

## uexXmlHttpMgr

管理网络请求，可以进行跨域访问。

### 方法

open 创建一个请求对象

setPostData 设置 post 请求数据

setCertificate 设置本次请求使用的数字证书

send 发送请求

setInputStream 设置一个本地文件地址

close 关闭一个请求对象

setHeaders 设置请求头

setBody 设置 post 请求发送的数据体

setAppVerify 设置是否在请求头中加入 appVerify 字段

### 监听方法

onData 请求发送完成的监听方法

onPostProgress 发送 Post 请求，发送进度改变的监听方法

### open

创建一个请求对象

uexXmlHttpMgr.open(id, method, url, timeOut )

说明：

创建一个请求对象，同一 id 只能被创建一次。

参数：

id: (*Number 类型*)必选 请求对象的唯一标识符

method: (*String 类型*)必选 请求方式，get 或 post

url: (*String 类型*)必选 请求地址

timeOut: (*Number 类型*)必选 请求的超时时间，单位为毫秒，默认 30\*1000 毫秒。为空或等于 0 时，取默认值

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

### send

发送请求

uexXmlHttpMgr.send(id, flag)

说明：

发送请求。

参数:

id: (*Number 类型*)**必选** 请求对象的唯一标识符

flag: (*Number 类型*)**可选** 本次请求的 flag。取值范围: 1、2、3。(仅 Android 支持)

当 flag 传入 1 时, 表示通知底层, 将本次请求的 header、状态码、结果等输出 log 到日志文件。

当 flag 传入 2 时, 表示通知底层, 如果本次请求发生了异常, 就将本次请求的异常输出 log 到日志文件。

当 flag 传入 3 时, 表示通知底层, 0x1 和 0x2 的操作均需要。

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

## close

关闭一个请求对象

uexXmlHttpMgr.close(id)

说明:

根据唯一标识符关闭请求对象。

参数:

id: (*Number 类型*)**必选** 请求对象的唯一标识符

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

## setPostData

设置 post 请求数据

uexXmlHttpMgr.setPostData(id,dataType, key, value)

说明:

设置 post 请求, 可以发送一次或多次。(仅 post 方式支持)

参数:

id: (*Number 类型*)**必选** 请求对象的唯一标识符

dataType: (*Number 类型*)**必选** 请求中要发送的数据格式, 详见 `CONTENT` 中 `XmlHttpRequest DataType`

key: (*String 类型*)**必选** 请求数据中的对应的键

value: (*String 类型*)**必选** 请求数据中的对应的值

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

### **setInputStream**

设置一个本地文件地址

`uexXmlHttpMgr.setInputStream(id,filePath)`

说明:

设置一个本地文件地址，`uexXmlHttpMgr` 对象向服务器发起 POST 请求时，直接将此文件二进制流提交至服务器，中间不带任何文件分割符、Mimeype 等 form 表单中的元素。(仅 post 方式请求支持)

参数:

`id`: (*Number 类型*)**必选** 请求对象的唯一标识符

`filePath` (*String 类型*)**必选** 文件路径，详见 CONSTANT 中 Path Types

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

### **setBody**

设置 post 请求发送的数据体

`uexXmlHttpMgr.setBody(id,data)`

说明:

设置 post 请求发送的数据体。(仅 post 方式请求支持)

参数:

`id`: (*Number 类型*)**必选** 请求对象的唯一标识符

`data`: (*String 类型*)**必选** 请求中要发送的数据体

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

### **setAppVerify**

设置是否在请求头中加入 appVerify 字段

`uexXmlHttpMgr.setAppVerify(id, isVerify)`

说明:

设置是否在请求头中加入 `appVerify` 字段, 字段格式如下 `appVerify:md5=appKey+appid;`  
`ts`=本次请求的时间戳,这些是插件内部处理。

参数:

