

Chương 6: Kỹ thuật lập trình AJAX

Giảng Viên: ThS. Tạ Việt Phương

1

1

Nội dung

- AJAX là gì?
- Cách dùng AJAX
- Một số ví dụ minh họa



Phát triển ứng dụng Web

2

2

AJAX là gì?

- AJAX: **A**synchronous **J**avascript **a**nd **X**ML. Là sự kết hợp giữa Javascript và XML
- AJAX không phải là một ngôn ngữ lập trình mà là một cách thức làm việc mới, giúp trang web trở nên linh hoạt và nhanh nhạy hơn bằng cách tải lại một phần dữ liệu mà không cần tải lại toàn bộ trang
- Chỉ cập nhật dữ liệu tại những nơi cần thiết trong trang web
- Dùng để nâng cao tốc độ xử lý của trang web.
- Giúp trang web động hơn
- Giúp trang web thân thiện với người dùng hơn, cải thiện trải nghiệm người dùng (User Experience)

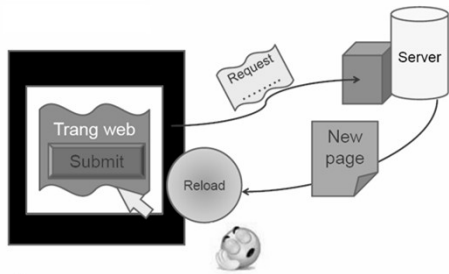


Phát triển ứng dụng Web

3

3

Trang web không dùng AJAX



Phát triển ứng dụng Web 4

4

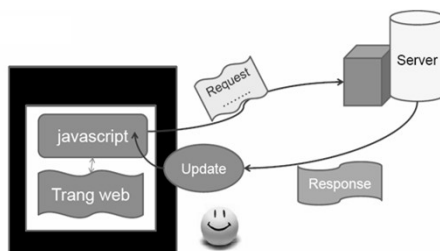
Trang web không dùng AJAX

- Hạn chế:
 - Khi người dùng thao tác thì server “ngủ” và ngược lại
 - Lãng phí thời gian, hiệu quả sử dụng thấp
 - Người dùng phải vừa làm vừa đợi: **gửi yêu cầu → đợi → nhận kết quả → gửi yêu cầu → đợi → ...** ⇒ Người dùng phải đợi lâu nếu yêu cầu xử lý lớn và server mất nhiều thời gian xử lý + Hiển thị không liên tục
 - Toàn bộ trang HTML mới được gửi từ server đến client
 - Không cần thiết vì có thể nhiều chi tiết trên trang mới vẫn như trang cũ
 - Lượng thông tin trao đổi giữa client-server lớn ⇒ chi phí truyền thông (thời gian, băng thông) lớn

Phát triển ứng dụng Web

5

Trang web dùng AJAX



Phát triển ứng dụng Web 6

6

Trang web dùng AJAX

- Ajax engine được cài trên client, làm nhiệm vụ giao tiếp trung gian giữa browser với web server
 - Browser gửi yêu cầu đến Ajax engine bằng lời gọi Javascript.
 - Ajax engine (Một lớp trung gian, được thực thi bằng XMLHttpRequest hoặc Fetch API, chịu trách nhiệm quản lý giao tiếp giữa trình duyệt và server) chuyển yêu cầu của Client thành HTTP request và gửi cho web server
 - Web server xử lý yêu cầu, thực thi mã backend rồi gửi kết quả cho Ajax engine ở dạng XML
 - Ajax engine biên dịch XML thành HTML và gửi HTML cho browser
- Một yêu cầu của người dùng chưa cần được giải quyết xong thì người dùng đã có thể đưa ra yêu cầu khác
 - Trao đổi giữa Browser với Ajax engine và giữa Ajax engine với Server để thực hiện các yêu cầu diễn ra không đồng bộ.



