УДК 004
https://doi.org/10.36906/AP-2020/10

ИСПОЛЬЗОВАНИЕ AJAX-ТЕХНОЛОГИИ ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ WEB-ФОРМ

Игнатов С. В.
Нижневартовский государственный университет
г. Нижневартовск, Россия
Мамедли Р. Э.
канд. физ.-мат. наук,
Нижневартовский государственный университет
г. Нижневартовск, Россия

Аннотация. Настоящая статья рассказывает о современной технологии создания интерактивных WEB-форм AJAX, о том, что она из себя представляет. Описаны примеры запроса на основе AJAX.

Ключевые слова: AJAX, JavaScript, XML, XMLHttpRequest, Fetch.

Технологии, главным образом IT, всегда стремились что-то упростить, сделать удобнее для человека. Особо требовательным к удобству пользователей является WEB-интерфейс. Для его обеспечения используются AJAX-технологии.

Сама аббревиатура AJAX означает Asynchronous Javascript and XML. JavaScript управляет динамическим контентом и взаимодействием с пользователем на WEB-сайте. XML — это расширяемый язык разметки, предназначенный для хранения и переноса данных. Как видно из аббревиатуры, AJAX это концепция использования JavaScript и XML для выполнения, в основном, асинхронных запросов к серверу. Асинхронный метод передачи данных позволяет браузеру продолжать работать во время выполнения запроса. Эта концепция позволяет обращаться к серверу без перезагрузки страницы, что уменьшает время отклика и увеличивает интерактивность интерфейса, а также уменьшает объем передаваемых данных. Примерами, демонстрирующими возможности AJAX, могут послужить поисковая строка в браузере, интерактивное отображение профиля на форуме, чат и т. д. [1-3].

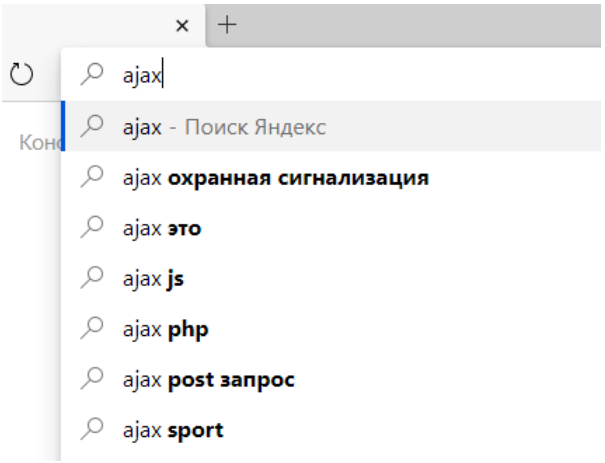

Рис. 1. Поисковая строка в браузере

К основным плюсам технологии, включая названные, относятся:

— Экономия трафика;
— Уменьшение нагрузки на сервер;
— Ускорение реакции интерфейса;
— Возможности для интерактивной обработки;
— Мультимедиа не останавливается.

Из минусов можно выделить:

— Динамически загружаемое содержимое недоступно поисковикам;
— Старые методы учета статистики сайтов становятся неактуальными;
— Плохое поведение на ненадежных соединениях;
— Риск фабрикации запросов другими сайтами.

При сопоставлении представленных плюсов и минусов можно определенно согласиться с тем что положительные стороны перечеркивают недостатки, которые как правило, можно устранить.

В обычной модели WEB-приложения пользователь сначала взаимодействует с элементом интерфейса, затем браузер отправляет запрос серверу и получает новую страницу от сервера, после чего страница перезагружается. При использовании AJAX эта модель слегка меняется: при взаимодействии с элементом страницы, JavaScript определяет, какую часть страницы необходимо обновить и после отправки запроса, сервер возвращает только необходимую часть данных, после чего JavaScript обновляет эту часть страницы. В схематическом представлении это выглядит следующим образом:

