

Comment envoyer des objets FormData avec des requêtes Ajax en jQuery? [dupliquer]



cette question a déjà une réponse ici:

- envoi de multipart / formdata avec jQuery ([https://over.wiki/jquery/"\).ajax](https://over.wiki/jquery/) (<https://over.wiki/ajax/>) 11 réponses

la norme XMLHttpRequest Level 2 (qui est toujours un brouillon de travail) définit l'interface `FormData`. Cette interface permet d'ajouter `File` aux objets XHR-requests (Ajax-requests).

Btw, c'est une nouvelle fonctionnalité dans le passé, le "caché-iframe-trick" (lire à ce sujet dans mon autre question).

C'est comme ça que ça marche (exemple):

```
var xhr = new XMLHttpRequest(),
    fd = new FormData();

fd.append( 'file', input.files[0] );
xhr.open( 'POST', 'http://example.com/script.php', true );
xhr.onreadystatechange = handler;
xhr.send( fd );
```

où `input` est un champ `<input type="file">`, et `handler` est le gestionnaire de succès pour la demande Ajax.

cela fonctionne magnifiquement dans tous les navigateurs (encore une fois, sauf IE).

maintenant, je voudrais faire fonctionner cette fonctionnalité avec jQuery. J'ai essayé ceci:

```
var fd = new FormData();
fd.append( 'file', input.files[0] );

$.post( 'http://example.com/script.php', fd, handler );
```

malheureusement, cela ne marchera pas (une erreur d' "invocation illégale" est lancée - screenshot est ici). Je suppose que jQuery s'attend à ce qu'un simple objet key-value représentant form-field-names / values, et l'instance `FormData` que je suis en train de passer soit apparemment incompatible.

Maintenant, puisqu'il est possible de passer d'un `FormData` instance `xhr.send()`, j'espère qu'il est aussi possible de la faire fonctionner avec jQuery.

mise à jour:

j'ai créé un "ticket de fonction" au Bug Tracker de jQuery. C'est ici: <http://bugs.jquery.com/ticket/9995>

on m'a suggéré D'utiliser un "Ajax préfiltre"...

mise à jour:

tout d'abord, permettez-moi de donner une démo démontrant quel comportement je voudrais atteindre.

HTML:

```
<form>
  <input type="file" id="file" name="file">
  <input type="submit">
</form>
```

JavaScript (<https://over.wiki/javascript/>):

```
$( 'form' ).submit(function ( e ) {
    var data, xhr;

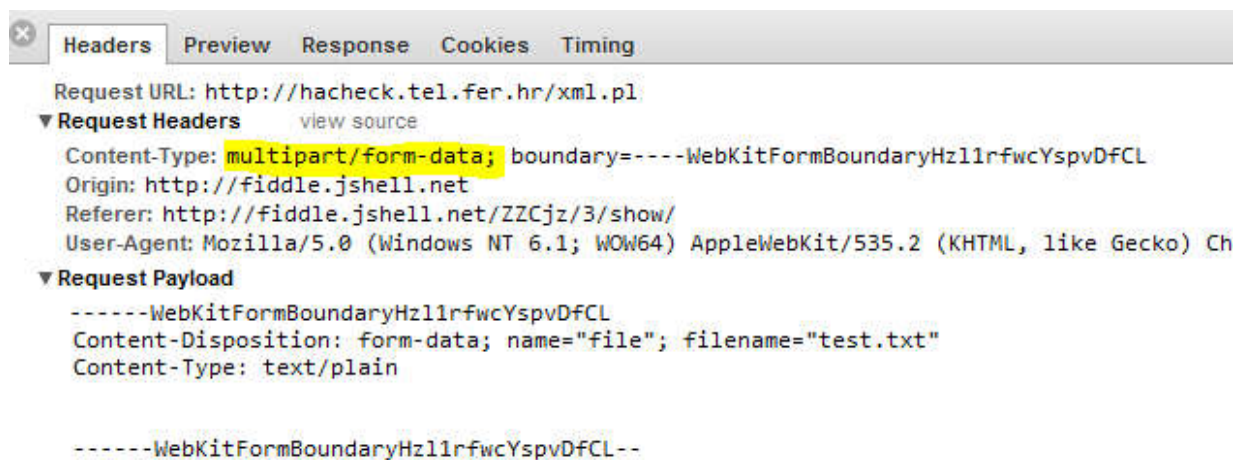
    data = new FormData();
    data.append( 'file', $( '#file' )[0].files[0] );

    xhr = new XMLHttpRequest();

    xhr.open( 'POST', 'http://hacheck.tel.fer.hr/xml.pl', true );
    xhr.onreadystatechange = function ( response ) {};
    xhr.send( data );

    e.preventDefault();
});
```