Phát triển ứng dụng Web

7

Trang web dùng AJAX

- Ưu điểm
 - Người dùng và server thực hiện một cách song hành
 - Không lãng phí thời gian, hiệu quả sử dụng cao
 - Người dùng không phải chờ làm vừa đợi
 - Hiện thị liên tục, không gây khó chịu
 - Chỉ phần khác biệt của trang mới so với trang cũ mới được gửi từ server đến client
 - Lượng thông tin trao đổi giữa client-server tối thiểu ⇒ tiết kiệm chi phí (thời gian, băng thông) truyền thông.



Phát triển ứng dụng Web

8

Trang web dùng AJAX

- Vì sao dùng Ajax
 - Để tạo ra các ứng dụng web mà giao tiếp của nó với người dùng diễn ra như giao tiếp của ứng dụng Winform với người dùng: liên tục.
 - Hiệu quả trong sử dụng và trong truyền thông




Phát triển ứng dụng Web

9

Bất đồng bộ

- **Bất đồng bộ (Asynchronous)** trong AJAX là khả năng xử lý yêu cầu (request) và nhận phản hồi (response) mà không làm gián đoạn hoặc chặn luồng chính (main thread) của trình duyệt.
- Điều này có nghĩa là:
 - Khi một yêu cầu AJAX được gửi đến server, trình duyệt không cần đợi server trả lời.
 - Trình duyệt vẫn có thể xử lý các thao tác khác của người dùng như nhấp chuột, điền biểu mẫu, hoặc di chuyển chuột.



Đại học Khoa học và Công nghệ Hà Nội

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ HÀ NỘI


UT-HCM

Phát triển ứng dụng Web10

10

Bất đồng bộ

- **Cách hoạt động bất đồng bộ**
 - Yêu cầu AJAX được gửi đi và chạy trong nền (background).
 - Trình duyệt không bị khóa trong quá trình server xử lý yêu cầu.
 - Khi server phản hồi xong, dữ liệu sẽ được xử lý bởi một callback function hoặc Promise, lúc đó giao diện mới được cập nhật.



Đại học Khoa học và Công nghệ Hà Nội

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ HÀ NỘI

UT-HCM


Phát triển ứng dụng Web11

11

Bất đồng bộ

- **Ví dụ:**

```
function loadData() {
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "server.php", true); // true: bất đồng bộ
  xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
      console.log("Dữ liệu nhận được:", xhr.responseText);
    }
  };
  xhr.send();
}
console.log("Yêu cầu được gửi!");
```



Đại học Khoa học và Công nghệ Hà Nội

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ HÀ NỘI

UT-HCM

Phát triển ứng dụng Web12

12

Cách dùng AJAX

- **Bước 1:** Tạo đối tượng XMLHttpRequest
- **Bước 2:** Gửi request lên Server
- **Bước 3:** Nhận Response từ Server gửi về
- **Bước 4:** Xử lý dữ liệu sau khi nhận được từ Server

13

Cách dùng AJAX

```
function Click(){
    var xmlhttp;
    //Khởi tạo đối tượng xmlhttp
    xmlhttp.open("GET","URL",true)
    xmlhttp.send(null);
    xmlhttp.onreadystatechange=function() {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            Xử lý Response trả về
        }
    }
}
```

echo("noi dung");

Client

Server

14

Khởi tạo đối tượng xmlhttp

- Mozilla, Safari, Chrome, IE7+
 - xmlhttp = new XMLHttpRequest();
- IE những Version trước
 - xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

15

Sử dụng AJAX

- Các phương thức của XMLHttpRequest
- open(method, server, async): đặt các tham số cho request
 - Method: Quy định hình thức vận chuyển dữ liệu đến server. Có 2 hình thức: POST/GET
 - Server: Trang web nhận và xử lý tham số sau đó trả kết quả về client thông qua thuộc tính.responseText
 - Async: Quy định cơ chế truyền thông (true: bất đồng bộ, false: đồng bộ)
- send: gửi request đến server



