

Unity
Thống nhất
Excellence
Vượt trội
Leadership
Tiên phong

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT



www.uel.edu.vn

**PHÁT TRIỂN WEB
KINH DOANH**

GV: *ThS Nguyễn Quang Phúc*

Phát triển web kinh doanh:

XML & JSON

ThS. Nguyễn Quang Phúc
phucnq@uel.edu.vn

NỘI DUNG

01

Giới thiệu về XML

02

Làm việc với dữ liệu XML

03

Làm việc với dữ liệu JSON

1. Giới thiệu về XML

- XML (eXtensible Markup Language): là ngôn ngữ đánh dấu (tương tự HTML) “mở rộng” dùng để lưu trữ và truyền tải dữ liệu.

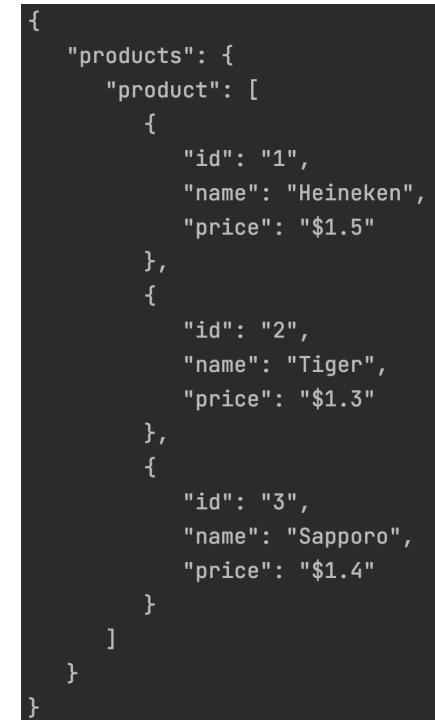
Lưu ý:

- Các thẻ mở phải có thẻ đóng tương ứng.
- Tên thẻ phân biệt chữ hoa, thường.
- XML document phải có một thẻ root.
- Giá trị các thuộc tính phải đặt trong nháy kép (hoặc nháy đơn).



```
<?xml version="1.0" encoding="UTF-8"?>
<products>
  <product>
    <id>1</id>
    <name>Heineken</name>
    <price>$1.5</price>
  </product>
  <product>
    <id>2</id>
    <name>Tiger</name>
    <price>$1.3</price>
  </product>
  <product>
    <id>3</id>
    <name>Sapporo</name>
    <price>$1.4</price>
  </product>
</products>
```

XML



```
{
  "products": [
    {
      "product": [
        {
          "id": "1",
          "name": "Heineken",
          "price": "$1.5"
        },
        {
          "id": "2",
          "name": "Tiger",
          "price": "$1.3"
        },
        {
          "id": "3",
          "name": "Sapporo",
          "price": "$1.4"
        }
      ]
    }
  ]
}
```

