

# Сетевые запросы

**№ урока:** 20 **Курс:** JavaScript Базовый

**Средства обучения:** Visual Studio Code  
Web Browser

## Обзор, цель и назначение урока

Научиться использовать метод `fetch` для отправки сетевых запросов. Изучить основную структуру HTTP протокола.

## Изучив материал данного занятия, учащийся сможет:

- Использовать `fetch` и `promise` для отправки сетевых запросов и обработки ответов.
- Понимать разницу между HTTP глаголами и использовать метод `fetch` для отправки запроса с нужным HTTP глаголом.
- Устанавливать заголовки в HTTP запрос и считывать заголовки из HTTP ответа.
- Отменять HTTP запрос.

## Содержание урока

1. HTTP протокол
2. Основы использования метода `fetch`
3. Настройка параметров HTTP запроса при работе с `fetch`
4. Отмена HTTP запроса

## Резюме

- **HTTP - HyperText Transfer Protocol** протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.
- Для анализа HTTP запроса и ответа можно использовать встроенный инструмент разработки в браузере (вкладка Network) или отдельными приложениями, например: Telerik Fiddler, Postman, Wireshark.
- HTTP запрос состоит из трех компонентов – стартовой строки, заголовков и тела запроса.
- Стартовая строка состоит из – HTTP метода, URI и версии протокола.
- Список HTTP методов или HTTP глаголов
  - GET**                      запрос представления ресурса
  - POST**                    отправка сущности
  - PUT**                     изменение сущности
  - DELETE**                удаление сущности
  - CONNECT**             устанавливает соединение
  - OPTIONS**              описание параметров соединения
  - HEAD**                  запрос заголовков без тела

## **PATCH**      частичное изменение ресурса

- Группы кодов состояния  
**1xx** информационный  
**2xx** успех  
**3xx** перенаправление  
**4xx** ошибка клиента  
**5xx** ошибка сервера
- **AJAX** (**A**synchronous **J**avaScript **a**nd **X**ML) – термин, определяющий сетевой запрос, сделанный с помощью JavaScript кода. AJAX запрос, это запрос выполненный через JavaScript код.
- XMLHttpRequest – функция для создания объектов, которые позволяют отправлять сетевые запросы и обрабатывать ответ от сервера. Данный подход можно считать устаревшим, но из преимуществ можно выделить высокий уровень поддержки в разных браузерах.
- **Fetch** – современный вариант создания сетевых запросов. Данный метод возвращает promise, что упрощает организацию кода обработки запроса, так как все сетевые запросы по умолчанию выполняются асинхронно.
- Результат, связанный с promise, который возвращает функция fetch – встроенный класс Response. Данный класс предоставляет несколько методов для работы с телом ответа. Все методы возвращают Promise с результатом, полученным из тела HTTP ответа:

response.text() - получение ответа как обычного текста

response.json() - получение объекта из ответа в формате JSON

response.formData() - возвращает тело ответа в виде FormData (объектное представление полей формы)

response.blob() - бинарные данные с типом

response.arrayBuffer() - низкоуровневое представление бинарных данных ArrayBuffer

- Для определения HTTP метода, при использовании метода fetch в объекте опций, который передается вторым параметром, нужно указать свойство method со строковым значением, соответствующим нужному HTTP методу:

```
fetch(url, {  
  method: 'POST'  
})
```

- Promise, который возвращается fetch, не переходит в состояние rejected если сервер вернул статус код ошибки 4xx или 5xx. Для проверки ответа сервера следует использовать свойство ok объекта response.
- Метод fetch не передает на сервер cookie до тех пор, пока не будет изменено значение параметра credentials.
- Хорошей практикой при написании кода, работающего асинхронно, является отображение индикатора асинхронной операции (пользователь будет понимать, что страница выполняет какое-то действие, а не зависла) и предоставление возможности

отменить асинхронную операцию (особенно если операция может выполняться длительное время).

- Для отмены сетевого запроса, запущенного через метод fetch, используется встроенный конструктор AbortController. Значение свойство signal объекта, созданного через этот конструктор, нужно передать как свойство signal в объект настроек функции fetch.
- При отмене асинхронной операции необходимо обработать исключение, которое будет выброшено в promise, связанном с асинхронной операцией.

### Закрепление материала

- Что такое AJAX?
- Опишите структуру HTTP запроса.
- Назовите основные HTTP глаголы (HTTP методы).
- Опишите структуру HTTP ответа.
- Назовите группы статус кодов, HTTP ответа.
- Какой способ отправки сетевых запросов, кроме метода fetch вы знаете?
- Как указать HTTP метод при отправке запроса через fetch?
- Как отменить сетевой запрос запущенный с помощью функции fetch?

### Самостоятельная деятельность учащегося

Выполните задания в директории Exercises\Tasks\020 Network Requests. Текст задач, находится в комментариях в тегах script.

### Рекомендуемые ресурсы

HTTP протокол

<https://ru.wikipedia.org/wiki/HTTP>

HTTP методы

<https://developer.mozilla.org/ru/docs/Web/HTTP/Methods>

Коды ответов HTTP

<https://developer.mozilla.org/ru/docs/Web/HTTP/Status>

XMLHttpRequest

<https://developer.mozilla.org/ru/docs/Web/API/XMLHttpRequest>

Использование Fetch

[https://developer.mozilla.org/ru/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/ru/docs/Web/API/Fetch_API/Using_Fetch)