# XMLHttpRequest

XMLHttpRequest是一个API,它为客户端与服务器之间传输数据的功能。（而不会使得页面刷新）

XMLHttpRequest构造函数

new XMLHttpRequest（）

## 实例的属性

statusText

withCredentials

在进行跨域请求时，是否使用认证信息（cookie 或者授权header）

## 实例的方法与属性

### 请求头相关

**设置请求头**

setRequestHeader

**获取响应头部**

getAllResponseHeaders() | getResponseHeader()

但是获取的头部是有限制的

1. 无论是否跨域， 标准规定不能获取 Set-Cookie Set-Cookie2
2. 对于跨域，标准规定只能获取‘simple response header’和‘Access-Control-Expose-Headers’中运行客户端访问的头部

### 请求状态

**状态码**

status

**请求状态**

readyState

（0： open未被调用 1：send未被调用； 2：响应状态已经返回； 3：响应主体正在下载 4：请求完成 ）

**请求状态变更**

onreadystatechange

**事件**当readyState改变时会调用它

**设置请求超时时间**

timeout

如果是一个sync同步请求，timeout必须设置为0，否则抛错。

send后仍然可设置timeout，起点仍然是send发送的时刻。

**请求开始与结束**

onloadstart

事件当 send时，才会触发，数据开始传输

onloadend

### 返回的数据

**指定返回数据类型**

level 1 中提供了overrideMimeType()方法。

其重写response中的content-type来实现

level 2 中提供了 responseType 属性

可以指定xhr.response的数据类型。更为强大

兼容问题：

**获取返回数据**

xhr.response

请求完成时，此属性才有正确值

xhr.responseText

只有当responseType 为 ‘text’ ， ‘’时，此属性才存在，否则调用此属性会报错。

xhr.responseXML

只有 responseType 为 ‘text’, ‘’ ,’document’时，此属性才存在，否则调用会报错。

## 同步请求

xhr.open(url,method, false)

同步请求的限制

timeout 必须为0

withCredentials必须为false 注意：同源时，设置这个不会有任何影响，都会发送cookie

responseType 必须为 “text/json” (不完全确定，所有都适用？)

## 上传下载

upload

上传过程**事件**

### 进度

xhr.onprogress

下载进度

xhr.upload.onprogress

上传进度

**abort（）**

立即终止请求

返回所有响应头信息或者指定响应头信息

**open( method , url , async, user, password )**

**send()**

所有事件绑定必须在send之前绑定