Json

2. Làm việc với dữ liệu XML

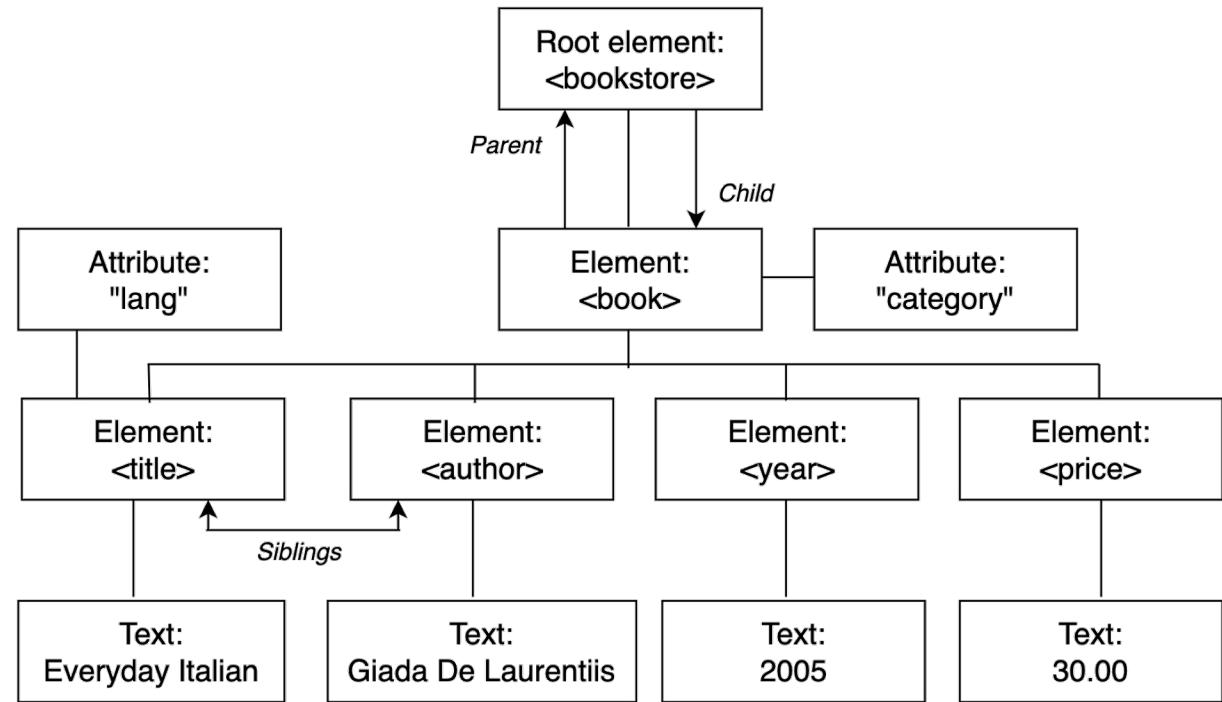
» XML DOM

- XML DOM: Element, Attribute, Text

```
<bookstore>
...
<book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
</book>

<book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
</book>

</bookstore>
```



2. Làm việc với dữ liệu XML

»» XML DOM

- ✓ Chuyển chuỗi thành XML Document

```
var parser = new DOMParser();
```

```
var xmlDoc = parser.parseFromString(txt,"text/xml");
```

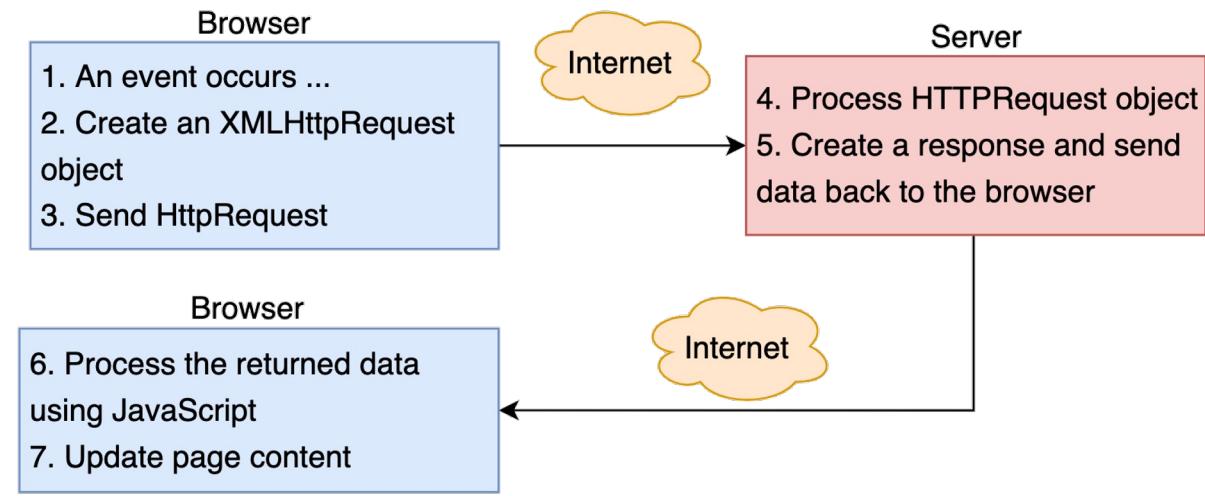
- ✓ Truy cập các phần tử trong XML theo cấu trúc DOM

```
<body>
    <div id="content"></div>
    <script>
        var xmlString = "<bookstore><book>";
        xmlString += "<title>Everyday Italian</title>";
        xmlString += "<author>Giada De Laurentiis</author>";
        xmlString += "<year>2005</year>";
        xmlString += "</book></bookstore>";
        var parser = new DOMParser();
        var xmlDoc = parser.parseFromString(xmlString, "text/xml");
        var content = document.getElementById("content");
        content.innerHTML = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
    </script>
</body>
```