Phát triển ứng dụng Web

16

Sử dụng AJAX

- GET:
 - Obj.open("GET", "demo.php?id=a&cat=b", true);
 - Obj.onreadystatechange = ten_ham_callback;
 - Obj.send(null);
- POST:
 - Obj.open("POST", "demo.php", true);
 - Obj.onreadystatechange = handleRequestStateChange;
 - Obj.send("id=a&cat=b");



Phát triển ứng dụng Web

17

Gửi request lên Server

- Gửi biến đi kèm bằng phương thức GET
 - Chuẩn bị URL
 - var url="getalbum.php";
 - url=url+"?emp_id="+emp_value;
 - Gởi nhiều biến đi kèm:
 - url=url+"?emp_id="+emp_value+"&"+"id="+Bingqilin"
- Gửi Request lên Server
 - xmlhttp.open("GET",url,true);
 - xmlhttp.send(null);



Phát triển ứng dụng Web 18

18

Gửi request lên Server

- Gửi biến đi kèm bằng phương thức POST
 - Chuẩn bị URL
 - `var url="getalbum.php";`
 - Chuẩn bị biến gửi lên
 - `params="emp_id="+emp_value;`
 - Gửi nhiều biến:
 - `params="emp_id="+emp_value+"&"+"id="+PTUDW";`
 - Gửi Request lên Server
 - `xmlHttp.open("POST",url,true);`
 - `xmlHttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");`
 - `xmlHttp.send(params);`

19

Nhận Response trả về từ Server

- `xmlHttp.onreadystatechange = Function;`
- Các trạng thái của thuộc tính **readyState**

Trạng thái	Ý nghĩa
0	Request chưa được khởi tạo
1	Request đã được thiết lập
2	Request đã được gửi
3	Request đang được xử lý
4	Request được xử lý xong

20

Nhận response trả về từ Server

- Nhận kết quả trả về
 - `xmlHttp.responseText`: khi server trả về là chuỗi;
 - `xmlHttp.responseXML`: khi server trả về là file dạng xml;
- Xử lý kết quả trả về
 - `document.getElementById('text').innerHTML = Kết quả lấy được;`

21

Nhận response trả về từ Server (dạng chuỗi)

• Cách viết code 1:

```
xmlHttp.onreadystatechange = function() {
    if (xmlHttp.readyState==4 && xmlHttp.status==200) {
        var showdata = xmlHttp.responseText;
        document.getElementById('text').innerHTML=showdata;
    }
}
```

22

Nhận response trả về từ Server (dạng chuỗi)

• Cách viết code 2:

```
xmlHttp.onreadystatechange=stateChanged ;
function stateChanged() {
    if (xmlHttp.readyState==4 && xmlHttp.status==200) {
        var showdata = xmlHttp.responseText;
        document.getElementById('text').innerHTML=showdata;
    }
}
```

23

Ví dụ: file index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>AJAX với Javascript</title>
    <meta charset="UTF-8">
    <script src="AJAXex.js"></script>
</head>
<body>
    <div id="test"></div>
    <form>
        <label>User Name: <input type="text" name="namefield" id="txtName"> </label>
        <label><br>Password: <input type="text" name="pwdfield" id="txtPass"> <br></label>
        <input name="submit" type="button" onClick="postData()" value="Lay du lieu bang POST">
    </form>
    <div id="targetDiv">
    </div>
</body>
</html>
```

24

Ví dụ: file Javascript xử lý AJAX

```
//Gửi bằng phương thức POST
function postData() {
    if (xmlHttp) {
        var obj = document.getElementById("targetDiv");
        var url = "data.php";
        var param = "name=" + txtName.value + "&pass=" + txtPass.value;
        xmlHttp.open("POST", url, true);
        xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
        xmlHttp.send(param);
        //Đón dữ liệu từ server trả về
        xmlHttp.onreadystatechange = function() {
            if (xmlHttp.readyState == 4 && xmlHttp.status == 200) {
                obj.innerHTML = "<p>" + xmlHttp.responseText + "</p>";
            }
        }
    }
}
```



Phát triển ứng dụng Web 25

25

Ví dụ: file gửi kết quả trên server

