無料電子ブック

学習

ajax

Free unaffiliated eBook created from
Stack Overflow contributors.

#ajax

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: ajax

It is an unofficial and free ajax ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ajax.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: ajaxをいめる

AJAX**J** avaScript ND **X** MLは、コードのをブロックすることなく、デ―タをすることをにします。くの、これは、XMLhttpRequestsをしてサ―バ―からのペ―ジまたはのをし、にjavascriptをしてそれをおよびすることでされます。

AJAXのノンブロッキングは、それがそのようなソフトウェアパタ―ンになるです。 javascriptはブラウザでブロックされているため、びしは、デ―タがされるかタイムアウトになるまで、びしのしなくなります。には、アプリケ―ションをア―キテクチャににさせ、アプリケ―ションがどれだけうまくするかをします。

AJAXびしは、、のやをするためにされていますが、は XMLHttpRequest についています。

Examples

インストールまたはセットアップ

AJAXとはですか

> AJAXはAsynchronous JavaScript and XMLのです。にえば、サ―バ―サイドスクリプトとするのはXMLHttpRequestオブジェクトのです。 JSON、XML、HTML、さらにはテキストファイルなど、さまざまなでをすることができます。 -Mozillaネットワ―ク2016

AJAXをするもなは、にサ―バ―とのをしているは、jQueryをすることです。

jQueryとはですか

> jQueryは、でがしたJavaScriptライブラリです。 HTMLドキュメントのトラバ―スや、イベント、アニメ―ション、Ajaxなどのを、のブラウザでするいやすいAPIをすると、はるかににすることができます。 -jquery.com

くのjQueryをしていないにとって、たちのをにするためにできるとえることができます。これは、じことをするためにかなければならないコ―ドのをするAJAXでのにです

jQueryをWebサイトにする

Ajaxをするがあるは、プロジェクトにjQueryをするがあります。 http://jquery.com/download/このリンクには、jqueryをするさまざまながあります。 jQueryのダウンロ―ドをすることも、CDNをすることもできます。 http://jquery.com/download/#jquery-39-s-cdn-provided-by-maxcdn 。しか

し、CDNをするとセキュリティのリスクがあります。プロジェクトがjqueryをするようにびすと、ハッカーがそのびしをできるようになります。あなたがダウンロードしたバージョンをうことができればもっといです。 HTMLプロジェクトにjqueryをするをてみましょう。それはです。のは、ダウンロードしたソースをすることです。 http://jquery.com/download/#jqueryダウンロードへのリンクをしてください。 jqueryをいたいだけなら、ダウンロードすることをおめします。 された**production jQuery 3.1.1**をダウンロードしてください。jquery-version.min.jsをなプロジェクトのjavascriptフォルダなどにしてください。のようにsrc = jquery / locationをします。

```
<head>

<script src="path/from/html/page/to/jquery.min.js"></script>

</head>
```

CDNのいをてみましょう。このリンクhttp://jquery.com/download/#using-jquery-with-a-cdnあなたは々なCDNコンテンツネットワークをることができます。

```
<head>

<script   src="https://code.jquery.com/jquery-3.1.1.min.js"   integrity="sha256-
hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="   crossorigin="anonymous"></script>

</head>
```

ごのとおり、CDNプロバイダがするタグをするだけです。 htmlページにいくつかのスクリプトをして、していることをします。

```
<script>
$(document).ready(function(){
    alert("jQuery Works")
});
</script>
```

jQueryがアラートをするは、しくしたことをします。

サーバーとするためのな**jQuery**の

`jQuery.ajax` APIの Webサイトから`jQuery.ajax`

```
$.ajax({
    method: "POST",
    url: "some.php",
    data: {
        name: "John",
        location: "Boston"
    },
    success: function(msg) {
        alert("Data Saved: " + msg);
    },
    error: function(e) {
        alert("Error: " + e);
```

```
    }
});
```

このコードは、jQueryのためにみやすく、がこっているのかをするのがです。

- `$.ajax` - このビットはjQueryの`ajax`をびします。
- `method: "POST"` - このは、POSTメソッドをしてサーバーとすることをしています。リクエストのについておみください
- `url` -をするためにされようとしているところ、このをします。あなたはどこかにリクエストをっています。それがアイデアです。
- `data` - かなりです。これはあなたのリクエストでするデータです。
- `success` -あなたがってデータをどのようにするかをするためにくここにこの`msg` こののように、は`msg`がされるアラートをしています。
- `error` - エラーメッセージをするための、または`ajax`リクエストがエラーをこしたときにするアクションをするためのです。
- `.done`わりに

```
success: function(result) {
    // do something
});
```

