

Кит на кот

выкидываем любимого ассистента, заменяем
на Коог и экономим токены, нервы и время



Николай Поташников

Архитектор решений, КУРС-ИТ

✉ consulting@yandex.ru 📍 @nmpotashnikoff

Joker<?>

“

Я в тюрьме. Одна стена — авангард, другая —
прошрое. Я хочу убежать

— Лигети Дьёрдь

О чем доклад?

О чем доклад?

Универсальные агентские системы

- Claude Code
 - Codex
 - OpenCode
 - ...
-

О чем доклад?

Универсальные агентские системы

- Claude Code
- Codex
- OpenCode
- ...

Средства сборки специализированных агентов

- Прямое использование API
- Spring AI, Koog
- LangChain / LangChain.js
- ...

О чем доклад?

Универсальные агентские системы

- Claude Code
- Codex
- OpenCode
- ...

Универсальная
агентская
система на
высокоуровневом
фреймворке Koog

Средства сборки специализированных агентов

- Прямое использование API
- Spring AI, Koog
- LangChain / LangChain.js
- ...

О чем доклад?

Универсальные агентские системы

- Claude Code
- Codex
- OpenCode
- ...

Универсальная агентская система на высокоуровневом фреймворке Koog

Средства сборки специализированных агентов

- Прямое использование API
- Spring AI, Koog
- LangChain / LangChain.js
- ...



чуть меньше универсальности → значительная экономия и точность

Вам будет интересен доклад

Вам будет интересен доклад

- Вы используете Koog или оцениваете возможность его использования для своих задач

Вам будет интересен доклад

- Вы используете Коог или оцениваете возможность его использования для своих задач
- Вы пишете агенты и собираете рабочие паттерны для своих задач

Вам будет интересен доклад

- Вы используете Koog или оцениваете возможность его использования для своих задач
- Вы пишете агенты и собираете рабочие паттерны для своих задач
- Вы постоянно ищете новые приёмы для эффективной работы с ИИ-агентами

План

План

- Как это работает

План

- Как это работает — точно не революция

План

- Как это работает — точно не революция
- Логика агентского workflow

План

- Как это работает — точно не революция
- Логика агентского workflow — проигранная битва за простоту

План

- Как это работает — точно не революция
- Логика агентского workflow — проигранная битва за простоту
- Коог в бою

План

- Как это работает — точно не революция
- Логика агентского workflow — проигранная битва за простоту
- Коог в бою — удобно, но нюансы есть

Мы здесь

- Как это работает
- Логика агентского workflow
- Коог в бою



Весь диалог содержится в файле Markdown, начинается с запроса пользователя

Структура проекта

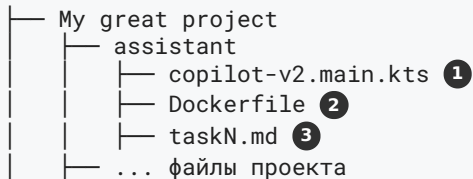
```
├── My great project
│   ├── assistant
│   │   ├── copilot-v2.main.kts ❶
│   │   ├── Dockerfile ❷
│   │   └── taskN.md ❸
│   └── ... файлы проекта
```

Структура проекта

```
├── My great project
│   ├── assistant
│   │   ├── copilot-v2.main.kts ❶
│   │   ├── Dockerfile ❷
│   │   └── taskN.md ❸
│   └── ... файлы проекта
```

- ❶ Весь копилот
- ❷ Спецификация песочницы
- ❸ Диалоги (задачи)

Структура проекта

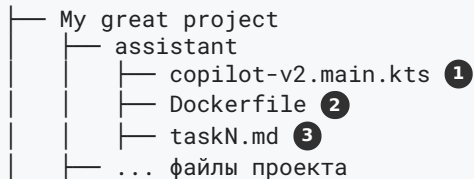


- 1 Весь копилот
- 2 Спецификация песочницы
- 3 Диалоги (задачи)



В IntelliJ IDEA я использую External Tools и горячую клавишу

Структура проекта



- 1 Весь копилот
- 2 Спецификация песочницы
- 3 Диалоги (задачи)



В IntelliJ IDEA я использую External Tools и горячую клавишу

```
kotlin -Xplugin=[kotlin]/lib/kotlinx-serialization-compiler-plugin.jar \  
copilot-v2.main.kts [путь к проекту]/assistant/test.md
```

Project

copilot-v2.main.kts

Dockerfile

epigraph.current.md

epigraph.md

openai.current.md

openai.md

openapi.vaml

Structure

As

Анализ цветов на странице jokerconf.com

Основные (брендовые) цвета

Тёмные / фоновые цвета

Светлые / фоновые цвета

Пастельные цвета

Синие / навигационные цвета

2026-joker-koog.adoc

colors.md

re-id.md

assessment-1.md

Какие цвета используются на этой странице

<https://jokerconf.com/>

As

Отлично! Я успешно получил и проанализировал страницу **jokerconf.com** с помощью Python-библиотек (`requests` + `BeautifulSoup`). Вот подробный отчёт:

🌈 Анализ цветов на странице jokerconf.com

Всего найдено **278** цветовых значений, из них **142** уникальных цвета.

🌈 Основные (брендовые) цвета

Цвет	Код	Описание	Частота
🌈	<code>#f3353f</code>	Ярко-красный (основной акцентный цвет Joker)	3 раза

2026-joker-koog > assistant > colors.md

3:1 LF UTF-8 4 spaces NORMAL

Project ▾ 2026-joker-koog.adoc M+ colors.md × M+ re-id.md M+ assessment-1.md

copilot-v2.main.kts Dockerfile M+ epigraph.current.md M+ epigraph.md M+ openai.current.md M+ openai.md M+ openapi.vaml

Structure

▾ As

▾ Анализ цветов на странице

- Основные (брендовые) цвета
 - Тёмные / фоновые цвета
 - Светлые / фоновые цвета
 - ● ● ● Пастельные цвета
 - Синие / навигационные цвета

```

1  Какие цвета используются на этой странице
2  https://jokerconf.com/
3
4  # As
5
6  Отлично! Я успешно получил и проанализировал страницу **jokerconf.com** с помощью
   Python-библиотек ('requests' + 'BeautifulSoup'). Вот подробный отчёт:
7
8  ## 🧠 Анализ цветов на странице jokerconf.com
9
10 Всего найдено **278** цветовых значений, из них **142** уникальных цвета.
11
12 ### 🎨 Основные (брендовые) цвета
13
14 | Цвет | Код | Описание | Частота |
15 |-----|-----|-----|-----|
16 | 🎨 | '#f3353f' | **Ярко-красный** (основной акцентный цвет Joker) | 3 раза |