Le code ci-dessus résulte en cette requête HTTP:



C'est ce dont j'ai besoin - je veux ce type de contenu "multipart/form-data"!

la solution proposée serait la suivante:

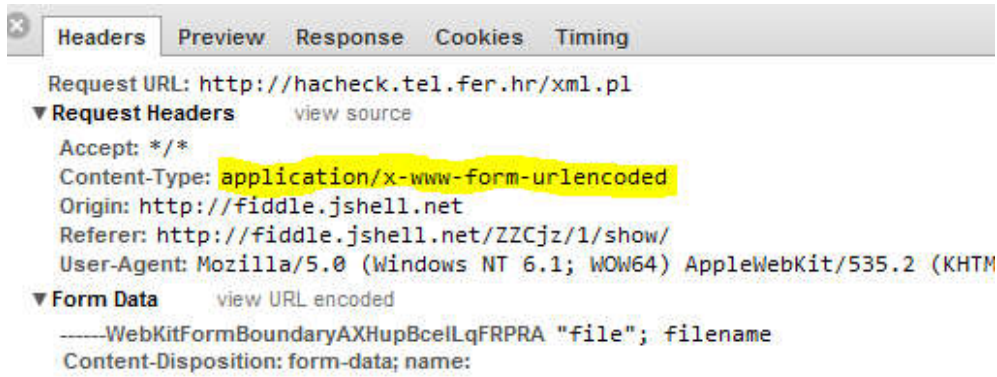
```
$( 'form' ).submit(function ( e ) {
    var data;

    data = new FormData();
    data.append( 'file', $( '#file' )[0].files[0] );

    $.ajax({
        url: 'http://hacheck.tel.fer.hr/xml.pl',
        data: data,
        processData: false,
        type: 'POST',
        success: function ( data ) {
            alert( data );
        }
    });

    e.preventDefault();
});
```

Cependant, il en résulte:



Comme vous pouvez le voir, le type de contenu est faux...

399

[ajax](https://webdevdesigner.com/tags/ajax/) (<https://webdevdesigner.com/tags/ajax/>) [html5](https://webdevdesigner.com/tags/html5/) (<https://webdevdesigner.com/tags/html5/>)

[javascript](https://webdevdesigner.com/tags/javascript/) (<https://webdevdesigner.com/tags/javascript/>) [jquery](https://webdevdesigner.com/tags/jquery/) (<https://webdevdesigner.com/tags/jquery/>)

[multipartform-data](https://webdevdesigner.com/agent_category/browse/id:3422/) (https://webdevdesigner.com/agent_category/browse/id:3422/)

demandé sur Community (<https://stackoverflow.com/users/-1/community>) 2011-08-07 22:06:39



10 réponses

je crois que vous pourriez faire comme ceci :

```
var fd = new FormData();
fd.append( 'file', input.files[0] );

$.ajax({
  url: 'http://example.com/script.php',
  data: fd,
  processData: false,
  contentType: false,
  type: 'POST',
  success: function(data){
    alert(data);
  }
});
```

paramétrer `processData` à `false` vous permet d'empêcher jQuery de transformer automatiquement les données en une chaîne de requête. Voir les docs (<https://api.jquery.com/jquery.ajax/>) pour plus d'info.

mettre `contentType` à `false` est impératif, car sinon jQuery le mettra incorrectement (<https://stackoverflow.com/a/5976031/33080>) .

