**I. Lý thuyết**

- **Servlet**: là chương trình chạy trên một Web hoặc ứng dụng máy chủ (Application Server) và hành động như một lớp trung gian giữa một yêu cầu đến từ một trình duyệt Web hoặc HTTP khách (Client) khác và cơ sở dữ liệu hoặc các ứng dụng trên máy chủ HTTP (HTTP Server).

Sử dụng Servlet, bạn có thể thu thập đầu vào từ người dùng thông qua các hình thức trang web, từ một cơ sở dữ liệu hoặc một nguồn khác, và tạo ra các trang web động.

\* Servlet là gì?

Servlet có thể được mô tả bằng nhiều cách, tùy thuộc vào ngữ cảnh:

Servlet là một công nghệ được sử dụng để tạo ra ứng dụng web.

Servlet là một API cung cấp các interface và lớp bao gồm các tài liệu.

Servlet là một thành phần web được triển khai trên máy chủ để tạo ra trang web động.

Có nhiều interface và các lớp trong API servlet như Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, ...

\* Công dụng:

- Nhận request từ client và lấy các thông tin từ request đã nhận

- Truy cập database để xử lý các nghiệp vụ và những phát sinh

- Thực hiện việc tạo, sau đó gửi request đến client hay tạo một request mới đến Servlet và JSP mới

\* Nhiệm vụ của Servlet

Servlet thực hiện các nhiệm vụ chính sau đây:

Đọc dữ liệu rõ ràng do khách hàng (trình duyệt) gửi. Điều này bao gồm một mẫu HTML trên một trang Web hoặc nó cũng có thể đến từ một applet hoặc một chương trình khách hàng HTTP tùy chỉnh.

Đọc dữ liệu yêu cầu HTTP ẩn được gửi bởi khách hàng (trình duyệt). Điều này bao gồm các cookie, loại phương tiện truyền thông và các chương trình nén mà trình duyệt hiểu được, v.v.

Xử lý dữ liệu và tạo ra các kết quả. Quá trình này có thể yêu cầu nói chuyện với một cơ sở dữ liệu, thực hiện một cuộc gọi RMI hoặc CORBA, gọi một dịch vụ Web, hoặc tính trực tiếp phản hồi.

Gửi dữ liệu rõ ràng (tức là tài liệu) tới khách hàng (trình duyệt). Tài liệu này có thể được gửi bằng nhiều định dạng, bao gồm văn bản (HTML hoặc XML), nhị phân (hình ảnh GIF), Excel, v.v ...

Gửi phản hồi HTTP ẩn cho khách hàng (trình duyệt). Điều này bao gồm nói với trình duyệt hoặc các trình khách khác loại tài liệu đang được trả về (ví dụ, HTML), thiết lập cookie và các tham số bộ nhớ đệm, và các tác vụ khác.

\* Gói Servlet

Java Servlets là các lớp Java chạy bởi một máy chủ web có một trình thông dịch hỗ trợ đặc tả Java Servlet.

Servlets có thể được tạo ra bằng cách sử dụng các gói javax.servlet và javax.servlet.http , đây là một phần chuẩn của phiên bản Enterprise của Java, một phiên bản mở rộng của thư viện lớp Java hỗ trợ các dự án phát triển quy mô lớn.

Các lớp này thực hiện các đặc tả Java Servlet và JSP.

Các servlet Java đã được tạo ra và biên dịch giống như các lớp Java khác. Sau khi cài đặt gói servlet và thêm chúng vào Classpath của máy tính, bạn có thể biên dịch các servlet bằng trình dịch Java của JDK hoặc bất kỳ trình biên dịch hiện tại nào khác.

Vòng đời:

Có 5 bước:

1. Tải lớp Servlet vào bộ nhớ.
2. Tạo đối tượng Servlet.
3. Gọi phương thức init() của Servlet.
4. Gọi phương thức service() của Servlet.
5. Gọi phương thức destroy() của Servlet.