```

2026-joker-koog > assistant > M+ colors.md 3:1 LF UTF-8 4 spaces NORMAL

Ответ ассистента под заголовком (# As)

Кит на кот | Joker<?>

The screenshot shows a VS Code editor interface. The top bar displays the project name '2026-joker-koog' and several open files: 'colors.md', 're-id.md', and 'assessment-1.md'. The 'colors.md' file is active, showing the following content:

```
1 Какие цвета используются на этой странице
2 https://jokerconf.com/
3
4 # As
5
6 Отлично! Я успешно получил и проанализировал страницу **jokerconf.com** с помощью Python-библ
```

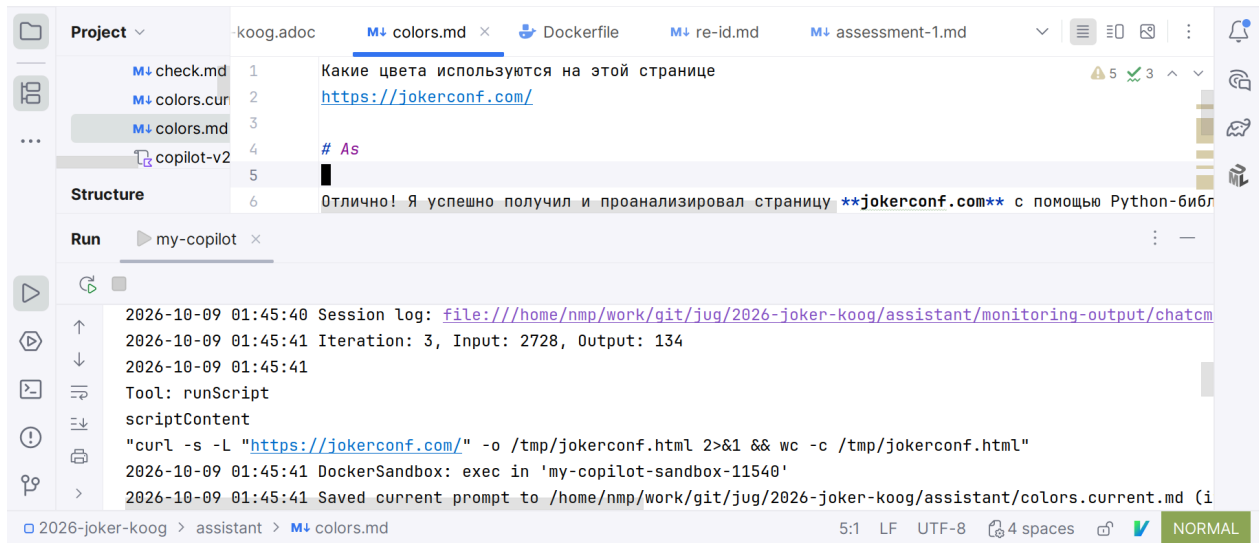
Below the editor is a 'Run' terminal window titled 'my-copilot'. It displays a session log with the following entries:

```
2026-10-09 01:45:40 Session log: file:///home/nmp/work/git/jug/2026-joker-koog/assistant/monitoring-output/chatcm
2026-10-09 01:45:41 Iteration: 3, Input: 2728, Output: 134
2026-10-09 01:45:41
Tool: runScript
scriptContent
"curl -s -L "https://jokerconf.com/" -o /tmp/jokerconf.html 2>&1 && wc -c /tmp/jokerconf.html"
2026-10-09 01:45:41 DockerSandbox: exec in 'my-copilot-sandbox-11540'
2026-10-09 01:45:41 Saved current prompt to /home/nmp/work/git/jug/2026-joker-koog/assistant/colors.current.md (i
```

The bottom status bar shows the file path '2026-joker-koog > assistant > colors.md', encoding '5:1 LF UTF-8', and a 'NORMAL' status indicator.

Лог содержит:

Кит на кот | Joker<?>



Project ▾ koog.adoc colors.md x Dockerfile re-id.md assessment-1.md

check.md 1 Какие цвета используются на этой странице 5 3 ^ ▾

colors.current.md 2 <https://jokerconf.com/>

colors.md 3

copilot-v2 4 # As

Structure 5

6 Отлично! Я успешно получил и проанализировал страницу **jokerconf.com** с помощью Python-библ

Run ▶ my-copilot x

2026-10-09 01:45:40 Session log: <file:///home/nmp/work/git/jug/2026-joker-koog/assistant/monitoring-output/chatcm>

2026-10-09 01:45:41 Iteration: 3, Input: 2728, Output: 134

2026-10-09 01:45:41

Tool: runScript

scriptContent

"curl -s -L "<https://jokerconf.com/>" -o /tmp/jokerconf.html 2>&1 && wc -c /tmp/jokerconf.html"

2026-10-09 01:45:41 DockerSandbox: exec in 'my-copilot-sandbox-11540'

2026-10-09 01:45:41 Saved current prompt to /home/nmp/work/git/jug/2026-joker-koog/assistant/colors.current.md (i

2026-joker-koog > assistant > colors.md 5:1 LF UTF-8 4 spaces NORMAL

Лог содержит: (1) ссылку на файл с текущим потоком ответа LLM

Project: koog.adoc | colors.md | Dockerfile | re-id.md | assessment-1.md

check.md | colors.current.md | colors.md | copilot-v2

Structure

Run: my-copilot

2026-10-09 01:45:40 Session log: <file:///home/nmp/work/git/jug/2026-joker-koog/assistant/monitoring-output/chatcm>