`id`: (*Number 类型*) 必选 请求对象的唯一标识符

`isVerify`: (*Number 类型*) 必选

0:不在请求头中增加 `appVerify` 字段, 对本次请求不做任何处理

1:在请求头中增加 `appVerify` 字段, `get`、`post` 都会增加

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

## **setHeaders**

设置请求头

`uexXmlHttpMgr.setHeaders(id, json)`

说明:

设置请求头

参数:

`id`: (*Number 类型*) 必选 请求对象唯一标识符

`json`: (*String 类型*) 必选 发送请求之前设置 `http` 的协议头, `json` 数据格式。

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

## **setCertificate**

设置本次请求使用的数字证书

`uexXmlHttpMgr.setCertificate(id, psw, path)`

说明:

设置本次异步请求使用的数字证书。

参数:

`id`: (*Number 类型*) 必选 请求对象的唯一标识符

`psw`: (*String 类型*) 必选 数字证书密码。当使用 `appcan` 默认证书时, 此参数为空(或“ ”)

`path`: (*String 类型*) 必选 证书路径, 支持 `file://`, `res://`, `wgt://` 等协议路径, 详见 `CONSTANT` 中 `Path Types`。当传入 ‘default’ 时, 本次请求将取 `appcan` 默认数字证书

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

## onData

请求发送完成的监听方法

uexXmlHttpMgr.onData(id, status, result, requestCode )

参数:

id: (*Number 类型*) 必选 请求对象的唯一标识符

status: (*Number 类型*) 必选 返回的状态 详见 CONSTANT 中 XmlHttpRequest Status

result: (*String 类型*) 必选 服务器返回的任意数据,包含错误

requestCode : (*Number 类型*) 可选 http 请求返回的状态码

版本支持:

3.0.0+

## onPostProgress

发送 Post 请求, 发送进度改变的监听方法

uexXmlHttpMgr.onPostProgress(id, progress)

参数:

id: (*Number 类型*) 必选 请求对象的唯一标识符

progress: (*Number 类型*) 必选 当有文件通过 POST 方式上传时, 该值表示上传进度

版本支持:

3.0.0+

示例:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>跨域异步请求接口</title>
```

```
<script type="text/javascript">
```

```
function sendGetRequest(){
```

```
    var
```

```
    url
```

```
=
```

```
"http://www.kedo.gov.cn/mobile/index/GetIndexInfo.jsp?jsoncallback=?&type=1";
```

```
    var headJson = '{"Content-type":"application/json;charset=utf-8"}';
```

```
    uexXmlHttpMgr.open("1", "get", url, "");
```

```
    uexXmlHttpMgr.setAppVerify("1","1");
```

```
    uexXmlHttpMgr.setCertificate("1","", "default");
```

```
    uexXmlHttpMgr.setHeaders("1",headJson);
```

```
    uexXmlHttpMgr.send("1");
```

```
}
```

```

function sendPostRequest(){
    var url = "http://192.168.1.130:8081/xxxx/public/masPro1/test";
    uexXmlHttpMgr.open("2", "POST", url, "");
    var v1 = "field";
    var file = "/storage/emulated/0/widgetone/tmp/12.jpg";
    uexXmlHttpMgr.setPostData("2", "0", "field", v1);
    //uexXmlHttpMgr.setInputStream("2", file);
    uexXmlHttpMgr.send("2");
}
function onData(id,state,result,requestCode){
    alert("onData->id = " + id + ",result = " + result);
    uexXmlHttpMgr.close(id);
}
function onPostProgress(id,inProgress){
    alert("onPostProgress->id = " + id + ",inProgress = " + inProgress);
}
window.uexOnload = function(){
    uexXmlHttpMgr.onData = onData;
    uexXmlHttpMgr.onPostProgress = onPostProgress;
}
</script>
</head>
<body>
    <input class="btn" type="button" value="发送 Get 请求" onclick=sendGetRequest()>
    <input class="btn" type="button" value="发送 Post 请求" onclick=sendPostRequest()>
</body>
</html>

```