Kodify Nano

Kodify Nano – ИИ-ассистент для программистов, помогающий в написании программного кода и решении других задач разработки, таких как рефакторинг, тестирование и документирование. Ассистент поддерживает основные языки программирования, включая Python, Java, JavaScript, C# и Go, и может переводить код с одного языка на другой.

Kodify Nano представляет собой LLM и плагин, встраиваемый в IDE от JetBrains (например, PyCharm) и Visual Studio Code. Установка плагина в IDE даёт разработчику возможность обращаться к LLM Kodify Nano для автоматической генерации кода на основе данных из редактируемого пользователем файла, а также для решения других задач. LLM модель Kodify Nano содержит 1.5 миллиарда параметров и может быть запущена на видеокартах от 4 Гб видеопамяти. Модель разработана для выполнения задач разработки кода с минимальными ресурсами. Она оптимизирована для быстрого и эффективного взаимодействия с пользователями, обеспечивая высокую производительность даже в условиях ограниченных ресурсов.

Использование образа

Вы можете запустить Kodify Nano на OLLAMA двумя способами:

- используя Docker;
- локально. При этом способе запуска модель отвечает быстрее, чем при запуске через Docker.

Способ 1. Запуск Kodify Nano на OLLAMA в Docker

1. Если у вас нет видеокарты NVIDIA, запустите OLLAMA следующей командой:

```
docker run -e OLLAMA_HOST=0.0.0.0:8985 -p 8985:8985 --name ollama -d ollama/ollama
```

Убедитесь, что у вас установлен и запущен Docker перед запуском команды.

Важно! Если порт 8985 уже занят, замените его на любой другой свободный порт. В этом случае также требуется изменить порт в конфигурации плагина. Смотрите раздел **"Изменение адреса порта в настройках плагина в средах Visual Studio Code и JetBrains"**.

Если у вас есть видеокарта NVIDIA и нужно добавить GPU в контейнер, выполните следующую команду:

```
docker run --runtime nvidia -e OLLAMA_HOST=0.0.0.0:8985 -p 8985:8985 --name ollama -d ollama/ollama
```

2. Загрузите модель в OLLAMA:

```
docker exec ollama ollama pull hf.co/MTSAIR/Kodify-Nano-GGUF
```

где "hf.co/MTSAIR/Kodify-Nano-GGUF" - имя модели. Подробнее о модели смотрите в https://huggingface.co/MTSAIR/Kodify-Nano-GGUF.

3. Задайте корректное имя для модели.

   Измените "hf.co/MTSAIR/Kodify-Nano-GGUF" на "kodify_nano" следующей командой:

```
docker exec ollama ollama cp hf.co/MTSAIR/Kodify-Nano-GGUF kodify_nano
```

4. Запустите модель:

```
docker exec ollama ollama run kodify_nano
```

Способ 2. Запуск Kodify Nano на OLLAMA локально

1. Скачайте OLLAMA с сайта командой:

```
https://ollama.com/download
```

2. Задайте порт:

```
export OLLAMA_HOST=0.0.0.0:8985
```

> **Важно!** Если порт 8985 уже занят, замените его на любой другой свободный порт. В этом случае также требуется изменить порт в конфигурации плагина. Смотрите раздел **"Изменение адреса порта в настройках плагина в средах Visual Studio Code и JetBrains"**.

3. Поднимите сервер OLLAMA.

```
ollama serve &
```

4. Загрузите модель, выполнив следующую команду:

```
ollama pull hf.co/MTSAIR/Kodify-Nano-GGUF
```

где "hf.co/MTSAIR/Kodify-Nano-GGUF" - имя модели. Подробнее о модели смотрите в https://huggingface.co/MTSAIR/Kodify-Nano-GGUF.

5. Задайте корректное имя для модели. Измените hf.co/MTSAIR/Kodify-Nano-GGUF на kodify_nano следующей командой:

```
ollama cp hf.co/MTSAIR/Kodify-Nano-GGUF kodify_nano
```

