

Цифриум

Технологическая карта

Умный код на Python. Базовый уровень

Модуль 3

Тема 4

Урок 6

**Интеграция API GPT в простую web-систему
(часть 2)**



Уважаемый коллега, обратите внимание!

Учёт присутствия ребёнка на уроке ведётся только по цифровому следу. Каждый ученик из группы на уроке должен передать цифровой след на нашей платформе.

Вы отвечаете за передачу следа учеником на нашей платформе во время урока.

Передавать цифровой след нужно в личном кабинете ученика.

Цифровой след засчитывается по итогу выполнения учеником трёх задач на платформе в ходе урока.

Обращаем внимание – в задачах с отправкой ответа в виде файла наша платформа поддерживает следующие форматы: .pdf, .doc, .png, .jpeg, .xlsx, .mp3, .py, .ipynb, .txt, .csv, .json, .xml, .sb3, .ino, .hex.

Контролируйте время урока, чтобы не пропустить момент, когда ученики должны залогиниться в личном кабинете и перейти к решению задач на платформе.

Проследите, чтобы каждый ученик залогинился в личном кабинете и выполнил 3 задачи на платформе. В противном случае, Вам не будет засчитано, что вы провели урок, занятие не будет оплачено, а ученику не будет выставлена отметка о присутствии на уроке.

Просим вас в ходе 45 минут урока проконтролировать, чтобы:

1. Каждый из учеников залогинился в личном кабинете.
2. Каждый ученик в разделе **«Решить задания»** отправил решения по 3 задачам – «Практическое задание 1», «Практическое задание 2» и «Практическое задание 3».

О занятии

Краткое содержание занятия:

1. Узнаете, что такое API и fetch-запросы.
2. Научитесь использовать `async/await` для асинхронного кода.
3. Изучите формат JSON для обмена данными.
4. Реализуете отправку запроса к GPT API с помощью `fetch`.
5. Обработаете JSON-ответ и обновите страницу.
6. Добавьте индикатор загрузки и обработку ошибок.

Цель:

Научиться использовать `fetch`-запросы для взаимодействия с GPT API и освоить `async/await` для управления асинхронным кодом.

Задачи:

1. Реализовать отправку асинхронного запроса к GPT API с помощью `fetch`.
2. Обработать JSON-ответ и динамически обновить область вывода на странице.

Термины

API (Application Programming Interface)

Интерфейс для общения между программами. Позволяет отправлять запросы и получать ответы от внешних сервисов.

`fetch`

Функция JavaScript для отправки HTTP-запросов в интернет. Используется для взаимодействия с API.

О ЗАНЯТИИ

JSON (JavaScript Object Notation)

Формат представления данных, похожий на словарь. Используется для обмена данными между клиентом и сервером.

async/await

Ключевые слова JavaScript для работы с асинхронным кодом. Позволяют ждать выполнения запроса без блокировки страницы.

HTTP-запрос

Обращение к серверу через интернет для получения или отправки данных.

Authorization

Заголовок HTTP-запроса, содержащий API-ключ для авторизации.

Bearer Token

Тип токена авторизации, используемый в заголовке Authorization.

Promise

Объект JavaScript, представляющий результат асинхронной операции (выполнится в будущем).

try/catch

Конструкция JavaScript для обработки ошибок в коде.

О занятии

Необходимые материалы:

1. HTML/JS-файлы с предыдущего занятия.
2. API-ключ (подготовить заранее).
3. Интернет-браузер и редактор кода.

Темы и время

ЭТАП	ВРЕМЯ
Приветствие	5 мин.
Теория	25 мин.
Итоги занятия	5 мин.
Получение фидбека. Рефлексия. Проверка знаний. Домашнее задание	5 мин.
Вопрос-ответ	5 мин.
Итого	45 мин.

Ход занятия

Номер слайда	Пояснение к слайду
1	Приветствие учеников. Добрый день, ребята! Сегодня мы продолжим работу с нашим web-интерфейсом для GPT и научимся отправлять реальные запросы к API через fetch. Помните, на прошлом уроке мы сделали форму, которая реагировала на клики? Сегодня мы научим её по-настоящему общаться с сервером. Это важный шаг, который позволит нашему приложению стать по-настоящему умным и интерактивным. Не переживайте, если код покажется сложным – мы разберём каждую строчку.
2	Титульный слайд. Тема занятия: «Интеграция API GPT в простую web-систему (часть 2)». Это вторая часть, где мы научимся работать с реальными API-запросами.
3	На прошлом занятии вы: Научились создавать базовый web-интерфейс, познакомились с HTML и JavaScript, создали форму «Задай вопрос», написали обработчик событий, научились работать с DOM и отлаживать код через консоль. Давайте быстро вспомним: кто покажет, как найти элемент по ID? А кто вспомнит, зачем нам нужна проверка на пустой ввод? Эти навыки станут фундаментом для сегодняшнего урока, когда мы добавим настоящую связь с искусственным интеллектом.
4	Сегодня на занятии вы: Узнаете, что такое API и fetch-запросы, научитесь использовать async/await для асинхронного кода, изучите формат JSON, реализуете отправку запроса к GPT API, обработаете ответ и обновите страницу, добавите индикатор загрузки и обработку ошибок. Звучит много, но на самом деле это логичная цепочка: берём вопрос пользователя, отправляем на сервер, ждём ответ и показываем его. Всё как в жизни, только быстрее.

