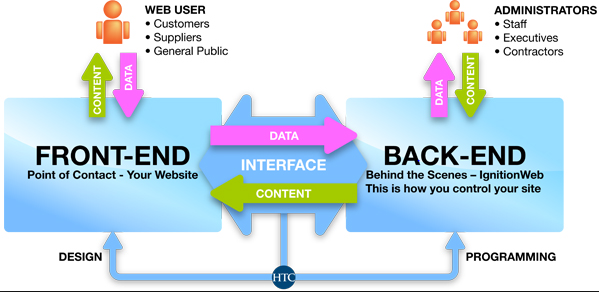
FRONT-END

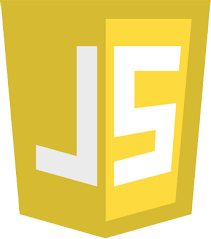
**I .FRONT-END:**

Phần front-end của một trang web là phần tương tác với người dùng. Tất cả mọi thứ bạn nhìn thấy khi điều hướng trên Internet, từ các font chữ, màu sắc cho tới các menu xổ xuống và các thanh trượt, là một sự kết hợp của HTML, CSS, và JavaScript được điều khiển bởi trình duyệt máy tính của bạn.

Các lập trình front-end chịu trách nhiệm cho giao diện của một trang web và kiến trúc những trải nghiệm của người dùng.

Để thực hiện được những mục tiêu đó, các lập trình viên front-end phải tinh thông 3 ngôn ngữ chính: HTML, CSS, và ngôn ngữ lập trình JavaScript.

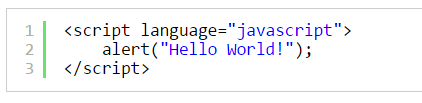


1. **JAVASCRIPT**
   1. **Giới thiệu**

* JavaScript là một ngôn ngữ lập trình kịch bản dựa trên đối tượng. Được sử dụng rộng rãi cho các trang web.
* Phần mở rộng cho tập tin của JavaScript là .js

Viết chương trình JavaScript đầu tiên:

Tất cả những đoạn mã Javascript đều phải đặt trong cặp thẻ **<script>...</script>**

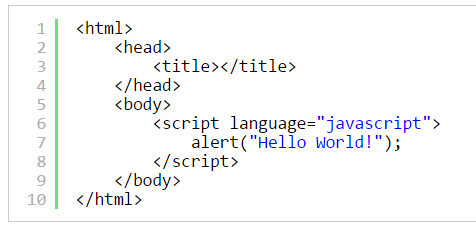


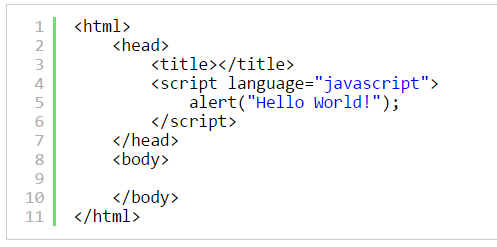
**Đặt thẻ script javascript ở đâu?**

Internal - viết trong file html hiện tại:

Thông thường chúng ta sẽ viết những đoạn mã javascript trên phần <**head**>, tuy nhiên đó không phải là điều kiện bắt buộc, nghĩa là bạn có thể đặt ở đâu tùy thích miễn là những đoạn mã đó phải được bao lại bằng thẻ <script>.

Đặt trong thẻ head, body:



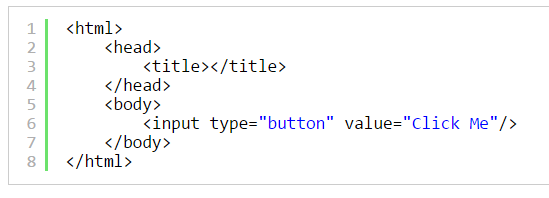


**External - viết ra một file js khác rồi import vào:**



**Viết chương trình Hello World!**

**Bước 1:** Tạo một file index.html và lưu tại bất kì vị trí nào desktop hay ổ D, ổ C phần mở rộng là .html và gõ nội dung sau vào:



Chạy lên bằng Chrome bạn thấy xuất hiện một button. Bây giờ ta sẽ viết ứng dụng khi click vào button đó thì sẽ xuất hiện một thông báo **Hello World** . Nhưng lưu ý với bạn là button đó có id=clickme.

