Запросы к серверу. Fetch API. Формат данных JSON.

Любое веб-приложение или сайт обменивается данными с обслуживающим его веб-сервером. В предыдущих темах мы уже разобрали, что запросы являются асинхронными функциями.

До внедрения промисов в ES2015 для взаимодействия с веб-сервером посредством протокола HTTP использовался (и в настоящее время используется в некоторых приложениях) объект XMLHttpRequest (XHR), в котором события (onload, onerror и т. д.) обрабатываются посредством коллбэков. Мы не будем останавливаться на изучении XHR, так как практически все современные браузеры имеют поддержку более современных методов работы с HTTP-запросами. Опционально с XHR можно ознакомится самостоятельно, например, на ресурсе MDN.

В ES2015 был реализован Fetch API с глобальным методом fetch(), возвращающим промис с объектом Response. Рекомендуется прочитать подробное руководство по Fetch API на ресурсе MDN [1], разобрав все приведённые там примеры.

Желательно также понять хотя бы поверхностно, что такое потоки «streams» в JS, так как тело ответа запроса есть поток чтения «stream». В статьях [2] и [3] даны неплохие объяснения с картинками и примерами.

Вопросы и задания

1. Опираясь на знания, полученные в одной из предыдущих тем – «Асинхронность на уровне движка JS. Цикл событий», какое неочевидное преимущество вы можете указать у Fetch API по сравнению с XHR исходя из факта, что fetch() возвращает промис, а XHR работает только с обработчиками событий на коллбэках?
2. Семантически, метод GET означает получение данных с сервера без изменений его состояния и, возможно, подключенной к нему базы данных. Напротив, метод POST используется в запросах, которые меняют состояние сервера (например, его локальные переменные) и, чаще всего, базу данных. Однако на практике часто реализуются GET-запросы с передачей некоторых данных, отличающиеся ответом сервера в зависимости от этих данных. Приведите пример такого запроса и объясните, почему такие запросы не реализуют через метод POST.
3. За счёт какой технологии методы-парсеры ответа запроса (то есть объекта Response), такие как .json(), .blob(), .text() и т. д., являются асинхронными? Какое преимущество дает эта технология?

Ссылки

1. https://developer.mozilla.org/en-US/docs/Web/API/Fetch\_API/Using\_Fetch
2. <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>
3. <https://developer.mozilla.org/en-US/docs/Web/API/Window/fetch>
4. https://developer.mozilla.org/en-US/docs/Web/API/Streams\_API/Concepts