2026-10-09 01:45:41 Iteration: 3, Input: 2728, Output: 134

2026-10-09 01:45:41

Tool: runScript

scriptContent

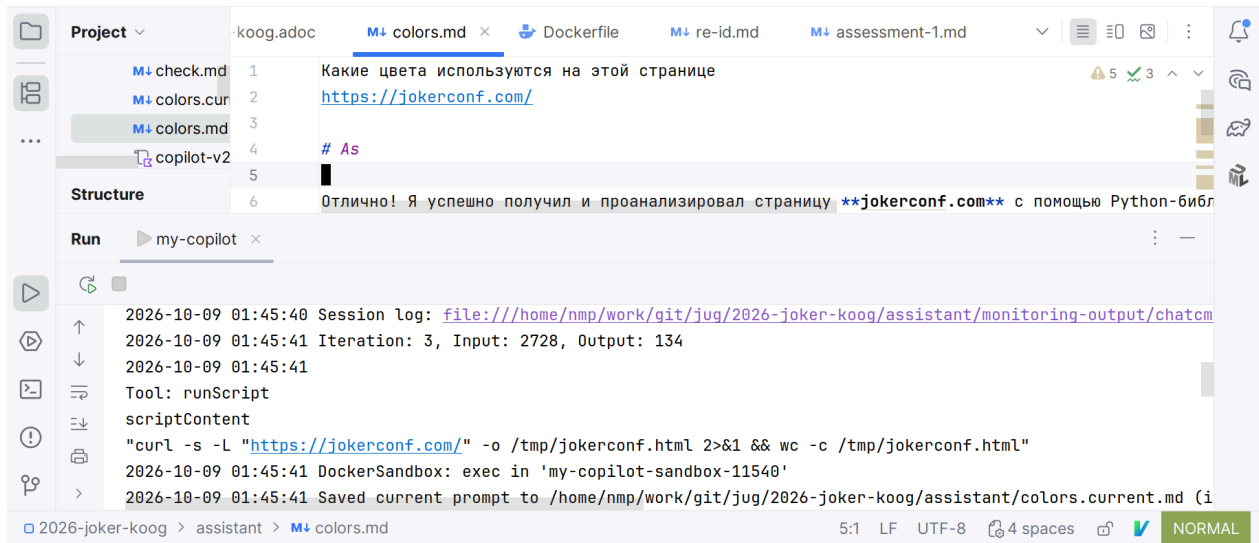
"curl -s -L "https://jokerconf.com/" -o /tmp/jokerconf.html 2>&1 && wc -c /tmp/jokerconf.html"

2026-10-09 01:45:41 DockerSandbox: exec in 'my-copilot-sandbox-11540'

2026-10-09 01:45:41 Saved current prompt to /home/nmp/work/git/jug/2026-joker-koog/assistant/colors.current.md (i

2026-joker-koog > assistant > colors.md | 5:1 LF UTF-8 4 spaces NORMAL

Лог содержит: (1) ссылку на файл с текущим потоком ответа LLM – (2) вызванные инструменты



Project: koog.adoc | colors.md | Dockerfile | re-id.md | assessment-1.md

check.md 1: Какие цвета используются на этой странице

colors.current.md 2: <https://jokerconf.com/>

colors.md 3:

copilot-v2 4: # As

Structure 5:

6: Отлично! Я успешно получил и проанализировал страницу ****jokerconf.com**** с помощью Python-библ

Run: my-copilot

2026-10-09 01:45:40 Session log: <file:///home/nmp/work/git/jug/2026-joker-koog/assistant/monitoring-output/chatcm>

2026-10-09 01:45:41 Iteration: 3, Input: 2728, Output: 134

2026-10-09 01:45:41

Tool: runScript

scriptContent

"curl -s -L "<https://jokerconf.com/>" -o /tmp/jokerconf.html 2>&1 && wc -c /tmp/jokerconf.html"

2026-10-09 01:45:41 DockerSandbox: exec in 'my-copilot-sandbox-11540'

2026-10-09 01:45:41 Saved current prompt to /home/nmp/work/git/jug/2026-joker-koog/assistant/colors.current.md (i

2026-joker-koog > assistant > colors.md | 5:1 LF UTF-8 4 spaces NORMAL

Лог содержит: (1) ссылку на файл с текущим потоком ответа LLM – (2) вызванные инструменты – (3) съединенные токены

Project: koog.adoc | colors.md | Dockerfile | re-id.md | assessment-1.md

check.md | colors.current.md | colors.md | copilot-v2

Structure

Run: my-copilot

2026-10-09 01:45:40 Session log: <file:///home/nmp/work/git/jug/2026-joker-koog/assistant/monitoring-output/chatcm>

2026-10-09 01:45:41 Iteration: 3, Input: 2728, Output: 134

2026-10-09 01:45:41

Tool: runScript

scriptContent

"curl -s -L "https://jokerconf.com/" -o /tmp/jokerconf.html 2>&1 && wc -c /tmp/jokerconf.html"

2026-10-09 01:45:41 DockerSandbox: exec in 'my-copilot-sandbox-11540'

2026-10-09 01:45:41 Saved current prompt to /home/nmp/work/git/jug/2026-joker-koog/assistant/colors.current.md (i

2026-joker-koog > assistant > colors.md | 5:1 LF UTF-8 | 4 spaces | NORMAL

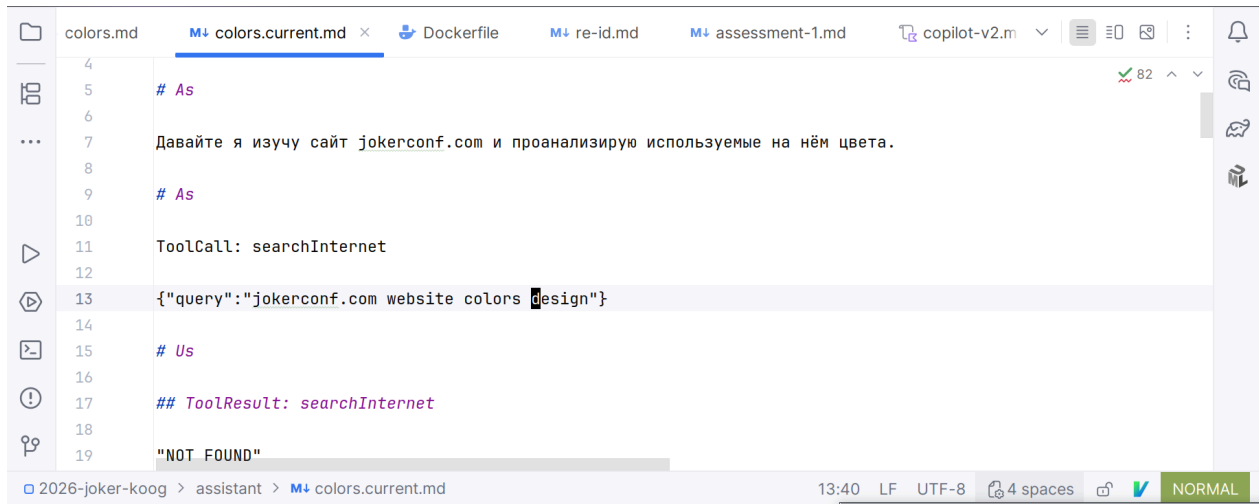
Лог содержит: (1) ссылку на файл с текущим потоком ответа LLM – (2) вызванные инструменты – (3) съединенные токены – (4) ссылку на технический диалог

```
Project ▾ -koog.adoc colors.md chatcmpl-9007f485-6233-49de-8adb-9b4222a88de4.md x
1 # Reasoning Content
2
3 Пользователь спрашивает о цветах, используемых на сайте jokerconf.com. Мне нужно получить
  доступ к HTML/CSS коду сайта, чтобы определить используемые цвета. Давайте попробуем
  загрузить страницу с помощью скрипта.
4
5 # Text Content
6
7 Давайте загрузим страницу и проанализируем её CSS-стили для определения цветов.
8
9 # Tool Calling Log
10
11 Tools were called

Structure
Reasoning Content
Text Content

log > assistant > monitoring-output > colors.md 2:1 LF UTF-8 4 spaces NORMAL
```

