We can’t change history, but we can change the future.
Be nice to each other. @robertnyman

Skip to main content

- About
- All posts
- Speaking
- Geek Meet
- Code
- RSS feeds

Using FormData to send forms with XHR as key/value pairs

Published on Monday, February 11, 2013

Being able to easily specify what to post with XMLHttpRequest is quite a powerful way of sending things to the server, using key/value pairs and `FormData`. However, many seem to have missed this gem, so I thought I’d outline it here.

The basics

The general approach is that you create a `FormData` object. Then you use the `append` method to add on any extra keys and their values. like this:

view sourceprint?

```
var form = new FormData();
form.append("myName", "Robert");
```

You then just send it with XMLHttpRequest (XHR) through the `send` method:

view sourceprint?

```
var xhrForm = new XMLHttpRequest();
xhrForm.open("POST", "getfile.php");
xhrForm.send(form);
```

The interesting thing with FormData is that you you aren’t limited to adding strings, but actually a number of different types:

- strings
- numbers (converted to strings when sent)
- files
- blobs

What is important to know, to be able to deal on the server side with a FormData form, is that is the same a regular form that has been sent with the encoding `multipart/form-data`.

Adding files and blobs

If you want to append a file, simplest way is accessing a file that the user has chosen through an `input` element of `type="file"`:

view sourceprint?

```
form.append("theFile", fileInput.files[0]);
```

Appending a blob

Working with blobs can be quite powerful, both in sending and receiving values. A `Blob` can be manually created by a reference to its contents and its type:

view sourceprint?

```
form.append("blobbie", new Blob([imgAsBlobRef], {"type": "image/png"}));
```

Create your blob's contents

You can also create a blob's contents on your own:

view sourceprint?

```
var xmlForBlob = ["<root><name>Robert</name></root>"],
        xmlBlob = new Blob(xmlForBlob, {"type" : "text/xml"});

form.append("xmlParts", xmlBlob);
```

Get image in page and create blob

You can also, for instance, get an image in a page through XHR and then send it through `FormData`:

view sourceprint?

```
// Getting a file through XMLHttpRequest as an arraybuffer and creating a Blob
var rhino = document.querySelector("#rhino");
if (rhino) {
        var xhr = new XMLHttpRequest(),
                blob;

        xhr.open("GET", "rhino.png", true);
        xhr.responseType = "blob";

        xhr.onreadystatechange = function () {
                if (xhr.readyState === 4 && xhr.status === 200) {
                        blob = xhr.response;
                        var form = new FormData();
                        form.append("blobbie", blob);

                        var xhrForm = new XMLHttpRequest();
                        xhrForm.open("POST", "getfile.php");
                        xhrForm.send(form);
                }
        };
        // Send XHR
        xhr.send();
}
```

Using Web Activities

I've written about Web Activities for the Mozilla Hacks blog before, and with that approach, it can be possible to access the camera of the device, take a picture and then get the result back as a blob.

Once you have it, you can send it to the server. In this case, we'll take a picture through Web Activities, send it to the server via FormData and then get the image back and present it in the current page:

view sourceprint?

```
var pick = new MozActivity({
          name: "pick",
          data: {
                  type: ["image/png", "image/jpg", "image/jpeg"]
          }
  });

pick.onsuccess = function () {?    var form = new FormData();
        form.append("blobbie",  this.result.blob);

        var xhrForm = new XMLHttpRequest();
        xhrForm.open("POST", "getfile.php");
        xhrForm.send(form);

        xhrForm.onreadystatechange = function () {
                if (xhrForm.readyState === 4) {
                        var img = document.createElement("img");
                        img.src = xhrForm.response;

                        var imagePresenter = document.querySelector("#image-presenter");
                        imagePresenter.appendChild(img);
                        imagePresenter.style.display = "block";
                }
        };
};
```

Reading out the form on the server

Remember in the beginning that I mentioned that the form is sent with the encoding `multipart/form-data`. This is how you could read out the name, value and contents of a blob posted through `FormData`:

view sourceprint?

```
<?php
    $fileName = $_FILES['blobbie']['name'];
    $fileType = $_FILES['blobbie']['type'];
    $fileContent = file_get_contents($_FILES['blobbie']['tmp_name']);
    $dataURL = 'data:' . $fileType . ';base64,' . base64_encode($fileContent);
    echo $dataURL;
?>
```

I found the above gem by Eric Bidelman in a gist. In general, you should be able to do this with any server side language. Just picked PHP here since it already runs on most servers.

A complete demo

I put together a complete example with getting an image in the page through XHR, posting it with `FormData`, reading out the contents and returning a data URL so the image can be presented in the page again.

Generally, an exercise to show how information can be sent back and forth.

The example is available at https://github.com/robnyman/robnyman.github.com/tree/master/html5demos/formdata

(GitHub pages doesn't – probably for good reasons – allow you to run PHP code to read out the file contents, but the page/layout with a broken image is in the FormData GitHub page).

