

Реализация живого поиска с использованием технологии AJAX

Брыкова Анастасия Леонидовна

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В статье разобран пример реализации живого поиска по БД. Он был реализован с помощью технологии AJAX.

Ключевые слова: AJAX; живой поиск, JQuery, HTML.

Implementation of live search using AJAX technology

Brykova Anastasia Leonidovna

Sholom-Aleichem Priamursky State University

Student

Abstract

In article, an example of realization of live search on a BD is disassembled. It was implemented using AJAX technology.

Keywords: AJAX; live search, JQuery, HTML.

Живой поиск – это поиск, который позволяет просматривать результаты непосредственно при вводе запроса.

Преимущества:

- 1) В Живом поиске результаты выводятся непосредственно при вводе, что позволяет экономить от 2 до 5 секунд при каждом запросе.
- 2) Даже если клиент не уверен в том, что хочет найти, система сама попытается выдать или дополнить запрос.
- 3) Возможность мгновенного переключения. Когда найденный результат удовлетворяет, можно перейти мгновенно, не нажимая ENTER.

В настоящее время тема актуальна и исследования в данной области интересуют большое количество людей. В статье Власов В.В., Крылов Б.А. [1] описаны решения и технологии, которые могут помочь разработчику интерфейсов WEB-приложений создавать мощные высокоскоростные инструменты взаимодействия между клиентом (браузером) и сервером. Зуев М.С., Бозюков А.Н. [2] рассматривают вопрос создания системы управления контентом, предназначенной для использования в учебном процессе в современных университетах. Обосновывается применение Web-технологий, технологии AJAX, защиты контента от несанкционированного доступа. Формулируются требования к разрабатываемой системе управления контентом. Зуев М.С., Пелихосов А.А. [3] разрабатывают защищенное web-

приложение на основе технологии AJAX. В статье Медведев Ю.С., Терехов В.В. [4] рассматривается достижение максимальной производительности аякс-приложений.

Для того чтобы использовать технологию AJAX, нужно подключить её к сайту с помощью кода (рис.1).

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
```

Рисунок 1. Подключение AJAX

Далее необходимо прописать структуру HTML документа в котором должна быть форма для ввода текста и блок для вывода результатов (рис.2).

```
<input type="text" name="search" placeholder="Поиск по сайту" class="search" autocomplete="off">
<div class="search_row"></div>
```

Рисунок 2. Разметка HTML документа

При вводе должен срабатывать скрипт на считывание этих букв. (рис.3).

```
$(function(){
    $('.search').bind("change keyup input click", function() {
        $.ajax({
            type: 'POST',
            url: "search.php",
            data: {'search':this.value},
            response: 'text',
            success: function(x){
                $(".search_row").html(x).fadeIn();
            }
        })
    })
    $(".search_row").hover(function(){
        $(".search").blur();
    })
    $(".search_row").on("click", "li", function(){
        s_user = $(this).text();
        $(".search_row").fadeOut();
    })
})
```

Рисунок 3. Обработчик AJAX

В приведённом коде сначала срабатывает функция bind, которая отвечает за любые изменения, происходящие в форме (ввод, изменение, щелчок и отпускание клавиш). Далее при вводе хотя бы 1 буквы запрос отправляется на обработку в PHP файл через AJAX. Структура запроса достаточно проста:

- 1) type - тип передачи(может быть GET или POST)
- 2) url - путь к файлу-обработчику.
- 3) data – 1 атрибут это название переменной, 2 атрибут это сама передаваемая переменная.
- 4) response – ответ запроса (может быть как success, так и error)

Теперь необходимо обработать принятые в обработчик переменные. (рис.4).

```
<?php
$mysqli = new mysqli('localhost','root','','search');
if(!empty($_POST["search"]))
{
    $search_row = trim(htmlspecialchars($_POST["search"]));
    $sql = $mysqli -> query("SELECT * FROM `post` WHERE `name` LIKE '%$search_row%'");
    while ($row =mysqli_fetch_array($sql))
    {
        echo "<a href='#' class='href'>".$row['name']."</a><br>";
    }
}
?>
```

Рисунок 4. Обработчик AJAX

Живой поиск происходит не по элементам, находящимся по сайту, а по базе данных. Поэтому сначала идёт подключение к БД, а потом идет проверка на пустоту принятой переменной. Если переменная не пуста, то убираются все пробелы, не нужные символы с неё и идет SQL запрос. Самое важное в нём это функция LIKE, которая начиная с первого символа ищет совпадения и выдаёт результат. Для более понятного представления выводимых элементов поиска можно добавить CSS. (рис.5).

```
.search_row{
    background: #FFF;
    border: 1px #000000 solid;
    width: 100px;
    border-radius: 5px;
    max-height:100px;
}
.href
{
text-decoration-line: none;
}
```

Рисунок 5. Стилизация

Идея заключалась в том, что живой поиск используется для перемещения пользователя по сайту (в данном примере по постам новостей) (рис.6).

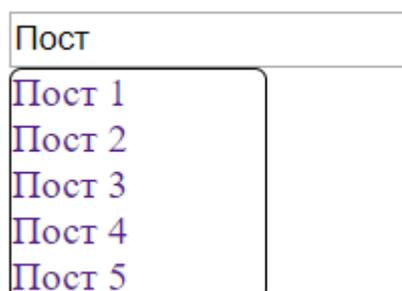


Рисунок 6. Обработчик AJAX

При вводе слова выдаются все возможные варианты. Живой поиск помогает при огромной нагрузке, когда искать вручную почти невозможно, листая десятки различных страниц с постами, чтобы найти тот, который нужен. Именно для этого и существует удобный способ поиска по сайту, который был реализован в данной статье.

Библиографический список

1. Власов В.В., Крылов Б.А. Расширение возможностей интерактивных пользовательских интерфейсов WEB-приложений с помощью технологии AJAX//Научно-технический вестник информационных технологий, механики и оптики. 2007. № 40. С. 173-181.
2. Зуев М.С., Бозюков А.Н. Разработка защищённой системы управления контентом, основанной на применении технологии AJAX// Естественные и технические науки. 2010. Т. 15. № 1. С. 260-261.
3. Зуев М.С., Пелихосов А.А. Электронное учебное пособие «разработка защищенного WEB-приложения на основе технологии ajax//Психолого-педагогический журнал Гаудеамус. 2011. Т. 2. № 18. С. 37-38.
4. Медведев Ю.С., Терехов В.В. К вопросу о достижении максимальной производительности ajax-приложений//Естественно-математические и технические науки. 2013. № 4 (125). С. 125-128.