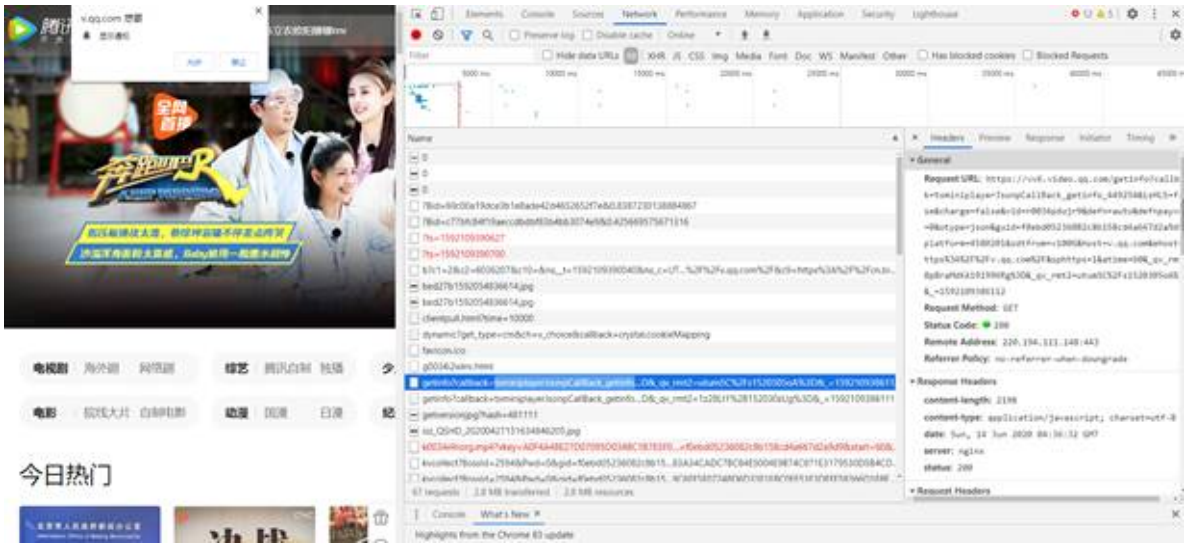
1. 请求

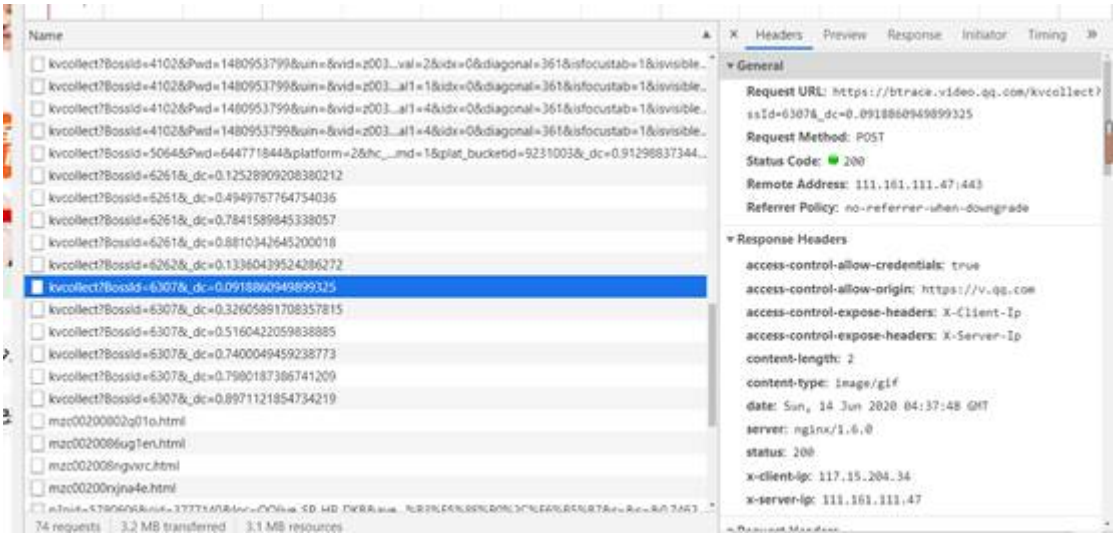
针对腾讯视频

此请求为GET请求

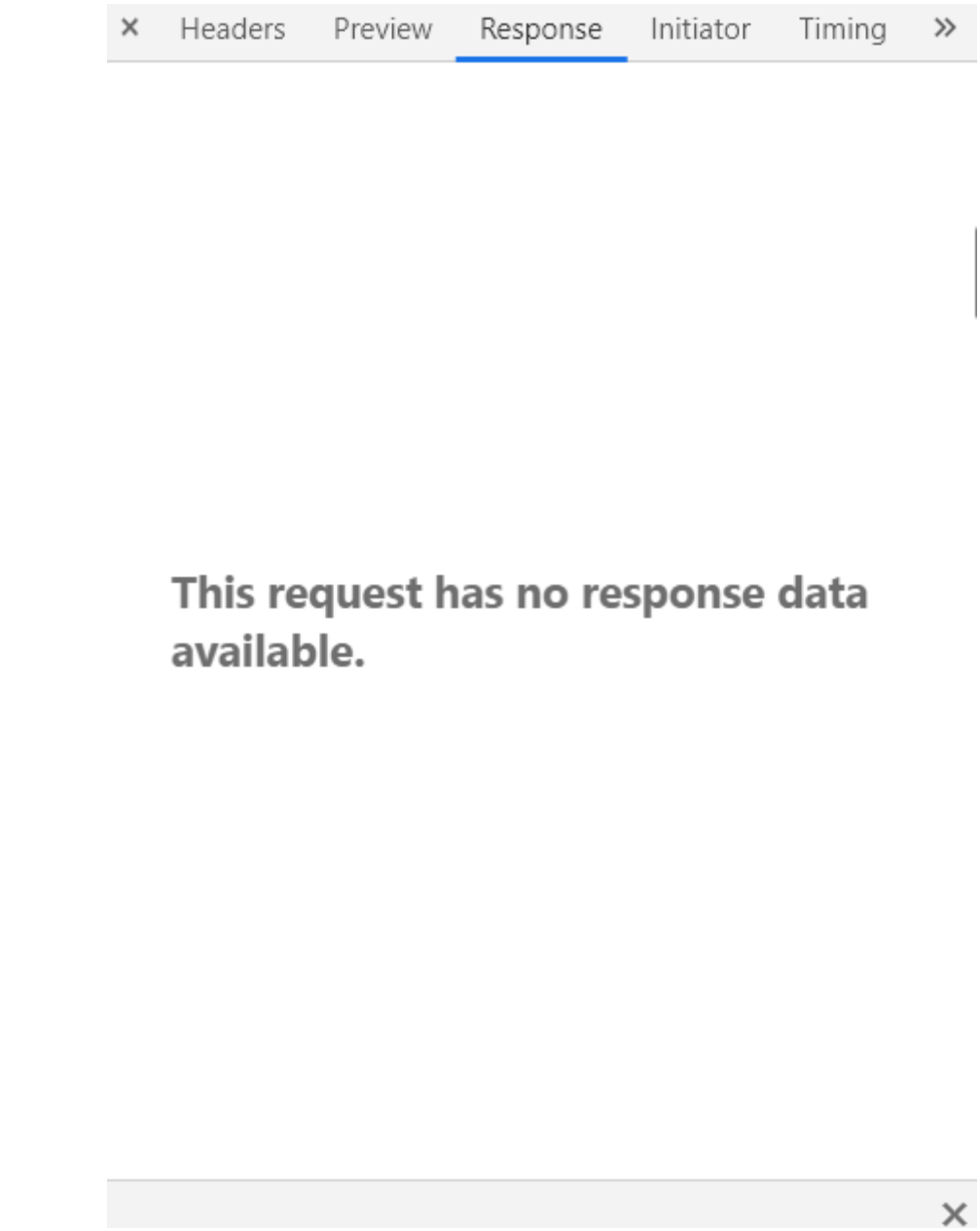


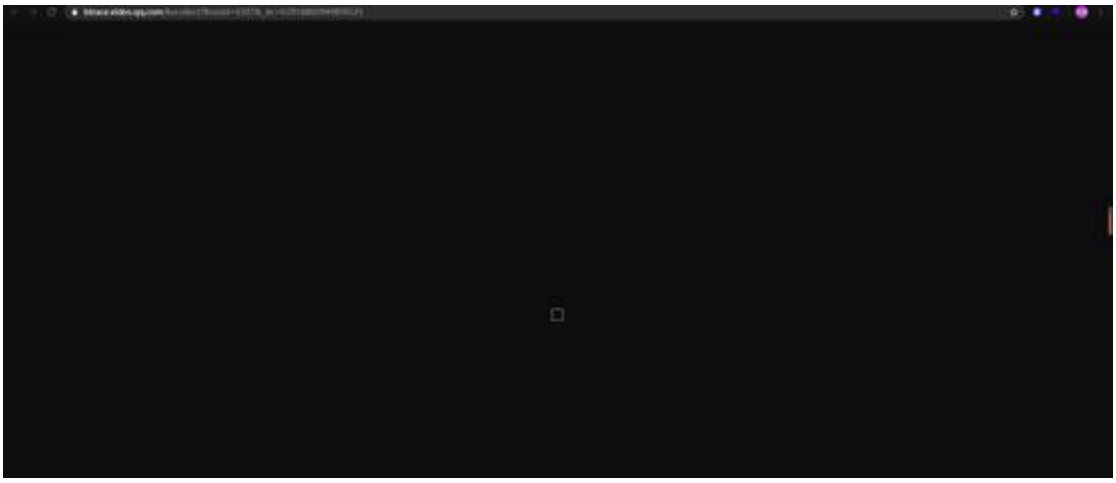
相应数据包

这是一个POST请求



无数据包

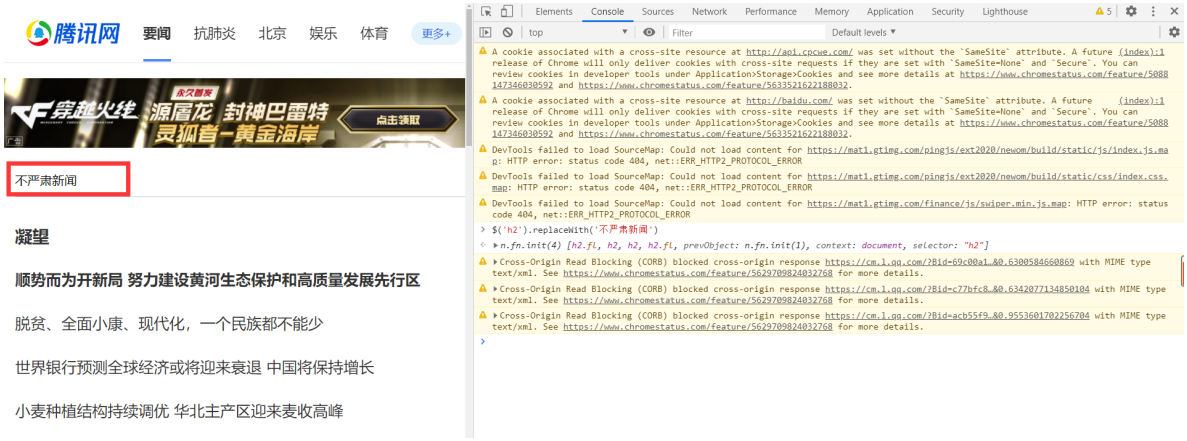