```
<?php
if (isset($_POST['name'])) {
    $username = $_POST['name'];
    $pass = $_POST['pass'];
    echo "<h3>Đăng nhập thành công với Post</h3>";
    $str = "";
    $str .= '<table border="1" cellspacing="0" cellpadding = "10">';
    $str .= '<tr><th>Username</th><th>Password</th></tr>';
    $str .= '<tr><td>'. $username . '</td><td>'. $pass . '</td></tr><';
    $str .= '</table>';
    echo $str;
}
```



Phát triển ứng dụng Web 26

26

Ví dụ: Kết quả



Phát triển ứng dụng Web 27

27

Nhận response trả về từ Server (file xml)

```
<?xml version="1.0" encoding="utf-8" ?>
<persons>
  <person>
    <name>John</name>
    <age>30</age>
  </person>
  <person>
    <name>Jim</name>
    <age>25</age>
  </person>
  <person>
    <name>Jane</name>
    <age>23</age>
  </person>
</persons>
```



28

Nhận response trả về từ Server (file xml)

• Cách 1:

```
xmlHttpRequest.onreadystatechange = function() {
  if (xmlHttpRequest.readyState == 4 && xmlHttpRequest.status == 200) {
    var names = xmlHttpRequest.responseXML.getElementsByTagName("person");
    var output = "<ul>";
    for (var i = 0; i < names.length; i++) {
      output += "<li>" + names[i].getElementsByTagName("name")[0].firstChild.nodeValue + "</li>";
      output += "<li>" + names[i].getElementsByTagName("age")[0].firstChild.nodeValue + "</li>";
    }
    output += "</ul>";
    obj.innerHTML = output;
  }
}
```



29

Nhận response trả về từ Server (file xml)

• Cách 2:

```
xmlHttpRequest.onreadystatechange = function() {
  if (xmlHttpRequest.readyState == 4 && xmlHttpRequest.status == 200) {
    var names = xmlHttpRequest.responseXML.getElementsByTagName("person");
    var str = "";
    str += '<table border="1" cellspacing="0" cellpadding="10">';
    str += '<tr><th>Name</th><th>Age</th></tr>';
    for (var i = 0; i < names.length; i++) {
      str += "<tr>";
      str += "<td>" + names[i].childNodes[0].textContent + "</td>";
      str += "<td>" + names[i].childNodes[1].textContent + "</td>";
      str += "</tr>";
    }
    str += "</table>";
    result.innerHTML = str;
  }
}
```



30

Tạo file xml từ php

```
<?php header('Content-type: application/xml');
$conn = mysqli_connect('localhost','root','','glv') or die ('Can not connect to mysql');
$query = mysqli_query($conn, 'select * from lop');
```

```
$doc = new DOMDocument('1.0');
$root = $doc->createElement('root');
$doc->appendChild($root);

if (mysqli_num_rows($query) > 0){
    while ($row = mysqli_fetch_array($query, MYSQLI_ASSOC)){
        $items = $doc->createElement('items');
        $name = $doc->createElement('username');
        $name->appendChild($doc->createTextNode($row['NAME']));
        $items->appendChild($name);
        $pass = $doc->createElement('password');
        $pass->appendChild($doc->createTextNode($row['PASSWORD']));
        $items->appendChild($pass);
        $root->appendChild($items);
    }
}
$xml = $doc->saveXML();
echo ($xml);

