



---

# Bài 10

# JQuery - Ajax

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT WITH LARAVEL

# Mục tiêu

---

- Trình bày được lợi ích của việc sử dụng JQuery
- Sử dụng được selector trong JQuery
- Sử dụng được jquery để làm việc với DOM
- Sử dụng được jquery để thực hiện được lời gọi AJAX
- Xử lý được kết quả trả về từ AJAX



# Giới thiệu về jQuery

---

- jQuery là một thư viện JavaScript rất phổ biến và dễ sử dụng. Thư viện này giúp việc lập trình JavaScript trở nên đơn giản hơn và giúp giảm bớt sự không tương thích JavaScript trên các trình duyệt khác nhau.
- Toàn bộ thư viện của jQuery được gói gọn trong một tệp .js
- Cập nhật thông tin, xem hướng dẫn sử dụng và tải các phiên bản jQuery tại trang <http://jquery.com>

# Sử dụng như thế nào



- Tải jQuery tại trang [jquery.com](http://jquery.com), tại đây có sẵn hai bản: nén (compressed) và không nén (uncompressed).
- Khi phát triển một ứng dụng web, Để hỗ trợ phát triển hoặc gỡ lỗi bạn nên chọn bản không nén
- Khi sản phẩm web của bạn đã đưa ra sử dụng rộng rãi, để tiết kiệm băng thông và cải thiện hiệu suất của trang web bạn nên chọn bản nén.
- Nhúng tệp thư viện jQuery (*tệp `jquery-...js`*) vào trang web của bạn. VD:

```
<head>
```

```
    <script src="jquery-3.2.0.min.js"></script>
```

```
</head>
```

# Cú pháp căn bản của jQuery

---



**`$(selector).action()`**

Trong đó:

- \$ là ký hiệu để định nghĩa\truy cập jQuery
- (*selector*) để truy vấn hoặc tìm phần tử HTML
- *action()* là hành động được jQuery định nghĩa sẵn để thực hiện với các phần tử

# Các loại selector

---



- **element** selector
- **#id** selector
- **.class** selector



# element selector

---

- **element** selector trong jQuery chọn các phần tử dựa vào tên của chúng.
- Tên của thẻ html chọn làm element selector
- Ví dụ: Chọn thẻ <p> làm selector
- Khi người dùng nhấp vào một nút, tất cả các phần tử <p> sẽ bị ẩn:

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("p").hide();  
    });  
});
```

# #id selector

---

- Sử dụng thuộc tính id của thẻ HTML để tìm phần tử cụ thể
- Viết ký tự # trước id của phần tử HTML để tạo #id selector
- Ví dụ: Tạo #id selector tên là test

```
<p id="test">This is another paragraph.</p>
```

```
<button>Click me</button>
```

- Khi người dùng nhấp vào một nút, phần tử với id = "test" sẽ bị ẩn:

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("#test").hide();  
    });  
});
```





# .class selector

---

- Sử dụng thuộc tính class của thẻ HTML để tìm phần tử cụ thể
- Viết dấu . trước tên của class để tạo .class selector
- Ví dụ: Tạo #id selector tên là test

```
<h2 class="test">This is a heading</h2>
```

```
<p class="test">This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
<button>Click me</button>
```

- Khi người dùng nhấp vào một nút, phần tử với class = "test" sẽ bị ẩn:

```
$(document).ready(function() {  
    $("button").click(function() {  
        $(".test").hide();  
    });  
});
```

# Sự kiện

---



Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

# Ví dụ - click



```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

# Hiệu ứng

---



- Hide
- Show
- Toggle
- Slide
- Fade
- Animate

# Ví dụ - hide/show



```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("#hide").click(function(){
        $("p").hide();
    });
    $("#show").click(function(){
        $("p").show();
    });
});
</script>
</head>
<body>