- ajax リクエスト

の**ajax**びし

このタイプのajaxびしでは、コードはびしがするまでしません。

```
$('form.ajaxSubmit').on('submit',function(){
    // initilization...
    var form   = $(this);
    var formUrl    = form.attr('action');
    var formType   = form.attr('method');
    var formData   = form.serialize();

    $.ajax({
        url: formUrl,
        type: formType,
        data: formData,
        async: true,
        success: function(data) {
            // .....
        }
    });
    //// code flows through without waiting for the call to complete
    return false;
});
```

Ajaxびし

このタイプのajaxびしでは、コーードはびしがするのをちます。

```
$('form.ajaxSubmit').on('submit',function(){
    // initilization...
    var form   = $(this);
    var formUrl    = form.attr('action');
    var formType   = form.attr('method');
    var formData   = form.serialize();
    var data = $.ajax({
            url: formUrl,
            type: formType,
            data: formData,
            async: false
        }).responseText;
    //// waits for call to complete
    return false;
});
```

XMLHttpRequestオブジェクトをしてされたなAjaxリクエスト

```
var httpRequest = new XMLHttpRequest();

httpRequest.onreadystatechange = getData;

httpRequest.open('GET', 'https://url/to/some.file', true);
httpRequest.send();

function getData(){
    if (httpRequest.readyState === XMLHttpRequest.DONE) {
        alert(httpRequest.responseText);
    }
}
```

`new XMLHttpRequest()`はしい*XMLHttpRequest*オブジェクトをします。これはリクエストをするものです

`onreadystatechange`ビットは、ステーータスがする`getData()`をびす`getData()`リクエストを`getData()`ます

`.open()`はリクエストをします。リクエストメソッド'**GET**'、'**POST**'など、クエリしているペーージのURL、オプションでリクエストがであるかどうか

`.send()`リクエストをします。これはオプションで`.send(data)`ようなサーバーにするデーータをけれます。

に、`getData()`はリクエストのステーータス がするたびにびされるべきです。 readyStateが**DONE**にしいは、サーバーからしたデーータだけである`responseText`にします。

は、MDNのGetting Started ガイドをしてください 。

なコールバックで、バニラの**javascript**で**ajax**をする

ここでは、vanilla javascriptes2015ではなくでかれたなajaxびしをするをします。

```
function ajax(url, callback) {
    var xhr;

    if(typeof XMLHttpRequest !== 'undefined') xhr = new XMLHttpRequest();
    else {
        var versions = ["MSXML2.XmlHttp.5.0",
                        "MSXML2.XmlHttp.4.0",
                        "MSXML2.XmlHttp.3.0",
                        "MSXML2.XmlHttp.2.0",
                        "Microsoft.XmlHttp"]

         for(var i = 0, len = versions.length; i < len; i++) {
            try {
                xhr = new ActiveXObject(versions[i]);
                break;
            }
            catch(e){}
         } // end for
    }

    xhr.onreadystatechange = ensureReadiness;

    function ensureReadiness() {
        if(xhr.readyState < 4) {
            return;
        }

        if(xhr.status !== 200) {
            return;
        }

        // all is well
        if(xhr.readyState === 4) {
            callback(xhr);
        }
    }

    xhr.open('GET', url, true);
    xhr.send('');
}
```

のようにできます。

```
ajax('myFile.html', function(response) {
    document.getElementById('container').innerHTML = response.responseText;
});
```

Ecmascript 6es2015ともばれますをするは、 **fetch**メソッドをして、をします。

```
fetch('myFile.json').then(function(res){
    return res.json();
 });
```

