|  |  |
| --- | --- |
| Post và Get | POST và GET là hai phương thức HTTP được sử dụng để gửi dữ liệu từ máy khách (client) đến máy chủ (server) trên mạng.  Phương thức GET được sử dụng để yêu cầu và lấy các tài nguyên từ máy chủ. Dữ liệu được truyền qua URL và có thể được lưu trong bộ nhớ cache của trình duyệt. Tuy nhiên, phương thức GET không nên được sử dụng để gửi dữ liệu nhạy cảm hoặc quan trọng vì các thông tin gửi đi có thể bị xem trộm thông qua đường dẫn URL.  Phương thức POST được sử dụng để gửi dữ liệu từ máy khách đến máy chủ, thông qua các trường dữ liệu được gửi trong phần thân của yêu cầu HTTP. Dữ liệu được mã hóa và không lưu trong bộ nhớ cache của trình duyệt. Phương thức POST thường được sử dụng để gửi dữ liệu nhạy cảm hoặc quan trọng như thông tin đăng nhập hoặc thông tin thẻ tín dụng.  Tóm lại, GET được sử dụng để lấy thông tin từ máy chủ và POST được sử dụng để gửi thông tin đến máy chủ.  - Tóm lại Get gửi thông tin qua thanh URL còn Post gửi ngầm và không hiển thị thông tin trên thanh URL nên sẽ an toàn thông tin hơn |
| JSP | JSP là viết tắt của JavaServer Pages là một công nghệ để phát triển các trang web động. JSP giúp các nhà phát triển chèn java code vào các trang HTML bằng cách sử dụng các thẻ JSP đặc biệt.  JSP là một kiểu Java servlet được thiết kế để tạo ra giao diện người dùng cho một ứng dụng Java web. Các nhà phát triển web viết các JSP như các tệp văn bản kết hợp mã HTML hoặc XHTML, các phần tử XML, các action và lệnh JSP.  Sử dụng JSP, bạn có thể thu thập dữ liệu đầu vào từ người dùng thông qua các Form của trang web, trình bày các bản ghi từ một cơ sở dữ liệu hoặc một nguồn khác, và tạo các trang web động. |
| Doget() | Phương thức DoGet được sử dụng để xử lý các yêu cầu GET gửi từ máy khách đến Servlet. DoGet được gọi khi Servlet nhận được một yêu cầu GET từ phía máy khách. Trong phương thức DoGet, các tham số của yêu cầu được truyền vào thông qua đối tượng HttpServletRequest và các phản hồi được trả về thông qua đối tượng HttpServletResponse.  - Nếu chúng gửi thông tin qua thanh URL để vào doget thì sẽ lộ thông tin. Nếu ta code thì vẫn có thể vào được dopost nhưng sẽ ảnh hưởng tới mô hình để lập trình wed |
| Dopost() | Phương thức DoPost được sử dụng để xử lý các yêu cầu POST gửi từ máy khách đến Servlet. DoPost được gọi khi Servlet nhận được một yêu cầu POST từ phía máy khách. Tương tự như DoGet, các tham số của yêu cầu được truyền vào thông qua đối tượng HttpServletRequest và các phản hồi được trả về thông qua đối tượng HttpServletResponse.  - Nếu dùng dopost thì thì chúng ta phải dùng các nút botton với kiểu submit trong phần body để gửi thông tin ngầm định để gửi thông tin tới sever để xử lý. |
| getParameter | Phương thức getParameter() được sử dụng trong Servlet của Java để lấy các tham số (parameter) mà máy khách đã gửi đến trên đường dẫn URL hoặc thông qua phương thức POST.    để lấy giá trị của tham số "username" được truyền qua phương thức POST, bạn có thể sử dụng đoạn mã sau trong Servlet của mình. ( username ta có thể lấy dữ liệu từ cái thẻ nhập dữ liệu như input trong html nhưng ta phải cài name của input giống như tên bển trong getParameter( ở đây ). |
| setAttribute | setAttributelà một phương thức trong JavaScript cho phép bạn đặt giá trị của một thuộc tính được chỉ định của một phần tử HTML.    Ở đây, elementlà tham chiếu đến một phần tử HTML, attributelà tên của thuộc tính bạn muốn đặt và valuelà giá trị bạn muốn đặt cho thuộc tính.    - ở đây student bên phải là truyền vào một đối tượng vào một tên biến student ở bên trái để ta có thể sử dụng trong jsp để truyền dữ liệu của đối tượng student đó vào html để lấy dữ liệu sau khi xử lý truyền vào html để truyền ra |
| getRequestDispatcher | Đối tượng này được sử dụng để chuyển tiếp yêu cầu từ một servlet này sang một servlet khác, trang JSP hoặc bất kỳ tài nguyên nào khác có thể xử lý yêu cầu và tạo phản hồi.  - Sau khi ta truyền một dữ liệu nào đó bằng setAttribute thì ta chuyển sang trang html hay jsp nào khác để in ra gửi về phía client hoặc ta gửi về một sẻvlet nào khác để xử lí |
