

## Применение AJAX технологии на веб сайты

*Кочитов Михаил Евгеньевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*студент*

### Аннотация

В данной статье рассматривается применение AJAX технологии на множество веб-сайтов. Эта технология позволяет обрабатывать сервером запросы пользователей сайтов большое количество раз, при этом не обновляя веб-страницу. Код AJAX рассмотрен подробнее на JavaScript, а код обработки запросов сервером - на PHP. В этой статье продемонстрирован собственный пример с регистрацией, используя технологию AJAX.

**Ключевые слова:** AJAX, технология, JavaScript, PHP, сервер, запрос, веб сайт

## Applying AJAX technology to Web sites

*Kochitov Mikhail Evgenevich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article discusses the use of AJAX technology on a variety of web sites. This technology allows the server to process requests for site users a large number of times, without updating the web page. The AJAX code is discussed in more detail in JavaScript, and the server processing code is in PHP. In this article, you will demonstrate your own example with registration using AJAX technology.

**Keywords:** AJAX, technology, JavaScript, PHP, server, query, web site

В настоящий момент большинство сайтов имеет форму регистрации и авторизации, вбивая каждый раз неправильные данные можно заметить, что вылезает конкретная ошибка и при том без перезагрузки самой веб-страницы, тоже самое и с авторизацией, если пользователь ввел неверно логин или пароль он также получит эту ошибку без какой-либо перезагрузки веб-страницы. Каждый раз, когда пользователь отправляет свои данные для регистрации или авторизации на сервер, то обработка этих данных происходит незаметно для самого пользователя, так как визуальное содержимое веб-страницы сайта не очистится, а останется таким какое есть, только с добавлением новых элементов (надписей, текста или изображений), иными словами данные пользователя отправляемыми для обработки на сервер не позволяют обновлять веб-страницу браузера, с которого сейчас сидит сам пользователь. Такая технология, которая позволяет обращаться к

серверу без обновления веб-страницы получила название AJAX (Asynchronous JavaScript And XML «А синхронизированный JavaScript и XML»). Эта технология дала удобство большинством сайтов, которыми пользуются пользователи, посещая их каждый раз и переходя на различные страницы самого сайта. Все формы регистрации и авторизации популярных сайтов и социальных сетей уже адаптированы под AJAX технологию, а также и другие функции, с которыми надо непосредственно обращаться к серверу не обновляя страницу. Благодаря AJAX взаимосвязи между клиентом (браузером) и сервером значительно улучшились, так как разработаны высокоскоростные инструменты, чтобы отправлять и получать информацию можно было гораздо быстрее, нежели без AJAX с обновлением страницы.

Теперь рассмотрим исследования других авторов, которые занимались AJAX технологией. В статье В.В. Власова и Б.А.Крылова описаны решения, которые помогут разработчику интерфейсов веб-приложений создать мощные высокоскоростные инструменты между клиентом (браузером) и сервером, применяя при этом технологию AJAX [1]. В.Н.Зимин и П.В.Федосов в своей статье рассмотрели использование технологии AJAX на веб-приложения в среде Microsoft Visual Studio [2]. Рассматривая статью А.А.Телегина, можно заметить, что AJAX технологии можно использовать в образовании [3]. В статье М.С.Зуева и А.Н.Бозюкова рассматривается вопрос создания системы управления контентом, предназначенной для использования в учебном процессе в современных университетах с помощью AJAX [4].

Далее приступим к созданию формы регистрации на странице, где применим AJAX технологию на отправку введенных данных на сервер и обработку их, а также получения ответа от сервера.

Для начала создадим HTML страницу и введем код, представленный на рисунке ниже:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>AJAX-технология</title>
    <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/2.0.3/jquery.min.js"></script>
```

Рис. 1. Загрузка библиотеки JQuery

На рисунке 1 представлен код HTML страницы, где загружаем библиотеку JQuery, в которой внедрена та самая AJAX технология. Эту библиотеку на собственный веб-проект, сайт или приложение стоит загружать последней версии. Сама библиотека представлена в виде огромного Java скрипта.

```

function registration(ajax_form, url) {
    jQuery.ajax({
        url: url,
        type: "POST",
        dataType: "html",
        data: jQuery("#"+ajax_form).serialize(),
        success: function(response) {
            result = jQuery.parseJSON(response);
            if (result.error=="notLogin")
            {
                document.getElementById("report").innerHTML="Введите логин!";
            }
            if (result.error=="notPass")
            {
                document.getElementById("report").innerHTML="Введите пароль!";
            }

            if (result.error=="lessPass")
            {
                document.getElementById("report").innerHTML="Слишком маленький пароль!";
            }

            if (result.error=="notMatchPass")
            {
                document.getElementById("report").innerHTML="Пароли не совпадают!";
            }

            if (result.error=="noErrors")
            {
                document.getElementById("report").innerHTML="Пользователь успешно зарегистрирован!";
            }
        },
        error: function(response) { // Данные не отправлены
        }
    });
}

```

Рис. 2. Обращение к серверу с помощью AJAX, используя JQuery библиотеку

Как видим на рисунке 2 изображен сам код, как отправлять данные на сервер с помощью AJAX, используя загруженную библиотеку JQuery. Функция Registration отправляет два аргумента это: registryData – id формы регистрации, в которой пишутся данные пользователем, url – путь к самому серверу (PHP файлу). После отправки данных на сервер, от него должен прийти ответ с конкретной ошибкой, которую допустил пользователь в форме регистрации. Функция «jQuery.parse.JSON» позволяет ответ от сервера в виде JSON структуры разбить на ассоциативный массив, в котором можно легко обнаружить конкретную ошибку по индексу массива. Сами ошибки будут отображаться под формой регистрацией. Виды ошибок можно рассмотреть на самом рисунке 2.

