

# HTML5 AJAX File Upload

timdread

*Tossug*, Sep. 14, 2010

# AJAX File Upload

- The ability to send files from browser to server without whole-page form submission
- Demand exists ever since rich web app. was being developed (circa. 2004)
- People tried to do this by other means
  - Flash-based resolution, e.g. SWFUpload
    - *Flash sucks* on non-IE browsers (NPAPI version)
    - *Even sucks* more on iOS
  - iframe and a form
    - Breaks back/forward buttons
    - Compromised program workflow
    - Doesn't show progress (except Chrome)



HTML5 come to the rescue!

:)

# Steps to send a file

1. Receive file from user
  - through `<input type="file" />`, or
  - File dropping
2. Read the file
3. Construct RFC 1867 multipart/form-data request body
  - or, not exactly
4. Send the file through `XMLHttpRequest`

# 1. Receive Files from User

- or, getting the `FileList` and `File` object

```
$( 'input[type=file]' ).bind( 'change',  
    function () {  
        handleFile( this.files[0] );  
    }  
);
```

```
$( 'div' )  
    .bind( 'dragover', function () {return false;} )  
    .bind( 'drop', function (ev) {  
        handleFile( ev.originalEvent.dataTransfer.files[0] )  
    } );  
});
```

## 2. Read the File

- **Firefox 3.0+:** `var bin = file.getAsBinary();`
  - Might block UI
  - Might throw error
- **Firefox 3.6**

```
var reader = new FileReader();
reader.onloadend = function (ev) {
    var bin = ev.target.result;
};
reader.onerror = function (ev) {
    switch (ev.target.error) { ... }
};
reader.readAsBinaryString(file);
```

- **Firefox 4, Chrome 5, Safari 5: Skip this step**

# 3. Construct request body

- Body of a POST request, defined in RFC 1867

```
--boundary-103973  
content-disposition: form-data; name="Filedata"; filename="a-image.png"  
Content-Type: image/png
```

%PNG

IHDR  .....(binary string)

```
--boundary-103973--
```

- We need to craft our own if there is no `FormData`
  - Firefox 3.0, Firefox 3.5, Firefox 3.6
  - UTF-8 file name issue

# 3. Construct request body (cont.)

- For browsers provide FormData

```
var formdata = new FormData();  
formdata.append('Filedata', file);
```

# 4. Send the file through xhr

- For FormData Browsers: `xhr.send(formData);`
  - Most leave automatically generated Content-Type header alone
- For self-crafted binary string request body (as data):  
`xhr.sendAsBinary(data);`
  - Most specify content-type along with request body, e.g.  
`multipart/form-data, boundary=boundary-103973`

# Comparison & Small Conclusion

Browser	Read	Form	Send
Fx 3.0/3.5	readAsBinary()	self-craft	sendAsBinary()
Fx 3.6	FileReader	self-craft	sendAsBinary()
Fx 4	FormData	FormData	send(formdata)
Chrome 5	FormData	FormData	send(formdata)
Safari 5	FormData	FormData	send(formdata)
Opera 10.6	<u>Not supported Yet</u>		
IE	<u>Wheeee!!!!</u>		

# Issue: UTF-8 file name

- Only applied to self-crafted request body
- Solution: conversion UTF-8 filename to binary string ourselves
  - `encodeURIComponent()` could to the trick

```
var encString = encodeURIComponent('土虱.png');  
// '%E5%9C%9F%E8%99%B1.png'  
String.fromCharCode(parseInt('E5', 16));  
// 'å': U+00E5, 0xE5 in ASCII  
...  
// åÉÿè™±.png
```

# Issue: Feature Detection

Browser	Read	Form	Send
Fx 3.0/3.5	<u>readAsBinary()</u>	self-craft	sendAsBinary()
Fx 3.6	FileReader	self-craft	sendAsBinary()
Fx 4	FormData	FormData	<u>send(formdata)</u>
Chrome 5	FormData	FormData	<u>send(formdata)</u>
Safari 5	FormData	FormData	<u>send(formdata)</u>

- Anything marked in **red** cannot be detected.
- Detection based on existence of `FileList` or `File` won't work.
- Solution: come up an `if (expression) {}` based on this table.

```
if (
  (XMLHttpRequest && XMLHttpRequest.prototype.sendAsBinary)
  || window.FormData
) { $(document.body).addClass('support_ajaxfileupload'); }
```

# Issue: Pre-processing

- Case: Thumbnail an image before sending it
  - Can be done with `FileReader` or `getAsBinary`
  - Won't work with `FormData` cause `File` is read-only and cannot be constructed.
  - Workaround
    - rewrite server endpoint for it to receive files as form fields or even request body itself
    - convert the data into base64 string so it would go through `xhr.send()`
  - Any suggestions?

# Issue: jQuery.ajax() integration

- Benefit: to have jQuery
  - call global ajax event handler
  - process the response data (with preferred dataType)
- Problems: jQuery is too smart *sometimes*
  - it defaults request content type to `application/x-www-form-urlencoded`  
but we need content-type being `multipart/form-data, boundary=boundary-103973`
  - it converts post data from object to string if it's an object
    - Wait! FormData is an object!
  - jQuery hard-coded `xhr.send()`
    - But we need `xhr.sendAsBinary()`

# Issue: jQuery.ajax integration (cont.)

- **Solution for content-type overwrite and data conversion**

```
settings.contentType = null;
settings.data = null;
settings.__beforeSend = settings.beforeSend;
settings.beforeSend = function (xhr, s) {
    s.data = formdata;
    if (s.__beforeSend)
        return s.__beforeSend.call(this, xhr, s);
}
```

- **based on jQuery-1.4.2.js line 5119**
- **Only needed if you are using FormData**

# Issue: jQuery.ajax integration (cont.)

- **Solution for hard-coded `xhr.send()`**

```
XMLHttpRequest.prototype.send = function (data) {  
    return this.sendAsBinary(data);  
}
```

- DO NOT USE `sendAsBinary` with `FormData`
- Use `sendAsBinary` to send ASCII string appears fine

- **Alternatively,**

```
settings.___beforeSend = settings.beforeSend;  
settings.beforeSend = function (xhr, s) {  
    xhr.send = xhr.sendAsBinary;  
    if (s.___beforeSend) return s.___beforeSend.call(this, xhr,  
s);  
}
```

(faint)

因為全部搞定實在是太麻煩了

I made a *jQuery plug-in* for  
everything above.

(dance)

# Issue: Interaction

- For file input control
  - Some CSS hack could overwrite the default look
  - Do disable and provide alternative when the browser is not supported
- For drag and drop control
  - *Do* provide visual indication when file is being dragging over the page
  - *Do* disable indication above (or any instruction before file being dragged into) if the browser don't support uploading
- For AJAX Uploading
  - Do provide an indicator
  - Progress is only supported by Firefox (for now)

# *Demo*

<http://timc.idv.tw/html5-file-upload/>

# Acknowledgment

- jnlin
- othree for [this tweet](#)
- Tossug

Q & A

*Thank you*