**Bước 2:** Viết javascript khi click vào button có id="clickme" thì thông báo lên màn hình.

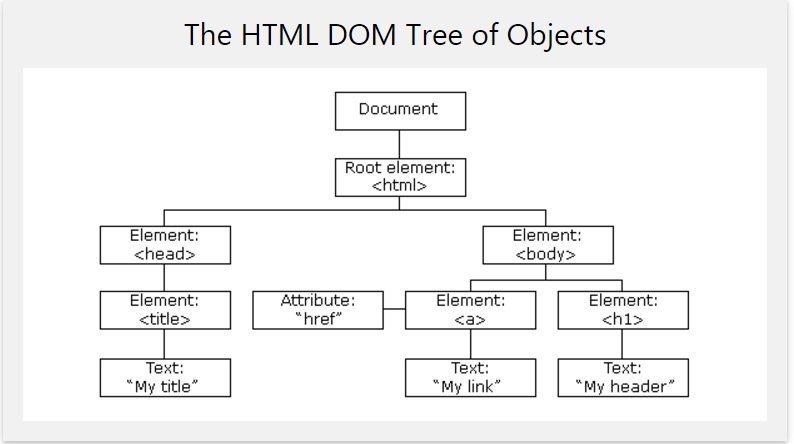


**DOM là gì? Các loại DOM trong Javascript**

Để thao tác được với các thẻ HTML thì nó phải thông qua một cơ chế ta gọi là DOM và ta hay gọi là Document Object Model.

DOM là viết tắt của chữ Document Object Model, dịch tạm ra là mô hình các đối tượng trong tài liệu HTML.

Trong mỗi thẻ HTML sẽ có những thuộc tính (Properties) và có phân cấp cha - con với các thẻ HTML khác. Sự phân cấp và các thuộc tính của thẻ HTML này ta gọi là [selector](http://freetuts.net/selector-la-gi-tim-hieu-css-selector-can-ban-345.html) và trong DOM sẽ có nhiệm vụ xử lý các vấn đề như đổi thuộc tính của thẻ, đổi cấu trúc HTML của thẻ, …



Trong hình thì tất cả các thẻ HTML sẽ được quản lý trong đối tượng document, thẻ cao nhất là thẻ html, tiếp theo là phân nhánh body và head. Bên trong head thì có những thẻ như style, title, ... và bên trong body thì là vô số các thẻ HTML khác.

Như vậy trong Javascript để thao tác với các thẻ HTML ta phải thông qua đối tượng **document** ví dụ:

**<html>**

**<body>**

<h1 id="main-content"></h1>

<**script** language="javascript">

**document**.getElementById("main-content").innerHTML = "Chào mừng các bạn đến với website học lập trình"

**</script>**

**</body>**

**</html>**

=> Đoạn code này có ý nghĩa rằng tìm thẻ có id="main-content"và gán nội dung HTML bên trong của thẻ này là dòng chữ "*Chào mừng các bạn đến với website học lập trình”*

**Các thể loại DOM trong JavaScript:**

**-**Việc xử lý và làm việc với đối tượng HTML rất phức tạp và đa dạng, chính vì vậy javascript có cung cấp cho chúng ta nhiều phương thức, đối tượng và mỗi thành phần như vậy sẽ có những nhiệm vụ riêng biệt.

**Danh sách chia nhóm DOM:**

* DOM document: có nhiệm vụ lưu trữ toàn bộ các thành phần trong tài liệu của website
* DOM element: có nhiệm vụ truy xuất tới thẻ HTML nào đó thông qua các thuộc tính như tên class, id, name của thẻ HTML
* DOM HTML: có nhiệm vụ thay đổi giá trị nội dung và giá trị thuộc tính của các thẻ HTML
* DOM CSS: có nhiệm vụ thay đổi các định dạng CSS của thẻ HTML
* DOM Event: có nhiệm vụ gán các sự kiện như onclick(), onload() vào các thẻ HTML
* DOM Listener: có nhiệm vụ lắng nghe các sự kiện tác động lên thẻ HTML đó
* DOM Navigation dùng để quản lý, thao tác với các thẻ HTML, thể hiện mối quan hệ cha - con của các thẻ HTML
* DOM Node, Nodelist: có nhiệm vụ thao tác với HTML thông qua đối tượng (Object).
  1. **Event Handler**

- Sự kiện là một hành động nào đó tác động lên trên đối tượng.

- Mỗi sự kiện chúng ta có thể có nhiều hành động khác nhau.