Поток ответа LLM внутри итерации: (1) проверка, что модель не зациклилась, (2) выявление ошибок и неточностей в исходном запросе, настройке агента, конфигурации песочницы



```
4
5 # As
6
7 Давайте я изучу сайт jokerconf.com и проанализирую используемые на нём цвета.
8
9 # As
10
11 ToolCall: searchInternet
12
13 {"query": "jokerconf.com website colors design"}
14
15 # Us
16
17 ## ToolResult: searchInternet
18
19 "NOT FOUND"
```

2026-joker-koog > assistant > colors.current.md 13:40 LF UTF-8 4 spaces NORMAL

Технический диалог: (1) возможность продолжить при сбое, (2) выявление ошибок и неточностей в исходном запросе, настройке агента, конфигурации песочницы

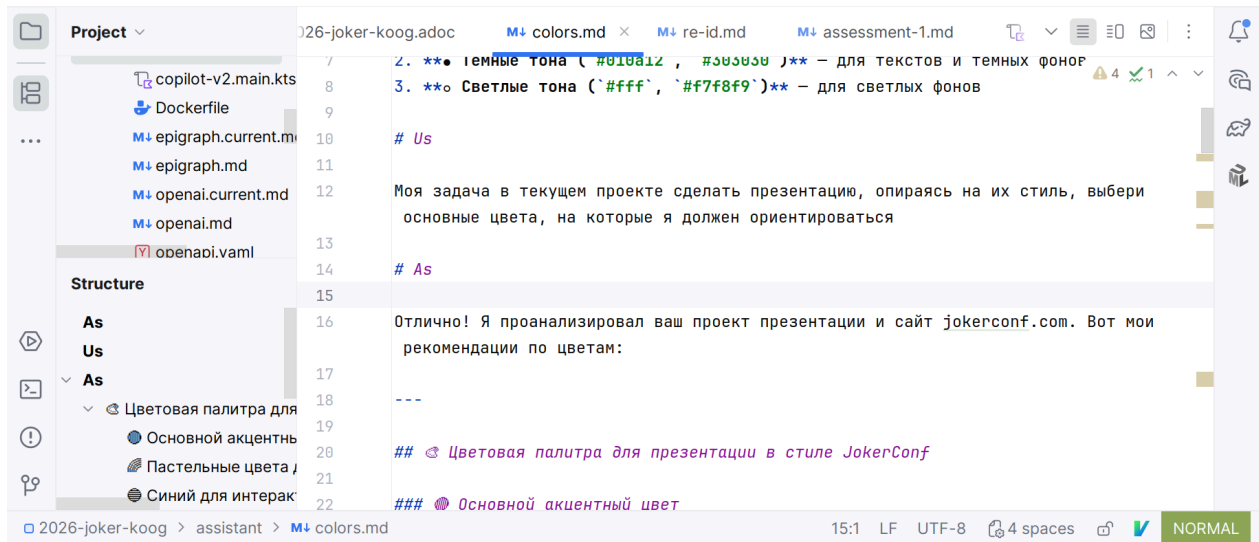
```
Project ▾ 2026-joker-koog.adoc colors.md re-id.md assessment-1.md
├── copilot-v2.main.kts
├── Dockerfile
├── epigraph.current.md
├── epigraph.md
├── openai.current.md
├── openai.md
└── openai.vaml

Structure
├── As
└── Us

1 Какие цвета используются на этой странице ✓
2 https://jokerconf.com/
3
4 # As
5
6 1. **● Красный (`#f3353f`)** — главный фирменный цвет, используется для акцентов и
7   кнопок
8 2. **● Тёмные тона (`#010a12`, `#303030`)** — для текстов и тёмных фонов
9 3. **○ Светлые тона (`#fff`, `#f7f8f9`)** — для светлых фонов
10
11 # Us
12
13 Моя задача в текущем проекте сделать презентацию, опираясь на их стиль, выбери
    основные цвета, на которые я должен ориентироваться
```

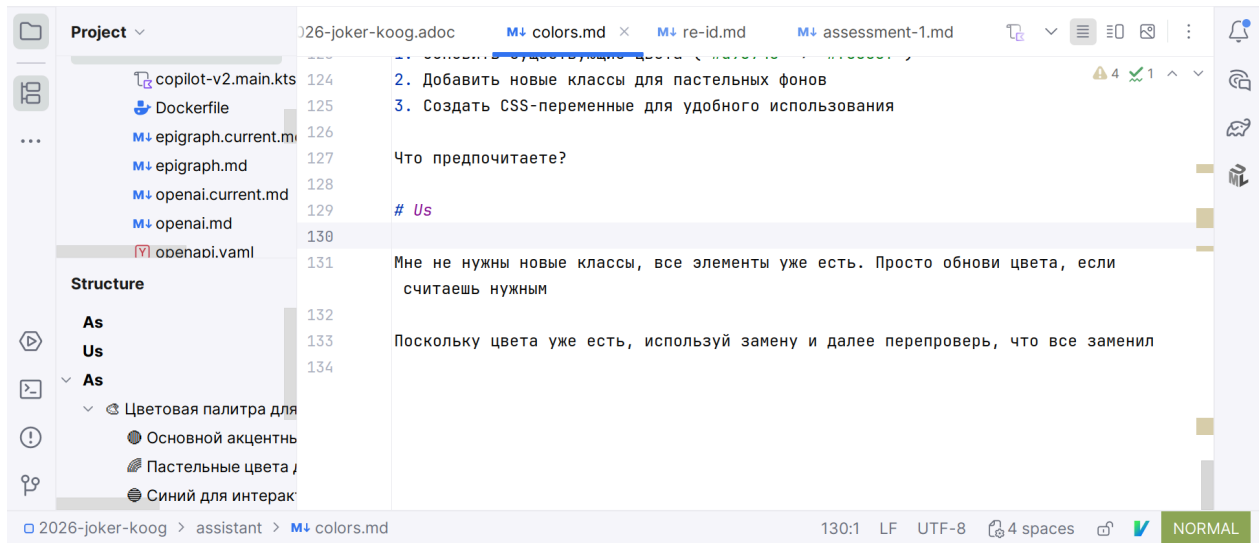
2026-joker-koog > assistant > colors.md 11:1 LF UTF-8 4 spaces NORMAL

- Ответ LLM можно упростить, это же текстовый файл
- Следующий вопрос — стандартно под заголовком # Us



Новый ответ – стандартно под заголовком # As

Кит на кот | Joker<?>



Ответ избыточный, но сокращать было лень



Задача решена

Кит на кот | Joker<?>

Это же просто линейный диалог?