2. Làm việc với dữ liệu XML

»» AJAX

- **AJAX (Asynchronous JavaScript And XML)** là kỹ thuật sử dụng kết hợp:
 - Đối tượng XMLHttpRequest của trình duyệt được tích hợp sẵn → để yêu cầu dữ liệu từ máy chủ web.
 - JavaScript và HTML DOM → để hiển thị hoặc sử dụng dữ liệu.
- **Chức năng:**
 - *Cập nhật trang web mà không cần tải lại trang.*
 - *Yêu cầu dữ liệu từ máy chủ - sau khi trang tải xong.*
 - *Nhận dữ liệu từ máy chủ - sau khi trang tải xong.*
 - *Gửi dữ liệu đến máy chủ - thực thi ở chế độ “background”.*

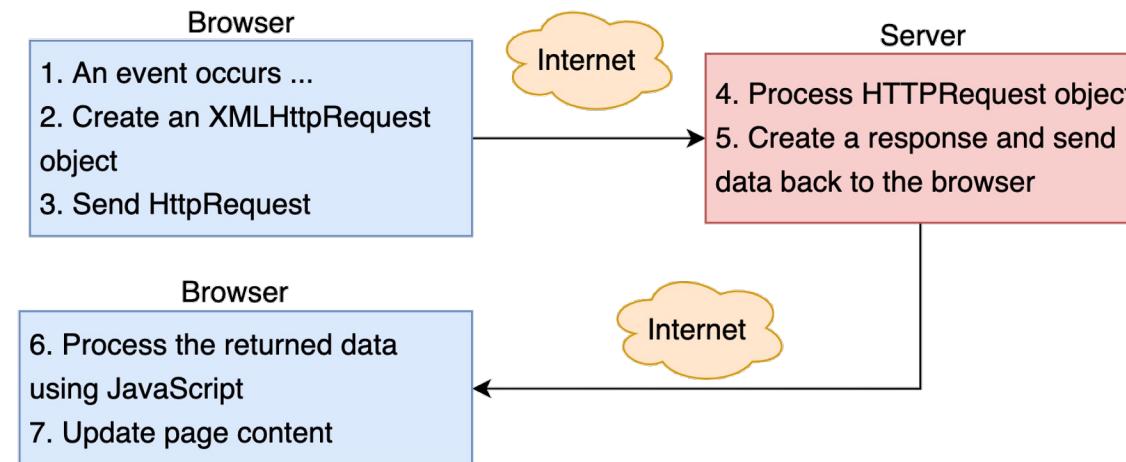


2. Làm việc với dữ liệu XML

»» AJAX

- Khởi tạo đối tượng XMLHttpRequest

```
let xhr = new XMLHttpRequest();
```



2. Làm việc với dữ liệu XML

»» AJAX ➤ Gửi Request

Method	Description
<code>open(<i>method</i>, <i>url</i>, <i>async</i>)</code>	<p>Specifies the type of request</p> <p><i>method</i>: the type of request: GET or POST <i>url</i>: the server (file) location <i>async</i>: true (asynchronous) or false (synchronous)</p>
<code>send()</code>	Sends the request to the server (used for GET)
<code>send(<i>string</i>)</code>	Sends the request to the server (used for POST)

2. Làm việc với dữ liệu XML

»» AJAX ➤ Gửi Request

```
xhr.open("GET", "products.xml", true);  
xhr.send();
```

```
let xhttp = new XMLHttpRequest();  
xhttp.open("GET", "products_get.asp?pId=68", true);  
xhttp.send();
```

```
let xhttp = new XMLHttpRequest();  
xhttp.open("POST", "demo_post.asp", true);  
xhttp.setRequestHeader("Content-type", "application/  
x-www-form-urlencoded");  
xhttp.send("fname=Henry&lname=Ford");
```

2. Làm việc với dữ liệu XML

»» **AJAX**

➤ Nhận Response

XMLHttpRequest Object Properties

Property	Description
onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found"
statusText	Returns the status-text (e.g. "OK" or "Not Found")