- Khi một trang tải, nó được gọi là một sự kiện(Event). Khi người dùng nhấn vào một nút, thì nhấn đó cũng là một sự kiện.

- Sự tương tác của JavaScript với HTML được xử lý thông qua các sự kiện xảy ra khi người sử dụng hoặc trình duyệt thao tác một trang.

**JavaScript có thể làm những gì?**

- Xử lý sự kiện có thể được sử dụng để xử lý, và xác minh, người dùng nhập vào, các hành động của người dùng, và các hành động trình duyệt:

- Những điều đó nên được thực hiện mỗi lần một trang tải

- Những điều đó nên được thực hiện khi trang được đóng lại

- Hành động cần được thực hiện khi người dùng nhấp chuột vào một nút

- Nội dung cần được xác nhận khi một dữ liệu người dùng đầu vào

- Nhiều phương pháp khác nhau có thể được sử dụng để cho việc JavaScript với các sự kiện:

- Thuộc tính sự kiện HTML có thể thực thi mã JavaScript trực tiếp

- Thuộc tính sự kiện HTML có thể gọi hàm JavaScript

- Bạn có thể gán các chức năng xử lý sự kiện của riêng bạn.

- Bạn có thể ngăn chặn các sự kiện được gửi hoặc bị xử lý.

**Ví dụ:**

**Bảng các sự kiện thông dụng trong JavaScript:**

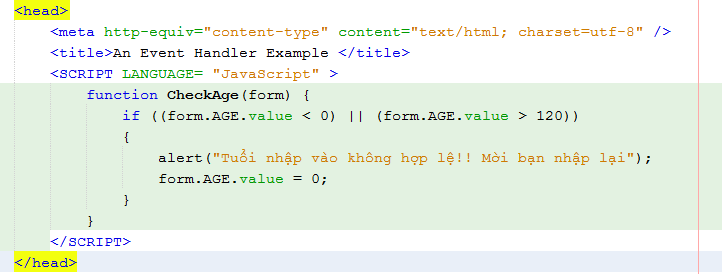
|  |  |  |
| --- | --- | --- |
| **STT** | **Event Name** | **Description** |
| **1** | **onclick** | **Xảy ra khi click vào thẻ HTML** |
| **2** | **ondbclick** | **Xảy ra khi double click vào thẻ HTML** |
| **3** | **onchange** | **Xảy ra khi giá trị (value) của thẻ HTML đổi. Thường dùng trong các đối thẻ form input** |
| **4** | **onmouseover** | **Xảy ra khi con trỏ chuột bắt đầu đi vào thẻ HTML** |
| **5** | **onmouseout** | **Xảy ra khi con trỏ chuột bắt đầu rời khỏi thẻ HTML** |
| **6** | **onmousemove** | **Xảy ra khi con chuột di chuyển bên trong thẻ HTML** |
| **7** | **onkeydown** | **Xảy ra khi gõ một phím bất kì vào ô input** |
| **8** | **onload** | **Sảy ra khi thẻ HTML bắt đầu chạy, nó giống như** [**hàm khởi tạo**](http://freetuts.net/ham-khoi-tao-va-ham-huy-trong-lap-trinh-huong-doi-tuong-php-36.html) **trong**[**lập trình hướng đối tượng**](http://freetuts.net/tag/lap-trinh-huong-doi-tuong) **vậy đó.** |
| **9** | **onkeyup** | **Xảy ra khi bạn gõ phím nhưng lúc bạn nhã phím ra sẽ được kích hoạt** |
| **10** | **onkeypress** | **Xảy ra khi bạn nhấn môt phím vào ô input** |