Это же просто линейный диалог?



Не совсем диалог: историю можно и нужно менять

Это же просто линейный диалог?



Не совсем диалог: историю можно и нужно менять

- Экономия контекста позволяет выйти за пределы коротких задач

Это же просто линейный диалог?



Не совсем диалог: историю можно и нужно менять

- Экономия контекста позволяет выйти за пределы коротких задач
- Опора на линейный транскрипт делает работу агента наблюдаемой и контролируемой

Настройка агента – просто код внутри скрипта

```
fun currentModel() = CModels().qwen37Plus
fun maxSizeText() = 40000
fun currentTools() = setOf<KFunction<String>>(
    MainAgentTools::readFile,
    MainAgentTools::readDirectory,
    //    MainAgentTools::modifyFile,
    MainAgentTools::askForClarification,
    ...
)
```

Настройка агента – просто код внутри скрипта

```
fun currentModel() = CModels().qwen37Plus
fun maxSizeText() = 40000
fun currentTools() = setOf<KFunction<String>>(
    MainAgentTools::readFile,
    MainAgentTools::readDirectory,
    //      MainAgentTools::modifyFile,
    MainAgentTools::askForClarification,
    ...
)
```

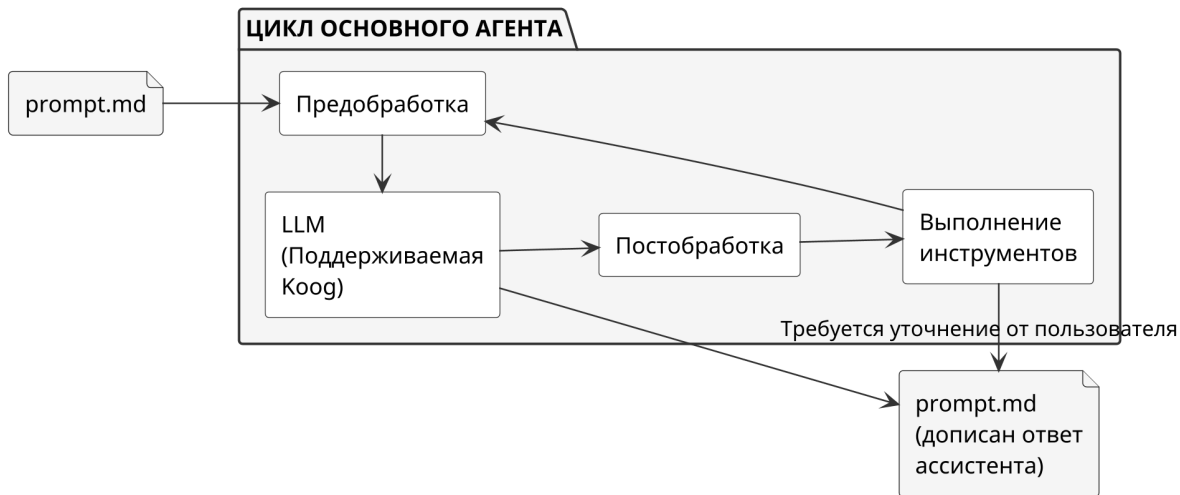


Копилот – скрипт внутри проекта, поэтому может быть легко модифицирован

Мы здесь

- Как это работает
- Логика агентского workflow
- Коог в бою

Логика агентского WorkFlow



Инструменты и ключевые решения по ним: readFile, readDir

Инструменты и ключевые решения по ним: readFile, readDir

- readFile — чтение файла
 - Модель определяет считываемый размер

Инструменты и ключевые решения по ним: readFile, readDir

- readFile – чтение файла
 - Модель определяет считываемый размер
- readDir – чтение директории
 - Формат ASCII-дерево. JSON тратил слишком много токенов

```
├── dir1
│   ├── file1 (23 Kb)
│   └── file2 (1 Kb)
```

Инструменты и ключевые решения по ним: modifyFile

Инструменты и ключевые решения по ним: modifyFile

- Два режима: (1) изменения применяются напрямую или (2) сохраняются в файле [имя исходного файла].proposal

Инструменты и ключевые решения по ним: modifyFile

- Два режима: (1) изменения применяются напрямую или (2) сохраняются в файле [**имя исходного файла**].proposal
- Изменения — это DIFF (чтобы экономить токены и не ломать файл)

Инструменты и ключевые решения по ним: modifyFile

- Два режима: (1) изменения применяются напрямую или (2) сохраняются в файле [**имя исходного файла**].proposal
- Изменения — это DIFF (чтобы экономить токены и не ломать файл)
- DIFF создается в формате JSON по схеме

Инструменты и ключевые решения по ним: modifyFile

- Два режима: (1) изменения применяются напрямую или (2) сохраняются в файле [**имя исходного файла**].proposal
- Изменения — это DIFF (чтобы экономить токены и не ломать файл)
- DIFF создается в формате JSON по схеме
- Но без Structured Output

Инструменты и ключевые решения по ним: modifyFile

- Два режима: (1) изменения применяются напрямую или (2) сохраняются в файле [**имя исходного файла**].proposal
- Изменения — это DIFF (чтобы экономить токены и не ломать файл)
- DIFF создается в формате JSON по схеме
- Но без Structured Output
- Если JSON битый пытаемся восстановить без LLM

Инструменты и ключевые решения по ним: modifyFile

- Два режима: (1) изменения применяются напрямую или (2) сохраняются в файле [имя исходного файла].proposal
- Изменения — это DIFF (чтобы экономить токены и не ломать файл)
- DIFF создается в формате JSON по схеме
- Но без Structured Output
- Если JSON битый пытаемся восстановить без LLM
- Structured Output — если не получилось десериализовать

