Get progress of an AJAX request

JQUERY PHP PROGRESS DOWNLOAD XHR AJAX UPLOAD JAVASCRIPT

So you write a cool AJAX application, but you're scared that people with über slow connections will get impatient and think that the site is broken. Obviously, the first thing to do is optimize by making sure you're not transferring too much data. That said, there are times when there is nothing much you can do about it, e.g. an image upload via ajax. The alternative would be to calculate the progress of the upload and show the user how much longer they have to wait

```
$.ajax({
    url: path,
    type: 'post',
    data: {payload: payload},
    xhr: function () {
        var xhr = $.ajaxSettings.xhr();
        xhr.onprogress = function e() {
            // For downloads
            if (e.lengthComputable) {
                console.log(e.loaded / e.total);
            }
        };
        xhr.upload.onprogress = function (e) {
            // For uploads
            if (e.lengthComputable) {
                console.log(e.loaded / e.total);
            }
        };
        return xhr;
    }
}).done(function (e) {
    // Do something
}).fail(function (e) {
    // Do something
});
```

NOTE

You can only use this calculate the progress of a download or upload. If your server is calculating a request slowly, there is no way of informing the user of any form of progress

The magic in the above code is the `xhr` property in the `$.ajax()` settings. What we do here is retrieve the original `xhr` object, and use the data therein to calculate the progress.

For uploads, the content-length is generally calculated for you by the browser, but for downloads, you have to make sure that you have a VALID (i.e. the correct amount of bytes) `content-length` header.

PHP

For the PHP developers, you can set the content-length header by doing the following

```
# E.G. For a large page
header('Content-Length: ' . mb_strlen($content));
```

```
# E.G. For a file
header('Content-Length: ' . filesize($download));
```

Written by Luke Madhanga