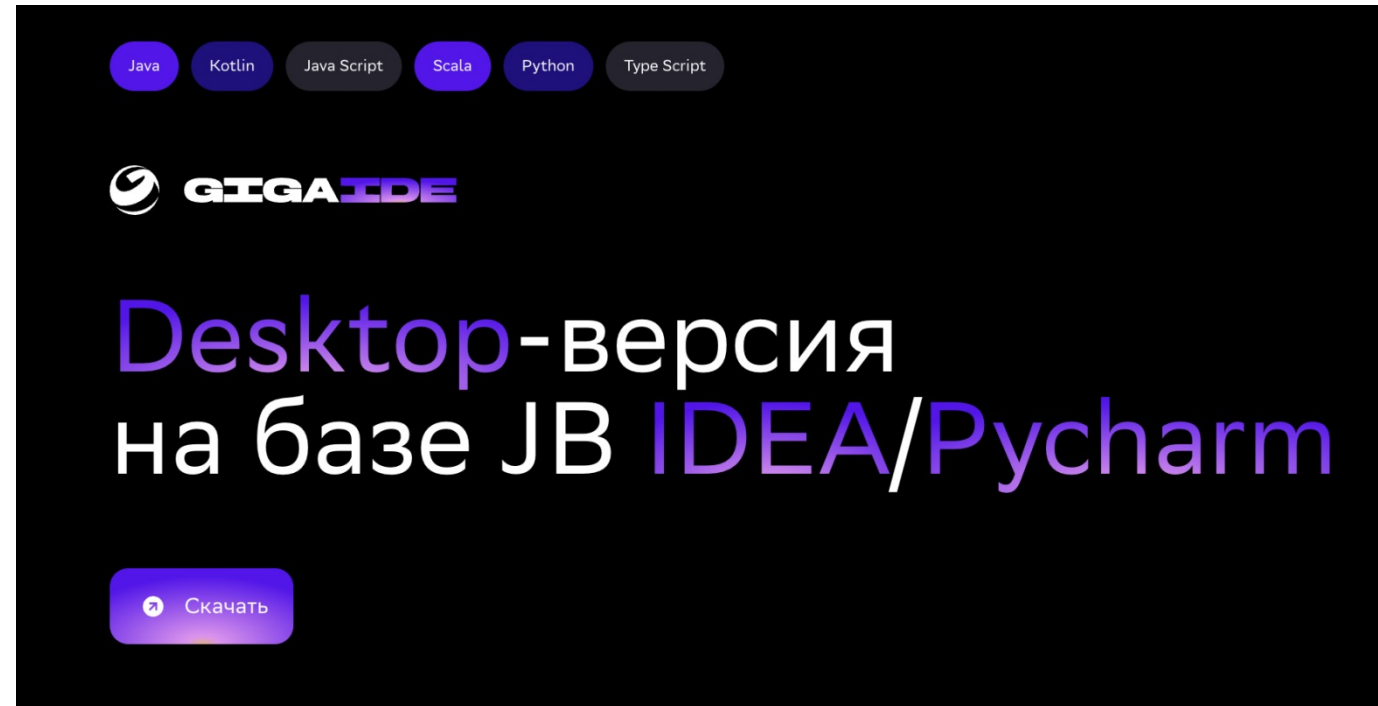
Gemini. Ограничения

- Нужен ключ для API для встраивания
- Доступен не во всех регионах



GigaCode

- AI-помощник от Сбер
- 1.3.359-ext



Особенности GigaCode AI

- Поддерживает разные языки (включая Kotlin)
- Плагин для разных IDE

Supported IDEs



VS Code



JetBrains IDEs



Android Studio



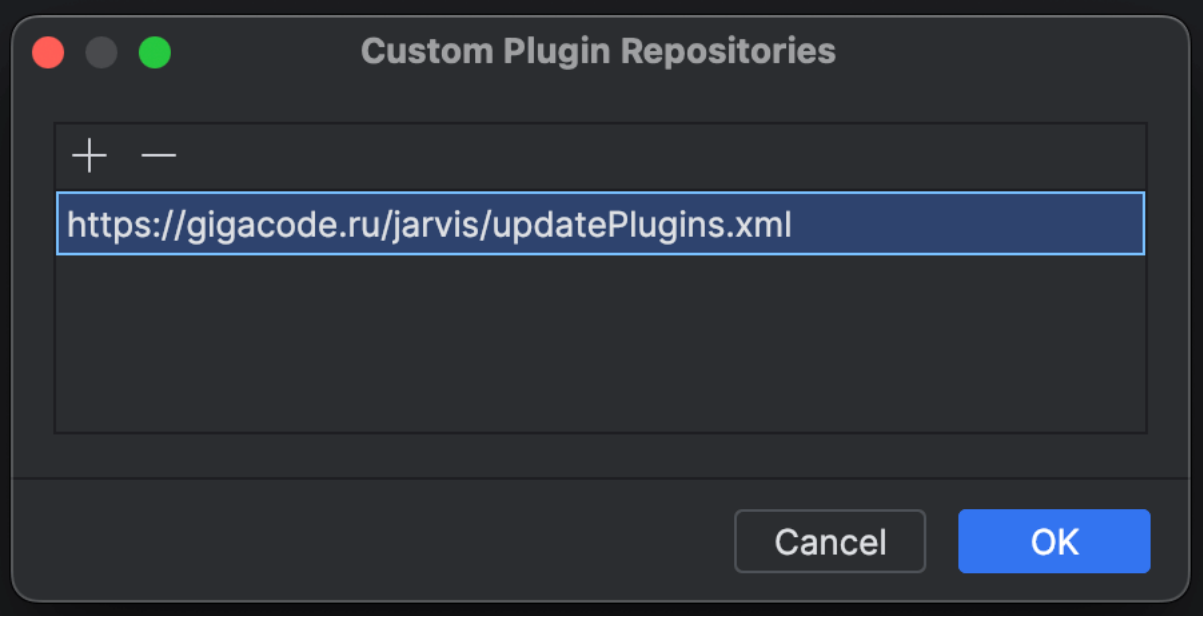
Jupyter Notebook

GigaCode AI

```
TS inline-completion.ts •
assistant > TS inline-completion.ts > UserManager > updateUser
11 class UserManager {
12     /**
13      * Updates user information
14      * @param id User ID
15      * @param updatedUser Object with updated user data
16      */
17     updateUser(id: number, updatedUser: Partial<User>): void {
18
19     }
20
21     /**
22      * Removes a user by ID
23      * @param id User ID
24      */
25     removeUser(id: number): void {
26         this.users = this.users.filter(user => user.id !== id)
27     }
28
29     /**
30      * Gets a user by ID
31      * @param id User ID
32      * @returns User or undefined if not found
33      */
34     getUserById(id: number): User | undefined {
35         return this.users.find(user => user.id === id)
36     }
37 }
```

GigaCode. Настройка и подключение

<https://gigacode.ru/jarvis/updatePlugins.xml>



GigaCode. Настройка и подключение

The screenshot shows the WordPress Plugins Marketplace interface. At the top, there are tabs for 'Plugins', 'Marketplace', 'Installed 1', and a settings gear icon. The 'Marketplace' tab is active. A search bar contains the text 'gigacode'. Below the search bar, it says 'Search Results (1)' and 'Sort By: Relevance'. A search result for 'GigaCode' is highlighted, showing a blue plug icon and an 'Install' button. To the right, the 'GigaCode' plugin details are shown, including a green 'Install' button and the number '1.3.186'. Below the details are tabs for 'Overview', 'What's New', and 'Reviews'.

GigaCode. Настройка и подключение



AI-ассистент разработчика, который ускоряет создание ПО

Для успешной установки плагина GigaCode требуется зарегистрировать ваше устройство. Для этого мы разместили подробную инструкцию по установке и активации плагина на вашем компьютере.

✔ GigaCode активирован

Устройство	IDE	Добавлено
✘ e09adee1cc04f41669c07eb5e1437cded...	Android Studio	28.06.2024

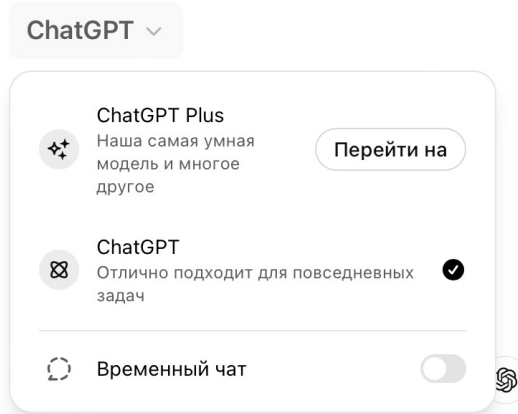
[Перейти к инструкции](#)

GigaCode. Базовые возможности

- Сгенерировать тесты
- Сгенерировать документацию
- Объяснение
- Что-то по запросу (код сгенерировать)

Chat GPT

<https://chatgpt.com/>



Напиши приложение Android для отображения списка новостей с newsapi.org по запросу everything. Для сетевого клиента используй Retrofit, для многопоточности coroutines. UI напиши на Jetpack Compose. Используй DI на Koin. Используй Material3 и Kotlin code-style

Чтобы создать Android-приложение для отображения списка новостей с использованием NewsAPI (<https://newsapi.org/>) по запросу "everything", используя Retrofit, Coroutines, Koin для DI, Jetpack Compose для UI и Material3, выполним следующие шаги:

1. Подключаем зависимости

В `build.gradle` добавляем необходимые зависимости:

```
gradle // Копировать код

// build.gradle (app)
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-kapt'
}

android {
    compileSdk 34

    defaultConfig {
        applicationId "com.example.newsapp"
        minSdk 24
        targetSdk 34
        versionCode 1
    }
}
```

Сообщить ChatGPT

ChatGPT может допускать ошибки. Рекомендуем проверять важную информацию.

Chat GPT. Плагин IDE

The screenshot shows an IDE interface with a code editor and a dropdown menu. The code editor contains Kotlin test code for a `NewsResponse` class. The dropdown menu lists various AI models and cloud providers.

```
// Arrange
val expectedResponse = Result.success(NewsResponse(/* initialize with test data */)
every { networkClient.getNews() } returns expectedResponse

// Act
val result = runBlocking { useCase() }

// Assert
result shouldBe expectedResponse
verify { networkClient.getNews() }
}

"should return error when network call fails" {
// Arrange
val expectedError = Result.failure<NewsResponse>(Exception("Network Error"))
every { networkClient.getNews() } returns expectedError

// Act
val result = runBlocking { useCase() }
```

CodeGPT

- o1-mini
- GPT-4o
- Claude 3.5 Sonnet
- Llama 3.1 (405B)
- DeepSeek Coder V2 - FREE
- GPT-4o mini - FREE

Cloud Providers

- OpenAI
- Custom: OpenAI
- Anthropic (Claude)
- Azure OpenAI
- Google (Gemini)

Local Providers

- CodeQwen2.5 Coder 1B (Q8)
- Ollama

Dependencies for Kotest and MockK in your `build.gradle` file.

- Gemini 1.0 Pro (32k)
- Gemini 1.0 Pro 001 (32k)
- Gemini 1.0 Pro Latest (32k)
- Gemini 1.0 Pro Vision Latest (16k)
- Gemini Pro (32k)
- Gemini Pro Vision (16k)
- Embedding 001 (2k)
- Text Embedding (2k)

9:4 LF UTF-8 GIGACODE 4 spaces

Chat GPT. Ограничения

- Платный, бесплатно только Chat GPT-4o mini
- Ограничение по локации

Портированные сервисы. Chad AI



Chad AI

Что может

Как работает

Отзывы

Тарифы

Для бизнеса

Вопросы

Блог

Вход

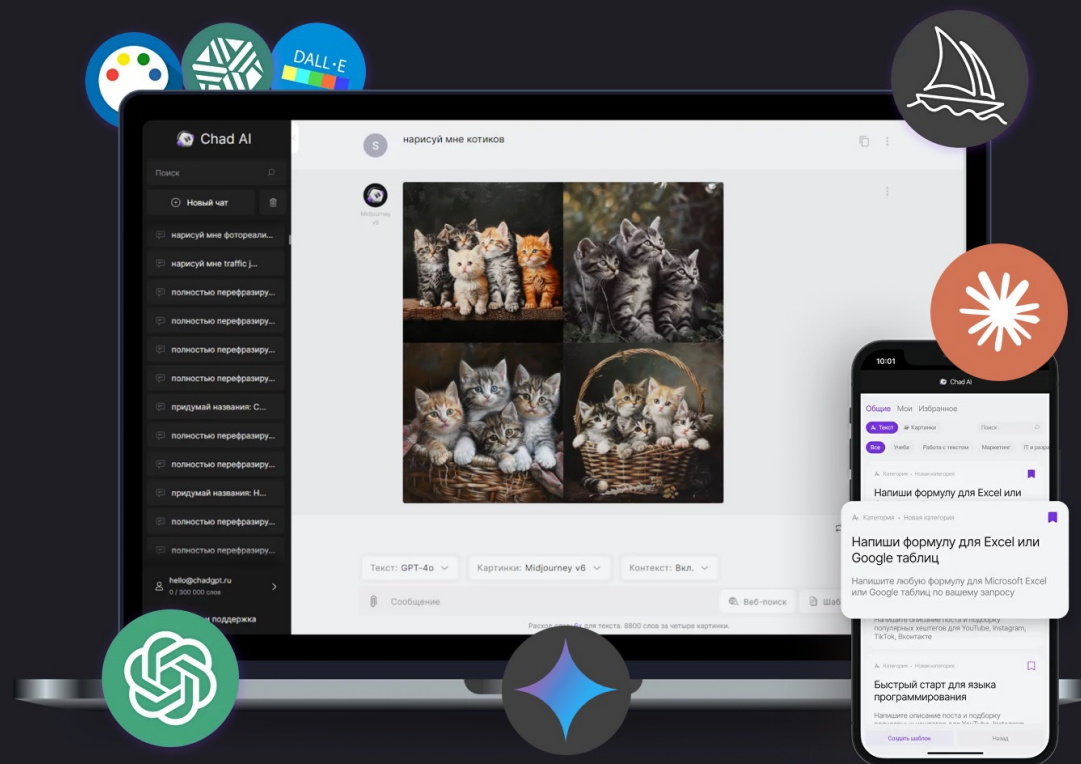
ChatGPT на русском и другие нейросети в **одном сервисе**

Chad работает без VPN и номера. Midjourney, Stable Diffusion, Dall-E и Chat GPT-4o.

Использовать бесплатно

★★★★★ 4,6 на основе 2000+ оценок в

T-Банк IOS & Android Telegram



Yandex GPT

Яндекс

- ^ Нейросети Яндекса
- Нейростат
- YandexGPT 3
- YandexGPT 2
- YandexART

A dark-themed banner with a starry background. It features the Yandex logo (a white circle with a dot) followed by the text 'YandexGPT 3' in large white font. Below this, the text 'Новое поколение генеративных текстовых нейросетей' is written in a smaller white font.

YandexGPT 3

Новое поколение генеративных текстовых нейросетей

Yandex GPT

Алиса Про

06 октября

Напиши Android приложение для отображения данных newsapi.org в списке. Раздели код на слои, выдели сетевой клиент на Retrofit. Экран с помощью Jetpack Compose

Я пока не умею писать код, но вот что у меня получилось.

Чтобы создать Android-приложение для отображения данных с сайта newsapi.org в списке, можно использовать следующие шаги:

- 1. Создание проекта.** Откройте Android Studio и создайте новый проект. Выберите тип проекта «Empty Compose Activity».
- 2. Подключение зависимостей.** В файле build.gradle добавьте зависимости для работы с Retrofit и Jetpack Compose. Пример:

```
dependencies {  
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
    implementation 'androidx.compose.ui:ui:1.3.0-beta02'  
    implementation 'androidx.compose.material:material:1.3.0-beta02'  
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.5.1'  
}
```

- 3. Создание интерфейса API.** Создайте интерфейс API, который будет использоваться для взаимодействия с newsapi.org. Пример:

Как вам помочь?