<p>If you click on the "Hide" button, I will disappear.</p>

<button id="hide">Hide</button>
<button id="show">Show</button>

</body>
</html>
```

[https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery\\_hide\\_show](https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_show)

---

# Ajax

# Ajax

---



- Viết tắt của Asynchronous JavaScript and XML
- Nó là một bộ các kỹ thuật thiết kế web giúp cho các ứng dụng web hoạt động bất đồng bộ – xử lý mọi yêu cầu tới server từ phía sau.

# Lợi ích

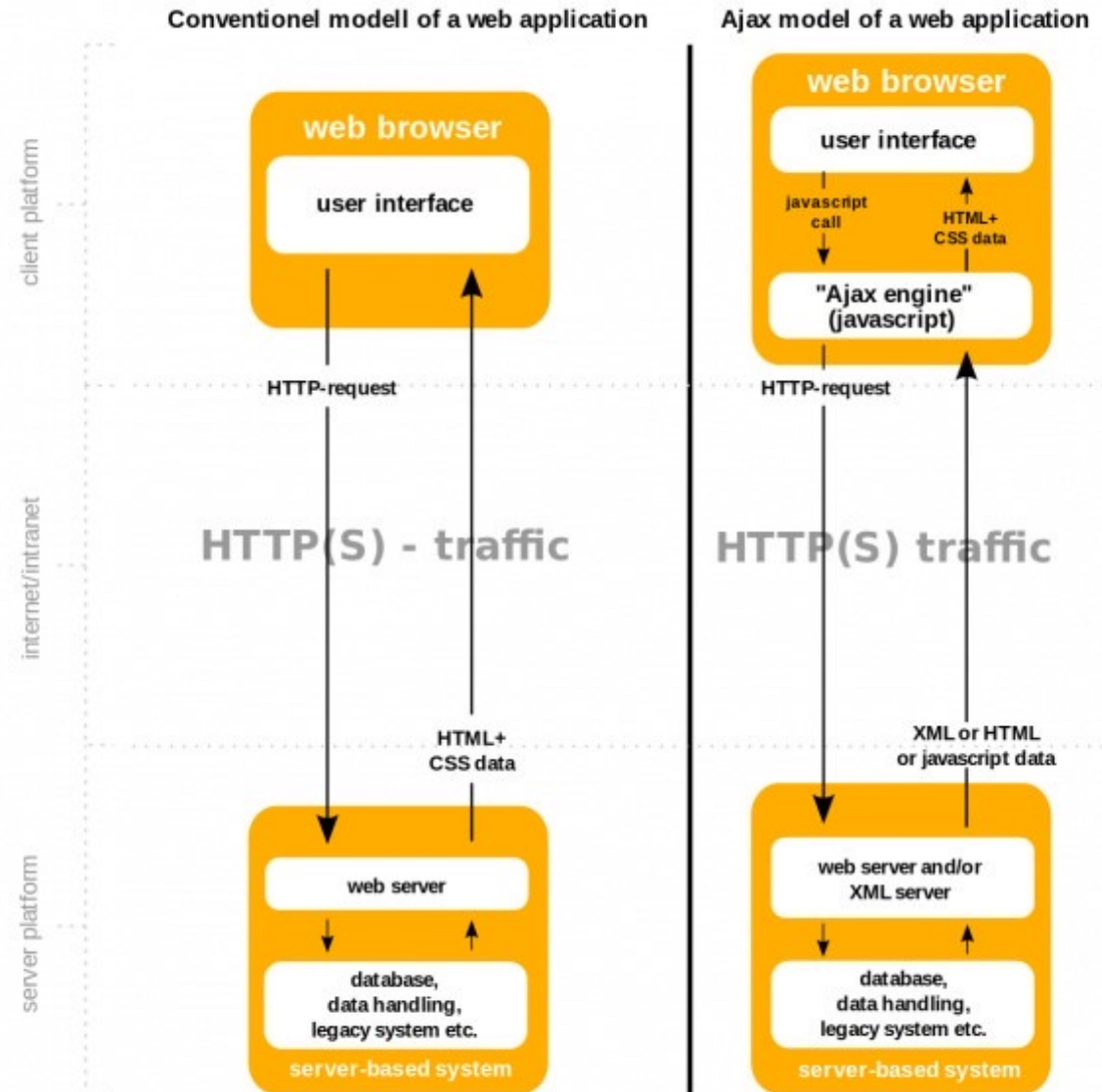
---



- Đọc dữ liệu từ máy chủ web - sau khi trang đã được tải
- Cập nhật một trang web mà không cần tải lại trang
- Gửi dữ liệu đến máy chủ web – chạy nền(background)



# Sơ đồ hoạt động





# Đối tượng XMLHttpRequest

---

- Đối tượng XMLHttpRequest được sử dụng để trao đổi dữ liệu với một Server ở Background.



# Đối tượng XMLHttpRequest

---

- Tạo đối tượng XMLHttpRequest

```
variable = new XMLHttpRequest();
```



# Ajax - Gửi dữ liệu đến Server

---

- Để gửi yêu cầu đến server, sử dụng các phương thức `open ()` và `send ()` của đối tượng `XMLHttpRequest`:

```
xhttp.open("GET", "ajax_info.txt", true);  
xhttp.send();
```



# Ajax - Gửi dữ liệu đến Server

Method	Mô tả
<code>open(method, url, async)</code>	Chỉ định loại yêu cầu  phương thức: loại yêu cầu: GET hoặc POST url: vị trí máy chủ (tệp) async: true (không đồng bộ) hoặc false (đồng bộ)
<code>send()</code>	Gửi request đến server với method GET
<code>send(string)</code>	Gửi request đến server với method POST



# Ajax – Xử lý response trả về

---

Sử dụng thuộc tính **responseText** hoặc **responseXML** của đối tượng XMLHttpRequest.

- responseText nhận dữ liệu phản hồi dưới dạng chuỗi.
- responseXML nhận dữ liệu phản hồi dưới dạng XML.

# Ajax – Xử lý response trả về

---



```
function loadDoc() {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            document.getElementById("demo").innerHTML =  
                this.responseText;  
        }  
    };  
    xhttp.open("GET", "ajax_info.txt", true);  
    xhttp.send();  
}
```



# Ajax – Xử lý response trả về

Property	Description
onreadystatechange	Xác định hàm được gọi khi thuộc tính readyState thay đổi
readyState	Giữ trạng thái của XMLHttpRequest. 0: yêu cầu không được khởi tạo 1: thiết lập kết nối máy chủ 2: nhận được yêu cầu 3: yêu cầu xử lý 4: yêu cầu kết thúc và phản hồi đã sẵn sàng
status	200: "OK" 403: "Forbidden" 404: "Page not found"
statusText	Trả về văn bản trạng thái (ví dụ: "OK" hoặc "Not Found")





# Tóm tắt bài học

---

- jQuery là một thư viện JavaScript rất phổ biến và dễ sử dụng. Thư viện này giúp việc lập trình JavaScript trở nên đơn giản hơn và giúp giảm bớt sự không tương thích JavaScript trên các trình duyệt khác nhau.
- Ajax là một kỹ thuật lập trình, xử lý các nghiệp vụ bất đồng bộ.

# Hướng dẫn thực hành

---



- Hướng dẫn bài tập, bài thực hành
- Chuẩn bị bài tiếp theo: **Internationalization**