



СИСТЕМА ГЕНЕРАЦИИ И АВТОДОПОЛНЕНИЯ ПРОГРАММНОГО КОДА НА
ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ (NEUROCODE)

Руководство пользователя ПО

Листов 11

СОДЕРЖАНИЕ

1. Общая информация.....	3
1.1. Назначение	3
1.2. Область применения	3
2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО ИСПОЛЬЗОВАНИЮ ЧАТА NEUROCODE	4
3. КОНТАКТЫ	11

1. ОБЩАЯ ИНФОРМАЦИЯ

1.1. Назначение

ПО предназначено для генерации и автоматического дополнения исходного кода программного обеспечения путем обработки заданий на естественном языке методами искусственного интеллекта в целях ускорения разработки программного обеспечения и снижения вероятности ошибок разработчиков.

1.2. Область применения

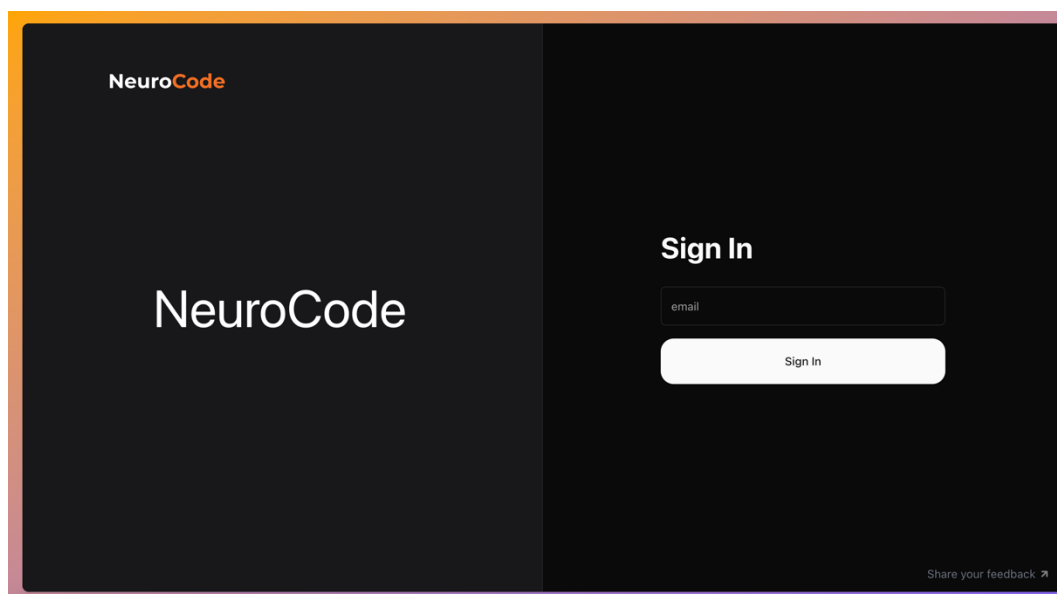
ПО обеспечивает возможность автоматизации процесса разработки ПО в высокотехнологичных компаниях, занимающихся разработкой собственных решений как для внешнего, так и для внутреннего использования.

2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО ИСПОЛЬЗОВАНИЮ ЧАТА NEUROCODE

2.1. Авторизация в пользовательском интерфейсе NeuroCode

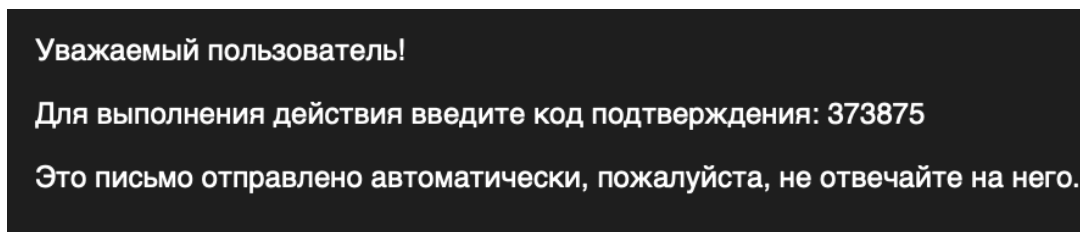
Для входа в пользовательский интерфейс ПО NeuroCode необходимо пройти процедуру авторизации в системе. Для этого необходимо перейти по ссылке <https://chat.axenix-neurocode.pro/login>, отобразится экран авторизации в пользовательском интерфейсе (Рисунок 1).