YALM

Yandex GPT. Ограничения

- Есть расширенный платный API
- Ограничения по локации (Яндекс сервисы)

Готовые сервисы хороши, но есть подвох

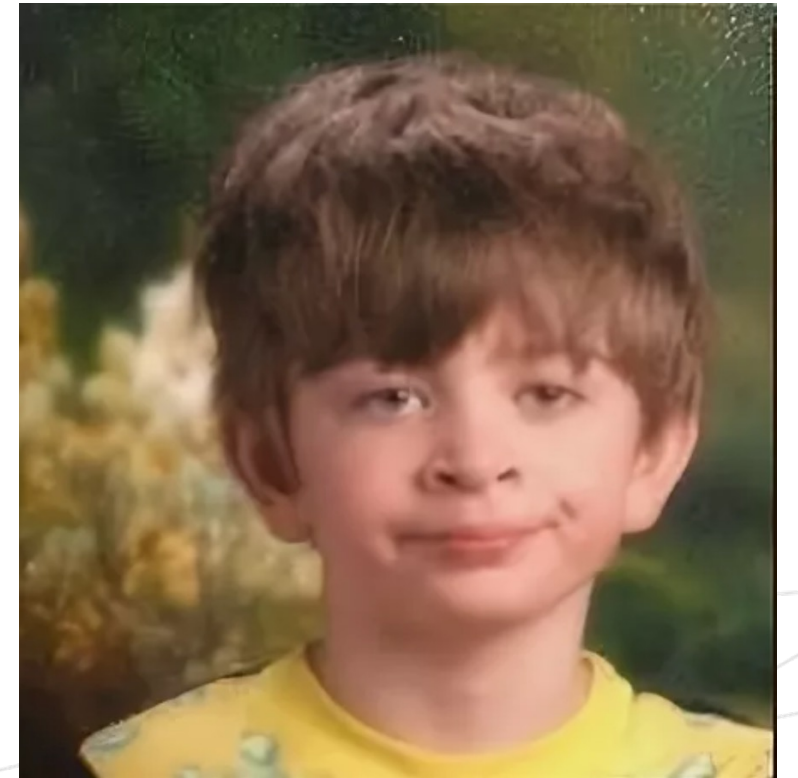


Внешний готовый сервис.Cons

- Ограниченные возможности кастомизации
- Ограниченные возможности обучения на собственных данных
- Нет доступа к специфичному контексту
- Оплата за использование

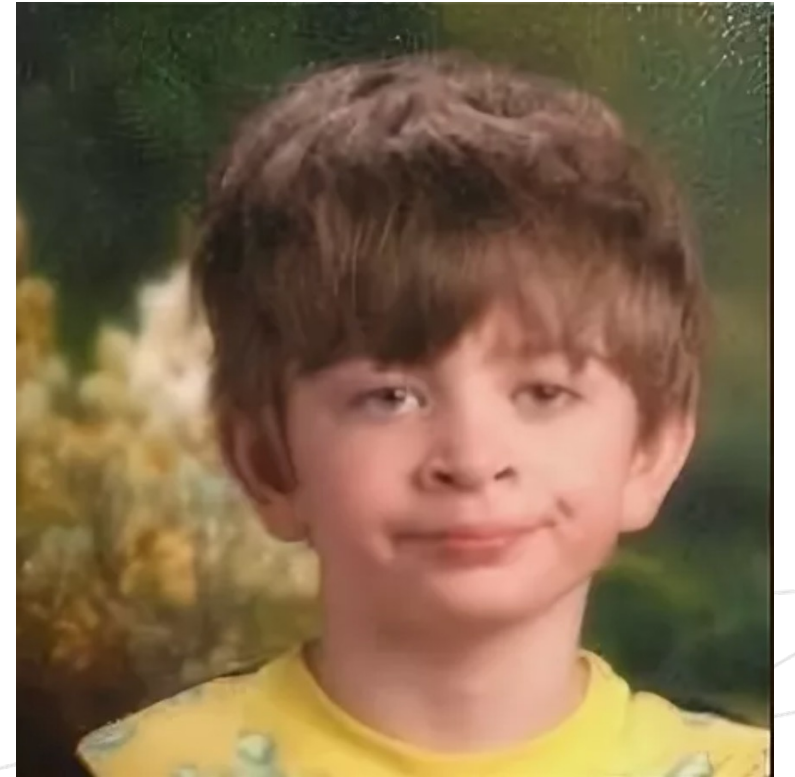
Внешний готовый сервис.Cons

- Запросы уходят в сеть и хранятся неизвестно где
- Ограничения по IP и локации



Внешний готовый сервис.Cons

- Утечка данных (каждый запрос идет в сеть)
- Нарушение лицензионных соглашений
- Нарушение NDA



Выход: Настраиваем свою LLM



Свой локальный LLM. Pro

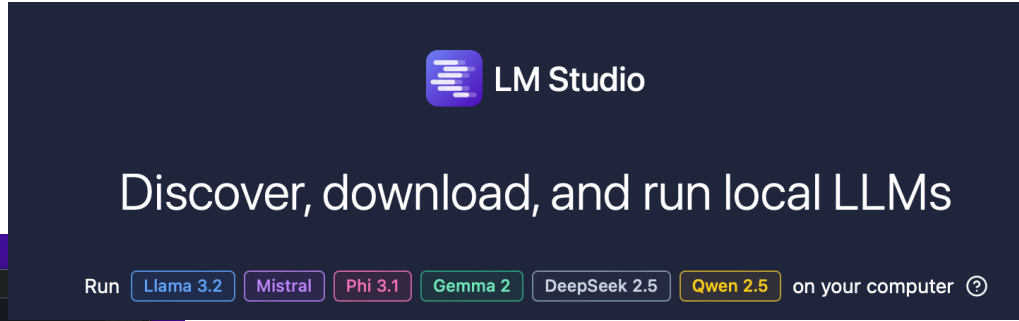
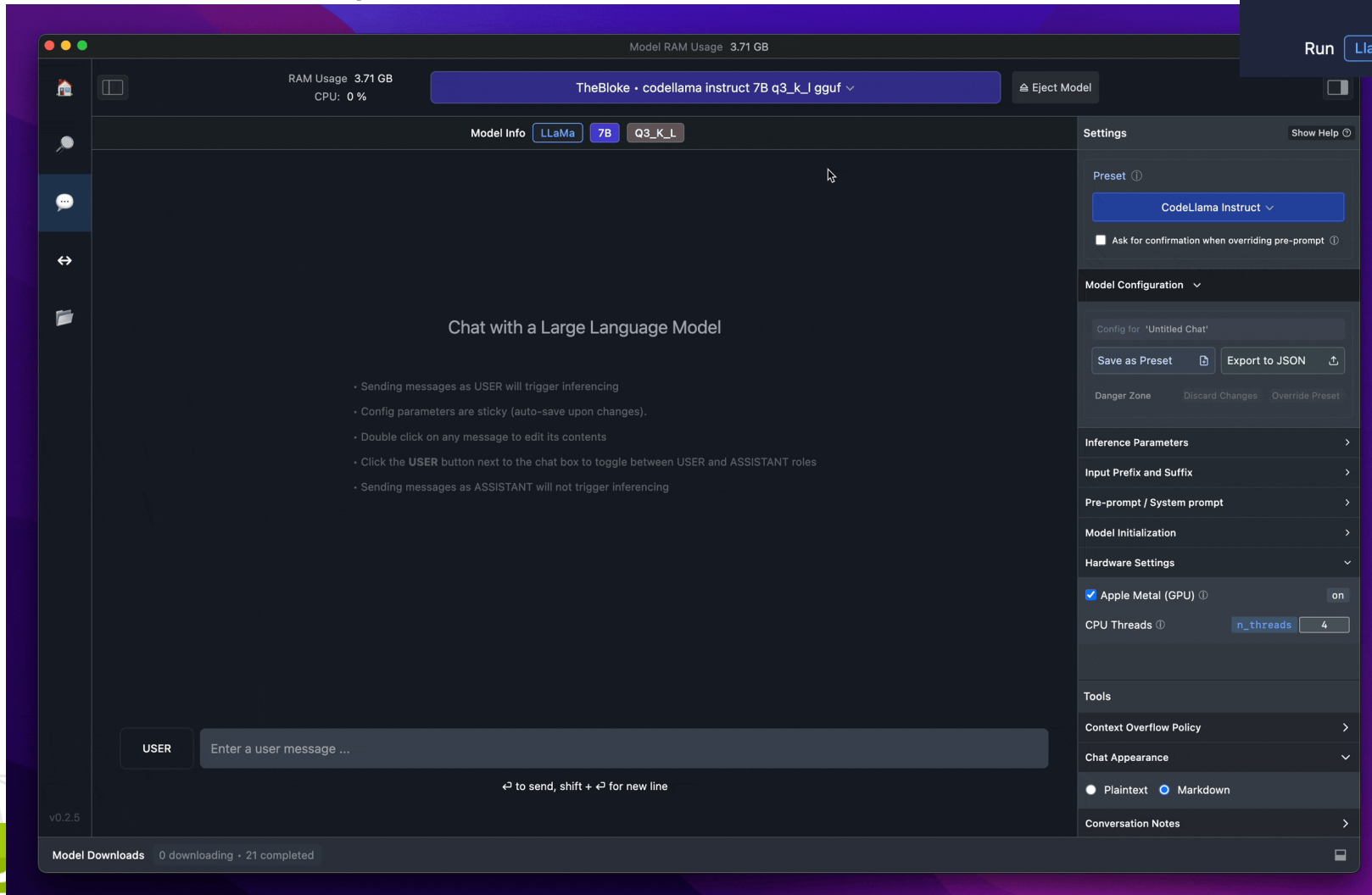
- Автономная работа без сети
- Независимость от внешних сервисов
- Гарантированная доступность и время отклика
- Возможность обучения специфическим задачам
- Даже бесплатно

Свой локальный LLM. Cons


- Сложность поддержки
- Надо учить
- Производительность

Свой локальный LLM. Инструменты

- LM Studio - среда



Свой локальный LLM. Инструменты

- Ollama 

Download Ollama



Linux

Windows

Run your first model

```
ollama run llama3.1
```

Run this command in your favorite terminal.

Finish

Download for macOS

Requires macOS 11 Big Sur or later

Свой локальный LLM. Модели и ресурсы

- Hugging Face
- [Promptingguide.ai](https://promptingguide.ai)
- LocalLLaMa

Open Source LLM

- Mistral-7B
- Llama-3
- Meta-LLama
- CogVLM2
- Phi-3
- Falcon...



Свое решение

- Meta-Llama
- LM Studio

The screenshot shows the LM Studio application window. The title bar indicates the model being used is 'lmstudio-community/Meta-Llama-3.1-8B-Instruct-GGUF'. The interface is divided into a chat area on the left and a code/output area on the right.

Chat History:

- 1. Jetpack Compose UI Design (1657 tokens)
- 2. **Android News API App (40680 tokens)**
- 3. Testing GetNewsUseCase with ... (2331 tokens)

User:

Напиши Android приложение для отображения данных newsapi.org в списке. Раздели код на слои, выдели сетевой клиент на Retrofit. Экран с помощью Jetpack Compose

Assistant (Meta-Llama-3.1-8B-Instruct-GGUF):

Чтобы реализовать приложение для отображения данных newsapi.org в списке, мы создадим следующие компоненты:

1. Сетевой клиент с помощью Retrofit.
2. Бизнес-логика, которая будет отвечать за получение и обработку данных.
3. Представление (экран) с помощью Jetpack Compose.

Сетевой клиент (Retrofit)

Сначала нам нужно создать сетевой клиент с помощью Retrofit.

Навигатор проекта:

```
src/main/java/com/example/newsapi/client/NetworkClient.java
```

Код:

```
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
```

Type a message and press Enter to send ...

User (%U) [attach icon] Insert (%I)

Context is 993.2% full ?

LM Studio 0.3.3 (Build 1) User Power User Developer SYSTEM RESOURCES USAGE: RAM: 4.50 GB | CPU: 0.00 %

Что использовать в Android

- GigaCode plugin
- Gemini plugin
- GPT plugin
- Собственный plugin + готовый сервис или своя LLM

Что использовать в Android




- GigaCode plugin
- Gemini plugin
- Собственный plugin + готовый сервис или своя LLM

Сравним решения






Да начнется челлендж!



Наши игроки

- Gemini 
- GigaCode 
- Chat GPT 

Наши игроки

- Gemini 
- GigaCode 
- Chat GPT 
- Yandex GPT 
- Meta-LLaMa 

Что не будем делать

- SDUI и JSON парсить

```
{
  "data": [
    {
      "type": "SCAFFOLD",
      "top_bar": [
        {
          "type": "APP_BAR",
          "children": [
            { /*...*/ }
          ]
        }
      ],
      "children": [
        {
          "type": "VERTICAL_LIST",
          "children": [
            {
              "type": "ROW",
              "children": [
                {
                  "type": "TEXT",
                  "value": "Hi 1"
                },
                {
                  "type": "IMAGE",
                  "value": "https://8pic.ir/uploads/flowers.png"
                }
              ]
            },
            //.....
          ]
        }
      ]
    }
  ]
}
```



Это было в прошлый раз

ДОКЛАД

Практика

Под капотом

31.05 / 14:30 – 15:15 (UTC+7)

Тулкиты кодогенерации Android vs iOS

RU



Презентация pdf

Спикеры



Анна Жаркова
Usetech

Ведущие



Дмитрий Гайдукевич
VK / Дзен

Условия

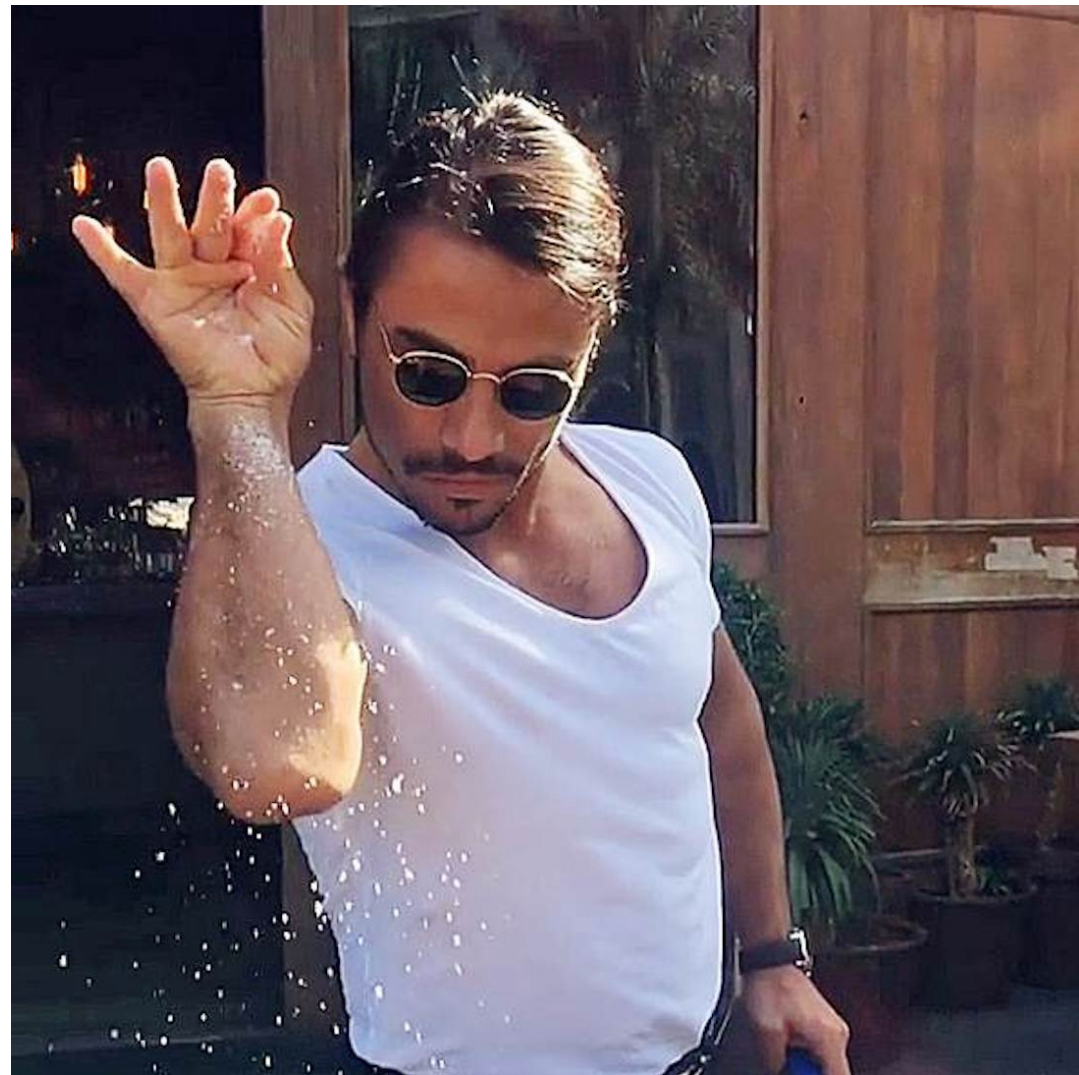
- Базовая комплектация сервиса
- Бесплатные возможности
- Обучение только в процессе
- **Дополнительные промпты только в особых случаях**

Что не будем сравнивать

- Мощность LLM
- Время выполнения
- Количество ложно-положительных результатов...
- Математику всякую...

