JSON и AJAX.

Лабораторная работа № 8.

ЦЕЛЬ РАБОТЫ

Ознакомление с JSON и AJAX, как с инструментами для получения данных для веб-страницы с сервера и применение их в JavaScript и их выводом

ПРОДОЛЖИТЕЛЬНОСТЬ

2 академических часа (1 занятие)

РЕЗУЛЬТАТ РАБОТЫ

Размещенный на Веб‐сервере и доступный по протоколу *http* документ рабочей версией веб-страницы.

ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ К РАБОТЕ

Результат лабораторной работы – веб-страница на которой представлены результаты работы JavaScript, в виде рабочей веб-страницы поиска фактов о котах.

Данные для страницы должны бать взяты со следующих сайтов:

* API endpoint: <http://cat-facts-api.std-900.ist.mospolytech.ru/facts> – для 1 части задания
* API endpoint: <http://cat-facts-api.std-900.ist.mospolytech.ru/autocomplete> – для 2 части задания

РЕКОМЕНДАЦИИ К СТРУКТУРЕ ПРОГРАММЫ

В данной лабораторной работе вам нужно доработать шаблон (см. файлы) страницы, загружающей записи посредством AJAX. Задание состоит из двух частей.

Часть 1

На странице есть форма для поиска записей. Вам нужно доработать скрипт main.js таким образом, чтобы эта форма заработала. Чтобы осуществить поиск записей по запросу, нужно добавить параметр q к запросу на сервер.

Часть 2

После реализации поиска добавьте на страницу функционал автодополнения запроса пользователя. После начала ввода запроса пользователем ему должны отображаться варианты возможных запросов, которые начинаются с введённых пользователем символов. Например, если пользователь введёт "m", ему могут быть предложены варианты "milk", "month", и т. д. Варианты автодополнения система генерирует автоматически на основе анализа истории запросов пользователей. Чтобы получить варианты автодополнения, необходимо выполнить запрос по указанному ниже адресу и передать в запросе параметр q -- текст поискового запроса, который на данный момент успел ввести пользователь. В ответ система пришлёт массив (в формате JSON) из 10 (или менее) наиболее популярных запросов, соответствующих указанному критерию. Для отображение возможных вариантов пользователю вам необходимо под полем ввода запроса добавить выпадающий список, как, например, это сделано в популярных поисковых системах (Google, Yandex, и др.). Оформление списка остаётся на ваше усмотрение. Пример возможного оформления представлен на картинке ниже. При клике пользователя по одному из предложенных вариантов его значение должно быть подставлено в поле ввода запроса, а выпадающий список должен быть скрыт.

СПРАВОЧНАЯ ИНФОРМАЦИЯ

|  |  |
| --- | --- |
| Работа с JSON | <https://developer.mozilla.org/ru/docs/Learn/JavaScript/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D1%8B/JSON> |
| AJAX | <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX> |
| XMLHttpRequest | <https://learn.javascript.ru/xmlhttprequest> |
| Cross-Origin Resource Sharing (CORS) | <https://developer.mozilla.org/ru/docs/Web/HTTP/CORS> |

КОНТРОЛЬНЫЕ ВОПРОСЫ К ЛАБОРАТОРНОЙ РАБОТЕ

Для успешной защиты работы помимо соответствующего требованиям результата необходимо уверенно отвечать на нижеперечисленные и другие вопросы, а также на контрольные вопросы всех предыдущих лабораторных работ.

1. Что такое JSON? Каковы его особенности?
2. Что такое AJAX? Каковы его особенности?
3. Из чего состоит AJAX?
4. Что такое XMLHttpRequest? В каких моментах используется? Особенности?
5. Что такое Cross-Origin Resource Sharing? Приведите примеры использования.

1. \*\*JSON (JavaScript Object Notation):\*\*

- \*\*Определение:\*\* JSON представляет собой формат обмена данными, основанный на подмножестве синтаксиса JavaScript. Он используется для передачи структурированных данных между программами.

- \*\*Особенности:\*\*

- Простота чтения и записи для человека.

- Легко парсится и генерируется программами.

- Поддерживает массивы и объекты.

- Не зависит от языка программирования, что делает его универсальным для обмена данными между различными языками.

2. \*\*AJAX (Asynchronous JavaScript and XML):\*\*

- \*\*Определение:\*\* AJAX - это подход к созданию интерактивных веб-приложений, позволяющий асинхронно обмениваться данными между браузером и сервером без перезагрузки страницы.

- \*\*Особенности:\*\*

- Асинхронность: Запросы и обработка данных выполняются асинхронно.

- Обновление части страницы: Позволяет обновлять только определенные части страницы, не перезагружая всю страницу.

- Использует технологии, такие как XMLHttpRequest или Fetch API для отправки запросов.

3. \*\*Составляющие AJAX:\*\*

- \*\*HTML и CSS:\*\* Для отображения данных на странице.

- \*\*JavaScript:\*\* Для обработки событий и выполнения асинхронных запросов.

- \*\*XMLHttpRequest или Fetch API:\*\* Для отправки запросов на сервер.

- \*\*Сервер:\*\* Для обработки запросов и возврата данных клиенту.

4. \*\*XMLHttpRequest:\*\*

- \*\*Определение:\*\* Это объект в JavaScript, который позволяет отправлять HTTP-запросы к серверу и получать ответы.

- \*\*Использование:\*\* Используется для асинхронной передачи данных между браузером и сервером без перезагрузки страницы.

- \*\*Особенности:\*\*

- Асинхронность: Запросы выполняются асинхронно, что позволяет избежать блокировки интерфейса.

- Обновление части страницы: Позволяет обновлять только необходимую часть страницы после получения данных от сервера.

5. \*\*CORS (Cross-Origin Resource Sharing):\*\*

- \*\*Определение:\*\* CORS - это механизм, позволяющий веб-странице запрашивать ресурсы с домена, отличного от того, с которого загружена страница.

- \*\*Примеры использования:\*\*

- Если веб-страница, загруженная из домена A, хочет сделать AJAX-запрос к домену B, то сервер на домене B должен включить заголовок `Access-Control-Allow-Origin` с доменом A.

- Пример заголовка в ответе сервера: `Access-Control-Allow-Origin: http://домен\_A.com`

- Это предотвращает запросы с других доменов без явного разрешения.

**В качестве ответа укажите ссылку на git-репозиторий с кодом сайта, а также ссылку на сайт, размещённый на хостинге.**