Пример сгенерированного proposal-файла

The screenshot shows an IDE with a project explorer on the left and a diff view in the center. The project explorer lists files: build.gradle.kts, build.gradle.kts.proposal, copilot-2.main.kts, gradlew, kotlin-meetup-2026-06-10.a, settings.gradle.kts, and white-course.css. The diff view compares build.gradle.kts (left) and build.gradle.kts.proposal (right). The diff highlights changes to the version, presentationFile, and buildscript repositories. The terminal at the bottom shows a command to find proposal files and its output.

build.gradle.kts	Line	build.gradle.kts.proposal	Line
version = "0.0.1"	65	version = "0.0.1"	65
val presentationFile	66	val presentationFile	66
= "kotlin-meetup-2026-06-10"	67	= project.name	67
buildscript {	68	buildscript {	68
repositories {	69	repositories {	69
	70	mavenLocal()	70

```
nmp@nmp-vb:~/work/git/al/my-assistant$ find "$(pwd)" -type f -name "*.proposal" -exec echo "file://{  
}" \;  
file:///home/nmp/work/git/al/my-assistant/build.gradle.kts.proposal  
nmp@nmp-vb:~/work/git/al/my-assistant$
```

Инструменты и ключевые решения по ним: search Internet,
askForClarification, analyzelImage

Инструменты и ключевые решения по ним: search Internet, askForClarification, analyzeImage

- searchInternet — поиск в интернете
 - Только через встроенные функции модели. Это самый быстрый способ

Инструменты и ключевые решения по ним: search Internet, askForClarification, analyzelImage

- searchInternet — поиск в интернете
 - Только через встроенные функции модели. Это самый быстрый способ
- askForClarification — самый простой способ объяснить модели требовать уточнений

Инструменты и ключевые решения по ним: search Internet, askForClarification, analyzelImage

- searchInternet — поиск в интернете
 - Только через встроенные функции модели. Это самый быстрый способ
- askForClarification — самый простой способ объяснить модели требовать уточнений
- analyzelImage — анализ изображения

Инструменты и ключевые решения по ним: runScript

Инструменты и ключевые решения по ним: runScript

- Выполняется только в докер-контейнере (лежит рядом со скриптом копилота)

Инструменты и ключевые решения по ним: runScript

- Выполняется только в докер-контейнере (лежит рядом со скриптом копилота)
- Содержимое докер-контейнера и адреса точек монтирования автоматически передаются в системный промпт

Инструменты и ключевые решения по ним: runScript

- Выполняется только в докер-контейнере (лежит рядом со скриптом копилота)
- Содержимое докер-контейнера и адреса точек монтирования автоматически передаются в системный промпт
- Как транслировать пути — пришлось делать отдельный кусок промпта. Хотя казалось бы...

Инструменты и ключевые решения по ним: runScript

- Выполняется только в докер-контейнере (лежит рядом со скриптом копилота)
- Содержимое докер-контейнера и адреса точек монтирования автоматически передаются в системный промпт
- Как транслировать пути — пришлось делать отдельный кусок промпта. Хотя казалось бы...
- Безопасность запуска скриптов определяется безопасностью контейнера

Инструменты и ключевые решения по ним: runScript

- Выполняется только в докер-контейнере (лежит рядом со скриптом копилота)
- Содержимое докер-контейнера и адреса точек монтирования автоматически передаются в системный промпт
- Как транслировать пути – пришлось делать отдельный кусок промпта. Хотя казалось бы...
- Безопасность запуска скриптов определяется безопасностью контейнера



Отдельный инструмент `runPredifinedBash`, обычно используется для сборки текущего проекта, может быть параметризован

Хуки: Пред- постобработка

Хуки: Пред- постобработка

- Redaction (убирание личных данных и секретов): регулярные выражения и `angrscan`

Хуки: Пред- постобработка

- Redaction (убирание личных данных и секретов): регулярные выражения и `angrscan`
- Выдаем ошибку, если агент пытается модифицировать файл не вычитав его или если он модифицировал файл, но не прочитал, что получилось

Хуки: Пред- постобработка

- Redaction (убирание личных данных и секретов): регулярные выражения и `angriscan`
- Выдаем ошибку, если агент пытается модифицировать файл не вычитав его или если он модифицировал файл, но не прочитал, что получилось
- Если файл перечитан заново, старую версию заменяем на сообщение о том, что файл считан позже

Хуки: Пред- постобработка

- Redaction (убирание личных данных и секретов): регулярные выражения и `angriscan`
- Выдаем ошибку, если агент пытается модифицировать файл не вычитав его или если он модифицировал файл, но не прочитал, что получилось
- Если файл перечитан заново, старую версию заменяем на сообщение о том, что файл считан позже



Контекстно-специфичный history compaction всегда лучше, чем универсальный

Мы здесь

- Как это работает
- Логика агентского workflow
- Коог в бою

Без Коог это было бы не возможно

Без Коог это было бы не возможно

- Создание пользовательских инструментов

Без Коог это было бы не возможно

- Создание пользовательских инструментов
- Идиоматичная поддержка Kotlin и Java

Без Коог это было бы не возможно

- Создание пользовательских инструментов
- Идиоматичная поддержка Kotlin и Java
- Комплексная трассировка

Без Коог это было бы не возможно

- Создание пользовательских инструментов
- Идиоматичная поддержка Kotlin и Java
- Комплексная трассировка
- Гибкие графовые рабочие процессы

Без Коог это было бы не возможно

- Создание пользовательских инструментов
- Идиоматичная поддержка Kotlin и Java
- Комплексная трассировка
- Гибкие графовые рабочие процессы



Успели заснуть?

Без Коог это было бы не возможно

- Создание пользовательских инструментов
- Идиоматичная поддержка Kotlin и Java
- Комплексная трассировка
- Гибкие графовые рабочие процессы



Успели заснуть?



Список можно дочитать на сайте Коог. Нюансы важнее

OpenAI – отраслевой стандарт: Серия 1

```
delta:
  content: ''
  reasoning_content: ''
  tool_calls:
    - id: call_5e39bc8f133a4500b0f330b8
      type: function
      index: 0
      function:
        name: readFile
        arguments: ''
```



Есть ли тут что-то подозрительное?

Есть

Есть

```
1. content = ''
```

Есть

1. `content = ''` — разрешено спецификацией

Есть

1. `content = ''` — разрешено спецификацией — Koog падает, нужен null

Есть

1. `content = ''` — разрешено спецификацией — Koog падает, нужен null — content тут вообще зачем?