JavaScript code

view sourceprint?

```
(function () {
    // Getting a file through XMLHttpRequest as an arraybuffer and creating a Blob
    var rhino = document.querySelector("#rhino");
    if (rhino) {
        var xhr = new XMLHttpRequest(),
            blob;

        xhr.open("GET", "rhino.png", true);
        /*
            Set the responseType to "blob".
            If it isn't supported in the targeted web browser,
            use "arraybuffer" instead and wrap the response
            with new Uint8Array() below
        */
        xhr.responseType = "blob";

        xhr.onreadystatechange = function () {
            if (xhr.readyState === 4 && xhr.status === 200) {
                /*
                    Create a blob from the response
                    Only needed if the responseType isn't already blob
                    If it's "arraybuffer", do this:

                    blob = new Blob([new Uint8Array(xhr.response)], {type: "image/png"});
                */
                blob = xhr.response;
                var form = new FormData();
                form.append("blobbie", blob);

                var xhrForm = new XMLHttpRequest();
                xhrForm.open("POST", "getfile.php");
                xhrForm.send(form);

                xhrForm.onreadystatechange = function () {
                    if (xhrForm.readyState === 4) {
                        console.log(xhrForm.response);
                        rhino.src = xhrForm.response;
                    }
                };
            }
        };
        // Send XHR
        xhr.send();
    }
})();
```

PHP code

view sourceprint?

```
<?php
        $fileName = $_FILES['blobbie']['name'];
        $fileType = $_FILES['blobbie']['type'];
        $fileContent = file_get_contents($_FILES['blobbie']['tmp_name']);
        $dataURL = 'data:' . $fileType . ';base64,' . base64_encode($fileContent);
        echo $dataURL;
?>
```

Web browser support

Really good, actually! `FormData` is supported in:

- Firefox 4+
- Google Chrome 7+
- Safari 5+
- Opera 12+
- Internet Explorer 10+ (planned support)

The support for `Blob` is:

- Firefox 13+
- Google Chrome 20+
- Safari 5.1+
- Opera 12.1+
- Internet Explorer 10+ (planned support)

Posted in Developing, HTML5/HTML/XHTML, JavaScript

13 Comments

- RSS feed for comments on this post
- RSS feed for comments on all posts

- *nyamsprod* says:
  February 11, 2013 at 17:14

  What do you mean by

  > Internet Explorer 10+ (planned support)

  Is it supported or not on IE10 ?

  Reply
- *Robert Nyman* says:
  February 11, 2013 at 17:48

  nyamsprod,

  It’s planned to be in Internet Explorer 10. Since I don’t work for Microsoft, I can’t guarantee if they will ship it or not.

  Reply
- *Brian LePore* says:
  February 11, 2013 at 18:30

  Robert,

  IE 10 launched with Windows 8. I think that was what prompted nyamsprod’s comment.

  Does this code retain EXIF and other meta data from JPEG files? That seems to be an issue with most other scripts I’ve seen that use Blob’s, though admittedly those are usually dynamically re-sizing a selected JPEG and uploading the re-sized content.

  Reply
- *Robert Nyman* says:
  February 12, 2013 at 10:19

  Brian,

  What I meant to say – and should have been clearer – is that it was supposed to ship with IE10, but I don’t know if it will ship on all platforms (I don’t know the status of IE10 on Windows 7, for instance).

  With EXIF data, it should be retained. This is sending the file in question – as a file, blob or similar – so no tampering should take place.

  Reply
- *Robert Lomeland* says:

February 12, 2013 at 10:24

Hi, I tried the example, I tried both the git version and ripped your example site (which is also having a problem, but not the same. getfile.php returnes 405 not allowed).

All testing is done using Firefox 18.0.2
I use .Net and not php so my serverside code runs in `getfile.aspx`.

The problem: the Form has no elements.
If I inspect `Request.Form I find that Request.Form.Keys.Count == 0`

If I add this to the script, just after the blob should have been appended: `form.append("robert", "test")` I do get 1 item in the posted form, namely my "robert" key..

Any ideas?

Reply

- *Robert Nyman* says:
  February 12, 2013 at 11:46

  Robert,

  Yes, as mentioned in the article, GitHub pages doesn't allow you to un PHP pages for that, so you are expected to get a 405.

  When it comes to .NET and reading out forms posted as multipart/form-data, I recommend reading the Sending HTML Form Data article.

  Reply
- *najib* says:
  October 17, 2014 at 15:52

  hi i love the code i will to try if it will work for me so thank you for sharing

  Reply
- *jayden* says:
  March 18, 2015 at 15:52

  hi robert,
  thanx for nice example, got the basics working out of the box in like 10 minutes, but only after copying the source directly from the demo site (http://robertnyman.com/html5/fileapi-upload/fileapi-upload.html), tried it before with the snippets you present here, but that wasnt working.
  one odd thing though, when i choose a file, it seems that it is processed twice, as the chosen image is presented in the preview twice… hm, any idea?
  gonna keep on trying, if i find reason myself first, i gonna post it here.
  Thanx again

  Reply
  - *jayden* says:
    March 18, 2015 at 16:03

    hey, i think it was just a stupid mistake on my side. i had the script placed within a loop, so it existed within the html twice.
    and probably the snippets from here work the same as on the demo page, i just had the script inserted before the html when i first tried the code from this very site, so it was not working. but now it is, and i must say, till now it looks great!! thanks again!!!

    Reply
    - *Robert Nyman* says:
      April 30, 2015 at 12:20

      Glad to hear it worked out!

      Reply
- *Michele* says:
  November 2, 2015 at 16:12

  Dear Robert, sorry for the previous question…I solved the stupid issue.Thanks anyway for the attention. michele

  Reply
- *Michele* says:
  November 2, 2015 at 16:12

  Dear Robert, sorry for the previous question…I solved the stupid issue.Thanks anyway for the attention. Thanks a lot for the great example michele

  Reply
- *Aditya* says:
  April 14, 2016 at 9:52

  Hi, how do you retrieve the "myName" key pair that is "Robert" data on server. we also appended that in form. But how to retrieve that?????????

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.

- @robertnyman
- Upcoming speaking
- Photos on Flickr