Что будем сравнивать

- Качество выполнения конечной задачи

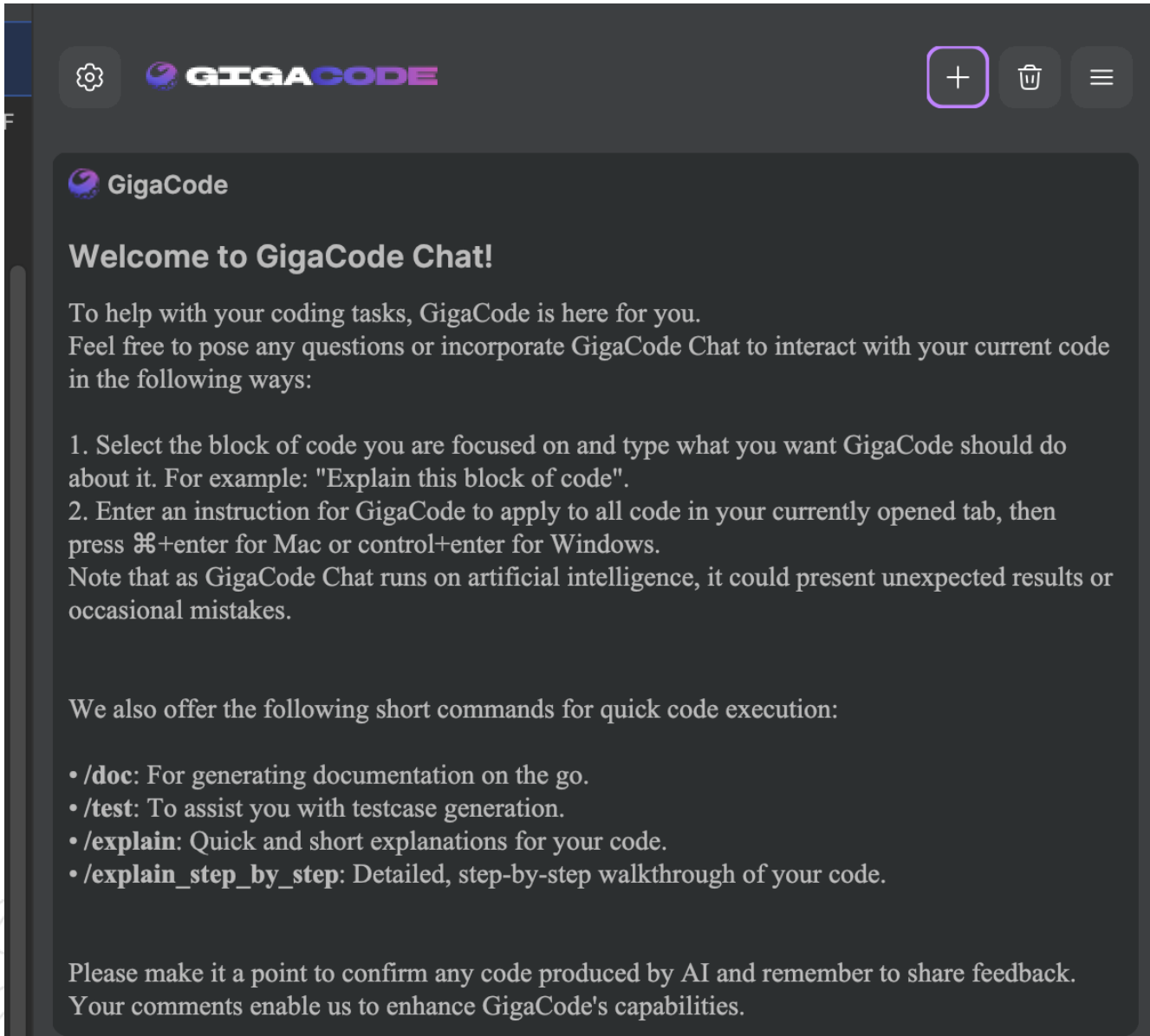


Простые задачи

Базовые кейсы:

- Тесты
- Документация

GigaCode. Базовые возможности



The screenshot shows the GigaCode Chat interface. At the top, there is a header with a settings gear icon, the GigaCode logo, and three utility icons: a plus sign (highlighted with a purple border), a trash can, and a hamburger menu. Below the header, the chat area displays a welcome message and instructions for using the chat.

GigaCode

Welcome to GigaCode Chat!

To help with your coding tasks, GigaCode is here for you. Feel free to pose any questions or incorporate GigaCode Chat to interact with your current code in the following ways:

1. Select the block of code you are focused on and type what you want GigaCode should do about it. For example: "Explain this block of code".
2. Enter an instruction for GigaCode to apply to all code in your currently opened tab, then press ⌘+enter for Mac or control+enter for Windows.

Note that as GigaCode Chat runs on artificial intelligence, it could present unexpected results or occasional mistakes.

We also offer the following short commands for quick code execution:

- **/doc**: For generating documentation on the go.
- **/test**: To assist you with testcase generation.
- **/explain**: Quick and short explanations for your code.
- **/explain_step_by_step**: Detailed, step-by-step walkthrough of your code.

Please make it a point to confirm any code produced by AI and remember to share feedback. Your comments enable us to enhance GigaCode's capabilities.

GigaCode. Базовые возможности

- Сгенерировать тесты
- Сгенерировать документацию
- Объяснение
- Что-то по запросу (код сгенерировать)

GigaCode. Документация для кода

You



/doc

 GigaCode

```
/**
 * This class represents a repository for news data.
 * It uses the `NewsApiService` to fetch news data.
 *
 * @property newsApiService The `NewsApiService` instance used to fetch news data.
 */
class NewsRepository(private val newsApiService: NewsApiService) {

    /**
     * This function fetches news data based on the given query.
     *
     * @param query The query string used to fetch news data.
     * @return The `NewsResponse` object containing the fetched news data.
     */
    suspend fun getNews(query: String): NewsResponse {
        return newsApiService.getEverything(query)
    }
}
```

GigaCode. Тесты

Unit-тесты
/test

```
@ExtendWith(MockitoExtension::class)
class NewsRepositoryTest {

    @Test
    fun `getNews returns correct response`() = runBlocking {
        // Arrange
        val mockNewsApiService = mockk<NewsApiService>()
        val newsRepository = NewsRepository(mockNewsApiService)
        val query = "testQuery"
        val expectedResponse = NewsResponse()

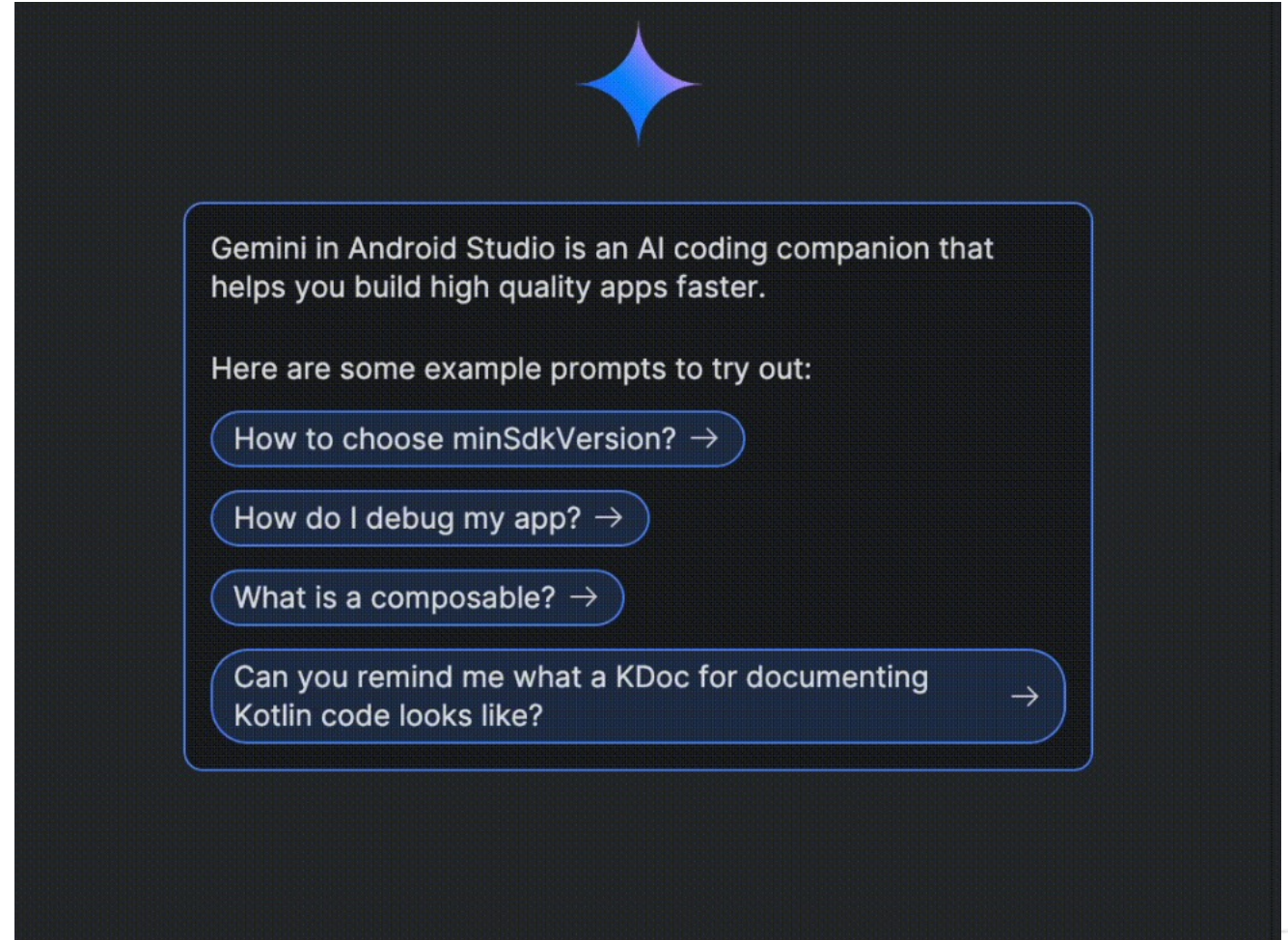
        every { mockNewsApiService.getEverything(query) } returns expectedResponse

        // Act
        val actualResponse = newsRepository.getNews(query)

        // Assert
        assertEquals(expectedResponse, actualResponse)
    }
}
```

Gemini. Базовые возможности

Есть примерные промпты-
рекомендации

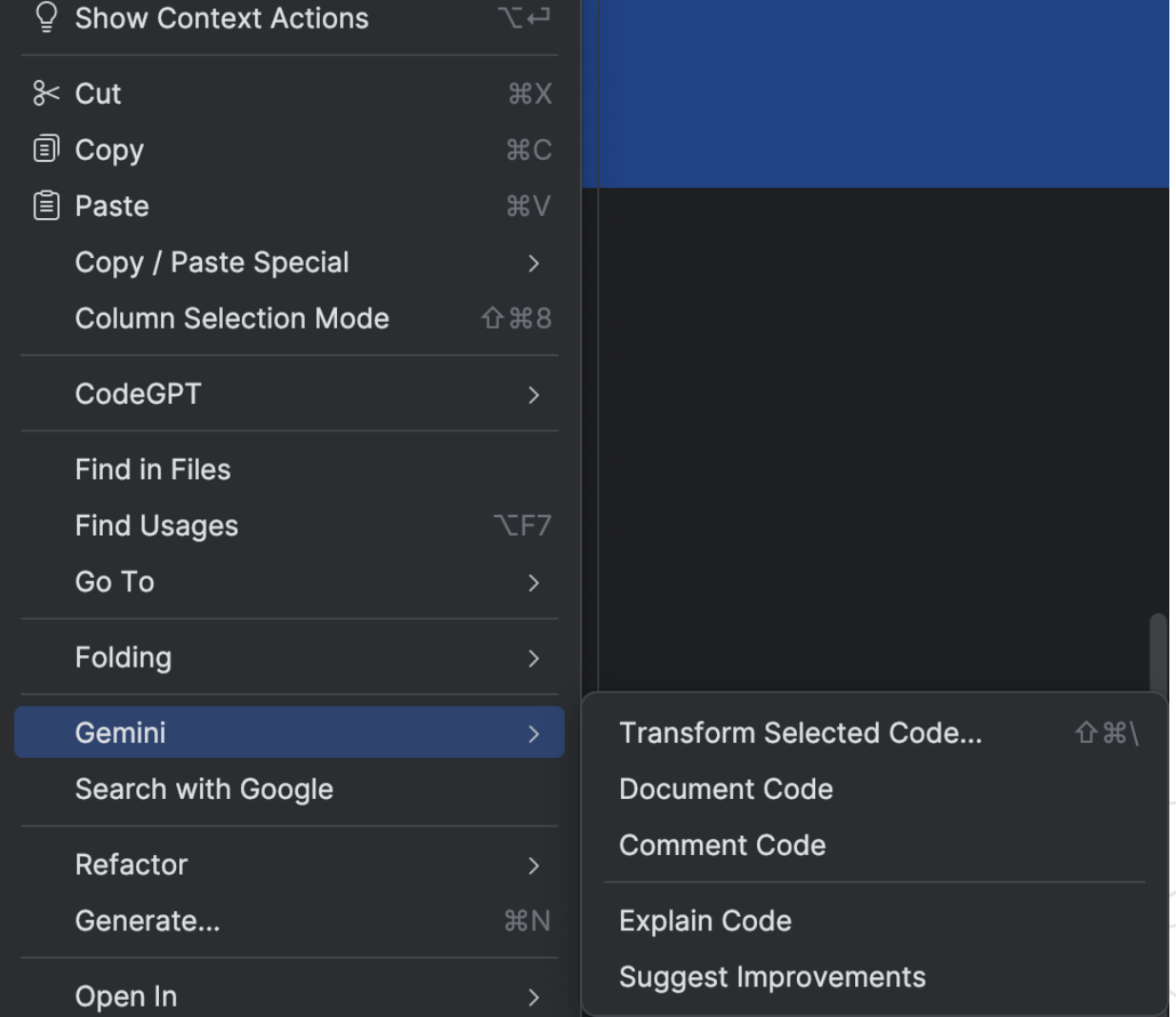


Gemini. Базовые возможности

Плагин работает с контекстным меню

```
class NewsRepository(private val newsApiService: NewsApiService) {
    suspend fun getNews(query: String): NewsResponse {
        return newsApiService.getEverything(query)
    }
}

interface NewsService {
    @GET("everything")
    suspend fun getNews(): NewsResponse
}
```



The screenshot shows an IDE context menu with the following items:

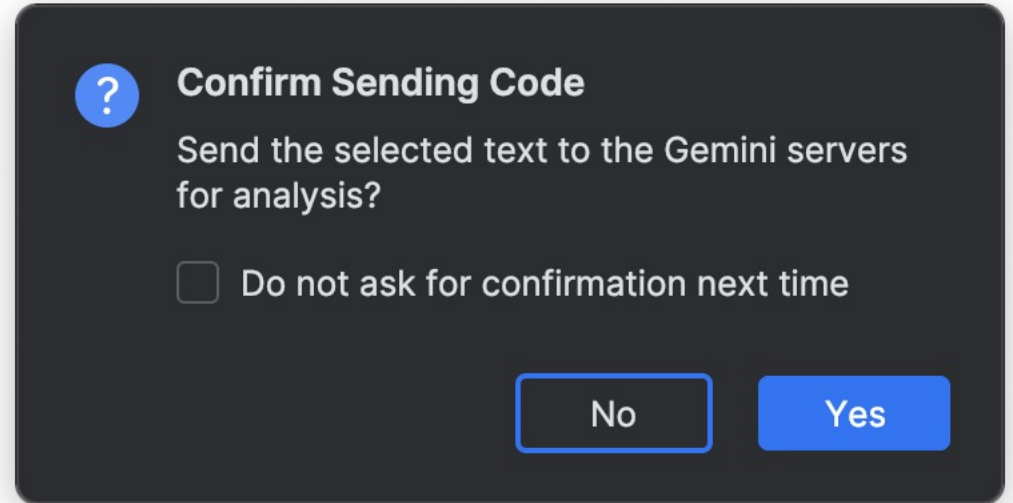
- Show Context Actions (⌘⇧↵)
- Cut (⌘X)
- Copy (⌘C)
- Paste (⌘V)
- Copy / Paste Special (>)
- Column Selection Mode (⇧⌘8)
- CodeGPT (>)
- Find in Files
- Find Usages (⇧⌘F7)
- Go To (>)
- Folding (>)
- Gemini (>)** (highlighted)
- Search with Google
- Refactor (>)
- Generate... (⌘N)
- Open In (>)

The Gemini submenu is open, showing the following options:

- Transform Selected Code... (⇧⌘\)
- Document Code
- Comment Code
- Explain Code
- Suggest Improvements

Gemini. Базовые возможности

При отправке через контекстное меню спрашивает подтверждение



Gemini. Базовые возможности

KDoc документация



Okay, let's generate documentation for the provided code:

```
/**
 * A repository responsible for fetching news articles from a remote data
 * source.
 *
 * This class uses a [NewsApiService] to make network requests and retrieve
 * news data.
 * It provides a single function, [getNews], to fetch news articles based on
 * a query.
 *
 * @property newsApiService The API service used to make network requests.
 */
class NewsRepository(private val newsApiService: NewsApiService) {

    /**
     * Fetches news articles based on the provided query.
     *
     * This function is a suspending function, meaning it can be paused and
     * resumed,
     * making it suitable for network operations. It calls the
     * [getEverything] function
     * on the [newsApiService] to retrieve the news data.
     *
     * @param query The search query for news articles.
     * @return A [NewsResponse] object containing the fetched news articles.
     */
    suspend fun getNews(query: String): NewsResponse {
        return newsApiService.getEverything(query)
    }
}
```

Gemini. Тесты



```
Generate test for code class NewsRepository(private val newsApiService: NewsApiService) {  
    suspend fun getNews(query: String): NewsResponse {  
        return newsApiService.getEverything(query)  
    }  
}
```