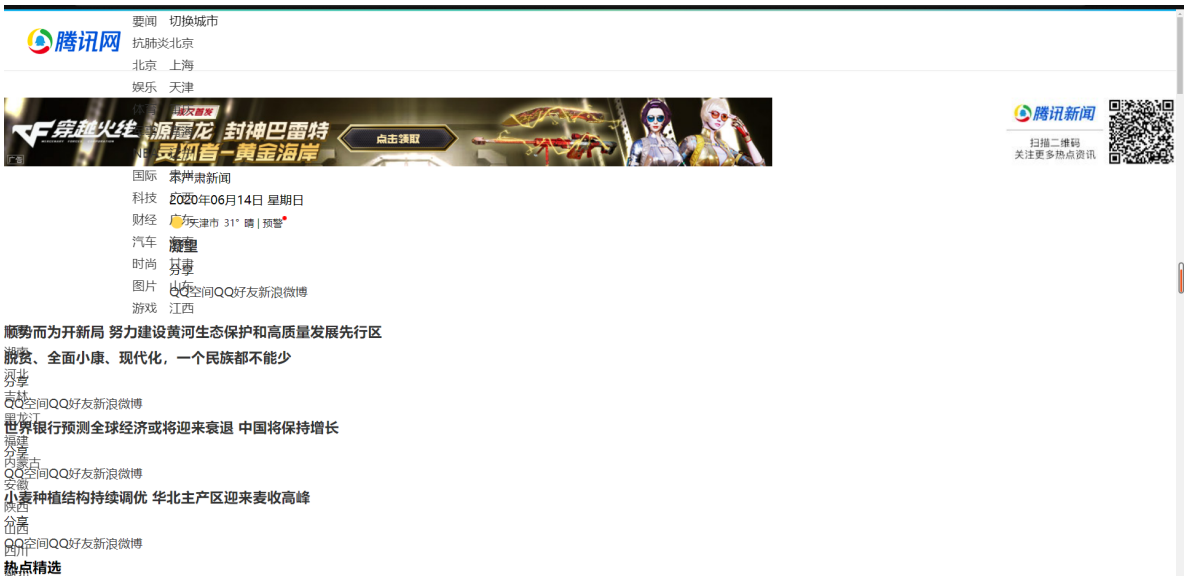
2.JQuery

替换后：

（1）$('h2').replaceWith('不严肃新闻') 替换文本



（2）$('div').removeAttr('name class') 删除属性



（3）$('ul').empty() 清空后代节点

3.浏览器插件

功能描述：快速过超星学习通视频（PS：不是自己用的，只是朋友的学习通视频很变态帮忙做了;仍然熟记上课的几条原则，求过审）

Background.js:

```
chrome.runtime.onInstalled.addListener(function () {
    chrome.declarativeContent.onPageChanged.removeRules(undefined, function () {
        chrome.declarativeContent.onPageChanged.addRules([{
            conditions: [
                new chrome.declarativeContent.PageStateMatcher({
                    pageUrl: {
                        urlContains: 'chaoxing.com/mycourse/studentstudy?'
                    }
                })
            ],
            actions: [new chrome.declarativeContent.ShowPageAction()]
        }]);
    });
});
```