```

</script>
</head>
<body>
<h4>Форма регистрации</h4>
<form action="" method=post id=registryData>
    Логин: <input type=text name=login><br>
    Пароль: <input type=password name=pass><br>
    Подтверждение пароля: <input type=password name=confPass><br>
    <input type=button value="Отправить" onclick="registration('registryData', 'test_ajax.php')">
</form>
<div id=report></div>
</body>
</html>

```

Рис. 3. HTML форма регистрации

Глядя на рисунок 3 можно заметить, что на нем изображен HTML код формы регистрации, в которой включены элементы текстового поля, где надо вводить логин пароль, подтверждение пароля и кнопка «Отправить», щелкая по ней собственно, и отправляет введенные данные в текстовых полях на сервер, обращаясь к AJAX.

```
<?php
$login = $_POST['login'];
$password = $_POST['pass'];
$password2 = $_POST['confPass'];
if (($login) and ($password))
{
    $len = strlen($password);
    if ($len >= 5)
    {
        if ($password==$password2)
        {
            $result = array(
                'error' => "noErrors"
            );
            echo json_encode($result);
            exit;
        }
        else
        {
            $result = array(
                'error' => "notMatchPass"
            );
            echo json_encode($result);
        }
    }
    else
    {
        $result = array(
            'error' => "lessPass"
        );
        echo json_encode($result);
    }
}
}
```

Рис. 4. Код обработки данных сервером на PHP, в случае, если логин и пароль были введены

На рисунке 4 рассматривается код уже самого сервера, который обрабатывает полученные регистрационные данные пользователя. Можно заметить, что в начале объявляются переменные, к которым присваиваются полученные данные через метод POST с формы регистрации. Далее по этим переменным идет уже проверка на ввод этих данных, что если логин и пароль были введены, то идет проверка на длину пароля, а также на совпадение двух одинаковых паролей, в случае если эти ошибки были проигнорированы, то сервер дает ответ, что данные от пользователя были правильными и этот логин был зарегистрирован.

```
else
{
    if (!$login)
    {
        $result = array(
            'error' => "notLogin"
        );
        echo json_encode($result);
        exit;
    }

    if (!$password)
    {
        $result = array(
            'error' => "notPass"
        );
        echo json_encode($result);
        exit;
    }
}
```

Рис. 5. Код обработки данных сервером на PHP, в случае, если логин или пароль не были введены

Как можно заметить на рисунке 5 изображен код сервера, если пользователь случайно отправил не введенный логин или пароль, ответ от сервера придет пользователю с конкретной ошибкой (с не введенным логином или паролем). При отправке ответа обратно к пользователю сервер использует функцию «`json_encode`», которая преобразует ассоциативный массив в саму JSON структуру, чтобы ее можно было отправить обратно к пользователю с применением AJAX для преобразования JSON обратно в ассоциативный массив с конкретной ошибкой.

Теперь проверим работоспособность AJAX технологии в браузере, которую использовали в собственной форме регистрации. Все результаты (ответы от сервера в виде ошибок или успеха), представлены на рисунке ниже:

#### Форма регистрации

Логин:

Пароль:

Подтверждение пароля:

Введите логин!

Рис. 6. Не введен логин

**Форма регистрации**

Логин:   
Пароль:   
Подтверждение пароля:

Введите пароль!

Рис. 6. Не введен пароль

**Форма регистрации**

Логин:   
Пароль:   
Подтверждение пароля:

Слишком маленький пароль!

Рис. 7. Введен маленький пароль

**Форма регистрации**

Логин:   
Пароль:   
Подтверждение пароля:

Пароли не совпадают!

Рис. 7. Введенные пароли не совпадают

**Форма регистрации**

Логин:   
Пароль:   
Подтверждение пароля:

Пользователь успешно зарегистрирован!

Рис. 8. Успешная регистрация

Таким образом, использование AJAX технологии дает большое преимущество и возможности над разработкой собственного интерактивного интерфейса в веб приложениях, на веб сайте и т.д. AJAX дает разработать инструменты для быстрой передачи и получения информации между клиентом (браузером) и сервером, тем самым не позволяя у клиента (браузера) обновить страницу. Также был рассмотрен собственный пример с

формой регистрации, применяя технологию AJAX, что после отправки регистрационных данных клиент (браузер) не позволил обновить страницу, а сразу же получил конкретный ответ от сервера.

### **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. Зимин В.Н., Федосов П.В. Разработка веб-приложений на основе технологии AJAX в среде Microsoft Visual Studio // Научно-технический вестник информационных технологий, механики и оптики. 2008. № 51. С. 103-107.
3. Телегин А.А. Использование AJAX в образовании // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2006. № 7. С. 160-161.
4. Зуев М.С., Бозюков А.Н. Разработка защищённой системы управления контентом, основанной на применении технологии AJAX // Вестник Тамбовского университета. Серия: Естественные и технические науки. 2010. Т. 15. № 1. С. 260-261.