

# Introduction to Servlet

Servlet là gì?

Cách hoạt động của Servlet

Tìm hiểu vòng đời của Servlet

Phương thức xử lý một request của Servlet container và web server

## Servlet là gì?

Tùy theo ngữ cảnh sử dụng mà Servlet được định nghĩa bằng nhiều cách khác nhau như:

- Servlet chính là công nghệ được dùng để thiết lập ra các ứng dụng web.
- Servlet được xem là một API cung cấp các interface, lớp và cả các tài liệu.
- Servlet cũng là một thành phần website được lập trình viên triển khai trên máy chủ, phục vụ cho mục đích tạo các trang web động.

Với Servlet, bạn được phép thu thập dữ liệu đầu vào của người dùng thông qua website. Cụ thể, chúng sẽ hiển thị bản ghi từ một nguồn cơ sở dữ liệu hay từ nguồn khác.

Servlet sử dụng công nghệ mạnh mẽ và cho phép mở rộng. Mặc dù trước khi Servlet ra đời, đã có ngôn ngữ kịch bản CGI (viết tắt của Common Gateway Interface) được dùng làm ngôn ngữ lập trình phổ biến, tuy nhiên, nó vẫn còn tồn tại nhiều bất cập. Do đó, Servlet được nghiên cứu và phát triển để có thể khắc phục những hạn chế mà CGI mang lại.

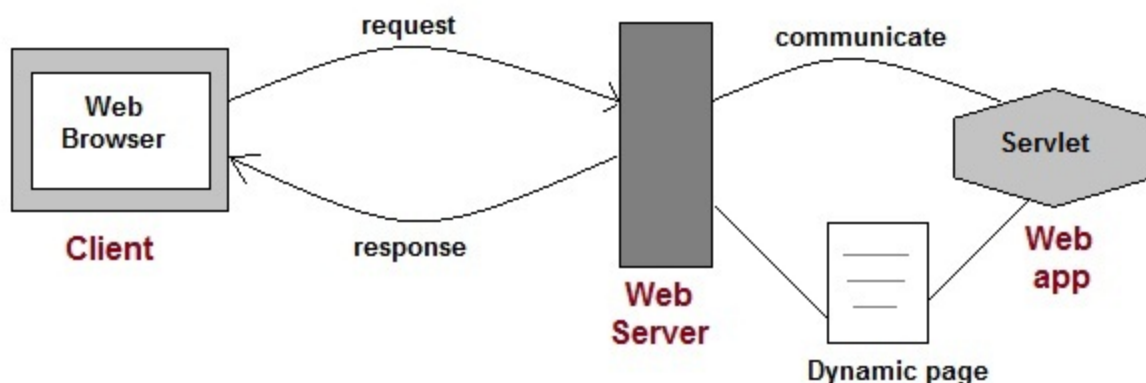
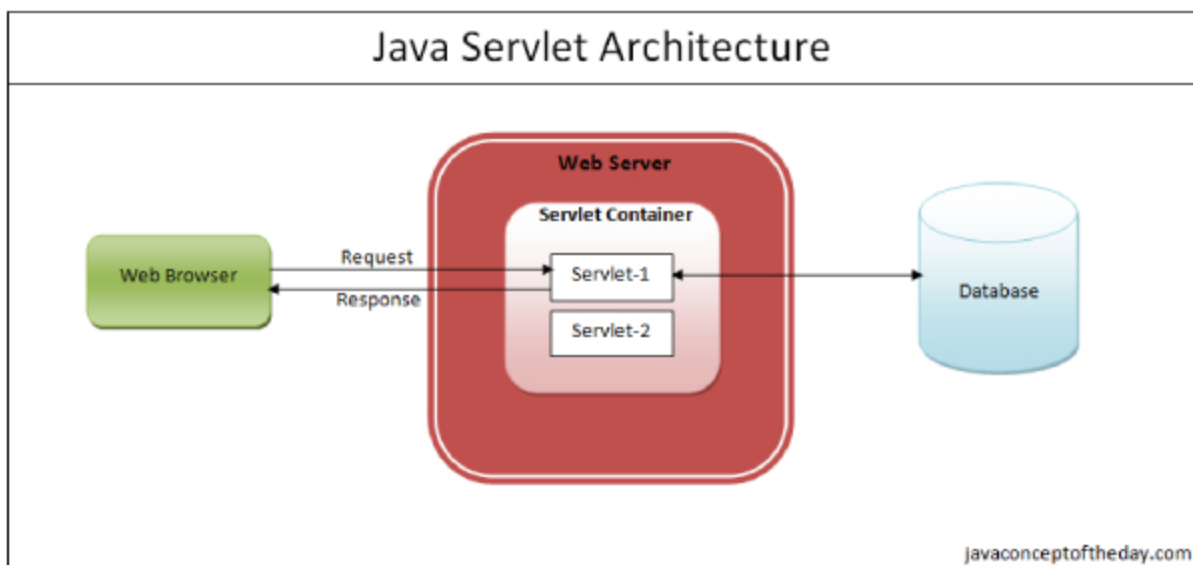
## Cách hoạt động của Servlet

Khi một yêu cầu được gửi từ phía người dùng, máy chủ web (như Apache Tomcat) nhận biết URL hoặc URI và xác định rằng yêu cầu cần được xử lý bởi một Servlet cụ thể.

Quá trình này bao gồm các bước sau:

1. **Nhận yêu cầu:** Servlet container (ví dụ: Tomcat) nhận yêu cầu và xác định Servlet tương ứng để xử lý nó.