2. Làm việc với dữ liệu XML

»» AJAX ➤ Nhận Response

```
let xhr = new XMLHttpRequest();
xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {
        //Xử lý khi load được dữ liệu
    } else {
        //Xử lý khi không load được dữ liệu
    }
}
```

2. Làm việc với dữ liệu XML

»» **AJAX** ➤ Vd: load danh sách sản phẩm

```
function loadDoc() {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', './data/products.xml', false);
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            //Xử lý dữ liệu nhận được
            updatePage(this);
        } else {
            alert("Không lấy được dữ liệu")
        }
    }
    xhr.send();
}
```

products.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
    <product>
        <id>1</id>
        <name>Heineken</name>
        <price>19000</price>
    </product>
    <product>
        <id>2</id>
        <name>Tiger</name>
        <price>18000</price>
    </product>
    <product>
        <id>3</id>
        <name>Sapporo</name>
        <price>21000</price>
    </product>
    <product>
        <id>4</id>
        <name>Corona Extra</name>
        <price>19000</price>
    </product>
</products>
```

2. Làm việc với dữ liệu XML

»» **AJAX** ➤ Vd: load danh sách sản phẩm

```
function updatePage(xhr) {
    let htmlStr = "";
    let element = document.getElementById("content");
    var xmlDoc = xhr.responseXML;
    var itemLayout = "";
    var products = xmlDoc.getElementsByTagName("product");
    for (i = 0; i < products.length; i++) {
        itemLayout += '<div class="wrap__item">' +
            products[i].getElementsByTagName("id")[0].childNodes[0].
            nodeValue + '<br><b>' +
            products[i].getElementsByTagName("name")[0].childNodes[0].
            nodeValue + '</b><br><br>' +
            products[i].getElementsByTagName("price")[0].childNodes[0].
            nodeValue +
            '</div>';
    }
    document.getElementById("content").innerHTML = itemLayout;
}
```

2. Làm việc với dữ liệu XML

»» **AJAX** ➤ Vd: load danh sách sản phẩm

The screenshot shows a web page titled "The XMLHttpRequest Object". On the left, there is a vertical toolbar with icons for Home, Star, Minus, Refresh, WhatsApp, Instagram, Plus, and Heart. In the center, a button labeled "Get list of products" is visible. Below it, four product cards are displayed in a 2x2 grid:

Rank	Product Name	Price
1	Heineken	19000
2	Tiger	18000
3	Sapporo	21000
4	Corona Extra	19000

2. Làm việc với dữ liệu XML

»» **AJAX** ➤ Vd: load danh sách CDs https://www.w3schools.com/xml/cd_catalog.xml

```
function loadDoc() {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', './data/cd_catalog.xml', false);
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            //Xử lý dữ liệu nhận được
            updatePage(this);
        } else {
            alert("Không lấy được dữ liệu")
        }
    }
    xhr.send();
}
```