Okay, let's generate a Kotest test for the NewsRepository class you provided.
First, make sure you have the Kotest dependency added to your project:

```
dependencies {  
    testImplementation("io.kotest:kotest-runner-junit5:5.5.5") // for kotest  
        framework  
    testImplementation("io.kotest:kotest-assertions-core:5.5.5") // for  
        kotest core assertions  
    testImplementation("io.mockk:mockk:1.13.3") // for mocking  
}
```

Gemini. Тесты

Спорное

Now, here's the Kotest test for NewsRepository:

```
import ...

class NewsRepositoryTest : FunSpec({
    val newsApiService = mockk<NewsApiService>()
    val newsRepository = NewsRepository(newsApiService)

    test("getNews should return news response from api service") {
        val query = "technology"
        val expectedResponse = NewsResponse(
            status = "ok",
            totalResults = 100,
            articles = emptyList() // You can add sample articles here
        )

        coEvery { newsApiService.getEverything(query) } returns
            expectedResponse

        val actualResponse = newsRepository.getNews(query)

        actualResponse shouldBe expectedResponse
    }
})
```

План челленджа

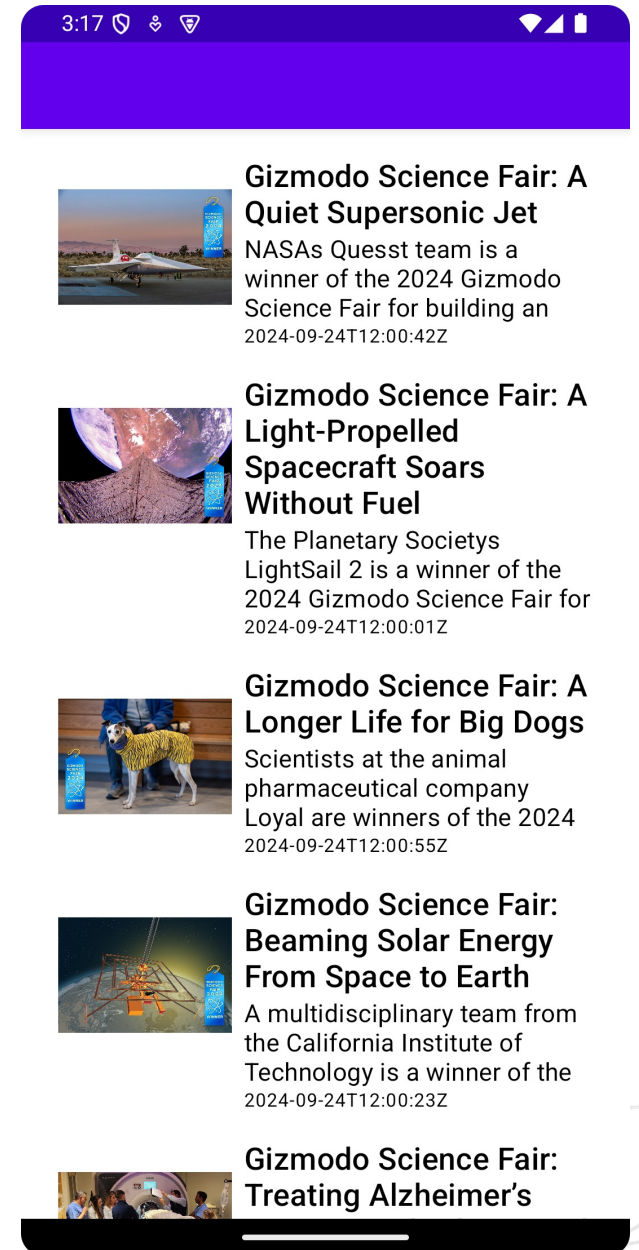
Приложение новостей на Android

1 Модели данных

2. UI (не SDUI)

3 Retrofit

4 ViewModel



Генерация моделей

Newsapi.org

```
← All articles about Tesla from the last month, sorted by recent first →  
  
GET https://newsapi.org/v2/everything?q=tesla&from=2024-09-10&sortBy=publishedAt&apiKey=API_KEY  
  
{  
  status: "ok",  
  totalResults: 4206,  
  - articles: [  
    - {  
      - source: {  
        id: null,  
        name: "ETF Daily News"  
      },  
      author: "MarketBeat News",  
      title: "Cantor Fitzgerald Reaffirms "Neutral" Rating for Tesla (NASDAQ:TSLA)",  
      description: "Cantor Fitzgerald reissued their neutral rating on shares of Tesla (NASDAQ:TSLA – Free Report) in a report published on Monday, Benzinga reports. The firm currently has a $245.00 price target on the electric vehicle producer's stock. A number of other equitie...",  
      url: https://www.etfdailynews.com/2024/10/10/cantor-fitzgerald-reaffirms-neutral-rating-for-tesla-nasdaqtsla/,  
      urlToImage: https://www.americanbankingnews.com/wp-content/timthumb/timthumb.php?src=https://www.marketbeat.com/logos/tesla-inc-logo.png?v=20221020135629&w=240&h=240&z=2,  
      publishedAt: "2024-10-10T08:47:38Z",  
      content: "Cantor Fitzgerald reissued their neutral rating on shares of Tesla (NASDAQ:TSLA – Free Report) in a report published on Monday, Benzinga reports. The firm currently has a $245.00 price target on the ... [+6145 chars]"  
    },  
    - {  
      - source: {  
        id: null,  
        name: "ETF Daily News"      },  
      author: "MarketBeat News",  
      title: "Cantor Fitzgerald Reaffirms "Neutral" Rating for Tesla (NASDAQ:TSLA)",  
      description: "Cantor Fitzgerald reissued their neutral rating on shares of Tesla (NASDAQ:TSLA – Free Report) in a report published on Monday, Benzinga reports. The firm currently has a $245.00 price target on the electric vehicle producer's stock. A number of other equitie...",  
      url: https://www.etfdailynews.com/2024/10/10/cantor-fitzgerald-reaffirms-neutral-rating-for-tesla-nasdaqtsla/,  
      urlToImage: https://www.americanbankingnews.com/wp-content/timthumb/timthumb.php?src=https://www.marketbeat.com/logos/tesla-inc-logo.png?v=20221020135629&w=240&h=240&z=2,  
      publishedAt: "2024-10-10T08:47:38Z",  
      content: "Cantor Fitzgerald reissued their neutral rating on shares of Tesla (NASDAQ:TSLA – Free Report) in a report published on Monday, Benzinga reports. The firm currently has a $245.00 price target on the ... [+6145 chars]"  
    }  
  ]  
}
```

Генерация моделей. Chat GPT

Трудности перевода

Сгенерируй модели для ответа на запрос everything с newsapi.org на Kotlin

Для того чтобы сделать запрос на News API (<https://newsapi.org>) на языке Kotlin, можно использовать библиотеку HTTP клиента, такую как `OkHttp`. Пример ниже показывает, как можно сгенерировать запрос для получения всех новостей по ключевому слову "everything" через NewsAPI.

Генерация моделей. Chat GPT

Модель = бизнес-логика

Шаги:

1. Загрузите библиотеку OkHttp в `build.gradle.kts` :

kotlin

Копировать код

```
dependencies {  
    implementation("com.squareup.okhttp3:okhttp:4.9.1")  
}
```

Генерация моделей. Chat GPT

Модель = бизнес-логика

```
import okhttp3.OkHttpClient
import okhttp3.Request
import okhttp3.Response
import java.io.IOException

fun getNews(apiKey: String, query: String): String? {
    val client = OkHttpClient()

    // Формируем URL запроса
    val url = "https://newsapi.org/v2/everything?q=$query&apiKey=$apiKey"

    // Создаем запрос
    val request = Request.Builder()
        .url(url)
        .build()

    // Выполняем запрос и обрабатываем ответ
    return try {
        val response: Response = client.newCall(request).execute()
        if (response.isSuccessful) {
            response.body?.string()
        } else {
            println("Ошибка запроса: ${response.code}")
            null
        }
    } catch (e: IOException) {
        println("Ошибка соединения: ${e.message}")
        null
    }
}
```


Генерация моделей. Chat GPT

Меняем запрос

Сгенерируй модели Data классы для ответа на запрос `everything` с `newsapi.org` на Kotlin

- Для обработки ответа от API NewsAPI в Kotlin удобно использовать библиотеку `Gson` для парсинга JSON в объектные модели данных. Ответ на запрос `everything` включает поля, такие как статьи, авторов, источники, заголовки и другие. Мы создадим несколько `data class` для представления этих данных.

Генерация моделей. Chat GPT

Меняем запрос

Результат

Вот пример классов данных, которые соответствуют JSON структуре ответа на запрос `everything`.

```
kotlin Копировать код

// Основной ответ от NewsAPI
data class NewsResponse(
    val status: String,           // Статус ответа (например, "ok")
    val totalResults: Int,       // Общее количество результатов
    val articles: List<Article>  // Список статей
)

// Класс для статьи
data class Article(
    val source: Source,         // Источник статьи
    val author: String?,       // Автор статьи (может быть null)
    val title: String,         // Заголовок статьи
    val description: String?,  // Описание статьи (может быть null)
    val url: String,           // Ссылка на статью
    val urlToImage: String?,   // Ссылка на изображение (может быть null)
    val publishedAt: String,    // Дата публикации
    val content: String?       // Основной контент статьи (может быть null)
)
```

Генерация моделей. Gemini

Прямой вопрос на русском



Сгенерируй модели для ответа на запрос everything с newsapi.org на Kotlin

Генерация моделей. Gemini

Прямой ответ на английском



Okay, here are the Kotlin data classes representing the response for the "everything" request from newsapi.org:

```
data class NewsResponse(  
    val status: String,  
    val totalResults: Int,  
    val articles: List<Article>  
)  
  
data class Article(  
    val source: Source,  
    val author: String?,  
    val title: String,  
    val description: String?,  
    val url: String,  
    val urlToImage: String?,  
    val publishedAt: String,  
    val content: String?  
)  
  
data class Source(  
    val id: String?,  
    val name: String  
)
```

Генерация моделей. Yandex GPT

Отказ...

Сгенерируй модели для ответа на запрос everything с newsapi.org на Kotlin

К сожалению, я не могу сгенерировать модели для ответа на запрос «everything» с NewsAPI.org на Kotlin, так как это требует более детального понимания структуры данных и требований проекта. Однако я могу предложить общий план действий:

1. **Изучите документацию:** ознакомьтесь с документацией NewsAPI.org, чтобы понять, какие данные доступны и как их можно получить.
2. **Создайте проект:** создайте новый проект на Kotlin и добавьте необходимые зависимости для работы с API.
3. **Получите токен доступа:** зарегистрируйтесь на сайте NewsAPI.org и получите токен доступа для вашего приложения.
4. **Напишите код:** напишите код для получения данных с помощью API и обработки полученных данных.
5. **Отобразите результаты:** отобразите полученные данные в удобном для пользователя виде.
6. **Тестирование:** проведите тестирование вашего кода, чтобы убедиться, что он работает корректно.
7. **Развертывание:** разверните ваше приложение на сервере или хостинге.

Генерация моделей. Yandex GPT

По своей инициативе написал код
сетевого клиента

Сам подобрал технологии

Вот пример кода для получения новостей с помощью NewsAPI.org:

```
import com.squareup.moshi.Moshi
import okhttp3.OkHttpClient
import okhttp3.Request
import java.io.IOException
import kotlin.concurrent.thread

class NewsApi {
    private val client = OkHttpClient()
    private lateinit var moshi: Moshi

    fun getNews(query: String): List<NewsItem> {
        val url = "https://newsapi.org/v2/top-headlines?country=us&"
        val request = Request.Builder().url(url).build()

        try {
            val response = client.newCall(request).execute()
            if (response.isSuccessful) {
                val newsJson = response.body!!.string()
                val data = moshi.adapter(NewsResponse::class.java).
                    return data.articles
            } else {
                throw IOException("Failed to fetch news")
            }
        } catch (e: IOException) {
            throw e
        }
    }
}
```

Генерация моделей. Yandex GPT

По своей инициативе написал код
сетевого клиента

Сам подобрал технологии

Вот пример кода для получения новостей с помощью NewsAPI.org:

```
import com.squareup.moshi.Moshi
import okhttp3.OkHttpClient
import okhttp3.Request
import java.io.IOException
import kotlin.concurrent.thread

class NewsApi {
    private val client = OkHttpClient()
    private lateinit var moshi: Moshi

    fun getNews(query: String): List<NewsItem> {
        val url = "https://newsapi.org/v2/top-headlines?country=us&"
        val request = Request.Builder().url(url).build()

        try {
            val response = client.newCall(request).execute()
            if (response.isSuccessful) {
                val newsJson = response.body!!.string()
                val data = moshi.adapter(NewsResponse::class.java)
                return data.articles
            } else {
                throw IOException("Failed to fetch news")
            }
        } catch (e: IOException) {
            throw e
        }
    }
}
```

Модель есть

Генерация моделей. Yandex GPT

Меняем запрос

Напиши код NewsResponse

Для написания кода модели NewsResponse на Kotlin, которая будет использоваться в примере получения новостей с помощью NewsAPI.org, можно использовать библиотеку Moshi для сериализации и десериализации JSON данных.

Вот пример кода для модели NewsResponse:

```
import com.squareup.moshi.Json
import java.util.*

data class NewsResponse(
    @Json(name = "status") val status: String,
    @Json(name = "totalResults") val totalResults: Int,
    @Json(name = "articles") val articles: List<NewsItem>
)
```

Здесь мы определяем класс NewsResponse с тремя свойствами:

- status — статус ответа от API;
- totalResults — общее количество результатов;
- articles — список объектов NewsItem, которые содержат информацию о новостях.



Скопировать ответ

Генерация моделей. Yandex GPT

Просим догенерировать модель

Напиши код NewsItem

Вот пример кода для модели NewsItem на Kotlin:

```
import com.squareup.moshi.Json

data class NewsItem(
    @Json(name = "author") val author: String?,
    @Json(name = "title") val title: String,
    @Json(name = "description") val description: String,
    @Json(name = "url") val url: String,
    @Json(name = "content") val content: String?
)
```

Здесь мы определяем класс NewsItem с пятью свойствами:

- author — автор новости;
- title — заголовок новости;
- description — описание новости;
- url — URL новости;
- content — содержание новости.



Скопировать ответ

Генерация моделей. Алиса Про

Попробуем другой продукт

Результат явно лучше

Сгенерируй модели для ответа на запрос everything с newsapi.org на Kotlin

```
// Модель для статьи
data class NewsItem(
    val title: String,
    val content: String,
    val date: String,
    val imageUrl: String
)

// Модель для списка статей
data class NewsList(
    val articles: List<NewsItem>
)

// Модель для состояния пользовательского интерфейса
data class NewsUiState(
    val articles: List<NewsItem> = emptyList(),
    val userMessages: List<String> = emptyList()
)
```

Эти модели помогут вам структурировать данные, полученные с newsapi.org, и использовать их в вашем приложении.

Генерация моделей. GigaCode

Модели и ретрофит

You

Сгенерируй модели для ответа на запрос everything с newsapi.org на Kotlin

Генерация моделей. GigaCode

Модели и ретрофит

```
data class Source(  
    val id: String?,  
    val name: String  
)
```

```
data class NewsResponse(  
    val status: String,  
    val totalResults: Int,  
    val articles: List<Article>  
)
```

```
data class Article(  
    val source: Source,  
    val author: String?,  
    val title: String,  
    val description: String?,  
    val url: String,  
    val urlToImage: String?,  
    val publishedAt: String,  
    val content: String?  
)
```

Генерация моделей. GigaCode

Модели и ретрофит



Для работы с API newsapi.org в Kotlin, вы можете использовать библиотеку Retrofit. Вот пример кода:

```
interface NewsAPIService {
    @GET("everything")
    fun getEverything(
        @Query("q") query: String,
        @Query("from") from: String,
        @Query("sortBy") sortBy: String,
        @Query("apiKey") apiKey: String
    ): Call<NewsResponse>
}
```

Генерация моделей. Meta-LLaMa

You

Все по делу, ничего лишнего

Сгенерируй модели для ответа на запрос everything с newsapi.org на Kotlin

Assistant

Meta-Llama-3.1-8B-Instruct-GGUF

Модели для ответа на запрос everything с newsapi.org на Kotlin:

Генерация моделей. Meta-LLaMa

Все по делу, ничего лишнего

Assistant

Модели для ответа на запрос everything с newsapi.org на Kotlin:

```
data class Source(  
    val id: String,  
    val name: String,  
    val description: String,  
    val url: String,  
    val category: String,  
    val language: String,  
    val country: String  
)  
  
data class Article(  
    val source: Source,  
    val author: String,  
    val title: String,  
    val description: String,  
    val url: String,  
    val publishedAt: String,  
    val content: String,  
    val urlToImage: String,  
    val id: String  
)  
  
data class NewsResponse(  
    val status: String,  
    val totalResults: Int,  
    val articles: List<Article>  
)
```

Summary. Справились без проблем

- Gemini
- GigaCode
- Алиса Про
- Meta LLaMa

Summary. Ошиблись, наделали лишнего

- Yandex GPT
- Chat GPT

Composable UI по запросу. Chat GPT

Пишем подробный запрос с описанием UI

Указываем импорты и нужное решение

Напиши Composable для экрана списка новостей. Вынеси строку списка в отдельный Composable. В элемента списка используй картинку 100 на 100 dp, справа от картинки расположи столбец из 3х текстовых полей (заголовков, контент и дату). В качестве моделей данных используй Article. Используй Material3 и импорты. Coil для загрузки изображения



Вот пример `Composable` экрана для отображения списка новостей, где каждая строка списка вынесена в отдельный `Composable`. Мы будем использовать библиотеку `Coil` для загрузки изображений и `Material3` для создания UI компонентов.