6. Запустите модель следующей командой:

```
ollama run kodify_nano
```

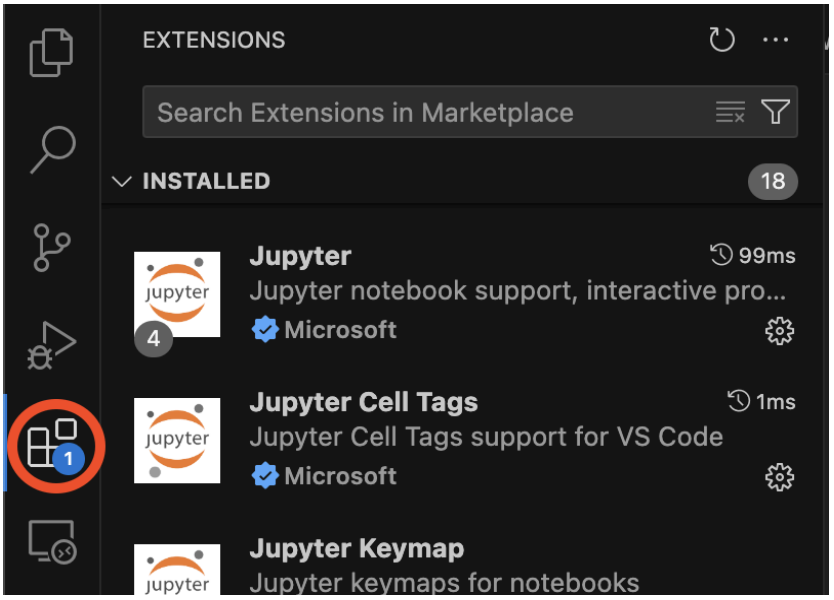
Список других моделей Kodify Nano:

- https://huggingface.co/MTSAIR/Kodify-Nano
- https://huggingface.co/MTSAIR/Kodify-Nano-CPU
- https://huggingface.co/MTSAIR/Kodify-Nano-GPTQ

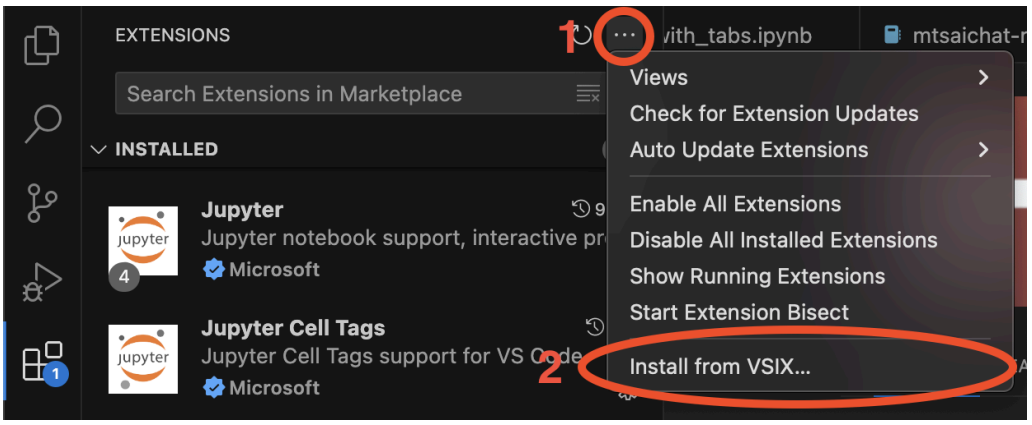
Установка плагина

Установка плагина в среде Visual Studio Code

1. Скачайте последнюю версию плагина Kodify для Visual Studio Code. Ссылку для скачивания вам предоставят.
2. Перейдите в раздел "Extensions" на левой панели инструментов.

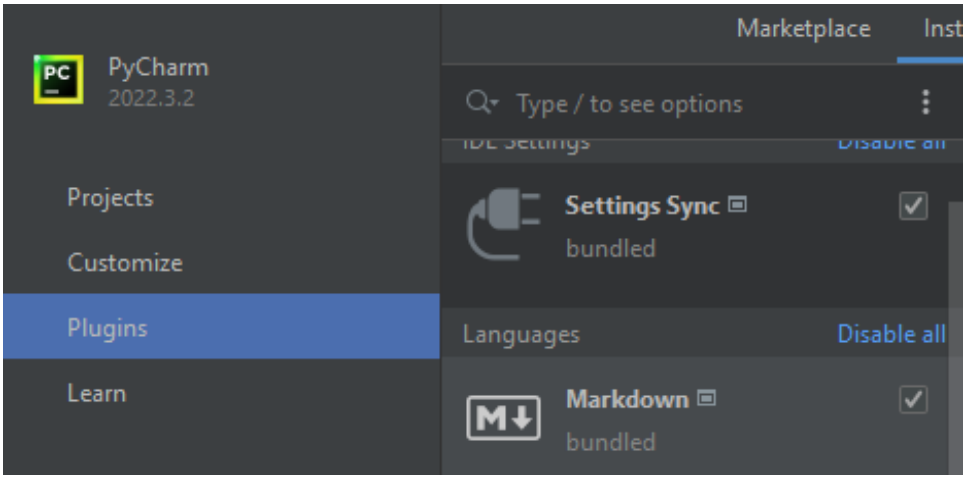


3. В диалоговом меню выберите пункт "Install from VSIX…" и укажите скачанный файл плагина.

