



[www.poly.edu.vn](http://www.poly.edu.vn)

## LẬP TRÌNH AJAX

---

## LẬP TRÌNH JAVA #4 (P7.1)

- Giới thiệu Ajax
- Giới thiệu Fetch API
- Xây dựng chức năng Upload bằng Ajax
- Xây dựng chức năng Download bằng Ajax



- ❑ Ajax là kỹ thuật lập trình phía client (client side) tương tác với phía server (server side) theo một kênh riêng mà không làm cho trang web bị tải lại (refresh)
- ❑ Tương tác với phía server có thể là
  - ❖ Download (tải về): Lấy dữ liệu từ server
  - ❖ Upload (tải lên): Gửi dữ liệu lên server
- ❑ Trong lập trình web phía client bạn có thể sử dụng JavaScript thuận hoặc 1 thư viện mở nào đó để thực hiện tương tác này
  - ❖ Fetch API: thuận JavaScript làm việc thuận tiện với mọi loại dữ liệu (text, binary, json...) được hỗ trợ bởi mọi trình duyệt hiện nay
  - ❖ Jquery, Axios....: thư viện mở, thường chỉ làm việc với JSON

❑ Fetch API là api JavaScript cung cấp cách giao tiếp với phía server

❑ Cú pháp:

```
var url = "địa chỉ url";
fetch(url).then(resp => {
    // xử lý lỗi dựa vào các thuộc tính resp
    resp.text().then(text => {
        // xử lý dữ liệu text đã chuyển đổi
    })
})
```

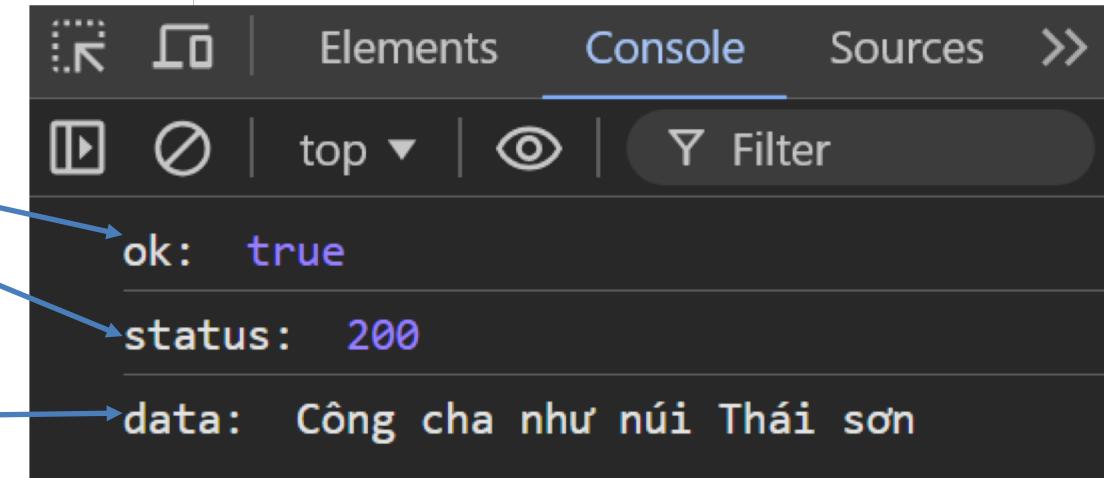
❑ Cú pháp thu gọn (bỏ qua xử lý lỗi)

```
var url = "địa chỉ url";
fetch(url).then(resp => resp.text()).then({
    // xử lý dữ liệu text đã chuyển đổi
})
```

```
@WebServlet("/ajax/fetch")
public class AjaxFetchServlet extends HttpServlet {
    @Override
    protected void service(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        resp.setCharacterEncoding("utf-8");
        resp.setContentType("text/plain");
        String data = "Công cha như núi Thái sơn";
        resp.getWriter().print(data);
    }
}
```