Composable UI по запросу. Chat GPT

Получаем адаптированную модель и точные Composable

(все не влезло в слайд)

```
// Модель данных для статьи
data class Article(
    val title: String,
    val content: String?,
    val publishedAt: String,
    val urlToImage: String?
)
```

```
// Основной экран для отображения списка новостей
@Composable
fun NewsListScreen(articles: List<Article>) {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text("Новости") }
            )
        }
    ) { padding ->
        LazyColumn(
            contentPadding = padding,
            modifier = Modifier.fillMaxSize(),
            verticalArrangement = Arrangement.spacedBy(8.dp)
        ) {
            items(articles) { article ->
                NewsListItem(article)
            }
        }
    }
}
```

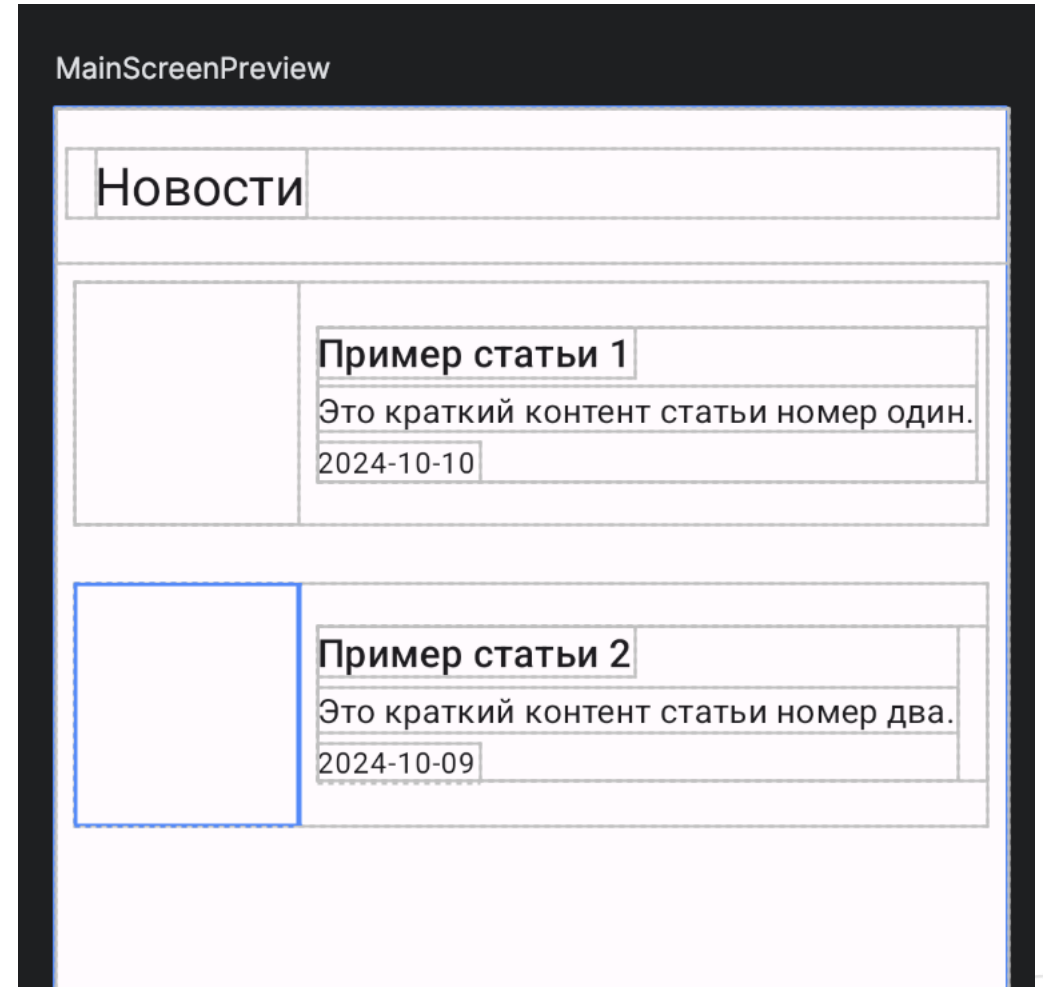


Composable UI по запросу. Chat GPT

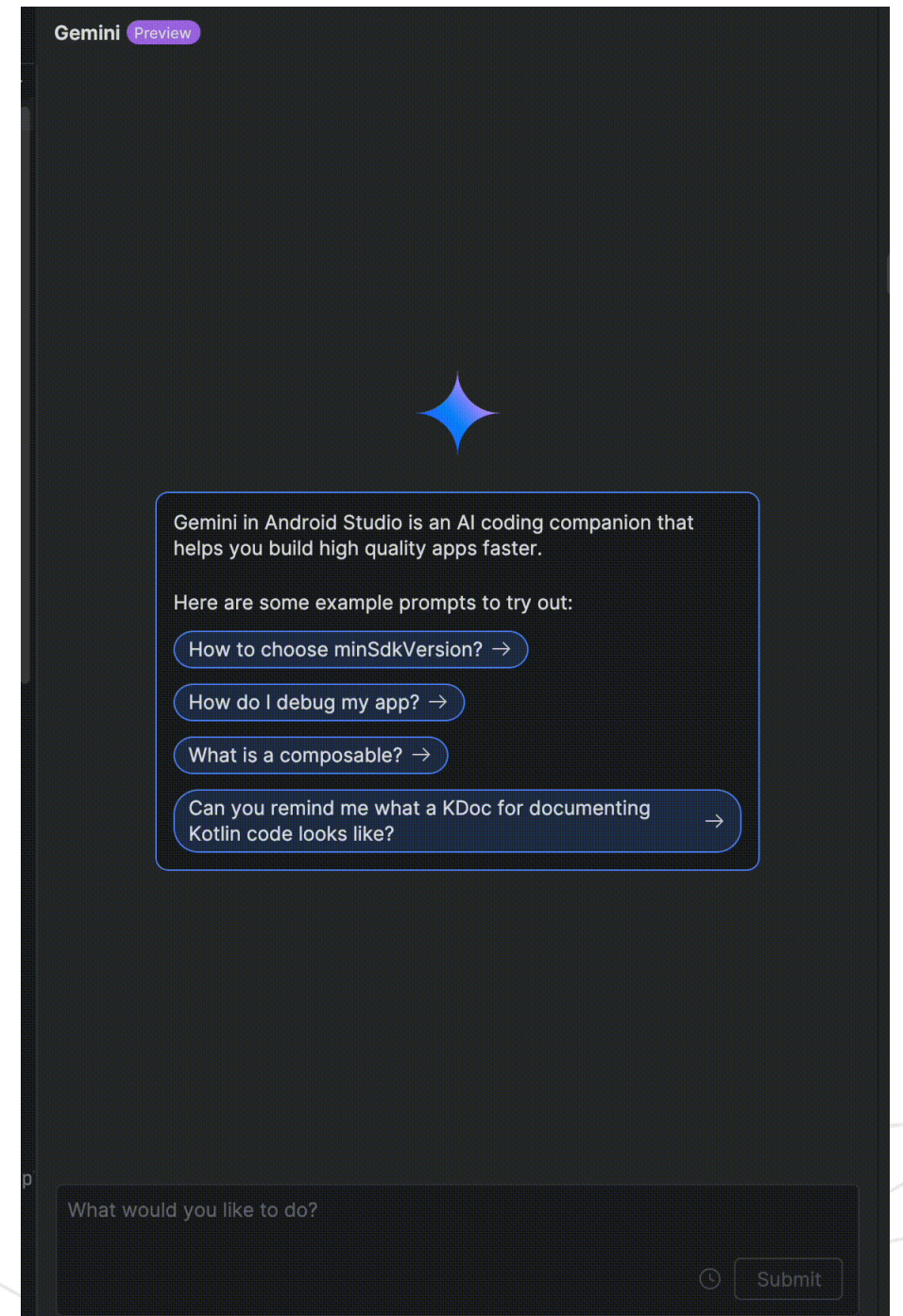
- NewsListScreen
- NewsItemScreen

Composable UI по запросу. Chat GPT

Получаем гибкую точную
верстку



Composable UI по запросу. Gemini

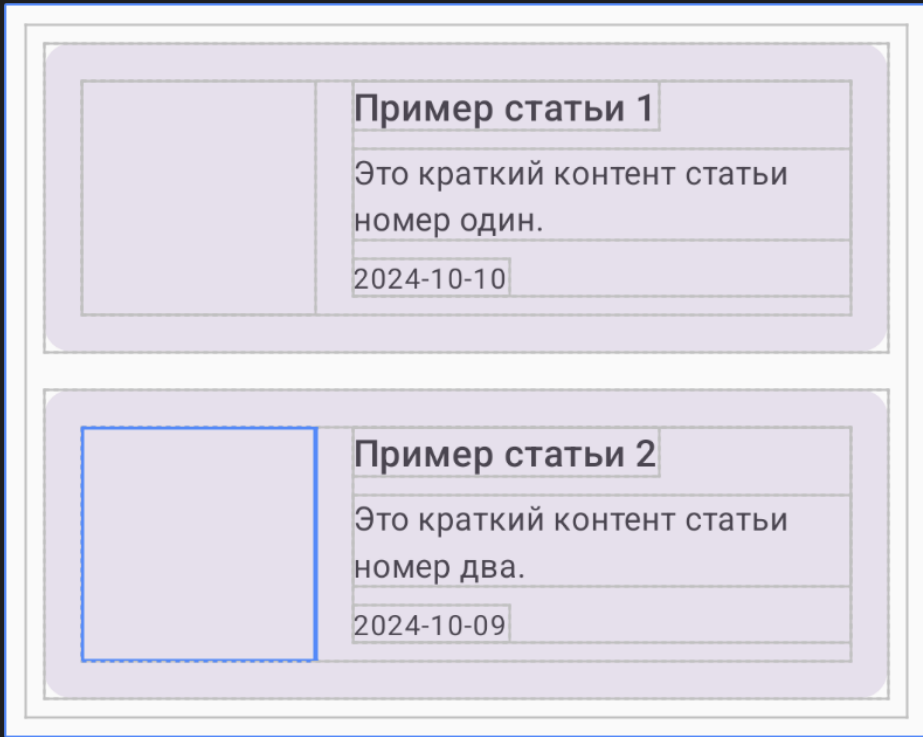


Composable UI по запросу. Gemini



Напиши Composable для экрана списка новостей. Вынеси строку списка в отдельный Composable. В элементе списка используй картинку 100 на 100 dp, справа от картинки расположи столбец из 3х текстовых полей (заголовок, контент и дату). В качестве моделей данных используй Article. Используй Material3 и импорты. Coil для загрузки изображения

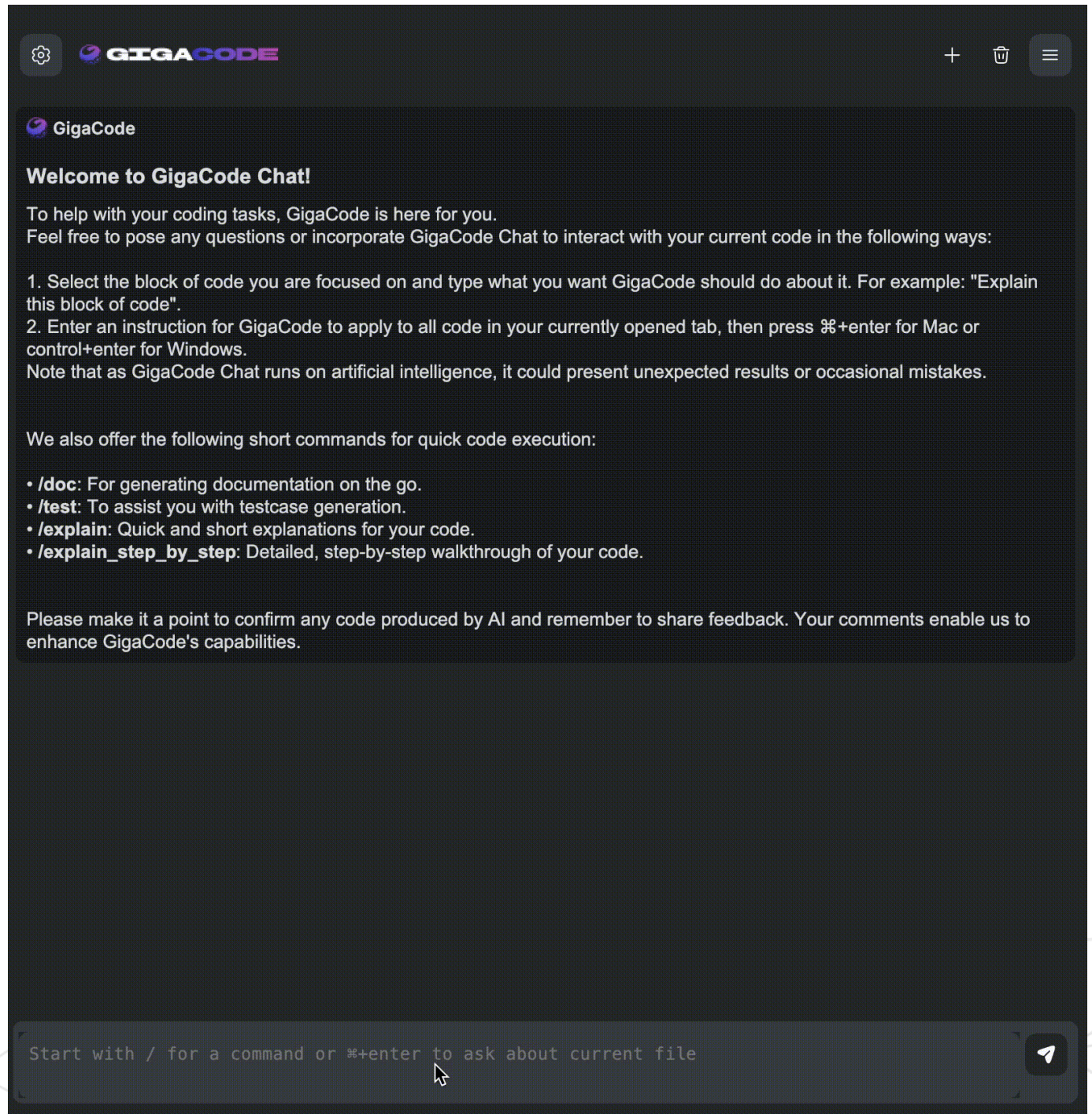
NewsListItemPreview





Composable UI по запросу. Gemini

- NewsListScreen
- NewsItemScreen

Composable UI по запросу. GigaCode



 **GIGACODE**

 GigaCode

Welcome to GigaCode Chat!

To help with your coding tasks, GigaCode is here for you.
Feel free to pose any questions or incorporate GigaCode Chat to interact with your current code in the following ways:


1. Select the block of code you are focused on and type what you want GigaCode should do about it. For example: "Explain this block of code".
2. Enter an instruction for GigaCode to apply to all code in your currently opened tab, then press `⌘+enter` for Mac or `control+enter` for Windows.

Note that as GigaCode Chat runs on artificial intelligence, it could present unexpected results or occasional mistakes.