md5.js

```
/*
 * A JavaScript implementation of the RSA Data Security, Inc. MD5 Message
 * Digest Algorithm, as defined in RFC 1321.
 * Version 2.1 Copyright (C) Paul Johnston 1999 - 2002.
 * Other contributors: Greg Holt, Andrew Kepert, Ydnar, Lostinet
 * Distributed under the BSD License
 * See http://pajhome.org.uk/crypt/md5 for more info.
 */

/*
 * Configurable variables. You may need to tweak these to be compatible with
```

```
 * the server-side, but the defaults work in most cases.
 */
var hexcase = 0;  /* hex output format. 0 - lowercase; 1 - uppercase        */
var b64pad  = ""; /* base-64 pad character. "=" for strict RFC compliance   */
var chrsz   = 8;  /* bits per input character. 8 - ASCII; 16 - Unicode      */

/*
 * These are the functions you'll usually want to call
 * They take string arguments and return either hex or base-64 encoded strings
 */
function hex_md5(s){ return binl2hex(core_md5(str2binl(s), s.length * chrsz));}
function b64_md5(s){ return binl2b64(core_md5(str2binl(s), s.length * chrsz));}
function str_md5(s){ return binl2str(core_md5(str2binl(s), s.length * chrsz));}
function hex_hmac_md5(key, data) { return binl2hex(core_hmac_md5(key, data)); }
function b64_hmac_md5(key, data) { return binl2b64(core_hmac_md5(key, data)); }
function str_hmac_md5(key, data) { return binl2str(core_hmac_md5(key, data)); }

/*
 * Perform a simple self-test to see if the VM is working
 */
function md5_vm_test()
{
  return hex_md5("abc") == "900150983cd24fb0d6963f7d28e17f72";
}

/*
 * Calculate the MD5 of an array of little-endian words, and a bit length
 */
function core_md5(x, len)
{
  /* append padding */
  x[len >> 5] |= 0x80 << ((len) % 32);
  x[(((len + 64) >>> 9) << 4) + 14] = len;

  var a =  1732584193;
  var b = -271733879;
  var c = -1732584194;
  var d =  271733878;

  for(var i = 0; i < x.length; i += 16)
  {
    var olda = a;
    var oldb = b;
    var oldc = c;
    var oldd = d;

    a = md5_ff(a, b, c, d, x[i+ 0], 7 , -680876936);
    d = md5_ff(d, a, b, c, x[i+ 1], 12, -389564586);
    c = md5_ff(c, d, a, b, x[i+ 2], 17,  606105819);
    b = md5_ff(b, c, d, a, x[i+ 3], 22, -1044525330);
    a = md5_ff(a, b, c, d, x[i+ 4], 7 , -176418897);
    d = md5_ff(d, a, b, c, x[i+ 5], 12,  1200080426);
    c = md5_ff(c, d, a, b, x[i+ 6], 17, -1473231341);
    b = md5_ff(b, c, d, a, x[i+ 7], 22, -45705983);
    a = md5_ff(a, b, c, d, x[i+ 8], 7 ,  1770035416);
    d = md5_ff(d, a, b, c, x[i+ 9], 12, -1958414417);
    c = md5_ff(c, d, a, b, x[i+10], 17, -42063);
    b = md5_ff(b, c, d, a, x[i+11], 22, -1990404162);
```

```
    a = md5_ff(a, b, c, d, x[i+12], 7 ,  1804603682);
    d = md5_ff(d, a, b, c, x[i+13], 12, -40341101);
    c = md5_ff(c, d, a, b, x[i+14], 17, -1502002290);
    b = md5_ff(b, c, d, a, x[i+15], 22,  1236535329);

    a = md5_gg(a, b, c, d, x[i+ 1], 5 , -165796510);