## VÍ DỤ FETCH API – CLIENTSIDE

```
var url = "http://localhost:8080/pro/ajax/fetch";
var options = {method: "GET"};
fetch(url, options).then(resp => {
    console.log("ok", resp.ok);
    console.log("status", resp.status);
    resp.text().then(text => {
        console.log("data", text)
    })
})
```



- ❑ Đối số resp trong .then(resp) chứa dữ liệu từ server side và các thông tin trạng thái khác
  - ❖ Thuộc tính ok cho biết trạng thái thực hiện có thành công hay không
  - ❖ Thuộc tính status chứa mã trạng thái từ server gửi về
  - ❖ Phương thức text() chuyển đổi dữ liệu từ server gửi về sang dạng văn bản.



# DEMOSTATION

---

- ❑ Servlet cung cấp một danh mục mã trạng thái để gửi về cho client.  
Sau đây là một số mã trạng thái thường dùng
  - ❖ `HttpServletResponse.SC_BAD_REQUEST`
    - 400: Request lỗi
  - ❖ `HttpServletResponse.SC_NOT_FOUND`
    - 404: Không tìm thấy
  - ❖ `HttpServletResponse.SC_OK`
    - 200: Thành công
  - ❖ `HttpServletResponse.SC_NO_CONTENT`
    - 204: Không có nội dung
- ❑ Gửi mã trạng thái về client
  - ❖ `resp.setStatus(HttpServletResponse.SC_OK);`

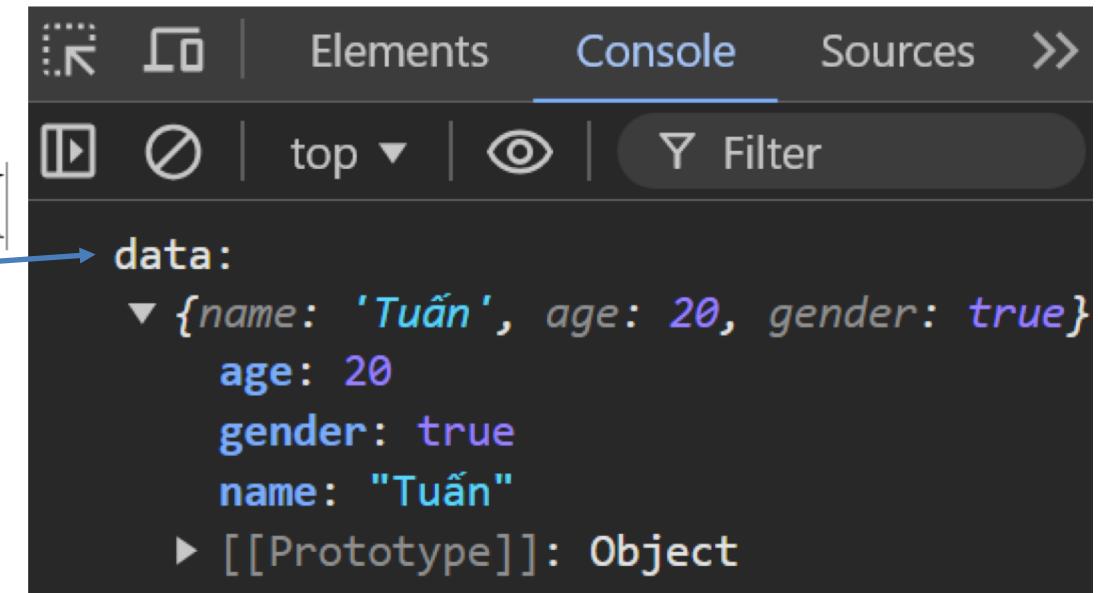
- ❑ Lệnh gọi fecth(url, options).then(resp) đối số resp là đối tượng chứa 4 phương thức cho phép bạn gọi để chuyển đổi sang dạng dữ liệu thích hợp để xử lý
  - ❖ resp.text()
    - Chuyển đổi sang dạng văn bản
  - ❖ resp.json()
    - Chuyển đổi sang đối tượng json
  - ❖ resp.arrayBuffer()
    - Chuyển đổi sang mảng dữ liệu nhị phân (mảng byte)
  - ❖ resp.blob()
    - Chuyển đổi sang đối tượng dữ liệu có cấu trúc file để có thể lưu vào đĩa