Есть

1. `content = ''` – разрешено спецификацией – Koog падает, нужен null – content тут вообще зачем?
2. есть `reasoning_content`

Есть

1. `content = ''` — разрешено спецификацией — Koog падает, нужен `null` — `content` тут вообще зачем?
2. есть `reasoning_content` — в спецификации отсутствует

Есть

1. `content = ''` — разрешено спецификацией — Koog падает, нужен `null` — `content` тут вообще зачем?
2. есть `reasoning_content` — в спецификации отсутствует — Koog не падает

Есть

1. `content = ''` — разрешено спецификацией — Koog падает, нужен `null` — `content` тут вообще зачем?
2. есть `reasoning_content` — в спецификации отсутствует — Koog не падает — просто игнорирует

Есть

1. `content = ''` — разрешено спецификацией — Koog падает, нужен `null` — `content` тут вообще зачем?
2. есть `reasoning_content` — в спецификации отсутствует — Koog не падает — просто игнорирует — ну т.е. вообще игнорирует

Пример исправления запроса на уровне интерсептора

```
val loggingSession = object: SSESession by session {  
    override val incoming: Flow<ServerSentEvent> =  
        session.incoming.map { event ->  
            event.copy(data = event.data?.replace("\"content\":\"", "\"content\":null  
        }).onEach { monitoring.processEvent(it.data) }  
        .onCompletion { monitoring.flushAll() }  
}
```

OpenAI – отраслевой стандарт: Серия 2

OpenAI – отраслевой стандарт: Серия 2

- `DashscopeChatCompletionRequest` содержит `enableSearch`. Включает поиск в Интернет?

OpenAI – отраслевой стандарт: Серия 2

- `DashscopeChatCompletionRequest` содержит `enableSearch`. Включает поиск в Интернет?
- Нет, требуется `"reasoning" : true`, иначе упадет... с понятным сообщением

OpenAI – отраслевой стандарт: Серия 2

- `DashscopeChatCompletionRequest` содержит `enableSearch`. Включает поиск в Интернет?
- Нет, требуется `"reasoning" : true`, иначе упадет... с понятным сообщением
- А еще `"search_options" : {"search_strategy" : "agent"}`, иначе сработает, но поиск не выполнит

OpenAI – отраслевой стандарт: Серия 2

- `DashscopeChatCompletionRequest` содержит `enableSearch`. Включает поиск в Интернет?
- Нет, требуется `"reasoning" : true`, иначе упадет... с понятным сообщением
- А еще `"search_options" : {"search_strategy" : "agent"}`, иначе сработает, но поиск не выполнит
- Или `max` если нужно дать агенту возможность выкачивать страницы

OpenAI – отраслевой стандарт: Серия 2

- `DashscopeChatCompletionRequest` содержит `enableSearch`. Включает поиск в Интернет?
- Нет, требуется `"reasoning" : true`, иначе упадет... с понятным сообщением
- А еще `"search_options" : {"search_strategy" : "agent"}`, иначе сработает, но поиск не выполнит
- Или `max` если нужно дать агенту возможность выкачивать страницы
- Впрочем, это в соответствии с документацией Alibaba Cloud, у меня `max` не работает

Так это просто Alibaba Cloud плохой?

Так это просто Alibaba Cloud плохой?

- OpenRouterChatCompletionRequest вообще не содержит опции `enableSearch`

Так это просто Alibaba Cloud плохой?

- OpenRouterChatCompletionRequest вообще не содержит опции `enableSearch`
- Но Koog поддерживает `additionalProperties`

Так это просто Alibaba Cloud плохой?

- OpenRouterChatCompletionRequest вообще не содержит опции `enableSearch`
- Но Koog поддерживает `additionalProperties`
- И, ура! Open Router поддерживает `{"tools" : {"type" : "openrouter:web_search" }}`

Так это просто Alibaba Cloud плохой?

- OpenRouterChatCompletionRequest вообще не содержит опции `enableSearch`
- Но Koog поддерживает `additionalProperties`
- И, ура! Open Router поддерживает `{"tools" : {"type" : "openrouter:web_search" }}`
- И даже `openrouter:web_fetch`

Так это просто Alibaba Cloud плохой?

- OpenRouterChatCompletionRequest вообще не содержит опции `enableSearch`
- Но Koog поддерживает `additionalProperties`
- И, ура! Open Router поддерживает `{"tools" : {"type" : "openrouter:web_search" }}`
- И даже `openrouter:web_fetch`
- Впрочем, если Вы используете свои инструменты, Koog серверные благополучно срежет

OpenAI – отраслевой стандарт: Серия 3

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет $2000000 + 500000$ рублей. Рассчитай общую стоимость в тысячах рублей

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет $2000000 + 500000$ рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет $2000000 + 500000$ рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?
- 2500

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет $2000000 + 500000$ рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?
- 2500
- Что ответить Qwen 3.x?

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет $2000000 + 500000$ рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?
- 2500
- Что ответить Qwen 3.x?
- Для начала ошибку: если мы требуем `json_object`, в запросе должно быть слово JSON

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет 2000000 + 500000 рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?
- 2500
- Что ответить Qwen 3.x?
- Для начала ошибку: если мы требуем `json_object`, в запросе должно быть слово JSON
- Если слово JSON есть, тогда 2500000

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет 2000000 + 500000 рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?
- 2500
- Что ответить Qwen 3.x?
- Для начала ошибку: если мы требуем `json_object`, в запросе должно быть слово JSON
- Если слово JSON есть, тогда 2500000
- QWEN настолько хуже GPT?

OpenAI – отраслевой стандарт: Серия 3

- Стоимость автомобиля составляет 2000000 + 500000 рублей. Рассчитай общую стоимость в тысячах рублей
- Что ответит GPT 5.x?
- 2500
- Что ответить Qwen 3.x?
- Для начала ошибку: если мы требуем `json_object`, в запросе должно быть слово JSON
- Если слово JSON есть, тогда 2500000
- QWEN настолько хуже GPT?
- Обязка, предоставляемая провайдерами не передает описания в модель

Явная отправка схемы в модель

Явная отправка схемы в модель

```
@Serializable
data class MyDataClass (
    @property:LLMDescription("Цена в тысячах рублей")
    price : String
)
```

Явная отправка схемы в модель

```
@Serializable
data class MyDataClass (
    @property:LLMDescription("Цена в тысячах рублей")
    price : String
)
```

Для qwen 3.x не работает

```
requestLLMStructured<MyDataClass>("Some user request")
```

Явная отправка схемы в модель

```
@Serializable
data class MyDataClass (
    @property:LLMDescription("Цена в тысячах рублей")
    price : String
)
```

Для qwen 3.x не работает

```
requestLLMStructured<MyDataClass>("Some user request")
```

Сработает

```
requestLLMStructured<MyDataClass>(
    "Some user request\n\nOutput JSON must conform to ", generateSchema<MyDataClass>()
)
```