Рисунок 1 – Экран авторизации в пользовательском интерфейсе



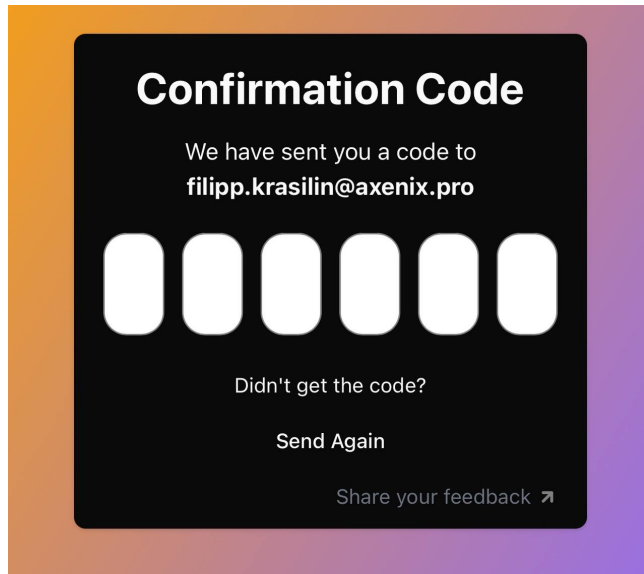
В поле «email» необходимо написать свой адрес электронной почты, после чего на указанный адрес придет одноразовый пароль в письме, пример которого представлен на Рисунке 2.

Рисунок 2 – Пример текста письма, содержащего одноразовый пароль для авторизации в NeuroCode



Далее необходимо ввести пришедший пароль в появившееся поле (Рисунок 3).

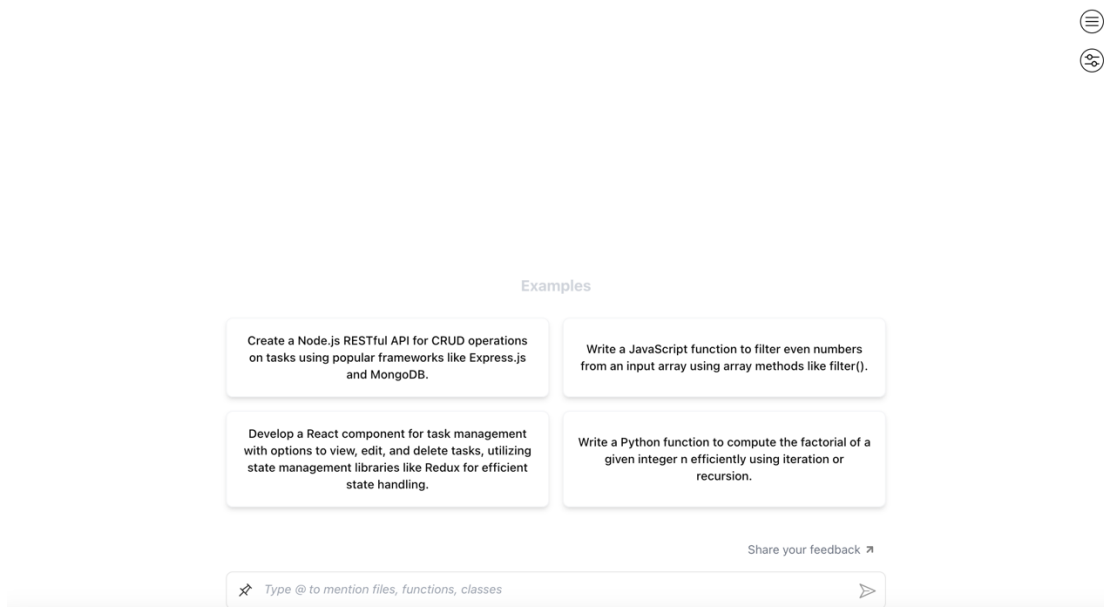
Рисунок 3 – Поле для введения одноразового пароля для авторизации в NeuroCode



2.2. Работа в чате NeuroCode

После успешного введения пароля откроется страница чата (Рисунок 4).

Рисунок 4 – Страница чата NeuroCode



Под надписью «Examples» представлены примеры запросов, которые можно выбрать для получения ответа в чате. Если нажать на один из них, то в чате начнет генерироваться ответ (Рисунок 5).