ServerSide

```
resp.setCharacterEncoding("utf-8");
resp.setContentType("application/json");
String data = "{\"name\":\"Tuấn\", \"age\":20, \"gender\":true}";
resp.getWriter().print(data);
```

ClientSide

```
var url = "http://localhost:8080/pro/ajax/fetch";
var options = {method:"GET"};
fetch(url, options)
  .then(resp => resp.json())
  .then(json => {
    console.log("data: ", json)
  })
}
```





# DEMOSTATION

---

# UPLOAD FILE VỚI FETCH API – CLIENTSIDE

```
<input id="photo" type="file">
<button onclick="upload()">Upload</button>
<script>
function upload(){
    var input = document.getElementById("photo");
    var formData = new FormData();
    formData.append('photo', input.files[0]);
    var url = "http://localhost:8080/Pro/ajax/fetch/upload";
    var options = { method: 'POST', body: formData }
    fetch(url, options).then(resp => resp.json()).then(json => {
        console.log("success", json);
    });
}
</script>
```

```
success
▼ {name: 'KQKD-Q224.png', type: 'image/png', size: 65128}
  name: "KQKD-Q224.png"
  size: 65128
  type: "image/png"
▶ [[Prototype]]: Object
```

# UPLOAD FILE VỚI FETCH API – SERVERSIDE

```
@MultipartConfig  
@WebServlet("/ajax/fetch/upload")  
public class UploadServlet extends HttpServlet {  
    @Override  
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)  
        throws ServletException, IOException {  
        Part part = req.getPart("photo");  
        String name = part.getSubmittedFileName(); // tên file  
        long size = part.getSize(); // kích thước file  
        String type = part.getContentType(); // kiểu file  
  
        String filename = req.getServletContext().getRealPath("/files/" + name);  
        part.write(filename); // lưu vào file trên server  
  
        resp.setCharacterEncoding("utf-8");  
        resp.setContentType("application/json");  
        String format = "{\"name\": \"%s\", \"type\": \"%s\", \"size\": %d}";  
        String responseData = String.format(format, name, type, size);  
        resp.getWriter().print(responseData);  
    }  
}
```



# DEMOSTATION

---

```
@WebServlet("/ajax/fetch/download")
public class DownloadServlet extends HttpServlet {
    @Override
    protected void service(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        resp.setCharacterEncoding("utf-8");
        resp.setContentType("text/plain");
        String data = "{\"name\":\"Tuấn\", \"age\":20, \"gender\":true}";
        resp.getWriter().print(data);
    }
}
```

# DOWNLOAD FILE VỚI FETCH API – CLIENTSIDE

```
<button onclick="download()">Download</button>
<script>
function download(){
    var url = "http://localhost:8080/pro/ajax/fetch/download";
    fetch(url).then(resp => resp.blob()).then(blob => {
        var blobUrl = URL.createObjectURL(blob); // Tạo URL từ blob
        saveAs(blobUrl, "data.txt"); // lưu xuống file data.txt
        URL.revokeObjectURL(blobUrl); // Xóa URL đã tạo ở trên
    })
}
function saveAs(url, filename) {
    var a = document.createElement("a"); // Tạo <a></a>
    a.href = url; // <a href=@url></a>
    a.download = filename; // <a href=@url download=@filename>
    document.body.appendChild(a); // bổ sung <a> vào <body>
    a.click(); // phát sinh sự kiện click vào <a>
    document.body.removeChild(a); // Xóa <a> khỏi <body>
}
</script>
```

data.txt

```
{"name":"Tuấn", "age":20,
"gender":true}
```



# DEMOSTATION

---

- ✓ Giới thiệu Ajax
- ✓ Giới thiệu Fetch API
- ✓ Xây dựng chức năng Upload bằng Ajax
- ✓ Xây dựng chức năng Download bằng Ajax





Cảm ơn