2. **Khởi tạo Servlet:** Nếu Servlet chưa được tạo, container sẽ khởi tạo nó trước khi xử lý yêu cầu.
3. **Xử lý yêu cầu:** Servlet xử lý yêu cầu bằng cách sử dụng các tham số truyền vào từ yêu cầu, tải dữ liệu từ cơ sở dữ liệu nếu cần, và thực hiện các thao tác xử lý khác.
4. **Tạo phản hồi:** Sau khi xử lý, Servlet tạo ra một phản hồi, có thể là mã HTML, XML hoặc JSON.
5. **Gửi phản hồi:** Phản hồi được gửi từ Servlet container đến máy chủ web, sau đó được truyền lại cho người dùng qua trình duyệt.

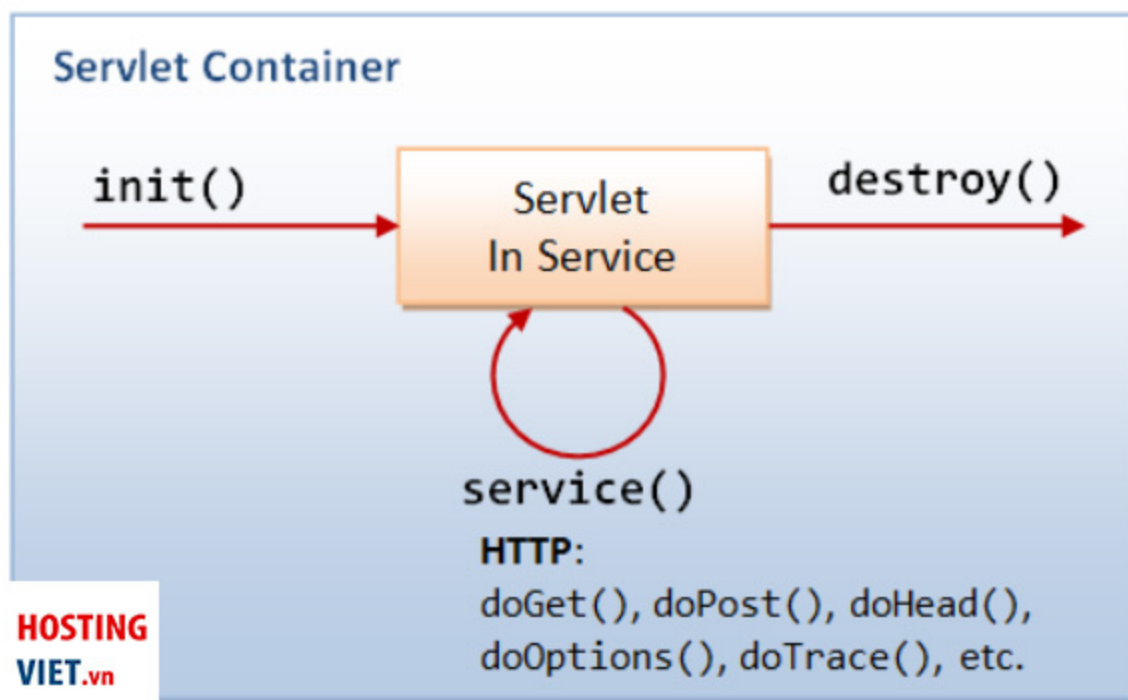


## Tìm hiểu vòng đời của Servlet

Web container có nhiệm vụ quản lý vòng đời của một Servlet. Nó sẽ tạo ra một phiên bản Servlet, rồi gọi ra `init()` method. Khi `init()` method được hoàn thành, Servlet sẽ ở trạng thái sẵn sàng cho bất kỳ yêu cầu dịch vụ nào đã nhận được từ clients.

Còn Container thực hiện xử lý yêu cầu thông qua việc tạo ra một thread mới cho từng yêu cầu mà nó được nhận từ thread pool trong Web Container, và sau khi nó tiến hành gọi `service()` method của Servlet. Trước khi tiến hành phá hủy instance, Container sẽ thực hiện lệnh gọi `destroy()` method. Sau khi phá hủy, Servlet bị chuyển thành rác chờ thu gom.

Tương tự như các chương trình Java khác, Servlet được chạy trong JVM. Trong khi đó, Servlet Container tham gia giải quyết những vấn đề phức tạp của HTTP request. Nó cũng chịu trách nhiệm tạo, thực hiện cũng như hủy Servlet.



## Phương thức xử lý một request của Servlet container và web server

Để xử lý một request bất kỳ, Servlet Container và web server trải qua các bước sau:

- **Bước 1:** Đầu tiên, máy chủ Web sẽ thực hiện nhận HTTP request.

- **Bước 2:** Web server chuyển tiếp yêu cầu đã nhận đến Servlet Container.
- **Bước 3:** Servlet tự động tiến hành lấy yêu cầu rồi tải chúng lên địa chỉ không gian Container (áp dụng trong trường hợp nó thuộc Container).
- **Bước 4:** Container thực hiện lệnh gọi `init ()` method của Servlet (chỉ gọi một lần khi Servlet tải lên lần đầu) để khởi tạo.
- **Bước 5:** Container tiến hành gọi `service ()` method của Servlet nhằm mục đích xử lý HTTP request. Điều này có nghĩa, chúng thực hiện việc đọc toàn bộ dữ liệu có trong yêu cầu, sau đó hình thành một response.
- **Bước 6:** Cuối cùng, máy chủ Web trả lại kết quả động tương ứng với vị trí yêu cầu.