    d = md5_gg(d, a, b, c, x[i+ 6], 9 , -1069501632);
    c = md5_gg(c, d, a, b, x[i+11], 14,  643717713);
    b = md5_gg(b, c, d, a, x[i+ 0], 20, -373897302);
    a = md5_gg(a, b, c, d, x[i+ 5], 5 , -701558691);
    d = md5_gg(d, a, b, c, x[i+10], 9 ,  38016083);
    c = md5_gg(c, d, a, b, x[i+15], 14, -660478335);
    b = md5_gg(b, c, d, a, x[i+ 4], 20, -405537848);
    a = md5_gg(a, b, c, d, x[i+ 9], 5 ,  568446438);
    d = md5_gg(d, a, b, c, x[i+14], 9 , -1019803690);
    c = md5_gg(c, d, a, b, x[i+ 3], 14, -187363961);
    b = md5_gg(b, c, d, a, x[i+ 8], 20,  1163531501);
    a = md5_gg(a, b, c, d, x[i+13], 5 , -1444681467);
    d = md5_gg(d, a, b, c, x[i+ 2], 9 , -51403784);
    c = md5_gg(c, d, a, b, x[i+ 7], 14,  1735328473);
    b = md5_gg(b, c, d, a, x[i+12], 20, -1926607734);

    a = md5_hh(a, b, c, d, x[i+ 5], 4 , -378558);
    d = md5_hh(d, a, b, c, x[i+ 8], 11, -2022574463);
    c = md5_hh(c, d, a, b, x[i+11], 16,  1839030562);
    b = md5_hh(b, c, d, a, x[i+14], 23, -35309556);
    a = md5_hh(a, b, c, d, x[i+ 1], 4 , -1530992060);
    d = md5_hh(d, a, b, c, x[i+ 4], 11,  1272893353);
    c = md5_hh(c, d, a, b, x[i+ 7], 16, -155497632);
    b = md5_hh(b, c, d, a, x[i+10], 23, -1094730640);
    a = md5_hh(a, b, c, d, x[i+13], 4 ,  681279174);
    d = md5_hh(d, a, b, c, x[i+ 0], 11, -358537222);
    c = md5_hh(c, d, a, b, x[i+ 3], 16, -722521979);
    b = md5_hh(b, c, d, a, x[i+ 6], 23,  76029189);
    a = md5_hh(a, b, c, d, x[i+ 9], 4 , -640364487);
    d = md5_hh(d, a, b, c, x[i+12], 11, -421815835);
    c = md5_hh(c, d, a, b, x[i+15], 16,  530742520);
    b = md5_hh(b, c, d, a, x[i+ 2], 23, -995338651);

    a = md5_ii(a, b, c, d, x[i+ 0], 6 , -198630844);
    d = md5_ii(d, a, b, c, x[i+ 7], 10,  1126891415);
    c = md5_ii(c, d, a, b, x[i+14], 15, -1416354905);
    b = md5_ii(b, c, d, a, x[i+ 5], 21, -57434055);
    a = md5_ii(a, b, c, d, x[i+12], 6 ,  1700485571);
    d = md5_ii(d, a, b, c, x[i+ 3], 10, -1894986606);
    c = md5_ii(c, d, a, b, x[i+10], 15, -1051523);
    b = md5_ii(b, c, d, a, x[i+ 1], 21, -2054922799);
    a = md5_ii(a, b, c, d, x[i+ 8], 6 ,  1873313359);
    d = md5_ii(d, a, b, c, x[i+15], 10, -30611744);
    c = md5_ii(c, d, a, b, x[i+ 6], 15, -1560198380);
    b = md5_ii(b, c, d, a, x[i+13], 21,  1309151649);
    a = md5_ii(a, b, c, d, x[i+ 4], 6 , -145523070);
    d = md5_ii(d, a, b, c, x[i+11], 10, -1120210379);
    c = md5_ii(c, d, a, b, x[i+ 2], 15,  718787259);
    b = md5_ii(b, c, d, a, x[i+ 9], 21, -343485551);

    a = safe_add(a, olda);
    b = safe_add(b, oldb);
```

```
    c = safe_add(c, oldc);
    d = safe_add(d, oldd);
  }
  return Array(a, b, c, d);

}

/*
 * These functions implement the four basic operations the algorithm uses.
 */
function md5_cmn(q, a, b, x, s, t)
{
  return safe_add(bit_rol(safe_add(safe_add(a, q), safe_add(x, t)), s),b);
}
function md5_ff(a, b, c, d, x, s, t)
{