```
<catalog>
<cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
</cd>
<cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
</cd>
...
</catalog>
```

2. Làm việc với dữ liệu XML

»» **AJAX** ➤ Vd: load danh sách CDs https://www.w3schools.com/xml/cd_catalog.xml

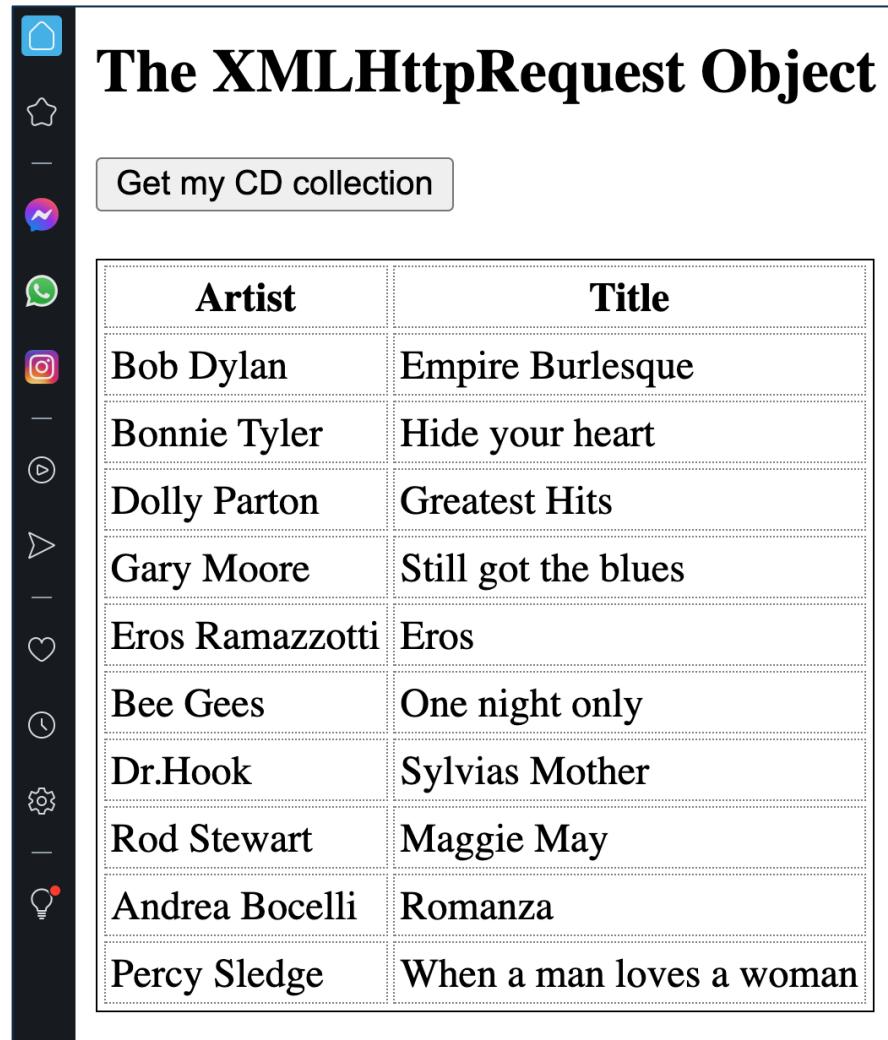
```
function updatePage(xhr) {
    let htmlStr = "";
    let element = document.getElementById("content");
    var xmlDoc = xhr.responseXML;
    var table = "<tr><th>Artist</th><th>Title</th></tr>";
    var cds = xmlDoc.getElementsByTagName("cd");
    for (i = 0; i < cds.length; i++) {
        table += "<tr><td>" +
            cds[i].getElementsByTagName("artist")[0].childNodes[0].nodeValue
            + "</td><td>" +
            cds[i].getElementsByTagName("title")[0].childNodes[0].nodeValue
            + "</td></tr>";
    }
    document.getElementById("content").innerHTML = table;
}
```

2. Làm việc với dữ liệu XML

»» **AJAX**

➤ Vd: load danh sách CDs

https://www.w3schools.com/xml/cd_catalog.xml



The XMLHttpRequest Object

Get my CD collection

Artist	Title
Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman

3. Làm việc với dữ liệu JSON

»» Giới thiệu JSON

- **JSON (JavaScript Object Notation)**: là một kiểu định dạng dữ liệu phổ biến được mô tả theo cặp “thuộc tính:giá trị” (key:value). JSON là một tiêu chuẩn mở để trao đổi dữ liệu trên web.