| Init() | Trong Java Web, từ khóa "init" thường được sử dụng để định nghĩa phương thức khởi tạo của một Servlet.  Một Servlet là một lớp Java được sử dụng để xử lý các yêu cầu từ một trình duyệt web và tạo ra các phản hồi tương ứng. Khi một Servlet được khởi động, phương thức "init" của nó sẽ được gọi để thực hiện việc khởi tạo các tài nguyên cần thiết cho Servlet.  Phương thức "init" có thể được định nghĩa trong Servlet bằng cách ghi đè phương thức này từ lớp cha HttpServlet. Phương thức "init" có thể có các tham số đầu vào, bao gồm đối tượng ServletConfig, cho phép truy cập thông tin cấu hình Servlet. |
| Super.init(); | Trong Java, từ khóa "super" được sử dụng để tham chiếu đến lớp cha của một lớp. Trong trường hợp của phương thức "init" trong Servlet, "super.init()" được sử dụng để gọi phương thức "init" của lớp cha HttpServlet.  Khi bạn định nghĩa một lớp Servlet, bạn thường sẽ ghi đè phương thức "init" từ lớp cha HttpServlet để thực hiện việc khởi tạo các tài nguyên cần thiết cho Servlet của bạn. Tuy nhiên, bạn vẫn muốn giữ lại một số chức năng của phương thức "init" của lớp cha. |
| JSP | -Là công nghệ được sử dụng để tạo web động, được viết dựa trên HTML, có thể nhúng JAVA để đổ dữ liệu  -> Về cở bản JSP cũng là servlet  là một công nghệ phát triển web được sử dụng để tạo các trang web động. JSP cho phép các lập trình viên sử dụng Java để tạo các trang web động bằng cách chèn mã Java vào các trang HTML. Khi một trang JSP được yêu cầu, server sẽ chạy mã Java trong trang và trả về một trang HTML được tạo ra bởi mã đó.  JSP có thể được sử dụng để tạo các trang web tương tác động với người dùng, như trang đăng nhập, trang giỏ hàng, trang đặt hàng và nhiều chức năng khác. JSP có thể được sử dụng với các công nghệ web khác như Servlet, JSTL (JSP Standard Tag Library), và JDBC (Java Database Connectivity) để tạo các ứng dụng web đầy đủ chức năng. |
| JSTL | - Là bộ thư viện thẻ được xây dựng dựa trên ngôn ngữ JAVA dùng cho JSP  - Thay vì phải nhúng code java trong ngôn ngữ JSP thì ta có thể dùng JSTL để thay thế  là một thư viện các thẻ (tags) tiêu chuẩn trong JSP được sử dụng để phát triển các ứng dụng web. JSTL được thiết kế để giúp lập trình viên giảm thiểu việc viết mã Java trực tiếp trong trang JSP bằng cách cung cấp các thẻ đơn giản để thực hiện các tác vụ phổ biến như lặp qua danh sách, kiểm tra điều kiện, định dạng số và ngày tháng...  JSTL có các phần tử và thuộc tính như HTML và XML, và chúng được sử dụng để đưa ra các chỉ dẫn cho việc xử lý và hiển thị dữ liệu trên trang JSP. Một số ví dụ về các thẻ JSTL bao gồm:  <c:forEach> để lặp qua một danh sách và hiển thị các giá trị  <c:if> để kiểm tra một điều kiện và hiển thị nội dung tương ứng  <c:choose> và <c:when> để kiểm tra nhiều điều kiện và hiển thị nội dung tương ứng  <c:set> để đặt một giá trị trong một biến và sử dụng nó trong trang JSP |
| Kết nối với SQL | dependencies **{** compileOnly('javax.servlet:javax.servlet-api:4.0.1')  implementation group: 'javax.servlet', name: 'jstl', version: '1.2'  implementation group: 'mysql', name: 'mysql-connector-java', version: '8.0.26' **}**  public class Repository {  private String jdbcUrl = "jdbc:mysql://localhost:3306/testjavaweb";  private String username = "root";  private String password = "12345";  private Connection connection;   public Repository() {  }   public Connection getConnectionJavaToDB() {  try {  Class.*forName*("com.mysql.cj.jdbc.Driver");  this.connection = DriverManager.*getConnection*(jdbcUrl, username, password);  return connection;  } catch (ClassNotFoundException e) {  e.printStackTrace();  } catch (SQLException e) {  e.printStackTrace();  }  return null;  } } |
| Statement | - Dùng để tạo câu lệnh truy vấn tĩnh SQL, nhưng sẽ tạo nên lỗ hổng bảo mật ( SQL injection )  - sever sẽ trả dữ liệu mở cho nên rất dễ để hack lấy dữ liệu cho nên sẽ không được bảo mật thông tin riêng tư cho client. |
| PreparedStatement | - Dùng câu lệnh truy vấn động ( Truyền tham số ? vào ), bởi vì tham số là ? nên sẽ ngăn chặn được SQL ịnection.  - PreparedStatement sẽ truyền kết quả của câu lệnh truy vấn SQL vào dấu chấm hỏi ( ? ) khi tơi được nơi cần đến thì sẽ giải dấu chấm hỏi đó ra để lấy dữ liệu gửi về client. Cho nên sẽ bảo mật thông tin hơn cho khách hàng |