https://itphutran.com/gioi-thieu-ve-servlet/

Servlet là thành phần chính được sử dụng để phát triển các chương trình dịch vụ Java ở phía máy chủ

+ HTTP Protocol là gì

HTTP là giao thức trả lời yêu cầu: khách hàng (ví dụ: trình duyệt web) gửi yêu cầu về máy chủ (server) (ví dụ: máy chủ web) phản hồi bằng câu trả lời. Mỗi yêu cầu bao gồm "dòng yêu cầu", một chuỗi các dòng "tiêu đề" và tùy chọn là "nội dung". Một yêu cầu trực tuyến điển hình trông như thế này:

GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1

Ba phần của dòng này là:

* phương thức yêu cầu HTTP (ví dụ: "GET"),
* URL (ví dụ: "<http://www.w3.org/pub/WWW/TheProject.html> "), và
* phiên bản giao thức (ví dụ: "HTTP/1.1")

**+ Do get và Do post khác nhau điểm nào, luồng thông tin đi vào do get và do post ra sao.**

doGet() và doPost() là HTTP yêu cầu xử lý bởi các lớp servlet.

Trong doGet(), tham số được nối vào URL và được gửi cùng với thông tin tiêu đề . Điều này không xảy ra trong trường hợp doPost(). Trong doPost(), các tham số được gửi riêng. Vì hầu hết các máy chủ web chỉ hỗ trợ một lượng thông tin giới hạn được đính kèm vào tiêu đề, kích thước của tiêu đề này không được vượt quá 1024 byte. doPost() không có ràng buộc này. Thông thường, các lập trình viên tìm thấy nó khó chọn giữa doGet() và doPost().

doGet() phải được sử dụng khi lượng dữ liệu nhỏ và dữ liệu không nhạy cảm như truy vấn phải được gửi dưới dạng yêu cầu. doPost() sẽ được sử dụng khi lượng dữ liệu nhạy cảm tương đối lớn phải được gửi đi. Ví dụ đang gửi dữ liệu sau khi điền vào biểu mẫu hoặc gửi id đăng nhập và mật khẩu.

PHƯƠNG THỨC GET VÀ POST

GIỐNG: đều gửi dữ liệu lên web server sau khi người dùng nhập thông tin form và submit

KHÁC

GET:

**used for post-processing data (post=after). For example: storing, updating, etc.**

- gửi về URL có thêm thông tin người dùng (mã hóa)

http://www.example.com/index.htm?**name=value1&name1=value1**

-> không dùng để lưu password /thông tin nhạy cảm

- nhanh hơn POST nhiều vì dữ liệu luôn được browser cached lại,

nếu đã có trong cached rồi thì trả về ngay mà ko cần đưa tới servers

- chỉ sử dụng khi đọc/tìm kiếm data

POST:

**used for retrieving data (during). For example: browsing.**

- truyền thông tin ngầm qua HTTP header (mã hóa giống như get)

- Không cache, ko bookmark, ko lưu trong lịch sử browser

- server luôn thực thi và trả kết quả cho client

- sử dụng khi đăng kí, đăng nhập, thêm xóa sửa data