**Bảng các chương trình xử lý sự kiện có sẵn của một đối tượng:**

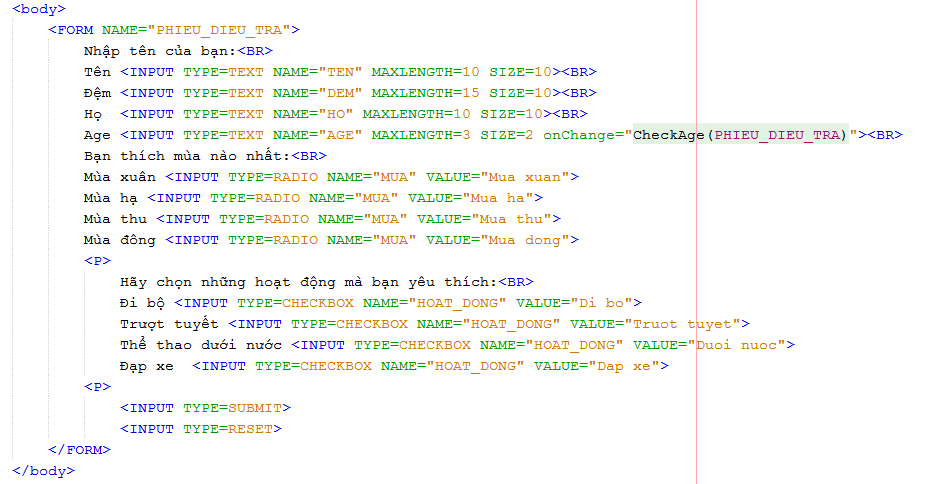
|  |  |
| --- | --- |
| **Đối tượng** | **Chương trình xử lý sự kiện có sẳn** |
| **Selection list** | **onBlur, onChange, onFocus** |
| **Text** | **onBlur, onChange, onFocus, onSelect** |
| **Textarea** | **onBlur, onChange, onFocus, onSelect** |
| **Button** | **onClick** |
| **Checkbox** | **onClick** |
| **Radio button** | **onClick** |
| **Hypertext link** | **onClick, onMouseOver, onMouseOut** |
| **Clickable Imagemap area** | **onMouseOver, onMouseOut** |
| **Reset button** | **onClick** |
| **Submit button** | **onClick** |
| **Document** | **onLoad, onUnload, onError** |
| **Window** | **onLoad, onUnload, onBlur, onFocus** |
| **Framesets** | **onBlur, onFocus** |
| **Form** | **onSubmit, onReset** |
| **Image** | **onLoad, onError, onAbort** |

Ví dụ sau là một đoạn mã script đơn giản của chương trình xử lý sự kiện kiểm tra giá trị đưa vào trong trường text. **Tuổi của người sử dụng được nhập vào trong trường này và chương trình xử lý sự kiện sẽ kiểm tra tính hợp lệ của dữ liệu đưa vào**. Nếu không hợp lệ sẽ xuất hiện một thông báo yêu cầu nhập lại. Chương trình xử lý sự kiện được gọi mỗi khi trường AGE bị thay đổi và focus chuyển sang trường khác.

**-Phần thẻ <head></head>**



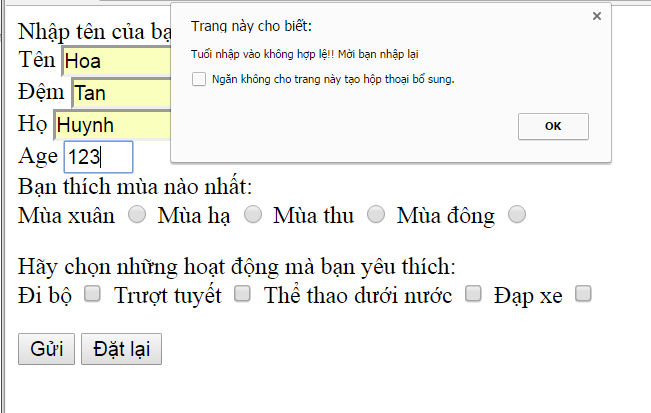
-**Phần thẻ <body></body>**



**Chạy trang web:**



**Khi nhập quá số tuổi qui định, Chương trình xử lí sẽ bắt sự kiện CheckAge() và thông báo bằng hàm alert().**

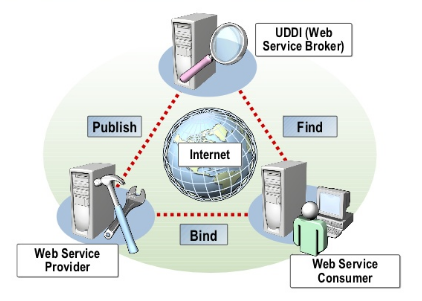


1. **WEB-SERVICE:**

Web Service là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML. Web service là tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu. Việc trao đổi dữ liệu trong một Web Service thường diễn ra thông qua giao thức TCP ở cổng 80, sử dụng phương thức POST trong giao thức HTTP chuẩn.