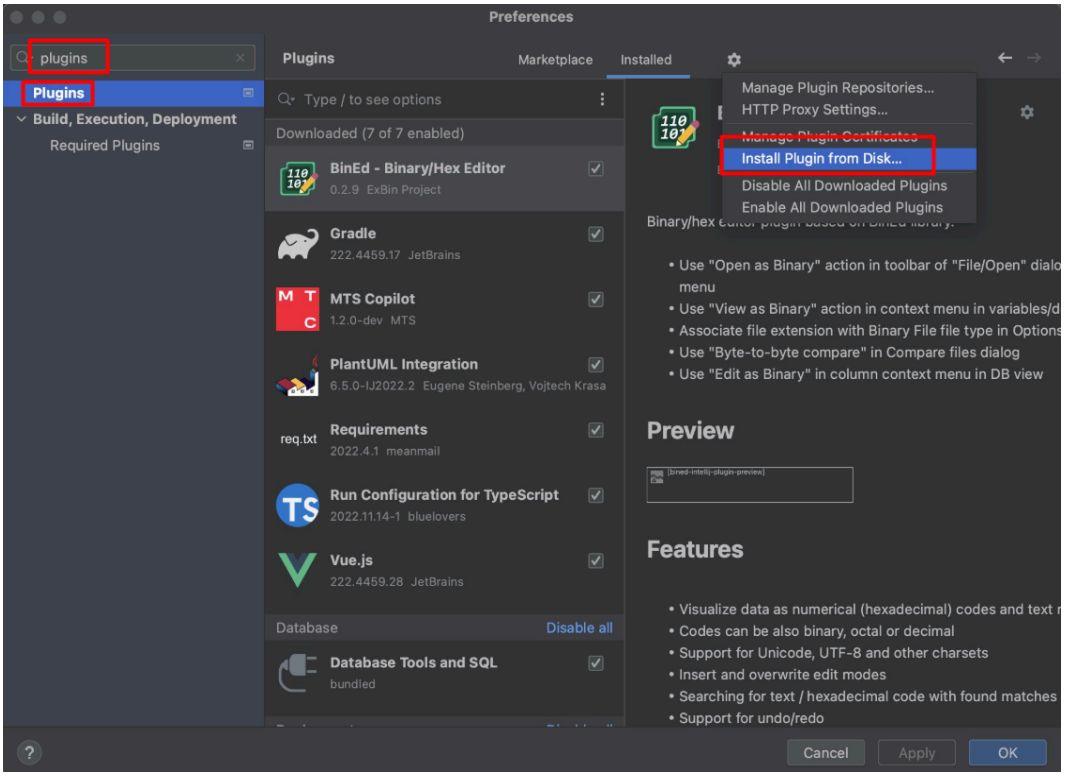


Установка плагина в среде JetBrains

1. Скачайте последнюю версию плагина Kodify для JetBrains. Ссылку для скачивания вам предоставят.
2. Запустите IDE и перейдите в настройки IDE.
3. Выберите пункт "Plugins" для перехода в настройку плагинов.



4. Нажмите на шестеренку в правом верхнем углу и выберите "Install Plugin from Disk…".

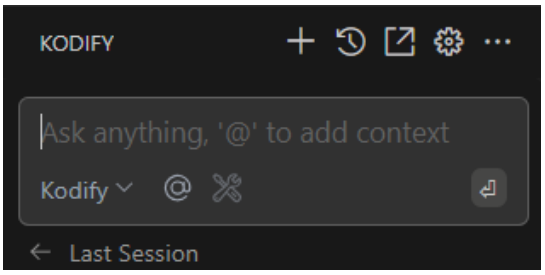


5. Выберите скачанный файл с плагином.
6. Подтвердите перезагрузку IDE после установки. Нажмите "Restart IDE".

