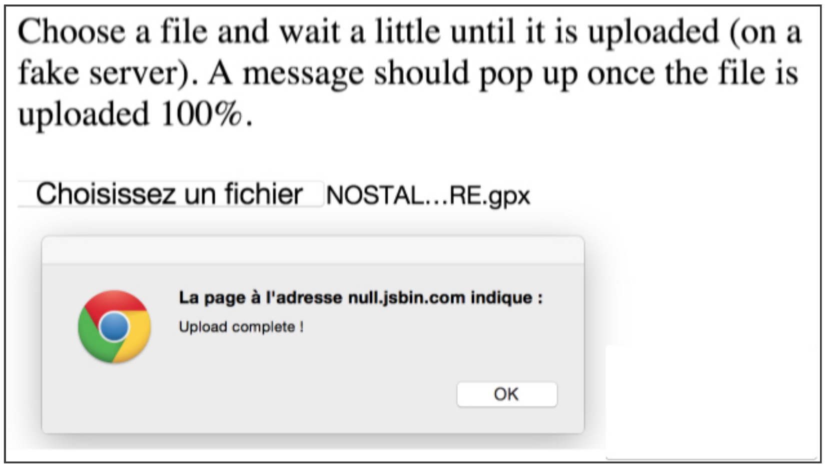
Uploading binary files with XHR2

Here is an example that uses a `FormData` object for uploading one or more files to an HTTP server.

Notice that the URL of the server is fake, so the request normally fails here. However, sending a file to a fake server takes time, and it is interesting to see how it works.

Another section of the course ("File API, drag and drop and XHR2"), to be seen later this week, will show full examples of real working code with server-side PHP source.

Try the example on JSBin:


Choose a file and wait a little until it is uploaded (on a fake server). A message should pop up once the file is uploaded 100%.

La page à l'adresse null.jsbin.com indique :

Upload complete !

Source code of the example:

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8" />
   <title>File upload with XMLHttpRequest level 2 and
HTML5</title>
 </head>

 <body>
```

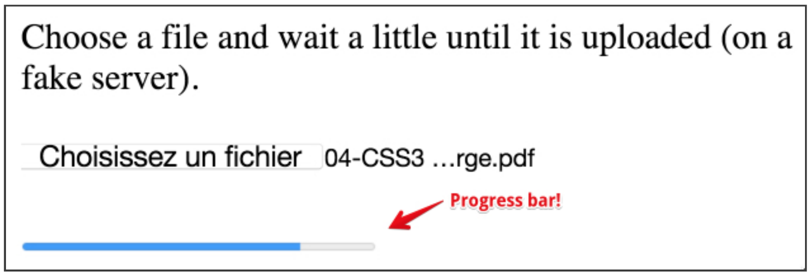
```
     <h1>Example of XHR2 file upload</h1>
10.   Choose a file and wait a little until it is uploaded (on a
    fake
11.   server). A message should pop up once the file is uploaded
    100%.
     <p>
     <input id="file" type="file" />
     </p>
     <script>
     var fileInput =document.querySelector('#file');

    fileInput.onchange = function() {
       var xhr = new XMLHttpRequest();
20.    xhr.open('POST', 'upload.html'); // With FormData,
21.                                     // POST is mandatory

       xhr.onload = function() {
         alert('Upload complete !');
       };

       var form = new FormData();
       form.append('file', fileInput.files[0]);
       // send the request
       xhr.send(form);
31. };
     </script>
     </body>
    </html>
```

Explanations:

- *Line 18*: callback called when a file has been selected.
- *Lines 19-20*: preparation of the XHR2 request.
- *Lines 27-30*: a `FormData` object is created (this is a container for parts in the multi part data that will be sent by the POST request). At *line 30*, the request is sent, with the `FormData`passed as a parameter (all data are sent).
- *Line 23*: when the file is completely uploaded, the `onload`listener is called and an

alert message is displayed.

MONITOR THE UPLOAD PROGRESS


Choose a file and wait a little until it is uploaded (on a fake server).

This is the same example, but this time we monitor the progress of the upload using a method similar to the one presented for monitoring file downloads:

- We use a `<progress>` element and its two attributes `value` and`max`.
- We also bind an event handler to the `progress` event that an XMLHttpRequest can trigger. The event has two properties:`loaded` and `total` that respectively correspond to the number of bytes that have been uploaded, and to the total number of bytes we need to upload (i.e., the file size).

Here is the code of such an event listener:

```
xhr.upload.onprogress = function(e) {
  progress.value = e.loaded; // number of bytes uploaded
  progress.max = e.total;    // total number of bytes in the
file
};
```

Try the example on JSBin:

Code from this example (nearly the same as previous example's code):

```
<!DOCTYPE html>
<html lang="en">
 <head>
```

```
   <meta charset="utf-8" />
   <title>HTML5 file upload with monitoring</title>
 </head>

 <body>
 <h1>Example of XHR2 file upload, with progress bar</h1>
Choose a file and wait a little until it is uploaded (on a
fake server).
 <p>
 <input id="file" type="file" />
 <br/><br/>
 <progress id="progress" value=0></progress>

 <script>
   var fileInput =document.querySelector('#file'),
   progress =document.querySelector('#progress');

   fileInput.onchange = function() {
     var xhr = new XMLHttpRequest();
     xhr.open('POST', 'upload.html');

     xhr.upload.onprogress = function(e) {
       progress.value = e.loaded;
       progress.max = e.total;
     };
     xhr.onload = function() {
     alert('Upload complete!');
   };

   var form = new FormData();
   form.append('file', fileInput.files[0]);

   xhr.send(form);
 };
 </script>
 </body>
</html>
```

The only difference between this and the previous example is the`onprogress` listener that updates the progress bar's `value` and `max`attributes.