```
{  
    "id": 1,  
    "name": "Heineken",  
    "price": 19000  
}
```

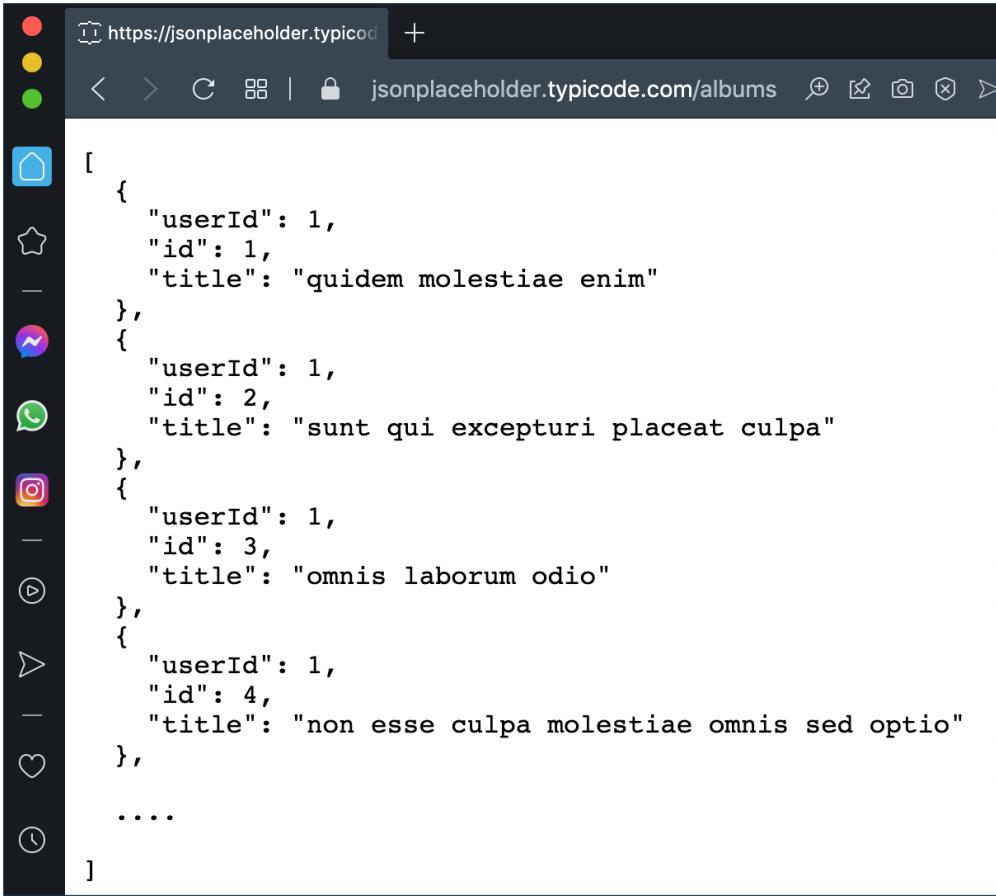
JSON Object

```
[  
    {  
        "id": 1,  
        "name": "Heineken",  
        "price": 19000  
    },  
    {  
        "id": 2,  
        "name": "Tiger",  
        "price": 18000  
    },  
    {  
        "id": 3,  
        "name": "Sapporo",  
        "price": 21000  
    }  
]
```

JSON Array

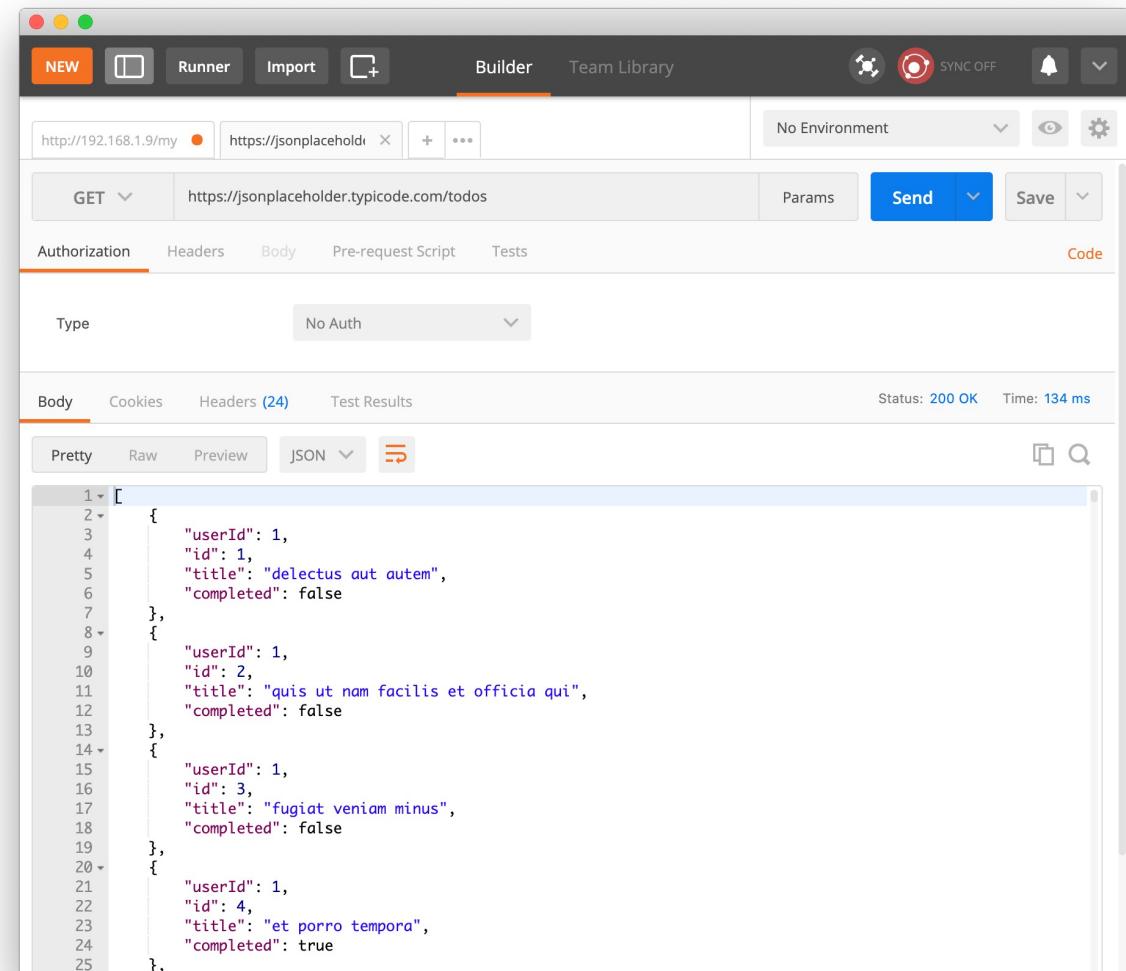
3. Làm việc với dữ liệu JSON

» Giới thiệu JSON



A screenshot of a web browser displaying JSON data from <https://jsonplaceholder.typicode.com/albums>. The JSON array contains five albums, each with an userId, id, and title.

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "quidem molestiae enim"  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "sunt qui excepturi placeat culpa"  
  },  
  {  
    "userId": 1,  
    "id": 3,  
    "title": "omnis laborum odio"  
  },  
  {  
    "userId": 1,  
    "id": 4,  
    "title": "non esse culpa molestiae omnis sed optio"  
  },  
  ....  
]
```