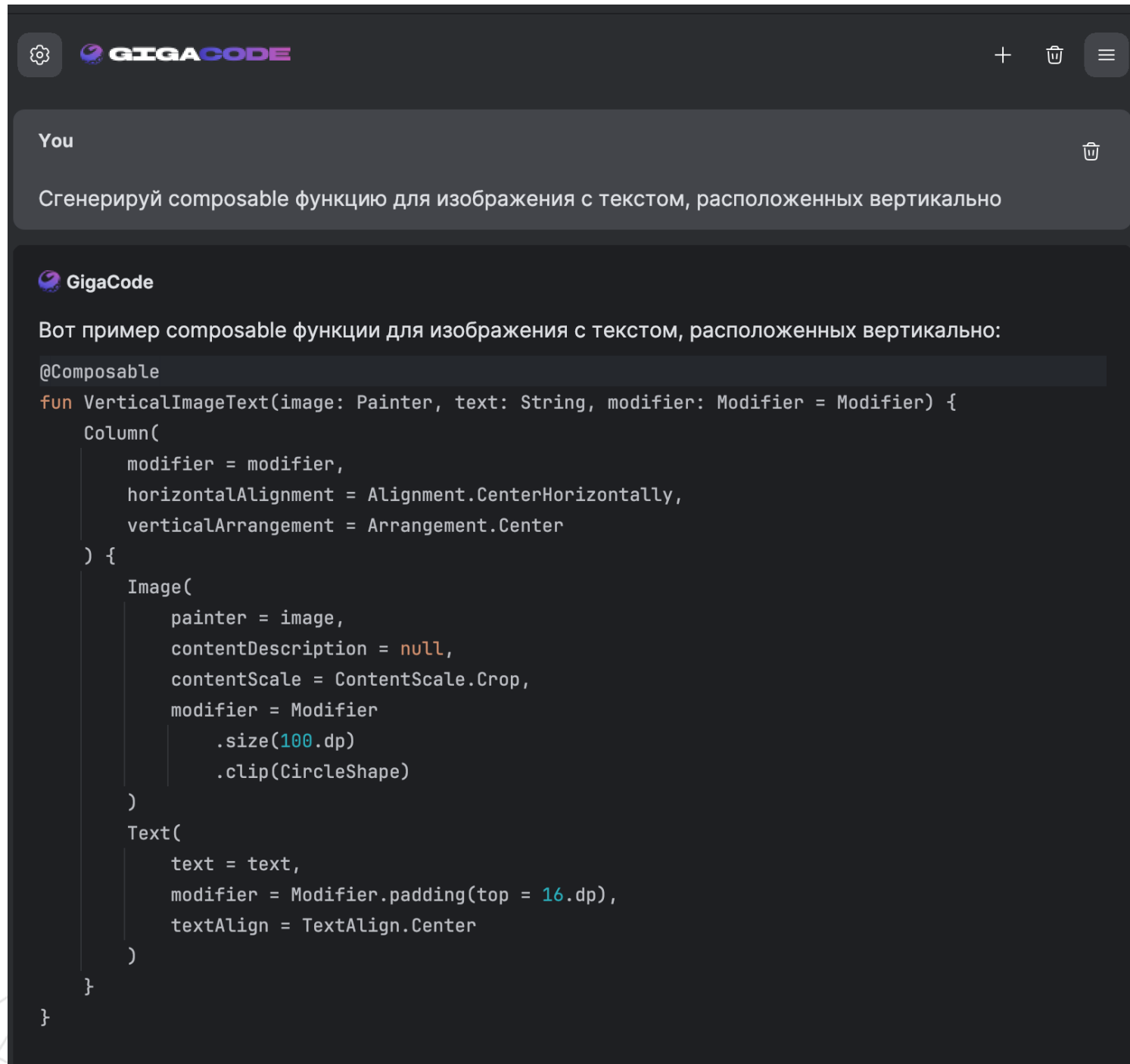
We also offer the following short commands for quick code execution:

- `/doc`: For generating documentation on the go.
- `/test`: To assist you with testcase generation.
- `/explain`: Quick and short explanations for your code.
- `/explain_step_by_step`: Detailed, step-by-step walkthrough of your code.

Please make it a point to confirm any code produced by AI and remember to share feedback. Your comments enable us to enhance GigaCode's capabilities.

Start with `/` for a command or `⌘+enter` to ask about current file 

Composable UI по запросу. GigaCode



The screenshot shows a chat window with a dark theme. At the top, there's a header with a settings gear, the GigaCode logo, and navigation icons. The chat history shows a message from 'You' asking to generate a composable function for vertically aligned image and text. Below that, a response from 'GigaCode' provides a Kotlin code snippet for a composable function named VerticalImageText. The code uses Column, Image, and Text composable functions to achieve the desired layout.

```
@Composable
fun VerticalImageText(image: Painter, text: String, modifier: Modifier = Modifier) {
    Column(
        modifier = modifier,
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Image(
            painter = image,
            contentDescription = null,
            contentScale = ContentScale.Crop,
            modifier = Modifier
                .size(100.dp)
                .clip(CircleShape)
        )
        Text(
            text = text,
            modifier = Modifier.padding(top = 16.dp),
            textAlign = TextAlign.Center
        )
    }
}
```

Composable UI по запросу. GigaCode

Аналогичный результат Gemini

NewsListItemPreview

Пример статьи 1

Это краткий контент статьи номер один.

2024-10-10

Пример статьи 2

Это краткий контент статьи номер два.

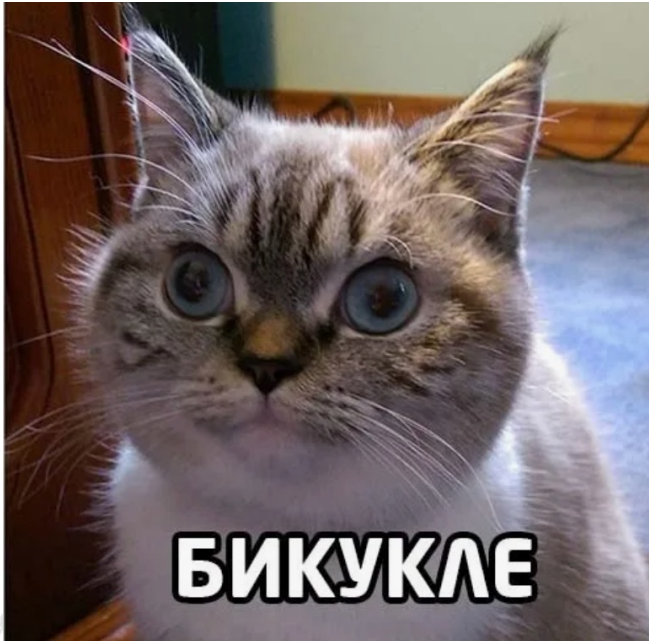
2024-10-09

Composable UI по запросу. Алиса Про

Алиса Про

Что-то пошло не так

Напиши Composable для экрана списка новостей. Вынеси строку списка в отдельный Composable. В элемента списка используй картинку 100 на 100 dp, справа от картинки расположи столбец из 3х текстовых полей (заголовок, контент и дату). В качестве моделей данных используй Article. Используй Material3 и импорты. Coil для загрузки изображения



Composable UI по запросу. Yandex GPT

Начал давать полный
развернутый ответ

Для создания экрана списка новостей с использованием Composable и Material 3 в Kotlin, вам потребуется выполнить следующие шаги:

1. **Создайте новый проект:** создайте новый проект на Kotlin и добавьте необходимые зависимости для работы с API, а также зависимости для Material 3 и Coil.
2. **Получите данные:** получите данные с помощью NewsAPI.org и преобразуйте их в список объектов Article.
3. **Определите компоуемый экран списка:** определите компоуемый экран ListScreen, который будет отображать список новостей.
4. **Импортируйте необходимые библиотеки:** импортируйте библиотеки для работы с Material 3, Coil и другими зависимостями.
5. **Создайте компоуемый элемент списка:** создайте компоуемый элемент ListItem, который будет представлять одну новость. В этом элементе вы можете использовать Coil для загрузки изображения и отображения его рядом с текстом.
6. **Отобразите список:** в компоуемом экране ListScreen отобразите список новостей, используя компоуемые элементы ListItem.
7. **Добавьте логику:** добавьте логику для обработки событий (например, нажатие на элемент списка) и обновления данных.
8. **Протестируйте:** проведите тестирование вашего кода, чтобы убедиться, что он работает корректно.
9. **Разверните:** разверните ваше приложение на сервере или хостинге.

Composable UI по запросу. Yandex GPT

Но потом устал

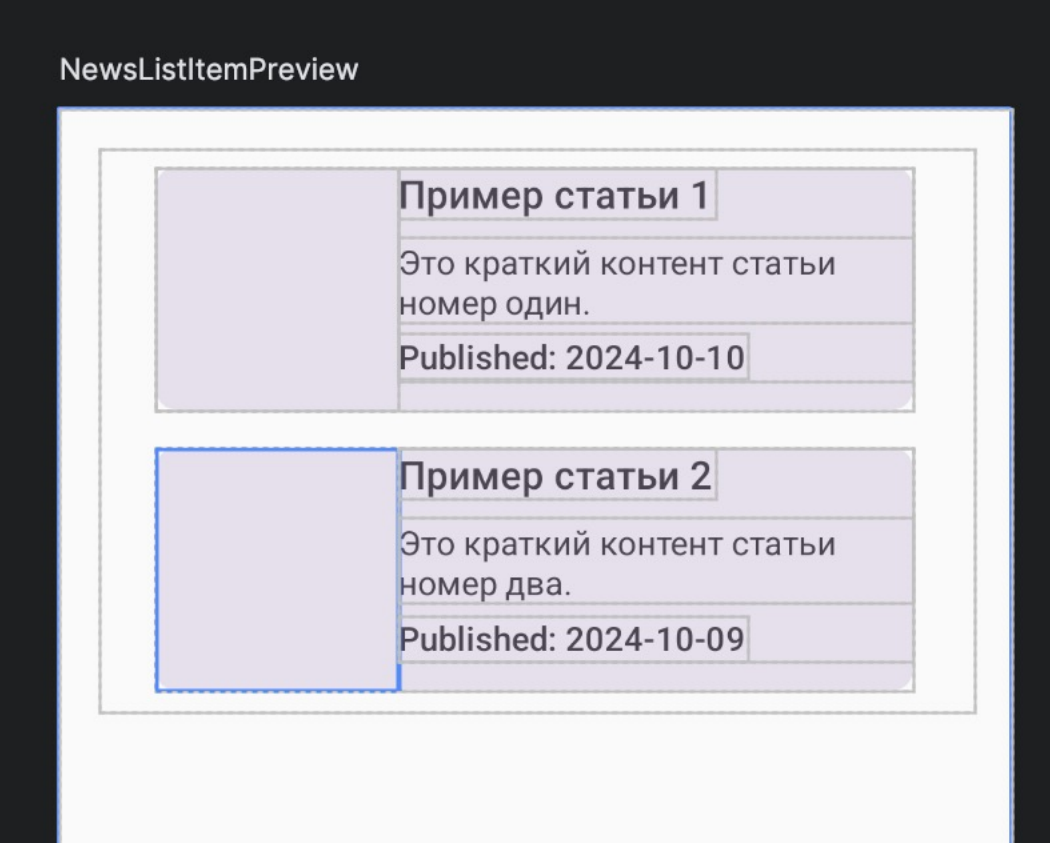


```
@Composable
fun ListScreen(news: List<Article>) {
    LazyColumn {
        items(news) { article ->
            ListItem(article)
        }
    }
}

@Composable
private fun ListItem(article: Article) {
    Surface(
        modifier = Modifier
            .fillMaxWidth()
            .padding(16.dp),
        shape = RoundedCornerShape(topStart = 16.dp, topEnd = 16.dp)
    ) {
        Row(verticalAlignment = Alignment.CenterVertically) {
            Image(
                painter = rememberAsyncImagePainter(
                    data = article.imageUrl,
                    builder = {
                        crossfade(true)
                        size(100.dp)
                    }
                )
            )
        }
    }
}
```

Composable UI по запросу. Meta LLaMa

Почти хорошо



Composable UI по запросу. Meta LLaMa

Но местами проблемы

Несуществующая логика

```
@Composable
fun CoilImage(
    data: String,
    contentDescription: String?,
    modifier: Modifier = Modifier,
) {
    val imageLoader = rememberCoilImageLoader()
    Image(
        painter = rememberAsyncImagePainter(model = data),
        contentDescription = contentDescription,
        modifier = modifier
            .align(Alignment.Center)
    )
}
```

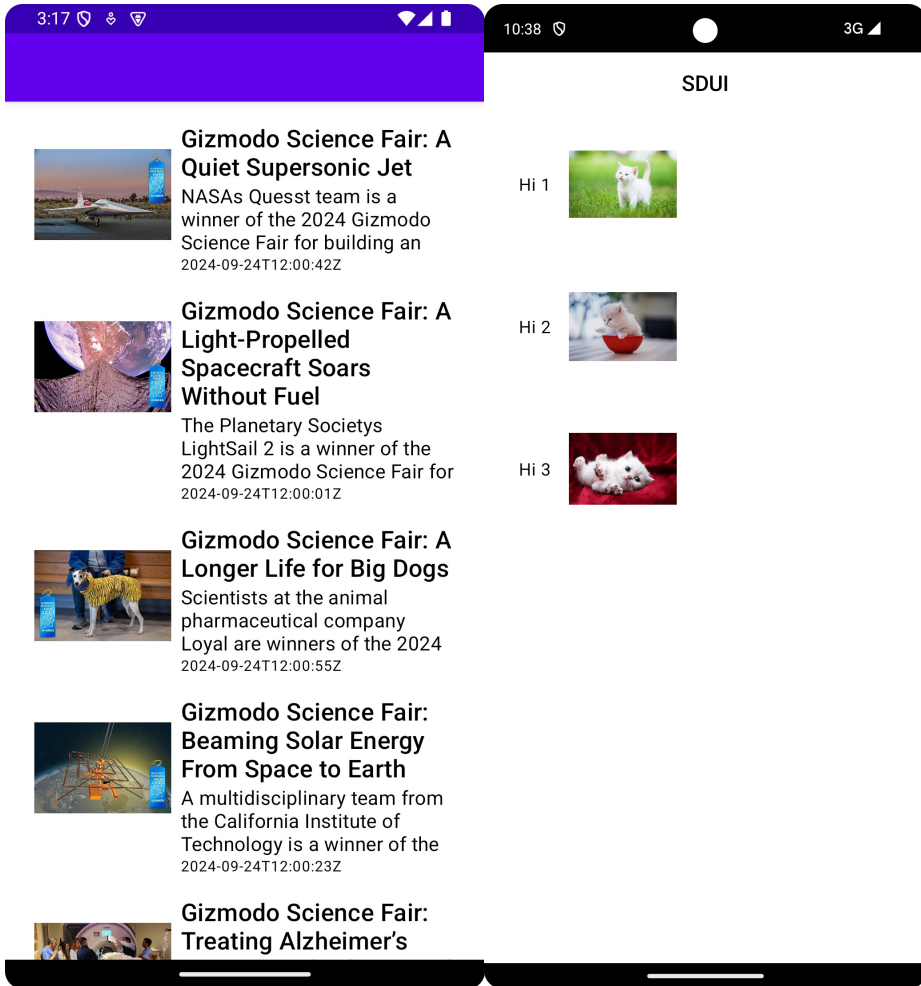

Общие ошибки

- В базе знаний старая версия Coil
- Пропадает часть параметров
- Местами едут стили

Summary

- Почти все справились
- Yandex GPT устал
- LLaMa торопится

Composable из UI



```
@Composable
fun ScienceFairScreen() {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text("Gizmodo Science Fair") },
                backgroundColor = MaterialTheme.colorScheme.primary
            )
        }
    ) { paddingValues ->
        ArticleList(
            modifier = Modifier.padding(paddingValues)
        )
    }
}
```

Composable из UI. Chat GPT



To create a composable UI for the screen you uploaded (which appears to be a news or article feed), we can utilize Jetpack Compose in Android. The following example outlines how to implement a similar design, with an app bar, a list of articles, images, and text content.



Gizmodo Science Fair: A Quiet Supersonic Jet

NASAs Quesst team is a winner of the 2024 Gizmodo Science Fair for building an
2024-09-24T12:00:42Z



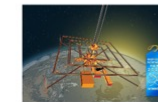
Gizmodo Science Fair: A Light-Propelled Spacecraft Soars Without Fuel

The Planetary Societys LightSail 2 is a winner of the 2024 Gizmodo Science Fair for
2024-09-24T12:00:01Z



Gizmodo Science Fair: A Longer Life for Big Dogs

Scientists at the animal pharmaceutical company Loyal are winners of the 2024
2024-09-24T12:00:55Z



Gizmodo Science Fair: Beaming Solar Energy From Space to Earth

A multidisciplinary team from the California Institute of Technology is a winner of the

Create composable for UI in uploaded screen

Composable из UI. Chat GPT

Сам создал data модели и мокк

```
data class Article(  
    val title: String,  
    val description: String,  
    val date: String,  
    val imageUrl: String  
)  
  
val articles = listOf(  
    Article(  
        "Gizmodo Science Fair: A Quiet Supersonic Jet",  
        "NASA's Quest team is a winner of the 2024 Gizmodo Science Fair for building a quiet supersonic jet",  
        "2024-09-24",  
        "https://example.com/jet_image.png"  
    ),  
    Article(  
        "Gizmodo Science Fair: A Light-Propelled Spacecraft Soars Without Fuel",  
        "The Planetary Society's LightSail 2 is a winner of the 2024 Gizmodo Science Fair for building a light-propelled spacecraft",  
        "2024-09-24",  
        "https://example.com/spacesail_image.png"  
    ),  
    // Add more articles...  
)
```

Composable из UI. Chat GPT

UI для элемента и списка

@Composable

Копировать код

```
fun ArticleItem(article: Article) {
    Row(modifier = Modifier.fillMaxWidth()) {
        Image(
            painter = rememberImagePainter(data = article.imageUrl),
            contentDescription = null,
            modifier = Modifier
                .size(100.dp)
                .padding(end = 8.dp),
            contentScale = ContentScale.Crop
        )
        Column(modifier = Modifier.weight(1f)) {
            Text(
                text = article.title,
                fontWeight = FontWeight.Bold,
                fontSize = 16.sp,
                maxLines = 2,
                overflow = TextOverflow.Ellipsis
            )
            Text(
                text = article.description,
                fontSize = 14.sp,
                maxLines = 2,
                overflow = TextOverflow.Ellipsis
            )
            Text(
                text = article.date,
                fontSize = 12.sp,
                color = MaterialTheme.colorScheme.onSurfaceVariant
            )
        }
    }
}
```



Composable из UI. Chat GPT

UI для элемента и списка

```
@Composable
fun ScienceFairScreen() {
    Scaffold(
        topBar = {
            TopAppBar(
                title = { Text("Gizmodo Science Fair") },
                backgroundColor = MaterialTheme.colorScheme.primary
            )
        }
    ) { paddingValues ->
        ArticleList(
            modifier = Modifier.padding(paddingValues)
        )
    }
}
```

```
@Composable
fun ArticleList(modifier: Modifier = Modifier) {
    LazyColumn(modifier = modifier.padding(16.dp)) {
        items(articles) { article ->
            ArticleItem(article)
            Spacer(modifier = Modifier.height(16.dp))
        }
    }
}
```

Composable из UI. Chat GPT

И Preview

ScienceFairScreenPreview

Gizmodo Science Fair

Gizmodo Science Fair: A Quiet Supersonic Jet

NASA's Quest team is a winner of the 2024 Gizmodo Science Fair for building an inn...
2024-09-24

Gizmodo Science Fair: A Light-Propelled Spacecraft Soars ...

