



# AJAX

Ba Nguyễn (Updated 2021)

# AJAX



**AJAX** (*Asynchronous JavaScript and XML*) - là một thuật ngữ. **AJAX** là một bộ các kỹ thuật, cho phép *client* và *server* giao tiếp, trao đổi dữ liệu với nhau, dữ liệu bao gồm JSON, XML, HTML, ...

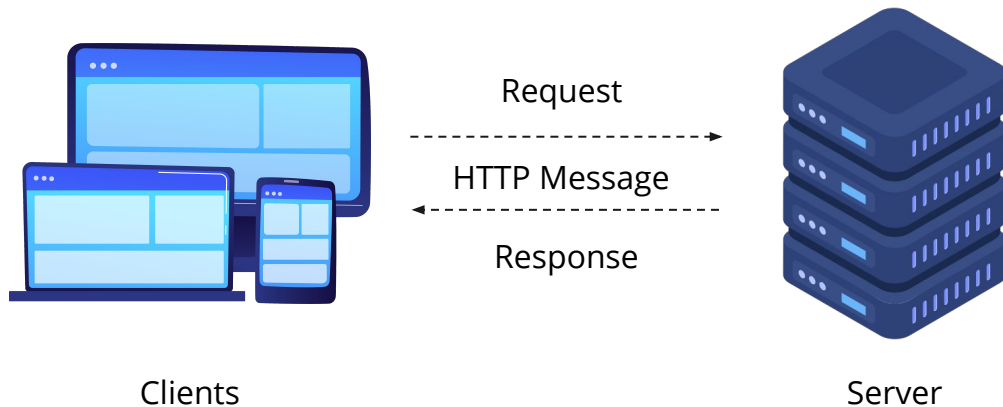
Điểm đặc biệt của **AJAX** là tính chất bất đồng bộ (*asynchronous*):

- Gửi yêu cầu tới *server* mà không cần làm mới lại trang web
- Nhận và xử lý dữ liệu từ *server*

Để gửi yêu cầu *HTTP* tới *server*, JavaScript sử dụng: `XMLHttpRequest()` hoặc `fetch()`.

# HTTP

**HTTP** (*Hypertext Transfer Protocol*) - là một giao thức truyền tải dữ liệu qua mạng theo mô hình *client - server*, thường được sử dụng để giao tiếp giữa trình duyệt mà máy chủ web



Tìm hiểu thêm về HTTP: [tutorialspoint.com/http](https://tutorialspoint.com/http)

# XMLHttpRequest

`XMLHttpRequest` là một object đặc biệt, cho phép gửi các yêu cầu HTTP trong JavaScript.

```
// Tạo một XMLHttpRequest object
let xhr = new XMLHttpRequest();
// Khởi tạo request
xhr.open(method, url); // methods: GET, POST, PUT, DELETE
// Gửi request
xhr.send();
// Khi request hoàn tất, và đã nhận response
xhr.onload = function () {};
// Khi không thể gửi request
xhr.onerror = function () {};
// Khi response đang được tải xuống
xhr.onprogress = function (e) {};
// Khi quá hạn thực hiện request, xhr.timeout = milisecon
xhr.ontimeout = function () {};
```

# XMLHttpRequest

Sau khi nhận được **response**, có thể truy cập dữ liệu thông qua một số thuộc tính:

```
// Khi request hoàn tất, và đã nhận response
xhr.onload = function () {
    // Status code: 200, 201, 301, 400, 404, ...
    xhr.status;

    // Status text: 200 → OK, 201 → Created, 404 → Not Found
    xhr.statusText;

    // Response data
    xhr.response; // xhr.responseText
};
```

# XMLHttpRequest

Một số thuộc tính cấu hình cho `XMLHttpRequest`

```
// Định dạng kiểu dữ liệu cho response  
xhr.responseType = "json"; // "", "text", "document", "blob", ...  
// Đặt thời hạn cho request  
xhr.timeout = 10000;  
// Kiểm soát kết nối tới các trang bên ngoài  
// Bao gồm cookies, auth, ...  
xhr.withCredentials = true;  
  
xhr.open(method, url);  
xhr.send();
```

# Custom Header

`XMLHttpRequest` cho phép tùy chỉnh header cho **request** và đọc header từ **response** với một số phương thức:

```
// Tùy chỉnh HTTP Header  
// xhr.setRequestHeader(name, value)  
xhr.setRequestHeader("Content-Type", "application/json");  
  
xhr.open(method, url);  
xhr.send();  
  
xhr.getResponseHeaders("Content-Type");
```



Danh sách header được phép tùy chỉnh: [mdn/header\\_name](https://developer.mozilla.org/en-US/docs/HTTP/Headers)

# Sending Data

Để gửi dữ liệu (với **POST** request), có thể sử dụng **FormData** hoặc **JSON**

```
<form action="" name="login">
  <input type="text" name="user">
  <input type="password" name="password">
</form>
<script>
  const data = new FormData(document.forms.login);
  data.append("key", "value");
  // xhr.send(data)
  // Content-Type: multipart/form-data
</script>
```



# Sending Data

Để gửi dữ liệu (với **POST** request), có thể sử dụng **FormData** hoặc **JSON**

```
let data = JSON.stringify({  
  username: " ... ",  
  password: " ... ",  
});
```

```
xhr.setRequestHeader("Content-Type", "application/json");  
xhr.send(data);
```