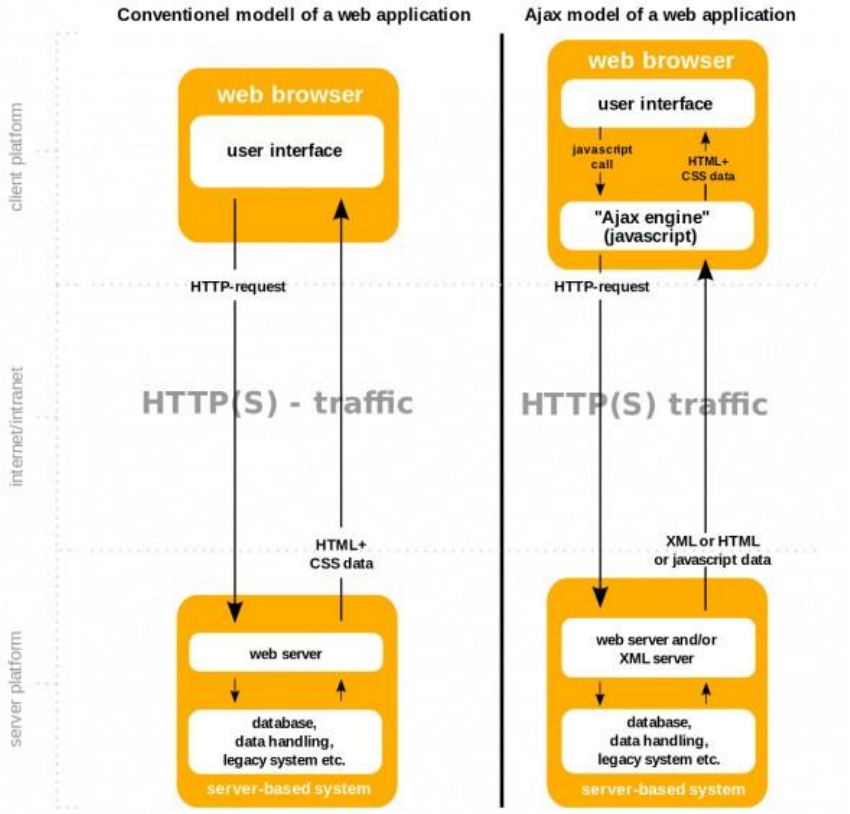

Рис. 2. Модель WEB-приложения

Реализовать AJAX можно разными способами, рассмотрим 2 из них. Например, на странице может быть создан экземпляр объекта XMLHttpRequest, с его помощью можно

отправить запрос на сервер, а также получить ответ в виде различного рода данных. Рассмотрим небольшой пример:

```
1. var request = new XMLHttpRequest();
2. request.open('GET', 'http://www. http://nvsu.ru/', true);
3. request.send(null);
```

В примере в первой строке создается экземпляр класса XMLHttpRequest. Во второй строке “GET” либо “POST” это тип запроса. В первом случае происходит обращение на сервер через URL, во втором в теле запроса в виде .send(body). “true/false” означает асинхронное и синхронное использование соответственно. В третьей строке .send отсылает запрос.

У XMLHttpRequest есть состояния, которые меняются по мере выполнения запроса. Текущее состояние можно посмотреть в свойстве .readyState.

```
1. UNSENT = 0; // исходное состояние;
2. OPENED = 1; // вызван метод open;
3. HEADERS_RECEIVED = 2; // получены заголовки ответа;
4. LOADING = 3; // ответ в процессе передачи (данные частично получены);
5. DONE = 4; // запрос завершен.
```

Аналогом является класс Fetch:

```
const request = fetch(url, {options})
```

где “url” это ссылка для отправки запроса, а “options” это дополнительные опции, без дополнительных указаний это будет “GET” запрос. Для отправки “POST” запроса нужно написать “method: ‘POST’,”. Минусами использования данного класса является пока что ограниченная по сравнению с XMLHttpRequest функциональность, например, fetch не поддерживает события прогресса. Поэтому невозможно сообщить о статусе загрузки файлов или аналогичных представлений больших форм. Помимо этого, fetch не поддерживается старыми браузерами.

Несмотря на название технологии, ответ от сервера может быть не только в виде XML, но и к примеру простым текстом или еще в каком-то другом виде.

Хотя в применении AJAX в целом нет ничего сложного, она, как и любой серьезный инструмент, имеет множество вариаций применения. Технология AJAX получила распространения благодаря компании Google и теперь является популярной технологией, в целом не имеющей аналогов.

Литература

1. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. СПб.: Питер, 2016. 768 с.
2. Дронов В. А. JavaScript и AJAX в Web-дизайне. СПб.: БХВ-Петербург, 2008. 736 с.
3. Пауэрс Ш. Добавляем Ajax: Пер. с англ. СПб.: БХВ-Петербург, 2009. 448 с.