The Planetary Society's LightSail 2 is a winner of the 2024 Gizmodo Science Fair.
2024-09-24


Composable из UI. Chat GPT

Ошибки:

- TopAppBar
- В базе знаний старая версия Coil

```
Scaffold(  
  topBar = {  
    TopAppBar(  
      title = { Text(text: "Gizmodo Science Fair") },  
      backgroundColor = MaterialTheme.colorScheme.primary  
    )  
  }  
) { paddingValues ->
```

Composable из UI. Chat GPT плагин

 annazharkova

Attached image ^

Screenshot_20241010_031803.png



Create a UI with Jetpack Compose for this image

 CodeGPT



Unknown API response. Code: 413, Body:

De

413 Request Entity Too Large

Composable из UI. Chat GPT плагин

Берем картинку попроще



SDUI

Hi 1



Hi 2



Hi 3



Composable из UI. Chat GPT плагин

Получаем код

Негибко, но окэй

```
@Composable
fun GreetingList() {
    val greetings = listOf("Hi 1", "Hi 2", "Hi 3")
    val imageResId = R.drawable.cat // Replace with your actual image resource ID

    Column(horizontalAlignment = Alignment.CenterHorizontally) {
        for (greeting in greetings) {
            Text(text = greeting, modifier = Modifier.padding(8.dp))
            Image(
                painter = painterResource(id = imageResId),
                contentDescription = null
            )
        }
    }
}
```

Composable из UI. Chat GPT плагин

Сравниваем



SDUI

Hi 1



Hi 2



Hi 3



GreetingListPreview

Hi 1



Hi 2



Hi 3



Composable из UI. Gemini. Web-версия

<https://gemini.google.com>

```
)  
Image(  
    painter = painterResource(id = R.drawable.image_2), // Repl  
    contentDescription = "Hi 2 image",  
    modifier = Modifier.padding(bottom = 16.dp)  
)  
  
Text(  
    text = "Hi 3",  
    style = MaterialTheme.typography.titleMedium,  
)  
Image(  
    painter = painterResource(id = R.drawable.image_3), // Repl  
    contentDescription = "Hi 3 image",  
    modifier = Modifier.padding(bottom = 16.dp)  
)  
}  
}  
  
@Preview  
@Composable  
fun SDUIPreview() {  
    SDUI()  
}
```

Используйте код [с осторожностью](#).

Key improvements and considerations:

- **Material 3:** Uses Material 3 typography and theming for a modern look and feel.
- **Kotlin code style:** Adheres to Kotlin code style conventions for readability and maintainability.

Введите запрос



Gemini может показывать неточную информацию, в том числе о людях. Обязательно перепроверяйте ответы.
[Ваша конфиденциальность и приложения Gemini](#)

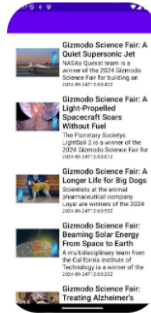
Composable из UI. Gemini. Web-версия

<https://gemini.google.com>

Успел эволюционировать



Create composable functions for UI in the uploaded screen. Use Jetpack Compose



Understanding the Screen:

Based on the image, the screen appears to be a list of news articles related to the Gizmodo Science Fair. Each article has a title, a brief description, and an image.

Defining Composable Functions:

Here's how we can break down the UI into composable functions:

1. **NewsCard:** A composable function to represent a single news article.
2. **NewsList:** A composable function to display a list of NewsCards.

Показать черновики 



Composable из UI. Gemini. Web-версия

Модель, UI и мокки

```
data class NewsArticle(  
    val title: String,  
    val description: String,  
    val imageUrl: String  
)
```


Composable из UI. Gemini. Web-версия

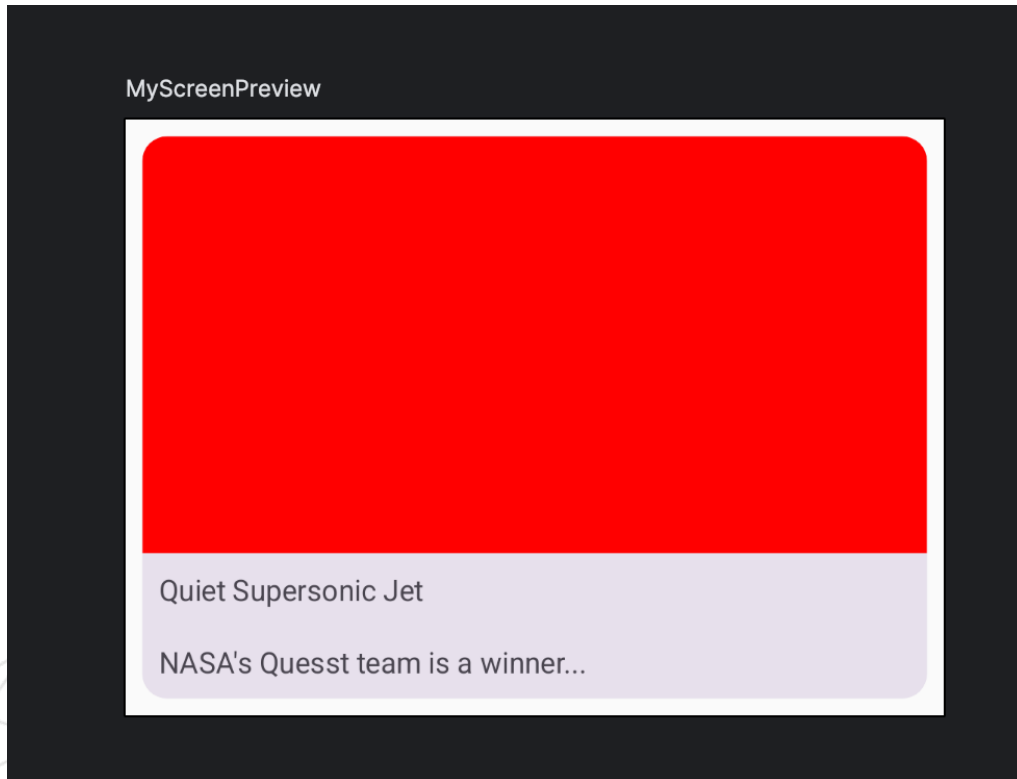
Модель, UI и мокки

```
@Composable
fun NewsList(newsArticles: List<NewsArticle>) {
    Column {
        newsArticles.forEach { newsArticle ->
            NewsCard(newsArticle)
        }
    }
}
```

```
@Composable
fun NewsCard(newsArticle: NewsArticle) {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(8.dp)
    ) {
        Column {
            Image(
                painter = rememberImagePainter(newsArticle.imageUrl),
                contentDescription = newsArticle.title,
                modifier = Modifier
                    .fillMaxWidth()
                    .height(200.dp),
                contentScale = ContentScale.Crop
            )
            Text(
                text = newsArticle.title,
                modifier = Modifier.padding(8.dp)
            )
            Text(
                text = newsArticle.description,
                modifier = Modifier.padding(8.dp)
            )
        }
    }
}
```

Composable из UI. Gemini. Web-версия

Модель, UI и мокки



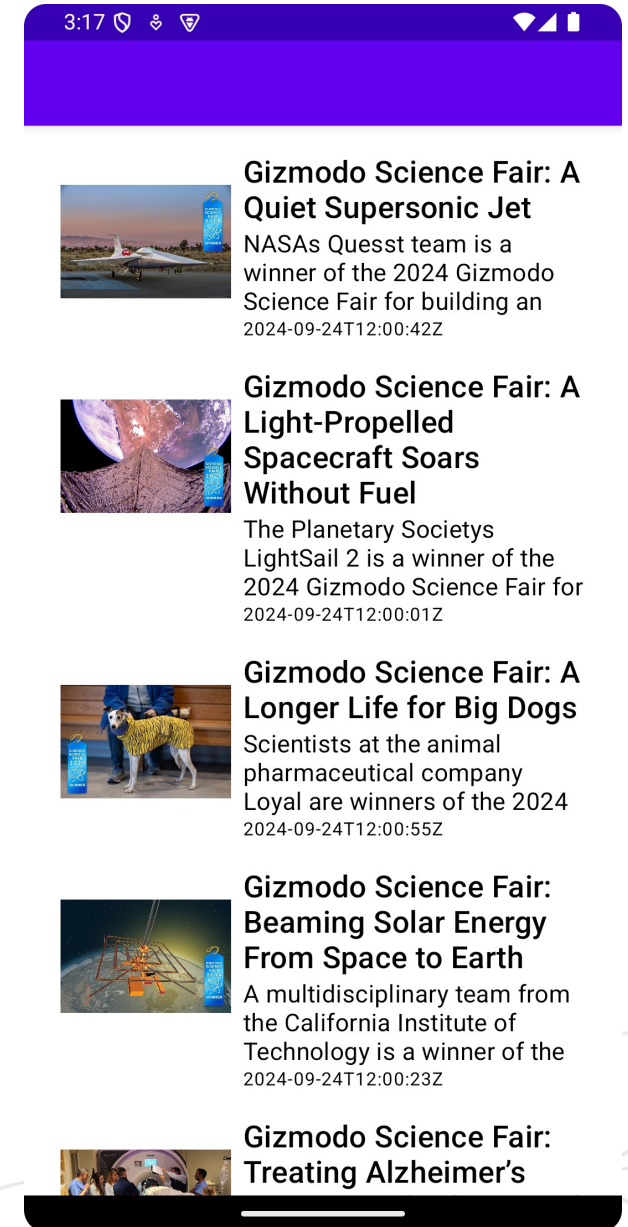
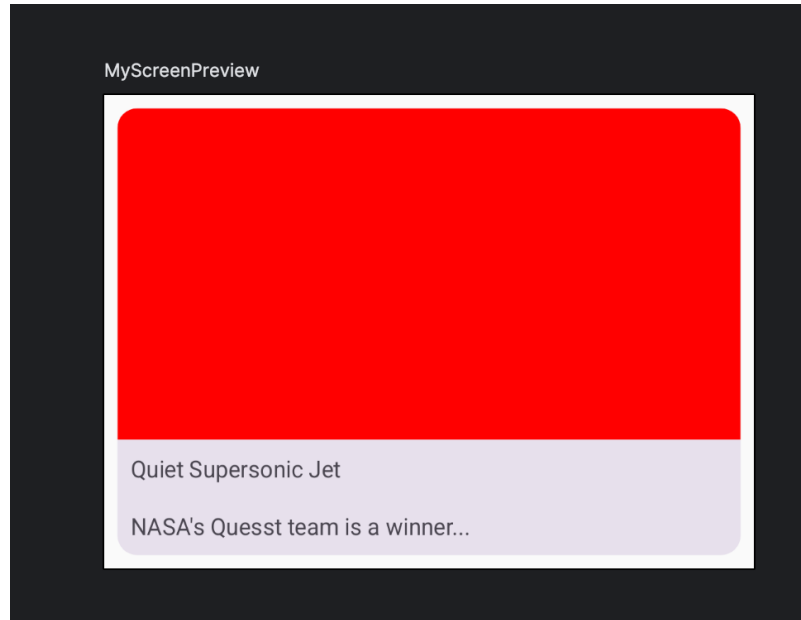
Kotlin

```
@Composable
fun MyScreen() {
    val newsArticles = listOf(
        NewsArticle("Quiet Supersonic Jet", "NASA's Quesst team is a winner..."),
        // ... other news articles
    )

    NewsList(newsArticles)
}
```

Composable из UI. Gemini. Web-версия

Сравним



Composable из UI. Gemini. Web-версия

Ошибки:

- Разметка
- Поле потерял
- Стили

Composable из UI. LM Studio

Нет инструментария для помощи моделям анализировать картинки



Screenshot_2024'

Опиши, что в pdf файле

Assistant `Meta-Llama-3.1-8B-Instruct-GGUF`

- ✓ Chosen context injection strategy: 'inject-full-content'. All content can fit into the context
- ✓ Done predicting

В PDF-файле "1728548560681 - 467.pdf" содержится следующая информация:

- На странице 0 имеется текстовая часть.
- В конце файла указано окончание его содержимого.

Если вы хотите узнать больше о содержании этого PDF-файла, вы можете попросить предоставление отдельных страниц или секций содержания.

Кто не взялся

- Yandex GPT – нет поля для ввода изображения
- GigaCode – аналогично
- Gemini плагин – аналогично
- LM Studio – только Pdf с текстом

Что имеем сейчас

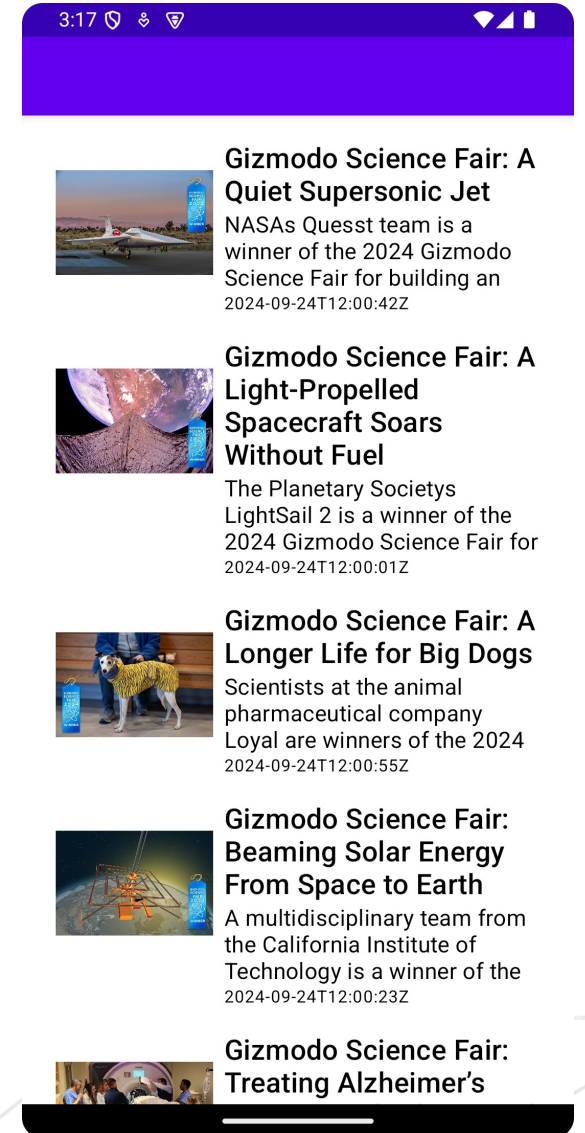
- Некоторые решения сразу дают больше, чем их просишь (бизнес-логика у GPT)
- Точечные и прямые вопросы обрабатывают хорошо все

А если все сразу?



Сгенерируем приложение полностью

Напиши приложение Android для отображения списка новостей с newsapi.org по запросу everything. Для сетевого клиента используй Retrofit, для многопоточности coroutines. UI напиши на Jetpack Compose. Используй DI на Koin. Используй Material3 и Kotlin code-style



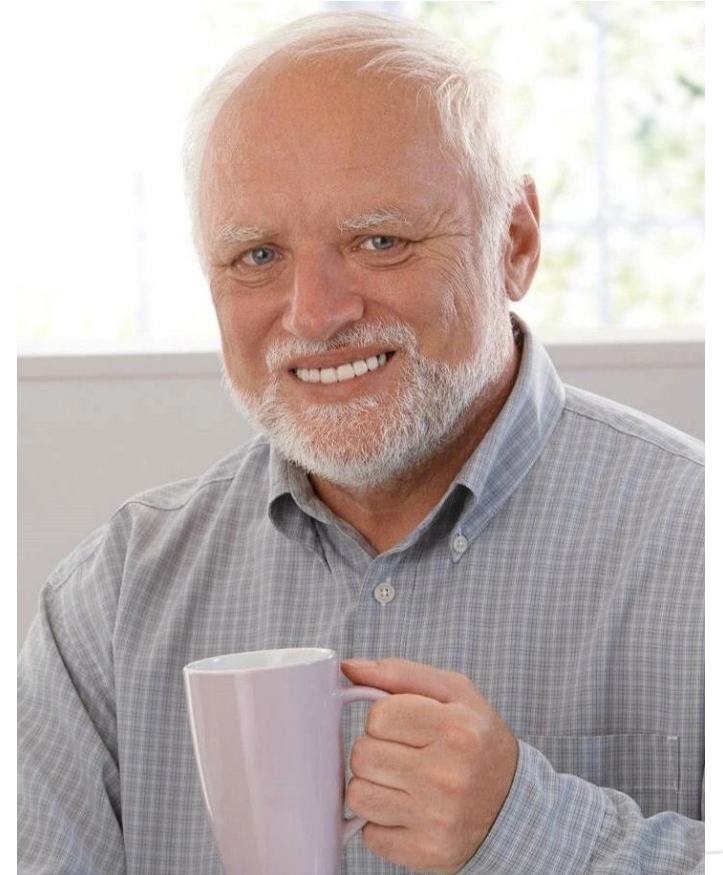
Спойлер: все (почти) справились

- Chat GPT
- Gemini
- Алиса Про
- Meta LLaMa

Спойлер: все (почти) справились

- Chat GPT
- Gemini
- Алиса Про
- Meta LLaMa

С нюансами...