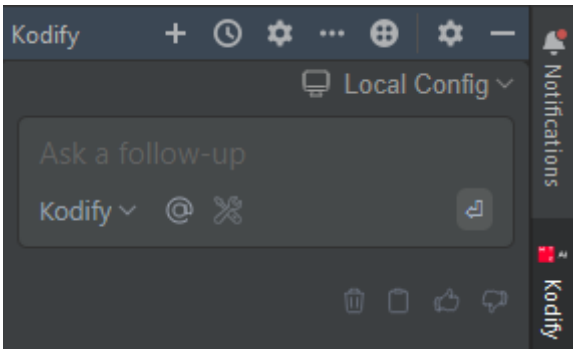
Изменение адреса порта в настройках плагина в средах Visual Studio Code и JetBrains

Если порт при поднятии Docker Compose отличается от порта **8985**, измените порт в файле *config.json*, выполнив следующие шаги:

1. Откройте в IDE любой файл.
2. Откройте боковую панель Kodify, используя следующую комбинацию клавиш:
   - в Visual Studio Code - "CTRL + L" ("CMD + L" на Mac);



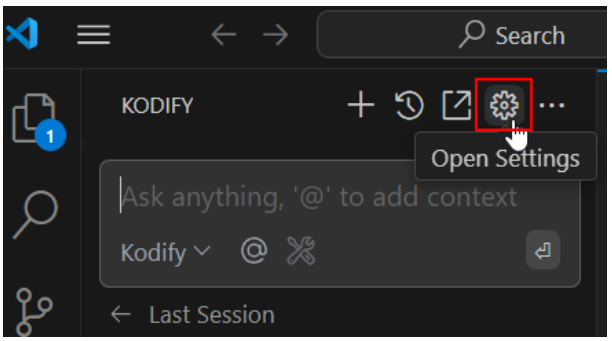
   - в JetBrains - "CTRL+ J" ("CMD + J" на Mac).



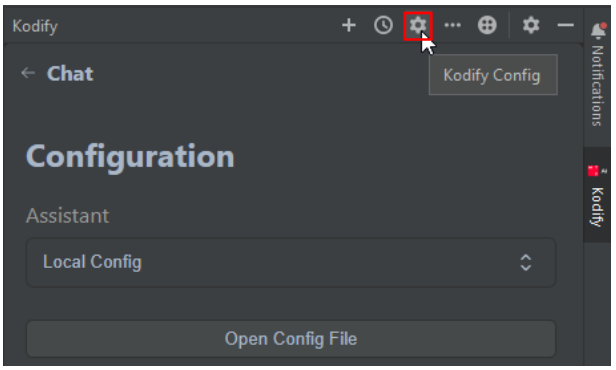
3. Откройте **конфигурационный файл *config.json*** одним из следующих способов:

Способ 1

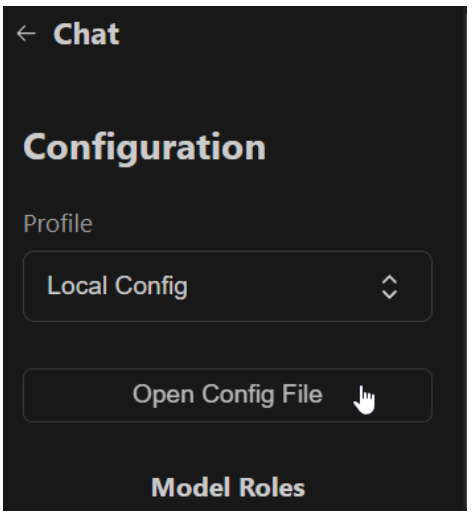
На боковой панели Kodify нажмите на кнопку "Open Settings" в Visual Studio Code



или "Kodify Config" в JetBrains.

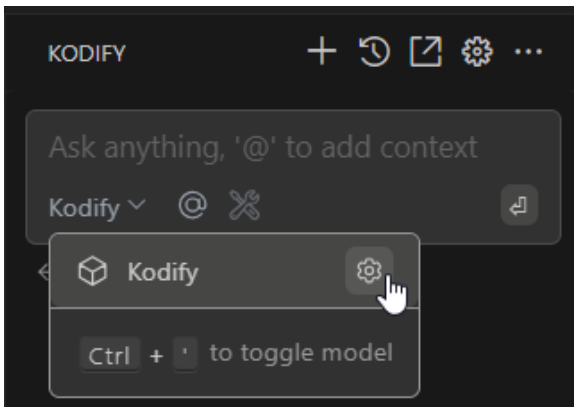


В разделе "Configuration" для Chat настроек нажмите кнопку "Open Config File".



Способ 2

На боковой панели раскройте меню Kodify и нажмите на кнопку "Open Settings" (шестеренка).



4. Измените в конфигурационном файле *config.json* адрес порта для поля **"apiBase"** в разделах конфигурации плагина **"tabAutocompleteModel"** и **"models"**.
5. Сохраните настройки файла *config.json* в IDE. Для этого, нажмите комбинацию клавиш **Ctrl+S** или выберите пункт меню IDE: **File -> Save**.