719

répondu pradeek (<https://stackoverflow.com/users/514873/pradeek>) 2017-05-29 20:44:49

il y a quelques techniques encore à être mentionnées disponibles pour vous. Commencez par définir la propriété `contentType` dans vos paramètres ajax.

Bâtiment sur pradeek exemple:

```

$('form').submit(function (e) {
    var data;

    data = new FormData();
    data.append('file', $('#file')[0].files[0]);

    $.ajax({
        url: 'http://hacheck.tel.fer.hr/xml.pl',
        data: data,
        processData: false,
        type: 'POST',

        // This will override the content type header,
        // regardless of whether content is actually sent.
        // Defaults to 'application/x-www-form-urlencoded'
        contentType: 'multipart/form-data',

        //Before 1.5.1 you had to do this:
        beforeSend: function (x) {
            if (x && x.overrideMimeType) {
                x.overrideMimeType("multipart/form-data");
            }
        },
        // Now you should be able to do this:
        mimeType: 'multipart/form-data',    //Property added in 1.5.1

        success: function (data) {
            alert(data);
        }
    });

    e.preventDefault();
});

```

dans certains cas, en forçant jquery ajax à faire des choses inattendues, l'événement `beforeSend` est un excellent endroit pour le faire. Pendant un certain temps, les gens utilisaient `beforeSend` pour outrepasser le `mimeType` avant que celui-ci soit ajouté dans jQuery en 1.5.1. Vous devriez être en mesure de Modifier à peu près n'importe quoi sur l'objet `jqXHR` dans l'événement `before send`.

25 

répondu BenSwayne (<https://stackoverflow.com/users/1002041/benswayne>) 2011-11-30 00:20:43

vous pouvez envoyer L'objet FormData dans la demande ajax en utilisant le code suivant,

```

$("form#formElement").submit(function(){
    var formData = new FormData($(this)[0]);
});

```

c'est très similaire à la réponse acceptée, mais une réponse réelle pour le sujet de la question. Ceci soumettra automatiquement les éléments de formulaire dans FormData et vous n'avez pas besoin d'ajouter manuellement les données à la variable FormData.

la méthode ajax ressemble à ceci,

```

$("form#formElement").submit(function(){
    var formData = new FormData($(this)[0]);
    //append some non-form data also
    formData.append('other_data', $("#someInputData").val());
    $.ajax({
        type: "POST",
        url: postDataUrl,
        data: formData,
        processData: false,
        contentType: false,
        dataType: "json",
        success: function(data, textStatus, jqXHR) {
            //process data
        },
        error: function(data, textStatus, jqXHR) {
            //process error msg
        },
    });
});

```

vous pouvez également passer manuellement le formulaire élément à l'intérieur de l'objet FormData comme paramètre comme ceci

```

var formElem = $("#formId");
var formData = new FormData(formElem);

```

J'espère que ça aidera. ;)

12 

répondu Lucky (<https://stackoverflow.com/users/1793718/lucky>) 2018-08-28 22:00:23

, Vous pouvez utiliser le \$.ajax beforeSend événement pour manipuler l'en-tête.

...

```

beforeSend: function(xhr) {
    xhr.setRequestHeader('Content-Type', 'multipart/form-data');
}

```

...

voir ce lien pour plus d'informations: [http://msdn.microsoft.com/en-us/library/ms536752\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536752(v=vs.85).aspx)
([https://msdn.microsoft.com/en-us/library/ms536752\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms536752(v=vs.85).aspx))

5 

répondu dmnkhnn (<https://stackoverflow.com/users/259234/dmnkhnn>) 2011-11-28 21:12:35

je pense que vous ne pouvez pas le faire en ajax pour prendre en charge tous les navigateurs, je pourrais dire bon de vérifier ce plugin Uploader ajax pour voir comment ils ont fait <http://valums.com/ajax-upload/>
(<http://valums.com/ajax-upload/>)