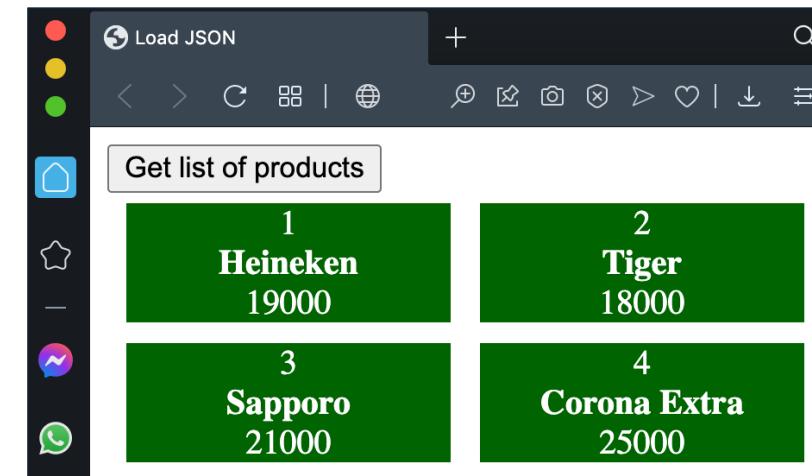
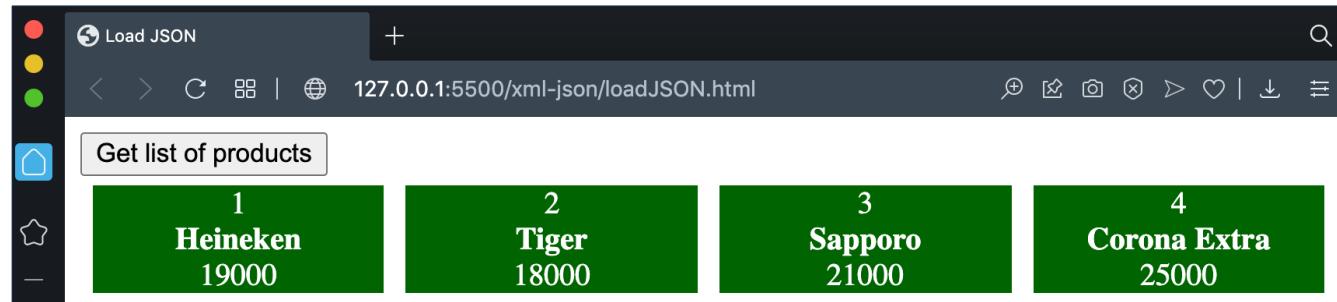
3. Làm việc với dữ liệu JSON