es2015についてのfurthnerのについては、のリンクにってください

TypeScriptをしたAJAXびしの

ショッピングカートにをする

のは、AJAXとTypeScriptをして、にデータベステーブルにまたはかをするをしています。

```
declare var document;
declare var xhr: XMLHttpRequest;

window.onLoad = () =>
{
    Start();
};

function Start()
{
    // Setup XMLHttpRequest (xhr).
    if(XMLHttpRequest)
    {
        xhr = new XMLHttpRequest();
    }
    else
    {
        xhr = new ActiveXObject("Microsoft.XMLHTTP");
    }

    AttachEventListener(document.body, "click", HandleCheckBoxStateChange);
}

function AttachEventListener(element: any, e, f)
{
    // W3C Event Model.
    if(element.addEventListener)
    {
        element.addEventListener(e, f, false);
    }
    else if(element.attachEvent)
    {
        element.attachEvent("on" + e, (function(element, f)
        {
            return function()
            {
                f.call(element, window.event);
            };
        })
        (element, f));
    }

    element = null;
}

function HandleCheckBoxStateChange(e)
{
    var element = e.target || e.srcElement;
```

```
    if(element && element.type == "checkbox")
    {
        if(element.checked)
        {
            AddProductToCart(element);
        }
        else
        {
            // It is un-checked.
            // Remove item from cart.
        }
    }
    else
    {
        break;
    }
}

AddProductToCart(e)
{
    var element = <HTMLInputElement>document.getElementById(e.id);

    // Add the product to the Cart (Database table)
    xhr.onreadystatechange = function()
    {
        if(xhr.readyState == 4)
        {
            if(xhr.status == 200)
            {
                if(element != null)
                {
                    console.log("200: OK");
                }
                else
                {
                    console.log(":-(");
                }
            }
            else
            {
                // The server responded with a different response code; handle accordingly.
                // Probably not the most informative output.
                console.log(":-(");
            }
        }
    }

    var parameters = "ProductID=" + encodeURIComponent(e.id) + "&" + "Action=Add&Quantity=" +
element.value;

    xhr.open("POST", "../Cart.cshtml");
    xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhr.setRequestHeader("Content-length", parameters.length.toString());
    xhr.setRequestHeader("Connection", "close");

    xhr.send(parameters);

    return e.id;
}
```

このをさせるには、にこのデータをデータベースにするようにサーバーのコードをしてください

。のコードでは、Cをしていて、 `Cart.cshtml`ファイルがあると`Cart.cshtml`ます。しかし、に`cshtml`を`php`にきえて、あなたがんだをしてのサーバーロジックをしてください。

オンラインでajaxをいめるをむ https://riptutorial.com/ja/ajax/topic/1082/ajaxをいめる

2: コールバック

Examples

「エラー」コールバックによるエラーの

サーバーによってにされると、エラーは2xxとはなるのHTTPステータスコード RFC 2616セクション10をでクライアントにされます。

のにすように、 `$.ajaxSetup()` からエラーをグローバルにすることをおめします。したがって、あなたのajaxびしからるすべてのエラーは、ajaxからにされます。

```
$.ajaxSetup({
    error: function (jqXHR, exception, errorThrown) {
        var message;
        var statusErrorMap = {
            '400': "Server understood the request, but request content was invalid.",
            '401': "Unauthorized access.",
            '403': "Forbidden resource can't be accessed.",
            '500': "Internal server error.",
            '503': "Service unavailable."
        };
        if (jqXHR.status) {
            message = statusErrorMap[jqXHR.status];
            if (!message) {
                message = "Unknown Error.";
            }
        } else if (exception == 'parsererror') {
            message = "Error.\nParsing JSON Request failed.";
        } else if (exception == 'timeout') {
            message = "Request Time out.";
        } else if (exception == 'abort') {
            message = "Request was aborted by the server";
        } else {
            message = "Unknown Error.";
        }

        // How you will display your error message...
        console.log(message);
        console.log(errorThrown);
    }
});
```

また、のエラーメッセージをっているときに、の`$.ajax()` `error`コールバックを「オーバーロード」することもできます。

```
$.ajax({
    url: './api',
    data: { parametersObject },
    type:'post',
    dataType: 'json',
    success:function(output){
        // Interpret success
```

```
    },
    error: function(xhr,textStatus,ErrorThrown){
        // Specific error will not be interpreted by $.ajaxSetup
    }
});
```

オンラインでコールバックをむ https://riptutorial.com/ja/ajax/topic/7175/コールバック

クレジット

S. No		Contributors
1	ajaxをいめる	acupajoe, adielhercules, Alessio Cantarella, Community, delete me, JhWebDevGuy, jignesh prajapati, MAZux, Menuka Ishan, Nicholas Qiao, TimTheEnchanter, Tolga Evcimen
2	コールバック	maxime_039