$xmlname = 'vd.xml';
$doc->save($xmlname);
}
}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <items>
    <username>1234</username>
    <password>PTUDW</password>
  </items>
  <items>
    <username>1235</username>
    <password>JAVA</password>
  </items>
</root>
```

Phát triển ứng dụng Web 31

31

AJAX với JQuery

- JQuery cung cấp 3 cách gửi yêu cầu lên Server khi sử dụng kỹ thuật lập trình AJAX:

- Hàm GET
- Hàm POST
- Hàm AJAX



Phát triển ứng dụng Web 32

32

Dùng hàm GET

- Dùng hàm `$.get` để gửi yêu cầu lên Server
- Cú pháp: `$.get(URL, callback)` ;
 - **URL**: Tập tin xử lý yêu cầu được gửi lên Server, kèm theo biến và giá trị đi kèm trong URL.
 - Các biến được ngăn cách bởi ký tự '&'.
 - Server dùng phương thức GET để lấy dữ liệu của biến đi kèm
 - **callback**: Phương thức nhận kết quả trả về từ Server



Phát triển ứng dụng Web 33

33

Lấy dữ liệu trả về từ Server

```
$.get("get.php?ten="+giá trị,function(data,status){
    $("#iddiv").html(data);
});
```

- o **data**: Dữ liệu trả về từ Server
- o **status**: status = "success" khi Server xử lý xong yêu cầu.
- o `$("#iddiv").html(data)`: Hiển thị dữ liệu trong tag <div>

34

Ví dụ

```
function load_get(){
    name = "test";
    $.get("result.php?name=" + name ,function(data,status){
        $('#result').html(data);
    });
}

<button name="btnGet" onclick="load_get()">Lấy dữ liệu với hàm
get()</button>
```

35

Dùng hàm POST

- Dùng hàm **\$.post** để gửi yêu cầu lên server
- Cú pháp **\$.post(URL,data,callback)** ;
 - o **URL**: Tập tin xử lý yêu cầu
 - o **Data**: dữ liệu gửi lên server
 - o **VD**:


```
{
    name: "Donald Duck",
    city: "Mac Donald"
}
```
 - o **Callback**: hàm nhận giữ dữ liệu trả về

36

Hàm nhận kết quả trả về

- **Callback:** hàm nhận dữ liệu trả về theo cú pháp sau:

```
function (data, status) {
    $('#iddiv1').html(data);
}
```

- data: Dữ liệu trả về từ Server
- status: Trạng thái của yêu cầu (Request), status = "success" khi yêu cầu được xử lý xong.

37

Ví dụ

```
function load_post() {
    $.post("result.php",
        {
            name: "Donald Duck",
            city: "Mac Donald"
        }, function (data, status) {
            $('#result').html(data);
        });
}
```

```
<button name="btnPost" onclick="load_post()">Lấy dữ liệu với hàm  
post()</button>
```

38

Mở rộng với async/await

- Thay vì sử dụng callback hoặc Promise, có thể viết mã AJAX bất đồng bộ dễ đọc hơn bằng async/await:
- ```

async function loadData() {
 try {
 const response = await
 fetch("server.php");
 if (response.ok) {
 const data = await response.text();
 console.log("Dữ liệu nhận được:",
 data);
 } else {
 console.error("Lỗi server:",
 response.status);
 }
 } catch (error) {
 console.error("Lỗi kết nối:", error);
 }
 loadData();
}
```

39

---

---

---

---

---

---

---

---

## Dùng hàm AJAX

- Dùng hàm \$.AJAX để gửi yêu cầu lên server
- Cú pháp

**\$.AJAX({url, type, datatype, data, success: function (result) })**

- o url: Tập tin xử lý yêu cầu
- o type: phương thức gửi "post" hoặc "get"
- o datatype: kiểu dữ liệu trả về, có thể là json, xml, script hoặc text
- o data: dữ liệu gửi lên server
- o success: hàm nhận giữ liệu trả về, tham số lưu trữ kết quả là result



40

---

---

---

---

---

---

---

---

## Ví dụ