  return md5_cmn((b & c) | ((~b) & d), a, b, x, s, t);
}
function md5_gg(a, b, c, d, x, s, t)
{
  return md5_cmn((b & d) | (c & (~d)), a, b, x, s, t);
}
function md5_hh(a, b, c, d, x, s, t)
{
  return md5_cmn(b ^ c ^ d, a, b, x, s, t);
}
function md5_ii(a, b, c, d, x, s, t)
{
  return md5_cmn(c ^ (b | (~d)), a, b, x, s, t);
}

/*
 * Calculate the HMAC-MD5, of a key and some data
 */
function core_hmac_md5(key, data)
{
  var bkey = str2binl(key);
  if(bkey.length > 16) bkey = core_md5(bkey, key.length * chrsz);

  var ipad = Array(16), opad = Array(16);
  for(var i = 0; i < 16; i++)
  {
    ipad[i] = bkey[i] ^ 0x36363636;
    opad[i] = bkey[i] ^ 0x5C5C5C5C;
  }

  var hash = core_md5(ipad.concat(str2binl(data)), 512 + data.length * chrsz);
  return core_md5(opad.concat(hash), 512 + 128);
}

/*
 * Add integers, wrapping at 2^32. This uses 16-bit operations internally
 * to work around bugs in some JS interpreters.
 */
function safe_add(x, y)
{
  var lsw = (x & 0xFFFF) + (y & 0xFFFF);
  var msw = (x >> 16) + (y >> 16) + (lsw >> 16);
```

```
  return (msw << 16) | (lsw & 0xFFFF);
}

/*
 * Bitwise rotate a 32-bit number to the left.
 */
function bit_rol(num, cnt)
{
  return (num << cnt) | (num >>> (32 - cnt));
}

/*
 * Convert a string to an array of little-endian words
 * If chrsz is ASCII, characters >255 have their hi-byte silently ignored.
 */
function str2binl(str)
{
  var bin = Array();
  var mask = (1 << chrsz) - 1;
  for(var i = 0; i < str.length * chrsz; i += chrsz)
    bin[i>>5] |= (str.charCodeAt(i / chrsz) & mask) << (i%32);
  return bin;
}

/*
 * Convert an array of little-endian words to a string
 */
function binl2str(bin)
{
  var str = "";
  var mask = (1 << chrsz) - 1;
  for(var i = 0; i < bin.length * 32; i += chrsz)
    str += String.fromCharCode((bin[i>>5] >>> (i % 32)) & mask);
  return str;
}

/*
 * Convert an array of little-endian words to a hex string.
 */
function binl2hex(binarray)
{
  var hex_tab = hexcase ? "0123456789ABCDEF" : "0123456789abcdef";
  var str = "";
  for(var i = 0; i < binarray.length * 4; i++)
  {
    str += hex_tab.charAt((binarray[i>>2] >> ((i%4)*8+4)) & 0xF) +
           hex_tab.charAt((binarray[i>>2] >> ((i%4)*8  )) & 0xF);
  }
  return str;
}

/*
 * Convert an array of little-endian words to a base-64 string
 */
function binl2b64(binarray)
{
  var tab = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
  var str = "";
```

```
  for(var i = 0; i < binarray.length * 4; i += 3)
  {
    var triplet = (((binarray[i   >> 2] >> 8 * ( i   %4)) & 0xFF) << 16)
                | (((binarray[i+1 >> 2] >> 8 * ((i+1)%4)) & 0xFF) << 8 )
                |  ((binarray[i+2 >> 2] >> 8 * ((i+2)%4)) & 0xFF);
    for(var j = 0; j < 4; j++)
    {
      if(i * 8 + j * 6 > binarray.length * 32) str += b64pad;
      else str += tab.charAt((triplet >> 6*(3-j)) & 0x3F);
    }
  }
  return str;
}
```

