# Send Request đến Server

## Định nghĩa

Để gửi một yêu cầu đến server, ta thường dùng phương thức open() và send() của đối tượng XMLHTTPRequest.

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| **open(method, url, async)** | Chỉ ra loại yêu cầu, đường dẫn, và có đồng bộ hay không   * method: loại request: GET hay POST * url: đường dẫn của file trên server * async: true (bất đồng bộ) hay false (đồng bộ) |
| **send(string)** | Gửi yêu cầu đến server  string: Chỉ được dùng cho POST request |

## Tính chất

### GET hay POST?

GET và POST khác nhau như thế nào và nên dùng vào khi nào? Trên lý thuyết, sử dụng GET khi request không làm thay đổi giá trị, nghĩa là khi có nhiều request đều sẽ trả về kết quả như nhau. Tóm lại, ta dùng GET để truy xuất dữ liệu từ server, hay nói cách khác là không làm thay đổi trạng thái trên server.

*Ví dụ:*

<html>

<head>

<script type="text/javascript">

function loadXMLDoc()

{

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("GET","demo\_get.asp",true);

xmlhttp.send();

}

</script>

</head>

<body>

<h2>AJAX</h2>

<div id="myDiv"></div>

<button type="button" onclick="loadXMLDoc()">Request data</button>

</body>

</html>

Ta dùng POST mỗi khi muốn thay đổi trạng thái trên server. Đó là khi ta cần cập nhật dữ liệu trên server, hay gửi một số lượng lớn dữ liệu đến server do POST không có giới hạn về kích thước dữ liệu được gửi.

*Ví dụ:*

<html>

<head>

<script type="text/javascript">

function loadXMLDoc()

{

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("POST","demo\_post.asp",true);

xmlhttp.send();

}

</script>

</head>

<body>

<h2>AJAX</h2>

<div id="myDiv"></div>

<button type="button" onclick="loadXMLDoc()">Request data</button>

</body>

</html>

Để POST dữ liệu như một HTML, ta cần phải thiết lập phần Content-Type header trên đối tượng XMLHttpRequest như sau:

xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

*Ví dụ:*

<html>

<head>

<script type="text/javascript">

function loadXMLDoc()

{

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function()

{

if (xmlhttp.readyState==4 && xmlhttp.status==200)

{

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;

}

}

xmlhttp.open("POST","demo\_post2.asp",true);

xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

xmlhttp.send("fname=Henry&lname=Ford");

}

</script>

</head>

<body>

<h2>AJAX</h2>

<div id="myDiv"></div>

<button type="button" onclick="loadXMLDoc()">Request data</button>

</body>

</html>

GET đơn giản hơn và nhanh hơn POST, và có thể được sử dụng trong hầu hết các trường hợp.

### Đường dẫn

Tham số url trong phương thức open() là một địa chỉ đến một file trên server.

File này có thể là bất kì loại file nào.

*Ví dụ:*

xmlhttp.open("GET","ajax\_test.asp",true);

### Asynchronous

Để truy cập dữ liệu theo kiểu bất đồng bộ (asynchronous), thì tham số async trong phương thức open() phải được thiết lập là true.