\* Servlet Container

- Là sử dụng Java để tự động tạo ra trang web ở phía máy chủ , xử lý các yêu cầu tính toán từ phía client và trả lại các kết quả theo request của client đó. Thực chất nó là một phần của một web server tương tác với các servlet. Đúng như tên gọi servlet container giống như một cái thùng chứa, chứa tất cả các servlet ở bên trong.

\* Môi trường làm việc Servlet:

- Là 1 lớp Java nên nó cần được thực thi trên máy ảo Java(gọi là JVM) thông qua một dịch vụ tên là Servlet engine. Servlet engine sẽ thực hiện tải lớp Servlet đầu tiên ma nó được yêu cầu hoặc tại thời điểm Servlet engine bắt đầu. Tiếp đến, servlet sẽ ngừng tải để tập trung nguồn lực xử lý các yêu cầu khi Servlet engine bị dừng hoặc tắt.

- Vai trò: Sử dụng Servlet cho phép JVM có thể tiến hành xử lý từng yêu cầu riêng lẻ trong mỗi chuỗi Java riêng biệt

\* Cách Servlet container web server xử lý một request: có 6 bước

- Máy chủ Web nhận HTTP request

- Máy chủ Web chuyển tiếp yêu cầu tới servlet container

- Servlet được lấy ra một cách tự động và tải lên địa chỉ không gian của container, nếu nó nằm trong container.

- Container gọi init () method của servlet để khởi tạo (được gọi một lần khi servlet được load lần đầu tiên)

- Container gọi service () method của servlet để xử lý HTTP request, tức là đọc dữ liệu trong yêu cầu và hình thành một response

- Web server trả lại kết quả động đúng với vị trí yêu cầu

\* Các công cụ hỗ trợ:

- Eclipse: Tích hợp Git và Apache Maven, đa nền tảng;

- NetBeans: Thêm các công cụ tái cấu trúc mã,đa nền tảng;

- Intellij: Hỗ trợ nhiều ngôn ngữ khác nhau, có tính năng chạy thử, đa nền tảng;

- Công cụ khác: Emmet: tối giản thời gian viết code

\* Môi trường:

- ApacheTomcat: Ứng dụng mã nguồn mở thực hiện các công nghệ Java Servlet và Java Server Pages, hoạt động như máy chủ độc lập, tính ổn định cao;

- GlassFish: có 1 giao diện đơn giản

\* Mô hình MVC: Model  Controller -> View -> User

- Model: Tương tác và truy xuất dữ liệu đến database

- Controller: Có nhiệm vụ điều khiển tương tác giữa Model và View cũng như xử lý logic nghiệp vụ(Business)

- View: Giao diện mà người dùng có thể nhìn thấy, thường View chỉ có một nhiện vụ là hiển thị dữ liệu.

- Yêu cầu được gửi đến Controller(Servlet). Controller tiếp nhận, xử lý và làm việc với dữ liệu thông qua Model. Cuối cùng chuyển tiếp đến View(JSP) để tạo giao diện người dùng

- View lấy dữ liệu từ Model (đã được tạo ra từ Controller trước đó) để tạo giao diện theo yêu cầu

- Model đóng vài trò cung cấp, cập nhật dữ liệu và chia sẻ giữa Controller và View

- **Spring boot**: là một dự án phát triển bởi JAV (ngôn ngữ java) trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng.

\* Để phát triển một ứng dụng web cơ bản HelloWorld sử dụng Spring framework bạn sẽ cần ít nhất 5 công đoạn sau:

Tạo một project sử dụng Maven với các dependency cần thiết của Spring MVC và Servlet API.

Một tập tin web.xml để khai báo DispatcherServlet của Spring MVC.

Một tập tin cấu hình của Spring MVC.

Một class Controller trả về một trang “Hello World” khi có request đến.

Cuối cùng là phải có một web server dùng để triển khai ứng dụng lên chạy.

\* Một số tính năng nổi bật của Spring Boot:

Tạo các ứng dụng Spring độc lập

Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần phải deploy ra file WAR)

Các starter dependency giúp việc cấu hình Maven đơn giản hơn

Tự động cấu hình Spring khi cần thiết

Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML …

II. Bài tập

1. Kết nối database
2. Session
3. MVC
4. Lambda Expression
5. Java Stream