

AJAX и работа с HTTP протоколом

№ урока: 9 Курс: JavaScript Advanced

Средства обучения: Компьютер с установленной Visual Studio

Обзор, цель и назначение урока

Задача урока – разобрать основные принципы построения приложения с использованием архитектуры AJAX. Научиться создавать внеполосные запросы на сервер, чтобы получать новый контент со стороны сервера не перегружая полностью страницу. Грамотно используя технологию AJAX можно сделать приложение “user friendly” повысить производительность и снизить нагрузку на сервер.

Изучив материал данного занятия, учащийся сможет:

- Выполнять запросы на сервер используя XMLHttpRequest
- Разрабатывать приложения используя архитектуру AJAX

Содержание урока

1. Создание объекта запроса.
2. Отправка синхронного и асинхронного запросов.
3. Отправка POST запроса.
4. Отправка запроса с использованием timeout функции.
5. Пример AJAX приложения.

Резюме

HTTP (Hyper Text Transfer Protocol) – протокол, который определяет как веб браузер должен запрашивать документы, как информация должна передаваться серверам и как веб-сервера должны отвечать на веб-запросы.

В первую очередь с протоколом HTTP работает веб браузер, в тоже время с HTTP протоколом может работать JavaScript сценарий.

HTTP запросы могут инициироваться, когда сценарий устанавливает значение свойства location или вызывает submit объекта form.

Для получения прямого доступа к протоколу HTTP используется объект XMLHttpRequest (поддерживается всеми современными браузерами).

Объект XMLHttpRequest является ключевым элементом архитектуры веб-приложений, известной как Ajax (Asynchronous JavaScript And XML).

Процесс взаимодействия с протоколом HTTP:

- 1) Создание объекта XMLHttpRequest
- 2) Определение и передача HTTP запроса на сервер
- 3) Пример ответа от сервера (может быть синхронным или асинхронным).

XMLHttpRequest поддерживается всеми браузерами.

Свойства XMLHttpRequest:

readyState – состояние HTTP запроса. Во время создания объекта инициализируется значением 0 и меняется в процессе использования объекта.

Состояние	Имя	Описание
0	Uninitialized	Начальное состояние объекта, который только что создан или возвращен в это состояние с помощью метода abort
1	Open	Метод open() уже вызван, но еще не вызван метод send(). Запрос еще не отправлен.

2	Start	Вызван метод <code>send()</code> . HTTP запрос отправлен серверу, но ответ еще не получен.
3	Receiving	Все заголовки ответа уже приняты. Продолжается прием тела ответа, но прием еще не получен.
4	Loaded	Ответ на запрос полностью получен.

responseText – тело ответа не включая заголовки. Если значение свойства `readyState` меньше 3, свойство возвращает ту часть тела, которая доступна на данный момент.

responseXML – ответ на запрос, который интерпретируется как XML документ и возвращается в виде объекта `Document`.

status – HTTP код состояния, например, 200 – в случае успеха, 404 - в случае отсутствия запрашиваемого документа.

statusText – свойство хранит описание HTTP кода ответа.

Методы `XMLHttpRequest`:

abort() – отменяет исполнение текущего запроса, закрывает соединение и прекращает сетевую активность.

getAllResponseHeaders() – возвращает все HTTP-заголовки в виде неразобранной строки.

getResponseHeader() – возвращает заголовок с указанным именем.

open() – инициализирует параметры HTTP запроса.

send() – отправляет HTTP запрос.

setRequestHeader() – устанавливает или добавляет HTTP заголовок.

Обработчики событий `XMLHttpRequest`:

onreadystatechange – срабатывает каждый раз, когда меняется значения свойства `readyState`. В зависимости от браузера, может многократно вызываться при значении `readyState` = 3.

Объект `XMLHttpRequest` может создавать запросы только к серверу, с которого был загружен сценарий. Например, если страница с `XMLHttpRequest` была загружена по адресу <http://edu.cbsystematics.com>, то сценарий этой страницы не сможет делать запросы к ресурсам по адресу <http://testprovider.com>. Сценарий сможет делать запрос только к тем ресурсам, которые находятся в том же домене, что и документ содержащий сценарий.

Закрепление материала

1. Что такое AJAX?
2. Назовите принцип использования объекта `XMLHttpRequest`.
3. Перечислите свойства объекта `XMLHttpRequest`.
4. Какие события есть у объекта `XMLHttpRequest` и как их использовать?
5. Назовите методы объекта `XMLHttpRequest`.
6. Назовите главное ограничение объекта `XMLHttpRequest`.

Дополнительное задание

Создайте две HTML страницы с разным контентом. Создайте страницу, в которой находится сценарий загружающий и отображающий контент двух страниц при нажатии на кнопку. Контент должен загружаться асинхронно.

Самостоятельная деятельность учащегося

Задание 1

Разработайте страницу с двумя полями ввода и кнопкой. Пользователь должен вводить в поля вводов целочисленные значения, а нажатие по кнопке должно приводить к асинхронному запросу на сервер. Запрашивать следует обработчик `CalcHandler.ashx`, который находится вместе с примерами в уроке номер 10, обработчик принимает два `post` параметра с именами 'a' и 'b'. Результат, который возвращает обработчик выведите на страницу.

Задание 2

Выведите курс евро и канадского доллара по отношению к доллару США. В качестве сервера используйте данные из ресурса <https://openrates.io>

Задание выполните двумя способами, с использованием XMLHttpRequest и fetch.

Рекомендуемые ресурсы

<http://ru.wikipedia.org/wiki/AJAX>

<http://javascript.ru/ajax/transport/xmlhttprequest>

http://www.w3schools.com/xml/xml_http.asp

<https://developer.mozilla.org/en-US/docs/DOM/XMLHttpRequest>

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