Что сделали

- Сами вывели ViewModel
- Добавили UiState
- Разделили UI на разные Composable
- Структурировали сетевой слой
- Разобрали на сложные модели данных

Что сделали

Добавили UiState (Gemini и Алиса)

```
sealed class NewsState {  
    object Loading : NewsState()  
    data class Success(val articles: List<Article>) : NewsState()  
    data class Error(val message: String) : NewsState()  
}
```

```
// Модель для состояния пользовательского интерфейса  
data class NewsUiState(  
    val articles: List<NewsItem> = emptyList(),  
    val userMessages: List<String> = emptyList()  
)
```

Что сделали

Сами вывели ViewModel
(пример GPT)

```
class NewsViewModel(private val newsApiService: NewsApiService) : ViewModel() {  
  
    private val _newsState = MutableStateFlow<List<Article>>(emptyList())  
    val newsState: StateFlow<List<Article>> = _newsState  
  
    fun getNews(query: String) {  
        viewModelScope.launch {  
            try {  
                val response = newsApiService.getNews(query)  
                _newsState.value = response.articles  
            } catch (e: Exception) {  
                // Обработка ошибки  
            }  
        }  
    }  
}
```

Chat GPT. Нюансы

Одно из полных решений
(почти сразу)

Разметка странная

```
@Composable
fun NewsItem(article: Article) {
    Column(modifier = Modifier.fillMaxWidth().padding(8.dp)) {
        Text(text = article.title ?: "No Title", style = MaterialTheme.typography)
        Spacer(modifier = Modifier.height(8.dp))
        article.urlToImage?.let {
            Image(
                painter = rememberAsyncImagePainter(it),
                contentDescription = null,
                modifier = Modifier.fillMaxWidth().height(200.dp),
                contentScale = ContentScale.Crop
            )
        }
        Spacer(modifier = Modifier.height(8.dp))
        Text(text = article.description ?: "No Description", style = MaterialTher
    }
}
```

Алиса Про. Ньюансы

Почти идеальное
решение

```
// ViewModel для управления состоянием пользовательского интерфейса
class NewsViewModel @Inject constructor(private val newsApi: NewsApi) {
    private val _uiState = MutableStateFlow(NewsUiState())
    val uiState: StateFlow<NewsUiState> = _uiState.asStateFlow()

    init {
        viewModelScope.launch {
            fetchNewsItems()
        }
    }

    private suspend fun fetchNewsItems() {
        val newsList = newsApi.fetchNewsItems()
        _uiState.update { it.copy(articles = newsList.articles) }
    }
}
```


Алиса Про. Ньюансы

Почти идеальное
решение

```
// ViewModel для управления состоянием пользовательского интерфейса
class NewsViewModel @Inject constructor(private val newsApi: NewsApi,
private val _uiState = MutableStateFlow(NewsUiState())
val uiState: StateFlow<NewsUiState> = _uiState.asStateFlow())

init {
    viewModelScope.launch {
        fetchNewsItems()
    }
}

private suspend fun fetchNewsItems() {
    val newsList = newsApi.fetchNewsItems()
    _uiState.update { it.copy(articles = newsList.articles) }
}
}
```

Алиса Про. Ньюансы

- Вызов событий ViewModel в init
- Никакого DI, но есть @Inject
- Опять Moshi

```
private val retrofit = Retrofit.Builder()  
    .baseUrl(BASE_URL)  
    .addConverterFactory(MoshiConverterFactory.create())  
    .build()
```

Алиса Про. Итог

- Отдельные доделки дополнительными промптами
- Временами откат назад
- Отдельные сниппеты ок
- Больше детализации для точного результата

Gemini. Ньюансы

- Все супер, но...

```
class NewsViewModel(private val repository: NewsRepository) : ViewModel() {
    private val _newsState = mutableStateOf<NewsState>(NewsState.Loading)
    val newsState: State<NewsState> = _newsState.asState()

    init {
        getNews(query: "technology") // Initial query
    }

    fun getNews(query: String) {
        viewModelScope.launch {
            _newsState.value = NewsState.Loading
            try {
                val response = repository.getNews(query)
                _newsState.value = NewsState.Success(response.articles)
            } catch (e: Exception) {
                _newsState.value = NewsState.Error(e.message ?: "Unknown error")
            }
        }
    }
}
```

Gemini. Нюансы

- Все супер, но опять init...
- Опять mutableStateOf

```
class NewsViewModel(private val repository: NewsRepository) : ViewModel() {  
    private val _newsState = mutableStateOf<NewsState>(NewsState.Loading)  
    val newsState: State<NewsState> = _newsState.asState()  
  
    init {  
        getNews(query: "technology") // Initial query  
    }  
  
    fun getNews(query: String) {  
        viewModelScope.launch {  
            _newsState.value = NewsState.Loading  
            try {  
                val response = repository.getNews(query)  
                _newsState.value = NewsState.Success(response.articles)  
            } catch (e: Exception) {  
                _newsState.value = NewsState.Error(e.message ?: "Unknown error")  
            }  
        }  
    }  
}
```

Gemini. Итог

- Отдельные доделки дополнительными промптами
- На реализацию ушло несколько часов
- Детализация запроса для точного результата

Meta LLaMA. Нюансы

Смешение Dagger и Koin для DI

```
class MyApp : Application() {  
    override fun onCreate() {  
        super.onCreate()  
  
        val appContext = this  
        val koinModule = module {  
            single { NewsApiModule(appContext).provideNewsApi() }  
        }  
  
        startKoin(applicationContext, listOf(koinModule))  
    }  
}
```

@KoinModule

```
class KoinConfig {  
    @Provides  
    fun provideAppContext(): Application = application
```

@Provides

```
fun provideNewsViewModel(newsApi: NewsApi): NewsViewModel =  
    NewsViewModel(newsApi)
```

```
}
```

Meta LLaMA. Нюансы

Лишний код в Retrofit

Call vs suspend

news_service.kt

```
interface NewsService {  
    @GET("everything")  
    suspend fun getNews(): NewsResponse  
}
```

```
private const val BASE_URL = "https://newsapi.org/v2/"  
  
class NewsApi private constructor(private val retrofit: Retrofit) {  
    fun getNews(): Call<NewsResponse> {  
        return retrofit.create(NewsService::class.java).getNews()  
    }  
  
    companion object {  
        fun create(): NewsApi {  
            val retrofit = Retrofit.Builder()  
                .baseUrl(BASE_URL)  
                .addConverterFactory(GsonConverterFactory.create())  
                .build()  
  
            return NewsApi(retrofit)  
        }  
    }  
}
```


Meta LLaMA. Нюансы

mutableStateOf в ViewModel

news_view_model.kt

```
class NewsViewModel(private val newsApi: NewsApi) : ViewModel() {
    val news = mutableStateOf<List<Article>>(emptyList())

    fun getNews() {
        viewModelScope.launch {
            try {
                val response = newsApi.getNews().execute()
                if (response.isSuccessful) {
                    val articles = response.body()?.articles ?: emptyList()
                    news.value = articles
                } else {
                    Log.e("NewsViewModel", "Error: ${response.code()}")
                }
            } catch (e: Exception) {
                Log.e("NewsViewModel", "Error: $e")
            }
        }
    }
}
```

Meta LLaMa. Итог

- Во время уточняющих промптов были откаты
- Фазы обучения по несколько часов (не все одобренные сэмплы сводились)
- Отдельные сниппеты ок



GigaCode. Ожидаем класс

You

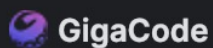
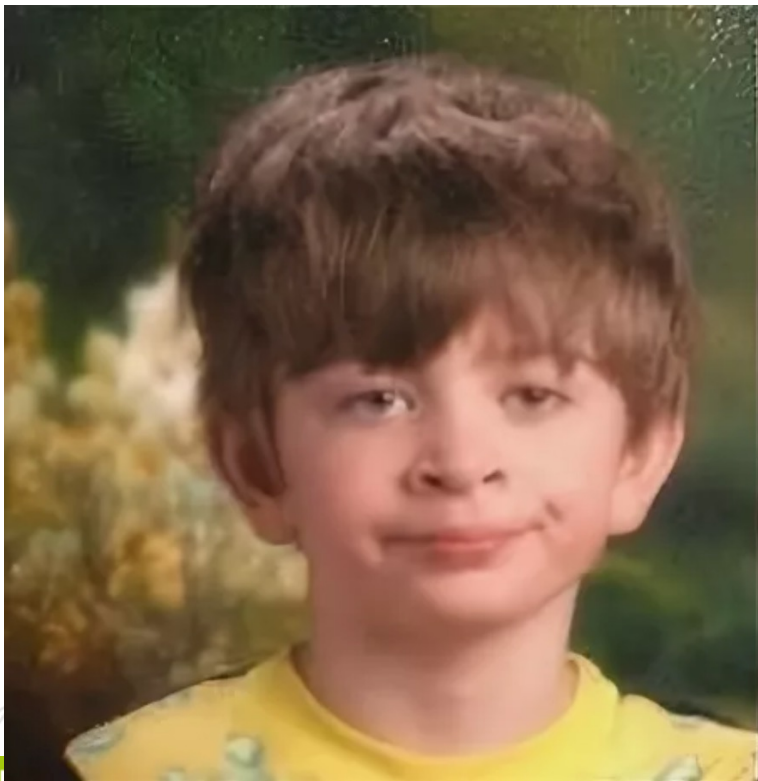


Напиши приложение Android для отображения списка новостей с newsapi.org по запросу everything. Для сетевого клиента используй Retrofit, для многопоточности coroutines. UI напиши на Jetpack Compose. Используй DI на Koin. Используй Material3 и Kotlin code-style



GigaCode. Делайте сами

Инструкция



Для создания приложения Android для отображения списка новостей с newsapi.org по запросу `everything`, вам нужно будет выполнить следующие шаги:

1. Создайте новый проект Android в Android Studio.
2. Добавьте зависимости Retrofit, Coroutines, Koin и Coil в файл `build.gradle`.
3. Создайте интерфейс `NewsAPIService` и классы данных `NewsResponse`, `Article` и `Source`.
4. Создайте экземпляр `NewsAPIService` с помощью Retrofit и корутин.
5. Создайте Composable для экрана списка новостей и Composable для элемента списка новостей.
6. Используйте Koin для внедрения зависимостей.
7. Используйте Coil для загрузки и отображения изображений.

GigaCode. Делайте сами

Пример с ретрофитом, а
дальше сами

Вот пример кода для создания экземпляра NewsAPIService:

```
val retrofit = Retrofit.Builder()
    .baseUrl("https://newsapi.org/v2/")
    .addConverterFactory(GsonConverterFactory.create())
    .build()

val newsAPIService: NewsAPIService = retrofit.create(NewsAPIService::class.java)

suspend fun getNews(query: String, from: String, sortBy: String, apiKey: String): NewsI
    return newsAPIService.getEverything(query, from, sortBy, apiKey).await()
}
```

YandexGPT

Напиши приложение Android для отображения списка новостей с newsapi.org по запросу everything. Для сетевого клиента используй Retrofit, для многопоточности coroutines. UI напиши на Jetpack Compose. Используй DI на Koin. Используй Material3 и Kotlin code-style

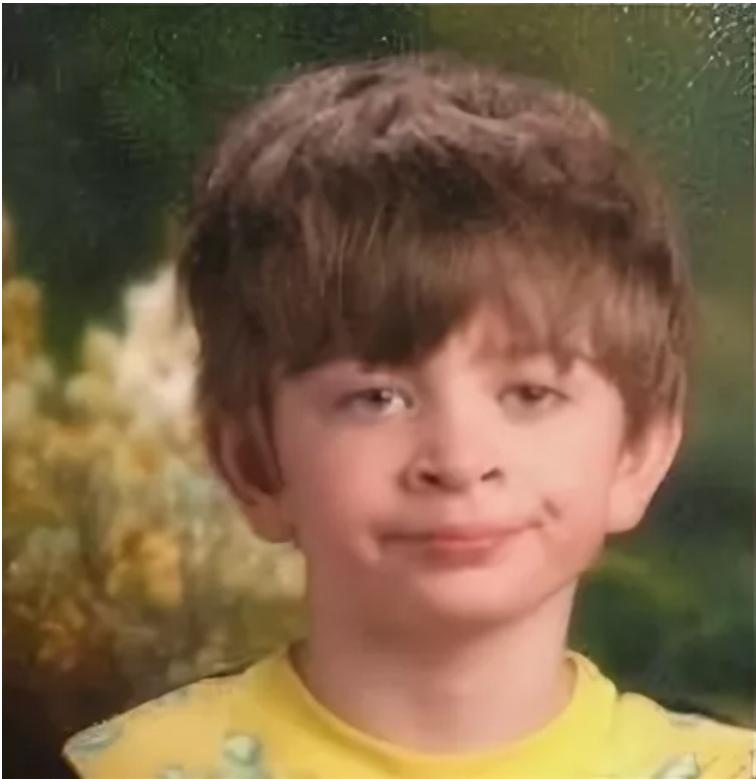
Для создания приложения Android для отображения списка новостей с NewsAPI.org по запросу «everything» вам потребуется выполнить следующие шаги:

1. **Создайте новый проект:** создайте новый проект на Kotlin и добавьте необходимые зависимости для работы с API, а также зависимости для Retrofit, Coroutines, Jetpack Compose, Koin, Material 3 и других библиотек.
2. **Получите данные:** получите данные с помощью NewsAPI.org и преобразуйте их в список объектов Article. Для этого вы можете использовать Retrofit и Coroutines.
3. **Определите компоуемый экран списка:** определите компоуемый экран ListScreen, который будет отображать список новостей.
4. **Импортируйте необходимые библиотеки:** импортируйте библиотеки для работы с Jetpack Compose, Material 3, Koin и другими зависимостями.
5. **Создайте компоуемый элемент списка:** создайте компоуемый элемент ListItem, который будет представлять одну новость. В этом элементе вы можете использовать Coil для загрузки изображения и отображения его рядом с текстом.
6. **Отобразите список:** в компоуемом экране ListScreen отобразите список новостей, используя компоуемые элементы ListItem.
7. **Добавьте логику:** добавьте логику для обработки событий (например, нажатие на элемент списка) и обновления данных.

Опять инструкция....

YandexGPT

Обрыв решения



```
import androidx.compose.foundation.Image
import androidx.compose.material3.*
import coil.compose.*
import kotlinx.coroutines.flow.collectAsState

@Composable
fun ListScreen(news: List<Article>) {
    LazyColumn {
        items(news) { article ->
            ListItem(article)
        }
    }
}

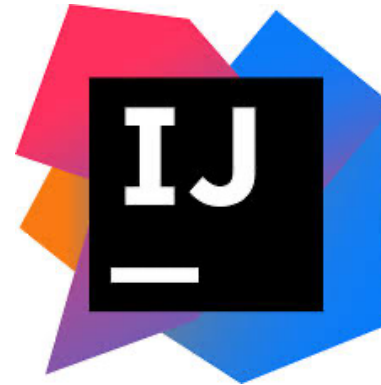
@Composable
private fun ListItem(article: Article) {
    Surface(
        modifier = Modifier
            .fillMaxWidth()
            .padding(16.dp),
        shape = RoundedCornerShape(topStart = 16.dp, topEnd = 16.dp)
    ) {
        Row(verticalAlignment = Alignment.CenterVertically) {
            Image(
                painter = rememberAsyncImagePainter(
```



Скопировать ответ

Чем пользоваться безопасно и без ограничений

- Своя LLM + свой плагин



Итог

- LLM умнеют и развиваются
- Есть готовые решения
- Можно развернуть свою LLM и научить ее
- Можно сделать свой плагин для доступа к своей LLM

Sources

- <https://developer.android.com/gemini-in-android>
- <https://gigacode.ru/>
- <https://ya.ru/ai/gpt-3>
- <https://habr.com/ru/articles/775842/>
- <https://openai.com/chatgpt/>
- <https://huggingface.co/>
- <https://www.promptingguide.ai/models/collection>



Спасибо за внимание!



@anioutkajarkova



azharkova
prettygeeknotes