/\*!

JSZipUtils - A collection of cross-browser utilities to go along with JSZip.

<http://stuk.github.io/jszip-utils>

(c) 2014 Stuart Knightley, David Duponchel

Dual licenced under the MIT license or GPLv3. See https://raw.github.com/Stuk/jszip-utils/master/LICENSE.markdown.

\*/

!function(e){"object"==typeof exports?module.exports=e():"function"==typeof define&&define.amd?define(e):"undefined"!=typeof window?window.JSZipUtils=e():"undefined"!=typeof global?global.JSZipUtils=e():"undefined"!=typeof self&&(self.JSZipUtils=e())}(function(){var define,module,exports;return (function e(t,n,r){function s(o,u){if(!n[o]){if(!t[o]){var a=typeof require=="function"&&require;if(!u&&a)return a(o,!0);if(i)return i(o,!0);throw new Error("Cannot find module '"+o+"'")}var f=n[o]={exports:{}};t[o][0].call(f.exports,function(e){var n=t[o][1][e];return s(n?n:e)},f,f.exports,e,t,n,r)}return n[o].exports}var i=typeof require=="function"&&require;for(var o=0;o<r.length;o++)s(r[o]);return s})({1:[function(require,module,exports){

'use strict';

var JSZipUtils = {};

// just use the responseText with xhr1, response with xhr2.

// The transformation doesn't throw away high-order byte (with responseText)

// because JSZip handles that case. If not used with JSZip, you may need to

// do it, see https://developer.mozilla.org/En/Using\_XMLHttpRequest#Handling\_binary\_data

JSZipUtils.\_getBinaryFromXHR = function (xhr) {

// for xhr.responseText, the 0xFF mask is applied by JSZip

return xhr.response || xhr.responseText;

};

// taken from jQuery

function createStandardXHR() {

try {

return new window.XMLHttpRequest();

} catch( e ) {}

}

function createActiveXHR() {

try {

return new window.ActiveXObject("Microsoft.XMLHTTP");

} catch( e ) {}

}

// Create the request object

var createXHR = window.ActiveXObject ?

/\* Microsoft failed to properly

\* implement the XMLHttpRequest in IE7 (can't request local files),

\* so we use the ActiveXObject when it is available

\* Additionally XMLHttpRequest can be disabled in IE7/IE8 so

\* we need a fallback.

\*/

function() {

return createStandardXHR() || createActiveXHR();

} :

// For all other browsers, use the standard XMLHttpRequest object

createStandardXHR;

JSZipUtils.getBinaryContent = function(path, callback) {

/\*

\* Here is the tricky part : getting the data.

\* In firefox/chrome/opera/... setting the mimeType to 'text/plain; charset=x-user-defined'

\* is enough, the result is in the standard xhr.responseText.

\* cf https://developer.mozilla.org/En/XMLHttpRequest/Using\_XMLHttpRequest#Receiving\_binary\_data\_in\_older\_browsers

\* In IE <= 9, we must use (the IE only) attribute responseBody

\* (for binary data, its content is different from responseText).

\* In IE 10, the 'charset=x-user-defined' trick doesn't work, only the

\* responseType will work :

\* http://msdn.microsoft.com/en-us/library/ie/hh673569%28v=vs.85%29.aspx#Binary\_Object\_upload\_and\_download

\*

\* I'd like to use jQuery to avoid this XHR madness, but it doesn't support

\* the responseType attribute : http://bugs.jquery.com/ticket/11461

\*/

try {

var xhr = createXHR();

xhr.open('GET', path, true);

// recent browsers

if ("responseType" in xhr) {

xhr.responseType = "arraybuffer";

}

// older browser

if(xhr.overrideMimeType) {

xhr.overrideMimeType("text/plain; charset=x-user-defined");

}

xhr.onreadystatechange = function(evt) {

var file, err;

// use `xhr` and not `this`... thanks IE

if (xhr.readyState === 4) {

if (xhr.status === 200 || xhr.status === 0) {

file = null;

err = null;

try {

file = JSZipUtils.\_getBinaryFromXHR(xhr);

} catch(e) {

err = new Error(e);

}

callback(err, file);

} else {

callback(new Error("Ajax error for " + path + " : " + this.status + " " + this.statusText), null);

}

}

};

xhr.send();

} catch (e) {

callback(new Error(e), null);

}

};

// export

module.exports = JSZipUtils;

// enforcing Stuk's coding style

// vim: set shiftwidth=4 softtabstop=4:

},{}]},{},[1])

(1)

});

;