Можно вообще без аннотаций

```
MainAgentTools::readFile.defineDescription {  
  ""  
  Gets file contents by path with numbered lines  
  
  ### Params  
  
  - ${"path" from this} -- absolute path to the file  
  - ${"maxSizeText" from this} -- tool returns first maxSizeText characters  
    recommended value to start with is ${maxSizeText()}"".trimIndent()  
}
```

Можно вообще без аннотаций

```
MainAgentTools::readFile.defineDescription {  
    ""  
    Gets file contents by path with numbered lines  
  
    ### Params  
  
    - ${"path" from this} -- absolute path to the file  
    - ${"maxSizeText" from this} -- tool returns first maxSizeText characters  
      recommended value to start with is ${maxSizeText()}"".trimIndent()  
}
```



Позволяет делать динамические описания инструментов

Можно вообще без аннотаций

```
MainAgentTools::readFile.defineDescription {  
    ""  
    Gets file contents by path with numbered lines  
  
    ### Params  
  
    - ${"path" from this} -- absolute path to the file  
    - ${"maxSizeText" from this} -- tool returns first maxSizeText characters  
      recommended value to start with is ${maxSizeText()}"".trimIndent()  
}
```



Позволяет делать динамические описания инструментов



"path" from this — + инфиксная функция `String.from(func: KFunction<*>): String {...}`

Коог (любой агентский framework) – это зло?

Коог (любой агентский framework) – это зло?

1. Ни один framework не сможет угнаться за нюансами стандарта

Коог (любой агентский framework) – это зло?

1. Ни один framework не сможет угнаться за нюансами стандарта
2. Необходимо работать на уровне HTTP response/request

Коог (любой агентский framework) – это зло?

1. Ни один framework не сможет угнаться за нюансами стандарта
2. Необходимо работать на уровне HTTP response/request
3. Необходимо активно вмешиваться в механизмы формирования промпта

Субъективно, с чем мне было приятно работать в Коог

Субъективно, с чем мне было приятно работать в Коог

1. Аннотация инструментов и классов

Субъективно, с чем мне было приятно работать в Коог

1. Аннотация инструментов и классов
2. Запуск LLM в разных режимах одной строчкой

Субъективно, с чем мне было приятно работать в Koog

1. Аннотация инструментов и классов
2. Запуск LLM в разных режимах одной строчкой
3. Prompt Assembly/Rewriting DSL

Субъективно, с чем мне было приятно работать в Koog

1. Аннотация инструментов и классов
2. Запуск LLM в разных режимах одной строчкой
3. Prompt Assembly/Rewriting DSL
4. Подключение к разным провайдерам

Субъективно, с чем мне было приятно работать в Koog

1. Аннотация инструментов и классов
2. Запуск LLM в разных режимах одной строчкой
3. Prompt Assembly/Rewriting DSL
4. Подключение к разным провайдерам
5. Поддержка MCP

Субъективно, с чем мне было приятно работать в Koog

1. Аннотация инструментов и классов
2. Запуск LLM в разных режимах одной строчкой
3. Prompt Assembly/Rewriting DSL
4. Подключение к разным провайдерам
5. Поддержка MCP
6. Поддержка Skill'ов

Субъективно, с чем мне было приятно работать в Koog

1. Аннотация инструментов и классов
2. Запуск LLM в разных режимах одной строчкой
3. Prompt Assembly/Rewriting DSL
4. Подключение к разным провайдерам
5. Поддержка MCP
6. Поддержка Skill'ов
7. Трассирование (с оговорками)

Выводы

Выводы

- Логика агента зависит не только от модели, но и от проксирующего сервиса, и от провайдера

Выводы

- Логика агента зависит не только от модели, но и от проксирующего сервиса, и от провайдера
- Учесть все эти особенности во FrameWork — невозможно, но Framework дает много полезных абстракций

Выводы

- Логика агента зависит не только от модели, но и от проксирующего сервиса, и от провайдера
- Учесть все эти особенности во FrameWork — невозможно, но Framework дает много полезных абстракций (для меня в первую очередь удобный вызов инструментов, structured output и prompt assembly/rewriting)

Выводы

- Логика агента зависит не только от модели, но и от проксирующего сервиса, и от провайдера
- Учесть все эти особенности во FrameWork – невозможно, но Framework дает много полезных абстракций (для меня в первую очередь удобный вызов инструментов, structured output и prompt assembly/rewriting)
- Если модель выдает верный ответ, это не значит, что она его выдает наилучшим образом: анализировать надо все: reasoning, промежуточные ответы, логику изменения контекста

Выводы

- Логика агента зависит не только от модели, но и от проксирующего сервиса, и от провайдера
- Учесть все эти особенности во FrameWork — невозможно, но Framework дает много полезных абстракций (для меня в первую очередь удобный вызов инструментов, structured output и prompt assembly/rewriting)
- Если модель выдает верный ответ, это не значит, что она его выдает наилучшим образом: анализировать надо все: reasoning, промежуточные ответы, логику изменения контекста
- Возвращаясь к эпиграфу: пока я в бегах — работаю не по-старому, но и авангардистом не назовешь. Скоро поймают и вернут?

Выводы

- Логика агента зависит не только от модели, но и от проксирующего сервиса, и от провайдера
- Учесть все эти особенности во FrameWork — невозможно, но Framework дает много полезных абстракций (для меня в первую очередь удобный вызов инструментов, structured output и prompt assembly/rewriting)
- Если модель выдает верный ответ, это не значит, что она его выдает наилучшим образом: анализировать надо все: reasoning, промежуточные ответы, логику изменения контекста
- Возвращаясь к эпиграфу: пока я в бегах — работаю не по-старому, но и авангардистом не назовешь. Скоро поймают и вернут?

<https://github.com/fiddlededee/my-copilot>

Источники

Источники

1. Mobius 2026: Автоматическое code review на базе OpenCode: LLM-агент в CI/CD (Александр Юдин)

Источники

1. Mobius 2026: Автоматическое code review на базе OpenCode: LLM-агент в CI/CD (Александр Юдин)
2. JPoint 2026: Koog — твой друг: специализированный агент на Kotlin (Максим Колпаков, Александр Нозик)

Источники

1. Mobius 2026: Автоматическое code review на базе OpenCode: LLM-агент в CI/CD (Александр Юдин)
2. JPoint 2026: Koog — твой друг: специализированный агент на Kotlin (Максим Колпаков, Александр Нозик)
3. JPoint 2026: ИИ в ревью кода: как перестать тюнить промпты и начать строить систему (Надежда Волкова)