```
$.AJAX({
 url : "result.php",
 type : "get",
 dataType:"text",
 data : {
 value: "AJAX với JQuery"
 },
 success : function (result){
 $('#result').html(result);
 }
});
```



41

---

---

---

---

---

---

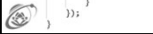
---

---

## Nhận response là file xml

```
function load_ajax_xml(){
 $.ajax({
 url: 'taikhoan.xml',
 type: 'get',
 dataType: 'xml',
 success: function (result){
 var str = '';
 str = 'ajax với XML đơ';
 str += '<table border="1" cellpadding="0" cellspacing="0">';
 str += '<tr><th>username</th><th>password</th></tr>';
 //lấy dữ liệu
 $(result).find('items').each(function (key,val){
 str += '<tr><td>';
 str += $(val).find('username').text();
 str += '</td>';
 str += '<td>';
 str += $(val).find('password').text();
 str += '</td></tr>';
 });
 str += '</table>';
 $('#result').html(str);
 },
 error: function (data, errorThrown){
 alert('error: ' + errorThrown);
 }
 });
}
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
 <items>
 <username>1234</username>
 <password>PTUDW</password>
 </items>
 <items>
 <username>1235</username>
 <password>JAVA</password>
 </items>
</root>
```



42

---

---

---

---

---

---

---

---

## Nhận response là file json

```
function load_ajax_json(){
 $.ajax ({
 url: 'data.json',
 type: 'get',
 datatype: 'json',
 success: function (result){
 var str = '';
 str = '<table với JSON <tr>';
 str += '<table border="1" cellspacing="0" cellpadding = "10">';
 str += '<tr><th>username</th><th>password</th></tr>';
 //kết quả là một object json
 //dùng loop
 $.each (result, function(key, item){
 str += '<tr><td>';
 str += item['name'];
 str += '</td>';
 str += '<td>';
 str += item['age'];
 str += '</td></tr>';
 });
 str += '</table>';
 $('#result').html(str);
 },
 error: function (data, errorThrown){
 alert('error:' + errorThrown);
 }
 });
}
```

```
{
 {
 "name": "John",
 "age": 23
 },
 {
 "name": "Jane",
 "age": 25
 },
 {
 "name": "Janet",
 "age": 30
 }
}
```

Phát triển ứng dụng Web 43

43

---

---

---

---

---

---

---

---

---

---

## Tạo file json từ php

```
<?php
//Connect database
$conn = mysqli_connect('localhost','root','','qlsv') or die ('Can not connect to mysql');
//get list member
$query = mysqli_query($conn, 'select * from lop');

//Biến result
$result = array();
if (mysqli_num_rows($query) > 0){
 while ($row = mysqli_fetch_array($query, MYSQLI_ASSOC)){
 $result[] = array(
 'username' => $row['MALOP'],
 'password' => $row['TENLOP']
);
 }
}
die (json_encode($result));
?>
```

Phát triển ứng dụng Web 44

44

---

---

---

---

---

---

---

---

---

---

## Q & A



**Cảm ơn đã theo dõi**

Hãy vọng cùng nhau đi đến thành công.

Phát triển ứng dụng Web 45

45

---

---

---

---

---

---

---

---

---

---

Bài tập nhóm (45p) – nhóm lý thuyết

**Xây dựng hệ thống quản lý công việc cá nhân (Task Manager)**

Mô tả: Nhóm Sinh viên sẽ xây dựng một ứng dụng quản lý công việc cá nhân. Người dùng có thể:

- Thêm công việc mới.
- Xem danh sách các công việc. Phân trang danh sách công việc.
- Xóa hoặc đánh dấu công việc đã hoàn thành.
- Tìm kiếm công việc theo từ khóa mà không cần tải lại trang.

Ứng dụng sẽ sử dụng PHP và MySQL để xử lý backend, kết hợp với AJAX (nếu được) để trao đổi dữ liệu giữa client và server.

**Nhóm SV thiết kế CSDL và phát triển trang web cho hệ thống này.**



---

---

---

---

---

---

---