Мережеві запити

№ уроку: 17 **Курс:** JavaScript Базовий

Засоби навчання:	Visual Studio Code Web Browser

Огляд, мета та призначення уроку

Навчитися використовувати метод fetch для надсилання мережевих запитів. Вивчити основну структуру протоколу HTTP.

Вивчивши матеріал цього заняття, учень зможе:

- Використовувати fetch і promise для надсилання мережевих запитів та обробки відповідей.
- Розуміти різницю між HTTP-дієсловами та використовувати метод fetch для надсилання запиту з потрібним дієсловом HTTP.
- Встановлювати заголовки в запит HTTP і зчитувати заголовки з відповіді HTTP.
- Скасовувати запит HTTP.

Зміст уроку

1. Протокол HTTP.
2. Основи використання методу fetch.
3. Налаштування параметрів HTTP-запиту під час роботи з fetch.
4. Скасування запиту HTTP.

Резюме

- **HTTP (HyperText Transfer Protocol)** – протокол прикладного рівня передання даних, спочатку – у вигляді гіпертекстових документів у форматі HTML, зараз використовується для передання довільних даних.
- Для аналізу HTTP-запиту та відповіді можна використовувати вбудований інструмент розробки у браузері (вкладка Network) або окремі застосунку, наприклад: Telerik Fiddler, Postman, Wireshark.
- HTTP-запит складається з трьох компонентів: **стартового рядка**, **заголовків** і **тіла запиту**.
- Стартовий рядок складається з: **HTTP-методу**, **URI** та **версії протоколу**.
- Список HTTP-методів, або HTTP-дієслів:
  - **GET** – запит представлення ресурсу.
  - **POST** – відправлення сутності.
  - **PUT** – зміна сутності.
  - **DELETE** – видалення сутності.
  - **CONNECT** – встановлює з'єднання.
  - **OPTIONS** – опис параметрів з'єднання.
  - **HEAD** – запит заголовків без тіла.
  - **PATCH** – часткова зміна ресурсу.

CyberBionic Systematics ®
19 Evgenia Sverstiuka Str., 5 floor
Kyiv, Ukraine

Tel.:0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **Групи кодів стану:**
  - **1xx** – інформаційний;
  - **2xx** – успіх;
  - **3xx** – перенаправлення;
  - **4xx** – помилка клієнта;
  - **5xx** – помилка сервера.

- **AJAX (Asynchronous JavaScript and XML)** – термін, який визначає мережевий запит, зроблений за допомогою JavaScript-коду. **AJAX-запит** – це запит, який виконаний через JavaScript-код.

- **XMLHttpRequest** – функція для створення об'єктів, які дають змогу надсилати мережеві запити та обробляти відповідь від сервера. Цей підхід можна вважати застарілим, але з переваг можна виділити високий рівень підтримки у різних браузерах.

- **Fetch** – найсучасніший варіант створення мережевих запитів. Цей метод повертає promise, що полегшує організацію коду обробки запиту, оскільки всі мережні запити за замовчуванням виконуються асинхронно.

- Результат, який пов'язаний із promise, який повертає **функція fetch** – вбудований клас Response. Цей клас надає кілька методів до роботи з тілом відповіді. Усі методи повертають Promise з результатом, отриманим з тіла HTTP-відповіді:

  - **response.text()** – отримання відповіді як звичайного тексту;
  - **response.json()** – отримання об'єкта з відповіді у форматі JSON;
  - **response.formData()** – повертає тіло відповіді у вигляді FormData (об'єктне представлення полів форми);
  - **response.blob()** – бінарні дані з типом;
  - **response.arrayBuffer()** – низькорівневе представлення бінарних даних ArrayBuffer.

- Для визначення методу HTTP під час використання методу fetch в об'єкті опцій, який передається другим параметром, потрібно вказати властивість method з рядковим значенням. Він має відповідати потрібному методу HTTP:

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

- Promise, який повертається fetch, не переходить у стан rejected, якщо сервер повернув статус-код помилки 4xx або 5xx. Для перевірки відповіді сервера варто використовувати властивість об'єкта response.

- Метод fetch не передає на сервер cookie до тих пір, поки не буде змінено значення параметра credentials.

- Хорошою практикою під час написання асинхронного коду є **зображування індикатора асинхронної операції** (користувач розумітиме, що сторінка виконує якусь дію, а не зависла) та **представлення можливості скасувати асинхронну операцію** (особливо якщо операція може виконуватись тривалий час).

- Для скасування запиту мережі, який запущений через метод fetch, використовується вбудований конструктор AbortController. Значення властивості signal-об'єкта, який

CyberBionic Systematics ®
19 Evgenia Sverstiuka Str., 5 floor
Kyiv, Ukraine
Tel.:0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

створений через цей конструктор, потрібно передати як властивість signal в об'єкт налаштувань функції fetch.

- Під час скасування асинхронної операції необхідно обробити виняток, який буде викинуто в promise, що зв'язаний з асинхронною операцією.

Закріплення матеріалу

- Що таке AJAX?
- Опишіть структуру запиту HTTP.
- Назвіть основні HTTP-дієслова (HTTP-методи).
- Опишіть структуру відповіді HTTP.
- Назвіть групу статус-кодів, HTTP-відповіді.
- Який спосіб надсилання мережевих запитів, окрім методу fetch, ви знаєте?
- Як вказати HTTP-метод під час надсилання запиту через fetch?
- Як скасувати мережевий запит, який запущений за допомогою функції fetch?

Самостійна діяльність учня

Виконайте завдання у директорії Tasks\020 Network Requests. Текст завдань розташований у коментарях, у тегах script.

Рекомендовані ресурси

HTTP-протокол
https://ru.wikipedia.org/wiki/HTTP

HTTP-методи
https://developer.mozilla.org/ru/docs/Web/HTTP/Methods

Коди відповідей HTTP
https://developer.mozilla.org/ru/docs/Web/HTTP/Status

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

Використання Fetch
https://developer.mozilla.org/ru/docs/Web/API/Fetch_API/Using_Fetch

CyberBionic Systematics ®
19 Evgenia Sverstiuka Str., 5 floor
Kyiv, Ukraine

Tel.:0 800 337 146
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com