4 

répondu nivanka (<https://stackoverflow.com/users/542815/nivanka>) 2011-11-29 09:18:30

je fais comme ça et c'est un travail pour moi, j'espère que cela aidera :)

```

<div id="data">
  <form>
    <input type="file" name="userfile" id="userfile" size="20" />
    <br /><br />
    <input type="button" id="upload" value="upload" />
  </form>
</div>
<script>
  $(document).ready(function(){
    $('#upload').click(function(){

      console.log('upload button clicked!')
      var fd = new FormData();
      fd.append( 'userfile', $('#userfile')[0].files[0]);

      $.ajax({
        url: 'upload/do_upload',
        data: fd,
        processData: false,
        contentType: false,
        type: 'POST',
        success: function(data){
          console.log('upload success!')
          $('#data').empty();
          $('#data').append(data);
        }
      });
    });
  });
</script>

```

4 répondu talalalshehri (<https://stackoverflow.com/users/3768665/talalalshehri>) 2015-03-28 15:17:47

si vous voulez soumettre des fichiers en utilisant ajax utilisez "jquery.form.js" Cette soumet tous les éléments de forme facilement.

échantillons <http://jquery.malsup.com/form/#ajaxSubmit> (<http://jquery.malsup.com/form/#ajaxSubmit>)

vue imprenable:

```

<form id='AddPhotoForm' method='post' action='../photo/admin_save_photo.php' enctype='multipart/form-data'>

```

```

<script type="text/javascript">
function showResponseAfterAddPhoto(responseText, statusText)
{
    information= responseText;
    callAjaxtoList();
    $("#AddPhotoForm").resetForm();
    $("#photo_msg").html('<div class="album_msg">Photo uploaded Successfully...</div>');
};

$(document).ready(function(){
    $('.add_new_photo_div').live('click',function(){
        var options = {success:showResponseAfterAddPhoto};
        $("#AddPhotoForm").ajaxSubmit(options);
    });
});
</script>

```

2 répondu Rohit (<https://stackoverflow.com/users/979713/rohit>) 2011-11-30 14:43:20

JavaScript:

```

function submitForm() {
    var data1 = new FormData($('input[name^="file"]'));
    $.each($('input[name^="file"]')[0].files, function(i, file) {
        data1.append(i, file);
    });

    $.ajax({
        url: "<?php echo base_url() ?>employee/dashboard2/test2",
        type: "POST",
        data: data1,
        enctype: 'multipart/form-data',
        processData: false, // tell jQuery not to process the data
        contentType: false // tell jQuery not to set contentType
    }).done(function(data) {
        console.log("PHP Output:");
        console.log( data );
    });
    return false;
}

```

PHP:

```

public function upload_file(){

    foreach ($_FILES as $key ) {

        $name =time().$key['name'];

        $path='upload/'.$name;

        @move_uploaded_file($key['tmp_name'],$path);

    }
}

```

2 répondu Rahul Yadav (<https://stackoverflow.com/users/4498317/rahul-yadav>) 2015-03-24 10:32:03

au lieu de - `fd.append('userfile', $('#userfile')[0].files[0]);`

utilisation - `fd.append('file', $('#userfile')[0].files[0]);`

2 

répondu user4757231 (<https://stackoverflow.com/users/4757231/user4757231>) 2015-04-07 10:40:45

la meilleure documentation et l'exemple que j'ai trouvé était ici https://developer.mozilla.org/en-US/docs/Web/Guide/Using_FormData_Objects (https://developer.mozilla.org/en-US/docs/Web/Guide/Using_FormData_Objects)

0 

répondu DoodleKana (<https://stackoverflow.com/users/1022837/doodlekana>) 2014-04-14 20:50:42

Licensed under over.wiki (<https://over.wiki/>) with attribution.



WebDevDesigner .com(<https://webdevdesigner.com/>)

(<http://hit.ua/?x=35276>)