



AJAX 2: JQuery AJAX

FDP 5.0



Nội dung

- ❖ Giới thiệu
- ❖ HTTP Request: GET vs POST
- ❖ HTTP Request: PUT vs AJAX



GIỚI THIỆU

- ❖ Viết mã AJAX thông thường có thể hơi phức tạp, vì các trình duyệt khác nhau có cú pháp khác nhau để triển khai AJAX.
- ❖ Bạn sẽ phải viết thêm mã để kiểm tra các trình duyệt khác nhau.
- ❖ jQuery AJAX ra đời giúp chúng ta có thể viết chức năng AJAX chỉ với một dòng mã duy nhất.

GIỚI THIỆU

❖ Để sử dụng jQuery AJAX cần cài đặt thư viện jQuery vào trong mã nguồn:

- Nhúng cdn thư viện jQuery

<https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js>

- Tải và cài lưu mã nguồn vào project tại <https://jquery.com/download/>

HTTP Request GET vs POST

❖ Hai phương thường dùng nhất để phản hồi các yêu cầu từ phía server

- GET – lấy dữ liệu từ server
- POST – gửi dữ liệu lên server

Cú pháp của GET: `$.get(url [, data] [, success] [, dataType])` hoặc `$.get([setting])` trong đó:

- ❖ **url**: địa chỉ gửi yêu cầu
- ❖ **data**: Dữ liệu cần gửi lên server
- ❖ **success**: hàm callback, sẽ được gọi khi yêu cầu lên server hoàn tất.
- ❖ **dataType**: là kiểu dữ liệu mà ta muốn nhận từ server. Gồm các giá trị: xml, json, script, text, html.

HTTP Request GET vs POST

❖ Ví dụ

```
$("#button").click(function(){  
    $.get("demo_test.asp", function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

Một trang được nhúng trên trang này cho biết

Data: This is some text from an external ASP file.

Status: success

OK

HTTP Request GET vs POST

Cú pháp của POST: `$.post(url [, data] [, success] [, dataType])` hoặc `$.post([setting])` trong đó:

- ❖ **url**: địa chỉ gửi yêu cầu
- ❖ **data**: Dữ liệu cần gửi lên server
- ❖ **success**: hàm callback, sẽ được gọi khi yêu cầu lên server hoàn tất.
- ❖ **dataType**: là kiểu dữ liệu mà ta muốn nhận từ server. Gồm các giá trị: xml, json, script, text, html.

HTTP Request GET vs POST

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.post("demo_test_post.asp",
    {
      name: "Donald Duck",
      city: "Duckburg"
    },
    function(data,status){
      alert("Data: " + data + "\nStatus: "
    });
  });
});
</script>
</head>
<body>

<button>Send an HTTP POST request to a page and get the result back</button>

</body>
</html>
```

Một trang được nhúng trên trang này cho biết

Data: Dear Donald Duck. Hope you live well in Duckburg.
Status: success

OK

Send an HTTP POST request to a page and get the result back

HTTP Request GET vs POST

Một điều quan trọng là nội dung của một POST request thì không bị giới hạn, chỉ cần server mà ta muốn gửi thông tin lên, nó hỗ trợ được bao nhiêu thì ta có thể gửi bấy nhiêu.

Như vậy, khi nào sử dụng GET và khi nào sử dụng POST? câu trả lời là do server quy định, nếu server quy định là GET thì chúng ta phải dùng \$.get, còn nếu server quy định là POST thì chúng ta phải dùng \$.post

HTTP Request: PUT vs AJAX

- ❖ Trong jQuery không có cung cấp shorthand method cho HTTP method là PUT. Do đó, chúng ta sẽ sử dụng \$.ajax
- ❖ Cú pháp của \$.ajax như sau:

```
$.ajax(url [, settings ])
```

- ❖ Trong đó url là bắt buộc
- ❖ Setting là các thuộc tính tùy chọn có thể thêm vào hoặc không
- ❖ Bảng thống kê sau bao gồm các setting thường xuyên sử dụng

HTTP Request: PUT vs AJAX

Name	Value/Description
accepts	A Boolean value indicating whether the request should be handled asynchronous or not. Default is true
async	Cho biết dữ liệu gửi đi có được xử lý đồng bộ hay không. Mặc định là có
contentType	Loại nội dung được sử dụng khi gửi dữ liệu đến máy chủ. Mặc định là: "application / x-www-form-urlencoded"
method	Phương thức gửi hoặc nhận dữ liệu: GET POST PUT
header	Cho phép gửi thêm header lên server
success	Một hàm sẽ được kích hoạt khi request gửi thành công
data	Dữ liệu cần gửi lên server
timeout	Giới hạn thời gian chờ mà request gửi đi.
dataType	Loại dữ liệu mà chúng ta mong đợi từ server. Có thể là: xml, json, script, hoặc html.

HTTP Request: PUT vs AJAX

❖ Đối với HTTP Request PUT chúng ta sử dụng method – PUT: cập nhật dữ liệu hiện đang có.

```
$.ajax({  
  url: 'http://example.com/api',  
  type: 'PUT',  
  data: "name=John&location=Boston",  
  success: function(data) {  
    alert('Load was performed.');  }  
});
```

HTTP Request: PUT vs AJAX

❖ Đối với HTTP Request PUT chúng ta sử dụng method – PUT: cập nhật dữ liệu hiện đang có.

```
$.ajax({  
  url: 'http://example.com/api',  
  type: 'PUT',  
  data: "name=John&location=Boston",  
  success: function(data) {  
    alert('Load was performed.');  }  
});
```

Thank you and
happy learning !!!