»» Đọc dữ liệu JSON

```
function loadJson() {
    let htmlString = '';
    var products = [
        {
            "id": 1,
            "name": "Heineken",
            "price": 19000
        },
        {
            "id": 2,
            "name": "Tiger",
            "price": 18000
        },
        {
            "id": 3,
            "name": "Sapporo",
            "price": 21000
        },
        {
            "id": 4,
            "name": "Corona Extra",
            "price": 25000
        }];

        for (let p of products) {
            htmlString += '<div class="item">' +
                p.id + '<br><b>' + p.name + '</b><br>' +
                p.price + '</div>';
        }

        let content = document.getElementById("content");
        content.innerHTML = htmlString;
    }
```

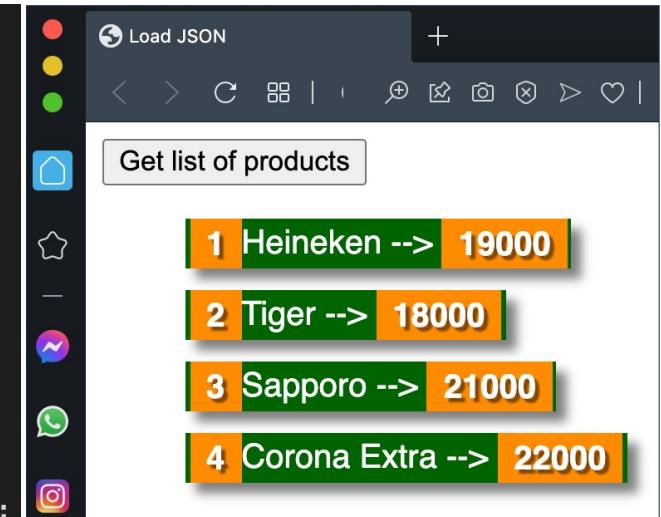


3. Làm việc với dữ liệu JSON

»» Đọc dữ liệu JSON

```
products.json •  
Examples > xml-json > data > products.json > ...  
1 {  
2   "products": [  
3     {  
4       "id": 1,  
5       "name": "Heineken",  
6       "price": 19000  
7     },  
8     {  
9       "id": 2,  
10      "name": "Tiger",  
11      "price": 18000  
12    },  
13    {  
14      "id": 3,  
15      "name": "Sapporo",  
16      "price": 21000  
17    },  
18    {  
19      "id": 4,  
20      "name": "Corona Extra",  
21      "price": 22000  
22  ]  
23 }
```

```
function loadJson() {  
  var myList = document.querySelector('ul');  
  fetch("./data/products.json")  
    .then(function(response) {  
      if (!response.ok) {  
        throw new Error("HTTP error, status = " +  
          response.status);  
      }  
      return response.json();  
    })  
    .then(function(json) {  
      for (let p of json.products) {  
        var listItem = document.createElement('li');  
        listItem.innerHTML = '<strong>' + p.id + '</strong>';  
        listItem.innerHTML += p.name;  
        listItem.innerHTML += ' --> <strong>' + p.price + '</strong>';  
        myList.appendChild(listItem);  
      }  
    })  
    .catch(function(error) {  
      alert(error.message);  
    });  
}
```



3. Làm việc với dữ liệu JSON

»» Đọc dữ liệu JSON

```
function loadJson() {
    var myList = document.querySelector('ul');
    fetch("https://jsonplaceholder.typicode.com/users")
        .then(function(response) {
            if (!response.ok) {
                throw new Error("HTTP error, status = " +
                    response.status);
            }
            return response.json();
        })
        .then(function(users) {
            for (let user of users) {
                var listItem = document.createElement('li');
                listItem.innerHTML = '<strong>' + user.id + '</
strong>';
                listItem.innerHTML += user.name;
                listItem.innerHTML += ' >> <strong>' + user.email + '</strong>';
                myList.appendChild(listItem);
            }
        })
        .catch(function(error) {
            alert(error.message);
        });
}
```

