Use XDomain instead. However, if you MUST use CORS:

IE8/9 CORS Polyfill

http://echo.jpillora.com/cors-test (cross domain)


```
{
  "time": "2018-07-30T12:16:32.098563445Z",
  "duration": "1.011096ms",
  "location": "CN-LAX",
  "ip": "113.108.226.41",
  "proto": "http",
  "host": "echo.jpillora.com",
  "method": "GET",
  "path": "/%60cors-test%60",
  "headers": {
    "accept": "*/*",
    "accept-encoding": "gzip",
    "accept-language": "zh-CN,zh;q=0.9",
    "cache-control": "no-cache",
    "origin": "http://jpillora.com",
    "pragma": "no-cache",
    "referer": "http://jpillora.com/xhook/example/ie-8-9-cors-polyfill.html",
    "user-agent": "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "via": "1.1 WEBPROXY06"
  },
  "body": ""
}
```


example2.txt (same domain)

```
this is the second text file example (example2.txt)
```

Warning: Due to limitations with IE's XDomainRequest:

1. requests MUST use the protocol of the current page

2. only these 3 events work: 'progress','timeout','error'

3. request headers do not work, and only the content-type response header works

4. only GET or POST may be used requests

5. requests must be async

6. 'withCredentials' will not work

Usage Guide

1. Include XHook

2. Install this before-hook

3. Make normal XMLHttpRequests, if user has crossdomain and IE8/9, they will be silently swapped out for XDR

```
xhook.before(function(request, callback) {
  //skip browsers that dont use XDR
  if(!window.XDomainRequest)
    return callback();
  //skip requests that aren't cross domain
  var url = request.url;
  var loc = window.location;
  var hostname = loc.hostname + (loc.port ? ":"+loc.port : "");
  if(!/^https?:\/\/([^\?\/]+)/.test(url) || RegExp.$1 === hostname)
    return callback();

  //if not GET, force POST
  var method = request.method;
  if(method !== 'GET') method = 'POST';
  //force same protocol
  url = url.replace(/^https?:/,loc.protocol);
```

```
  //request!
  var xdr = new window.XDomainRequest();
  xdr.timeout = request.timeout;
  //proxy events
  var proxy = function(e) {
    xdr['on'+e] = function() {
      request.xhr.dispatchEvent(e);
    };
  };
  var events = ['progress','timeout','error'];
  for(var i = 0; i < events.length; ++i )
    proxy(events[i]);
  //custom onload
  xdr.onload = function() {
    callback({
      status: 200,
      statusText: "OK",
      headers: {
        'Content-Type': xdr.contentType
      },
      text: xdr.responseText
    })
  };
  xdr.open(method, url);
  xdr.send(request.body);
  return
});
```