

# Lập trình Web Front-end 2

## | Front-end Web Development 2 |

*Nguyễn Huy Hoàng*

*[02 . 2019]*



Bài.

# AJAX

# AJAX

- **Asynchronous JavaScript And XML**
- Gửi yêu cầu lên server bằng JavaScript mà không reload trang.
- Có nhiều cách để thực hiện AJAX

# || AJAX cũ với XMLHttpRequest

```
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 với jQuery

```
$.ajax({  
    type: "GET",  
    url: "example.php",  
    data: post  
})  
.done(function (result) {  
  
});
```

# fetch()

- Là một cách hiện đại hiện nay
- Mạnh mẽ và dễ sử dụng
- Được hỗ trợ tốt bởi các trình duyệt hiện đại

Desktop						Mobile					
Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	Android webview	Chrome for Android	Firefox for Android	Opera for Android	Safari on iOS	Samsung Internet
42	14	39	No	29	10.1	42	42	39	29	10.3	4.0

# Cú pháp

## Dùng `await`

```
//Gửi request
const response = await fetch(url,
[options]);
//Nhận kết quả
const results = await response.json();
alert(results);
```

## Không dùng `await`

```
fetch(url, [options]) //Gửi request
.then(response => response.json())
//Nhận kết quả
.then(results => alert(results));
```

# Response

- `response.text()`: đọc và return kết quả theo kiểu text,
- `response.json()`: trả kết quả theo kiểu JSON
- `response.formData()`: return the response as FormData object
- `response.blob()`: return the response as Blob (binary data with type),
- `response.arrayBuffer()`: return the response as ArrayBuffer (low-level representation of binary data)



# POST requests

Để gửi một POST request, ta phải thêm options cho fetch:

- **method:** HTTP method (POST, PUT, DELETE, ...)
  - **header:** header cho request
  - **body:** body của request, gồm:
    - a string (VD: chuỗi JSON-encoded),
    - FormData object: để gửi dữ liệu theo form/multipart
- 
- Blob/BufferSource
  - URLSearchParams

# POST requests

Khi gửi JSON, ta dùng hàm **JSON.stringify**(data) để chuyển objects thành chuỗi JSON.

# Request header

```
headers: {  
    'Content-Type':    'application/json', // gửi request dạng JSON  
    'Accept':         'application/json' // nhận kết quả dạng JSON  
}
```

Có thể thêm bảng mã bằng cách: 'application/json; charset=utf-8'

Nếu gửi request dạng text, dùng: 'text/plain; charset=UTF-8'

# || Tổng hợp POST requests

```
const data = {id: productId};
const url = 'productbyid.php';
const response = await fetch(url, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json;charset=utf-8',
    'Accept': 'application/json'
  },
  body: JSON.stringify(data)
});
let result = await response.json();
console.log(result);
```

# || Nhận request từ fetch với PHP

```
$input = json_decode(file_get_contents('php://input'), true);  
$id = $input['id'];
```

# Thanks for your attention!



FACULTY OF  
**INFORMATION TECHNOLOGY**  
THU DUC COLLEGE OF TECHNOLOGY

**Phone:** (+848) 22 158 642

**Email:** [fit@tdc.edu.vn](mailto:fit@tdc.edu.vn)

**Website:** [fit.tdc.edu.vn](http://fit.tdc.edu.vn)

**Facebook:** [facebook.com/tdc.fit](https://facebook.com/tdc.fit)

**Youtube:** [youtube.com/fit-tdc](https://youtube.com/fit-tdc)