mooc.js

```
console.log('cxmooc-tools start');
//监听框架加载
document.addEventListener('load', function (ev) {
    var ev = ev || event;
    var _this = ev.srcElement || ev.target;
    if (_this.id == 'iframe') {
        showExpand(_this);
    }
}, true);
showExpand(document.getElementById('iframe'));
/**
 * 显示扩展按钮,并绑定事件
 * @param {iframe document} _this
 */
function showExpand(_this) {
    ans = _this.contentWindow.document.getElementsByClassName('ans-job-icon');
    for (var i = 0; i < ans.length; i++) {
        ans[i].style.width = '100%';
        var boom = createBtn('秒过视频');
        ans[i].appendChild(boom);
        boom.onclick = function () {
            //获取参数
            var mArg = _this.contentWindow.document.body.innerHTML;
            mArg = '{' + substrEx(mArg, 'mArg = {', ';');
            mArg = JSON.parse(mArg);
            console.log(mArg);
            get('/ananas/status/' + mArg.attachments["0"].property.objectid +
                '?k=318&_dc=' + Date.parse(new Date())).onreadystatechange =
function () {
                if (this.readyState == 4) {
                    if (this.status != 200) {
                        alert('未知错误');
                    } else {
                        //第二步
                        var json = JSON.parse(this.responseText);
                        var playTime = parseInt(json.duration - Math.random(1,
2));

                        var enc = '[' + mArg.defaults.clazzId + '][' +
mArg.defaults.userid + '][' +
```

```
                            mArg.attachments["0"].property._jobid + '][' +
mArg.attachments["0"].objectId + '][' +
                            (playTime * 1000).toString() + '][d_yHJ!$pdA~5][' +
(json.duration * 1000).toString() + '][0_' +
                            json.duration + ']';
                        enc = hex_md5(enc);
                        get('/multimedia/log/' + json.dtoken + '?clipTime=0_' +
json.duration +
                            '&otherInfo=' + mArg.attachments["0"].otherInfo +
                            '&userid=' + mArg.defaults.userid + '&rt=0.9&jobid='
+ mArg.attachments["0"].property._jobid +
                            '&duration=' + json.duration +
'&dtype=Video&objectId=' + mArg.attachments["0"].objectId +
                            '&clazzId=' + mArg.defaults.clazzId +
                            '&view=pc&playingTime=' + playTime +
'&isdrag=4&enc=' + enc).onreadystatechange = function () {
                            if (this.readyState == 4) {
                                if (this.status != 200) {
                                    alert('未知错误');
                                } else {
                                    alert('成功刷新后查看效果');
                                }
                            }
                        }
                    }
                }

            }
        }
    }
}

function substrEx(str, left, right) {
    var leftPos = str.indexOf(left) + left.length;
    var rightPos = str.indexOf(right, leftPos);
    return str.substring(leftPos, rightPos);
}

function createRequest() {
    var xmlhttp;
    if (window.XMLHttpRequest) {
        xmlhttp = new XMLHttpRequest();
    } else {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    return xmlhttp;
}

function get(url) {
    try {
        var xmlhttp = createRequest();
        xmlhttp.open("GET", url, true);
        xmlhttp.send();
    } catch (e) {
        return false;
    }
    return xmlhttp;
}
```

```
function createBtn(title) {
    var btn = document.createElement('button');
    btn.innerText = title;
    btn.style.outline = 'none';
    btn.style.border = '0';
    btn.style.background = '#7d9d35';
    btn.style.color = '#fff';
    btn.style.borderRadius = '4px';
    btn.style.padding = '2px 8px';
    btn.style.cursor = 'pointer';
    btn.style.fontSize = '12px';
    return btn;
}
```

start.js

```
window.onload = function () {
    //注入mooc.js
    injected('mooc.js');
    injected('md5.js');
}

function injected(file) {
    var path = 'js/' + file;
    var temp = document.createElement('script');
    temp.setAttribute('type', 'text/javascript');
    temp.src = chrome.extension.getURL(path);
    document.head.appendChild(temp);
}
```