Рисунок 5 – Пример ответа генерации при выборе одного из примеров запроса

You
Create a Node.js RESTful API for CRUD operations on tasks using popular frameworks like Express.js and MongoDB.
2025-01-24 13:35

Code Assistant
Давайте создадим простое приложение для управления задачами с помощью Node.js, Express.js и MongoDB.
Шаг 1: Установка необходимых пакетов
Сначала установите необходимые пакеты:

```
npm init -y
npm install express mongoose body-parser cors
```

Copy Apply

Шаг 2: Создание модели задачи
Создайте файл `task.model.js` и добавьте в него следующий код:

```
const mongoose = require('mongoose');
const taskSchema = new mongoose.Schema({
  title: String,
  description: String,
  status: Boolean
});
module.exports = mongoose.model('Task', taskSchema);
```

Copy Apply

Share your feedback ↗

✂ Type @ to mention files, functions, classes ➤

Ответ можно пролистать при помощи «мышки» или при нажатии на стрелку вниз. Каждый кусок кода можно скопировать при нажатии на кнопку «Сору». Если пролистать ответ вниз, то будут отражаться дополнительные кнопки для работы с полученным ответом (Рисунок 6).

Рисунок 6 – Работа со сгенерированным ответом в чате

```
mongoose.connect('mongodb://localhost/tasks', { useNewUrlParser: true,
useUnifiedTopology: true });

app.use(express.json());
app.use(cors());

app.use('/api', taskRoute);

app.listen(3000, () => {
  console.log('Сервер запущен на порту 3000');
});
```

Шаг 6: Запуск сервера

Запустите сервер командой `node server.js`. Теперь вы можете использовать Postman или другой инструмент для отправки запросов к вашему API.

Примеры запросов

- GET /api/tasks - Получить все задачи
- POST /api/tasks - Создать новую задачу
- PUT /api/tasks/:id - Обновить задачу по ID
- DELETE /api/tasks/:id - Удалить задачу по ID

Теперь у вас есть готовое приложение для управления задачами с помощью Node.js, Express.js и MongoDB!

2025-01-24 13:36

🔄 📄 🗑️ 🗨️ Apply Diff



1 2 3 4

Share your 5 ack ↗

✂ Type @ to mention files, functions, classes



Под номером 1 располагается кнопка, при нажатии которой происходит повторная генерация ответа по тому же самому запросу. Под номером 2 – кнопка копирования всего ответа. Под номером 3 – кнопка положительной оценки сгенерированного ответа. Под номером 4 – кнопка отрицательной оценки сгенерированного ответа. Под номером 5 – кнопка обратной связи, при нажатии на которую появляется всплывающее окно (Рисунок 7).

Рисунок 7 – Окно обратной связи

Send Feedback



Tell us about your experience, what went wrong, what you liked, what we can improve

name@gmail.com

Short info

Message



Click to upload or drag and drop

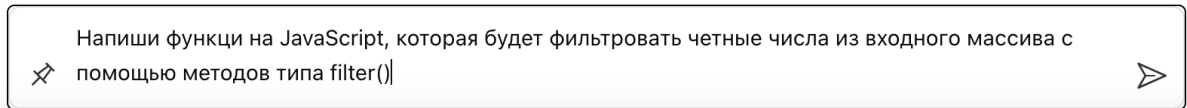
Max file size, 10mb

Max files, 5

В этом окне нужно указать адрес электронной почты, который был использован при авторизации, краткое описание проблемы, развернутое сообщение обратной связи, а также можно прикрепить до 5 файлов общим объемом не больше 10 мегабайтов.

В поле для введения чата можно ввести собственный запрос (Рисунок 8).

Рисунок 8 – Пример заполненного поля для введения запроса в чате



Напиши функции на JavaScript, которая будет фильтровать четные числа из входного массива с помощью методов типа filter()

После введения текста необходимо нажать на кнопку справа, и тогда произойдет генерация ответа по запросу из чата (Рисунок 9).

Рисунок 9 – Пример ответа на запрос в чате

You

Напиши функции на JavaScript, которая будет фильтровать четные числа из входного массива с помощью методов типа filter()

2025-01-24 14:05

Code Assistant

Вот пример функции на JavaScript, которая использует метод `filter()` для фильтрации четных чисел из входного массива:

```
function filterEvenNumbers(arr) {
  return arr.filter(num => num % 2 === 0);
}

// Пример использования:
const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
const evenNumbers = filterEvenNumbers(numbers);
console.log(evenNumbers); // Вывод: [2, 4, 6, 8, 10]
```

Copy Apply

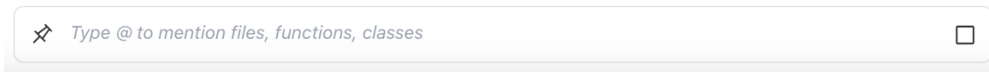
В этой функции `filterEvenNumbers` мы передаем входной массив `arr`. Метод `filter()` создает новый массив со всеми элементами, для которых функция-предикат возвращает `true`.