Ход занятия

5	<p>Цели и задачи урока. Цели: научиться использовать <code>fetch</code>-запросы для взаимодействия с GPT API и освоить <code>async/await</code> для управления асинхронным кодом. Задачи: реализовать отправку асинхронного запроса к API и обработать JSON-ответ для обновления страницы. Объясните ученикам, что <code>async/await</code> – это как волшебное слово «подожди», которое позволяет коду не спешить, пока сервер думает. Без этого страница бы зависала до получения ответа. <code>Fetch</code> используется для выполнения асинхронных сетевых запросов (получение/отправка данных на сервер)</p>
6	<p>План занятия. Что мы будем делать: узнаем, что такое API и как им пользоваться, научимся отправлять запрос через <code>fetch</code>, разберёмся с форматом JSON, реализуем функцию <code>askGPT</code> с подключением к GPT, выведем ответ на страницу и протестируем работу с разными вопросами. Мы будем двигаться по шагам, как в инструкции по сборке конструктора. Каждый шаг приближает нас к работающему приложению.</p>
7	<p>Что такое API. API – это способ отправить вопрос и получить ответ от другой программы, в нашем случае – от сервера с GPT. Пример: мы отправляем вопрос → GPT обрабатывает → возвращает текст ответа. Объясните простым языком: API – это как почтовый ящик или официант в ресторане: мы отдаём заказ, ждём немного, и нам приносят готовое блюдо. Сервер не видит нашу страницу, он понимает только специальные запросы через API.</p>
8	<p>Как работает <code>fetch</code>. <code>Fetch</code> – это встроенная в JavaScript функция для отправки запросов через интернет. Она работает так: указываем адрес URL, заголовки и данные, отправляем запрос, получаем результат в формате JSON и обрабатываем его через <code>await</code>. Объясните, что <code>async/await</code> упрощают работу с асинхронным кодом, позволяя писать его как обычный, без запутанных <code>.then()</code>. Это делает код чище и понятнее.</p>

Ход занятия

9	<p>Что такое JSON. JSON – это способ представления данных, похожий на словарь в кавычках. Пример: { "request": "Привет, как дела?" }. JSON используется для обмена данными между клиентом и сервером. Это текстовый формат, который легко читается и людьми, и программами. Объясните, что ключи всегда в кавычках, а значения могут быть строками, числами или массивами. Это универсальный язык для передачи информации в интернете.</p>
10	<p>Структура запроса к нашему API. В коде мы используем fetch с адресом <code>https://example-api.com/api</code>, методом POST, заголовками Authorization(процесс проверки прав и предоставления доступа определенному пользователю или группе лиц к конкретным ресурсам, функциям или данным в системе) с Bearer-токеном и Content-Type: application/json. В теле запроса передаём JSON с именем чата и версией модели. Разберите каждую часть: URL – куда отправляем, метод POST – что отправляем данные, заголовки – кто мы и в каком формате, тело – что именно отправляем. Токен – это как пароль, который доказывает серверу, что мы имеем право пользоваться сервисом.</p>
11	<p>Что делает наша функция askGPT. Алгоритм работы: читает вопрос из поля ввода, проверяет, не пустой ли ввод, рн блокирует поле, создаёт чат если нужно, отправляет сообщение через sendMessage, получает ответ и показывает его, в блоке finally разблокирует интерфейс. Важно: обработка ошибок через try/catch позволяет показать понятное сообщение вместо падения страницы. Это как страховка: если запрос не удался, программа аккуратно сообщит о проблеме. Продемонстрируйте код Js</p>
12	<p>Самостоятельная работа. Задание: вставьте свой API-ключ и URL сервера в начало файла script.js, откройте index.html в браузере, напишите вопрос в поле и нажмите Enter или кнопку, запишите, что ответила модель, проверьте обработку пустого ввода, посмотрите в консоли F12, какие запросы уходят и какие ответы приходят. Дайте время</p>

Ход занятия

	<p>студентам выполнить задание. Помогите тем, у кого возникли сложности. Убедитесь, что все получили доступ к API и смогли отправить запросы.</p>
13	<p>Итоги занятия. Сегодня вы научились использовать fetch-запросы для взаимодействия с GPT API, освоили async/await для управления асинхронным кодом, реализовали отправку запроса через sendMessage, обработали JSON-ответ и обновили страницу, добавили индикатор загрузки и обработку ошибок через try/catch/finally, создали полноценный веб-интерфейс для общения с GPT. Спросите у учеников, что им показалось сложным, что понравилось. Поаплодируйте классу за первую работающую интеграцию!</p>
14	<p>На следующем занятии вы: улучшите интерфейс – добавьте историю сообщений, чтобы видеть весь диалог, научитесь форматировать ответ с переносами строк, добавьте кнопку очистки чата и индикатор набора текста. Представьте: ваш чат будет помнить контекст беседы, как настоящий собеседник! Домашнее задание: поэкспериментируйте с разными вопросами и посмотрите, как модель отвечает на сложные и простые запросы. Всем спасибо, занятие окончено!</p>

Практика

Вопросы для самопроверки

- Что делает метод `fetch()`?
- Что означает `Authorization` в заголовке запроса?
- Что такое `JSON` и для чего он используется?
- Зачем нужны ключевые слова `async` и `await`?
- Что произойдёт, если вопрос не введён?
- Как обрабатывать ошибки в JavaScript?