Функция-предикат в данном случае проверяет, является ли число четным, используя оператор `%` (остаток от деления). Если остаток от деления числа на 2 равен 0, то число четное и функция возвращает `true`, что приводит к включению этого числа в результирующий массив.

2025-01-24 14:06

В процессе генерации можно ее приостановить посредством нажатия кнопки, которая появляется справа в текстовом поле (Рисунок 10).

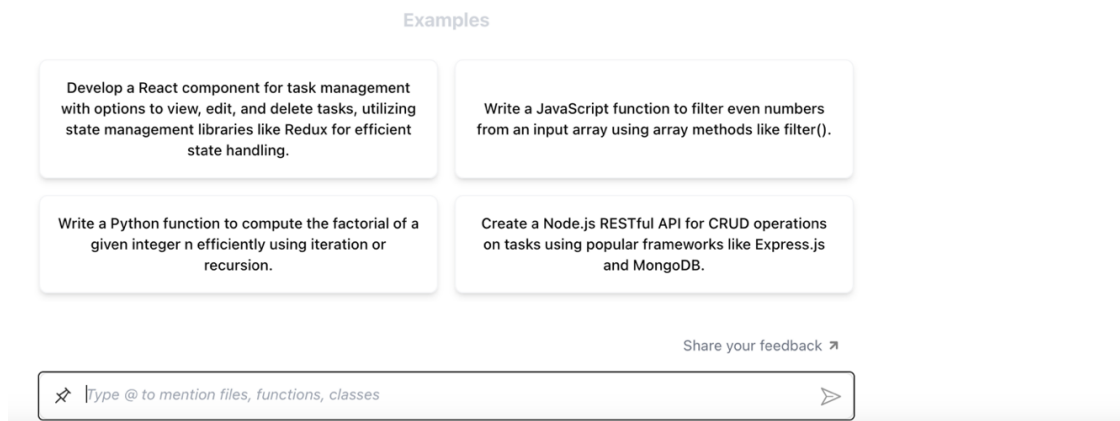
Рисунок 10 – Вид текстового поля чата в процессе генерации ответа



2.3. Настройки чата NeuroCode

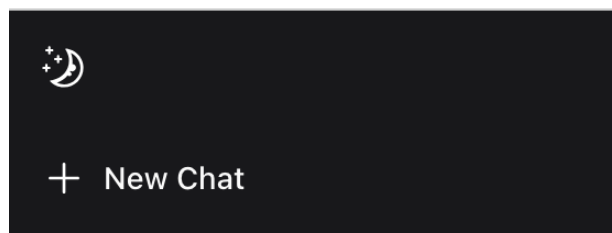
В чате также может осуществляться ряд настроек (Рисунок 11).

Рисунок 11 – Кнопки настроек в чате Neurocode



Сверху справа есть две кнопки настроек (выделены красным прямоугольником). При нажатии на первую появляется окно слева (Рисунок 12).

Рисунок 12 – Первое поле для настройки чата



В данном поле можно создать новый чат «New chat», старый чат(-ы) при этом останутся, в них можно будет вернуться в том же самом окне. Также можно сменить тему

на темную (или светлую, если темная уже выбрана) посредством нажатия на кнопку слева сверху.

При нажатии на вторую кнопку с Рисунка 11 откроется окно настроек справа (Рисунок 13).

Рисунок 13 – Второе поле настроек чата

Settings ×

1. Simple UI
 Enabled

2. Full Screen
 Enabled

3. Font size
Small ————— Big

4. Models
llama3.3:70b

5. Prompts

Prompt for all chats:
Will be used as a standard part for all generations

Prompt for current chat:
Will be used as a standard part for current chat generation

Update preprompts

Под номером 1 располагается чек-бокс (флажок), при нажатии на который применяется упрощенный пользовательский интерфейс чата. Под номером 2 располагается чек-бокс (флажок) применения режима полного UI, при нажатии на который применяется режим полного UI, то есть открывается чат на весь экран. Под номером 3 располагается бегунок для изменения размера шрифта, то есть он позволяет регулировать размер отражаемых в чате букв. Под номером 4 располагается выпадающий список для выбора LLM, которая будет обрабатывать запросы пользователя из чата. Под номером 5 располагаются поля для записи препромтов, которые позволяют ввести собственный препромт текстом для всех чатов (в первом текстовом поле) и для текущего чата (во втором текстовом поле), далее препромт будет отправляться в выбранную LLM при каждом запросе в чате.

3. КОНТАКТЫ

При возникновении проблем с подключением к проверочному экземпляру или любых вопросов по функциональности/архитектуре просьба обращаться по следующим контактам:

Телефон: +7 (495) 755-97-70

E-mail: Support_NeuroCode@axenix.pro