Задание 1

Модифицируйте функцию `askGPT()` так, чтобы она отправляла запрос к GPT API и выводила ответ на страницу.

Примеры выполнения заданий

Проверь себя по следующим критериям:

`fetch()` — делает HTTP-запросы к серверу (например, получает или отправляет данные).

`Authorization` в заголовке — передаёт учётные данные (чаще всего токен), чтобы сервер разрешил доступ к защищённым данным.

`JSON` — текстовый формат для хранения и передачи структурированных данных (например, объектов, массивов). Используется в API для обмена данными между клиентом и сервером.

`async/await` — упрощают работу с асинхронным кодом (например, `fetch`), позволяя писать его как синхронный, без `.then()`.

Если вопрос не введён — запрос может завершиться ошибкой, вернуть пустой/некорректный ответ или вызвать исключение — зависит от реализации.

Обработка ошибок в JavaScript:

```
try {  
  
    // код, который может вызвать ошибку  
  
} catch (error) {  
  
    // обработка ошибки  
  
}
```

Для `fetch` также нужно проверять `response.ok`, так как сетевые ошибки и HTTP-ошибки обрабатываются по-разному.

Задание 1

Открыть файлы `index.html` и `script.js` из прошлого урока.

Получить API-ключ (если ещё не получен).

Модифицировать функцию `askGPT()`: добавить `async`.

Добавить проверку на пустой ввод.

Примеры выполнения заданий

Написать fetch-запрос к API.

Обработать JSON-ответ и вывести на страницу.

Добавить обработку ошибок через try/catch.

Протестировать с несколькими вопросами.

Полное решение:

Файл: script.js (часть body)

```
const API_KEY = ""; // Вставьте ваш ключ или загружайте из env
```

```
    const MODEL = "qwen/qwen3-vl-30b-a3b-thinking";
```

```
    let messages = [];
```

```
    async function send() {
```

```
        const input = document.getElementById("input");
```

```
        const text = input.value.trim();
```

```
        if (!text) return;
```

```
        messages.push({ role: "user", content: text });
```

```
        addMessage("Вы: " + text);
```

```
        input.value = "";
```

```
        const res = await
```

```
        fetch("https://openrouter.ai/api/v1/chat/completions", {
```

```
            method: "POST",
```

```
            headers: {
```

Примеры выполнения заданий

```
    "Authorization": "Bearer " + API_KEY,
    "Content-Type": "application/json"
  },
  body: JSON.stringify({ model: MODEL, messages: messages })
});

const data = await res.json();
const reply = data.choices[0].message.content;
messages.push({ role: "assistant", content: reply });
addMessage("Qwen3-VL: " + reply);
}

function addMessage(text) {
  const div = document.createElement("div");
  div.textContent = text;
  document.getElementById("chat").appendChild(div);
}

function clear() {
  messages = [];
  document.getElementById("chat").innerHTML = "";
}
```

Примеры выполнения заданий

```
document.getElementById("input").addEventListener("keypress",  
function(e) {  
    if (e.key === "Enter") send();  
});
```

Сегодня на занятии вы:

- Научились использовать fetch-запросы для взаимодействия с GPT API.
- Освоили async/await для управления асинхронным кодом.
- Реализовали отправку асинхронного запроса к GPT API с помощью fetch.
- Обработали JSON-ответ и динамически обновили область вывода на странице.
- Добавили индикатор загрузки и обработку ошибок.
- Создали полноценный веб-интерфейс для общения с GPT.

На следующем занятии вы:

- Рассмотрите распространённые функции активации (ReLU, Sigmoid, Tanh).
- Поймете алгоритм обучения через прямой и обратный проходы на примере PyTorch.
- Выявите основные компоненты нейрона и их роль в сети.

Получение фидбека. Рефлексия. Проверка знаний

Попросите ребят решить интерактивные задачи по теме урока.

Ответы к интерактивным задачам

Практическое задание 1

Как называется функция JavaScript для отправки HTTP-запросов в интернет?

fetch

Практическое задание 2

Что такое JSON в контексте веб-разработки?

- Язык программирования для создания веб-сайтов.
- **Формат представления данных для обмена информацией между клиентом и сервером**
- База данных для хранения информации.
- Тип шифрования данных

Практическое задание 3

Что необходимо для отправки запроса к API? (выберите все правильные варианты)

- **API-ключ для авторизации**
- **URL адрес API (endpoint)**
- Библиотека json
- **Функция fetch или аналогичная**

Ответы к интерактивным задачам

Вопрос-ответ

Если во время подготовки или проведения урока возникают вопросы, их необходимо адресовать в методический чат.