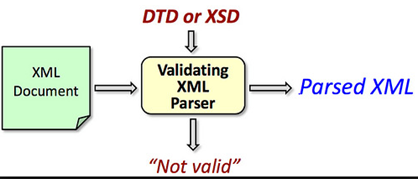
1. Mô hình hoạt động: gồm 3 thành phần chính:

Web Service Provider: Nhà cung cấp dịch vụ web.

* + Xây dựng và cài đặt đối tượng dịch vụ lên 1 Web Server và cung cấp bản mô tả dịch vụ để hướng dẫn mọi người cách khai thác. Ngôn ngữ sử dụng cho các bản mô tả dịch vụ là SDL-Service Description Language, với cấu trúc cú pháp của XML.
* Web Service Consumer: Nhà khai thac dịch vụ web.
  + Xem xét bản mô tả dịch vụ của nhà cung cấp, dực vào đó để xây dựng lớp trung gian truy xuất đối tượng dịch vụ và cuối cùng thiết kế giao diện cho ứng dụng.
* End User: Người dùng cuối.
  + Truy cấp trang web của các nhà khai thác dịch vụ và sử dụng một cách gián tiếp Web Service của Provider.

1. SOAP (Simple Object Access Protocol): một trong những giao thức “chuẩn” do W3C đưa ra dựa trên nền tảng XML cho việc xay dựng Web Service, là sự kết hợp giữa HTTP và XML.

* XML (eXtensible Markup Language): Ngôn ngữ đánh dấu mở rộng do W3C đề nghị.
  + khi đưa vào sử dụng cần thỏa mãn 2 điều kiện: dịnh dạng đúng và hợp lệ.
    - Định dạng đúng: tuân thủ đung chính xác cú pháp của ngôn ngữ XML.
    - Hợp lệ: các tên của các phần tử phải phù hợp với các quy định trong DTD hoặc SXD.
      * DTD (Document Type Declaration): một cách để miêu tả ngôn ngữ XML, Tuy nhiên phương pháp này có nhiều hạn chế như: không có nhiều dạng kiểu dữ liệu, không qui định được khoảng giá trị….
      * XSD (**XML Schema** Definition): hoàn toàn giống như DTD, được tạo ra để khắc phục tất cả các nhược điểm của DTD:
        + Định nghĩa cấu trúc các thành phần có trong XML.
        + Định nghĩa các thuộc tính có trong Schema.
        + Định nghĩa các thành phần con và thứ tự xuất hiện của chúng trong thành phần cha.
        + Định nghĩa một thành phần là rỗng hay có chứa text.
        + Định nghĩa kiểu dữ liệu cho các thành phần và thuộc tính.
        + Định nghĩa giá trị mặc định cho thuộc tính.



c. Api apache CXF:

* API là một **giao diện lập trình ứng dụng** (*Application Programming Interface* hay *API*) là một [giao diện](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) để xin cấp phát bộ nhớ và truy xuất tập tin.
* **Apache** hay là **chương trình máy chủ HTTP** là một chương trình dành cho máy chủ đối thoại qua giao thức [HTTP](https://vi.wikipedia.org/wiki/Hypertext_Transfer_Protocol). Apache chạy trên các hệ điều hành tương tự như [Unix](https://vi.wikipedia.org/wiki/Unix), Microsoft Windows, Novell Netware và các hệ điều hành khác. Apache đóng một vai trò quan trọng trong quá trình phát triển của mạng web thế giới.

Apache CXF là một bộ mã nguồn mở để giúp tạo ra các Web Service rất dễ dàng dựa trên các giao thức.



Các dịch vụ phát triển thông qua Apache CXF có thể nói chuyện với một loạt các giao thức, cho phép các nhà phát triển để xây dựng các ứng dụng đa chức năng có thể làm việc trên nhiều nền tảng và tầng giao vận.

Apache CXF là rất mở rộng, cũng là tài liệu, trực quan và dễ sử dụng.

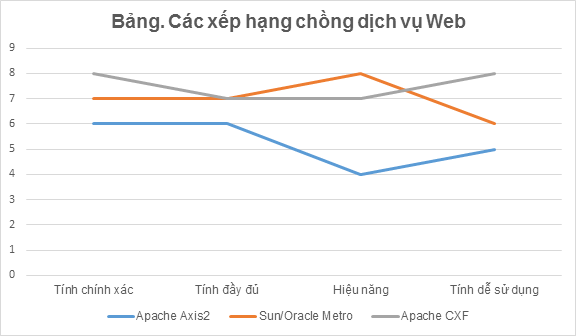
Các dịch vụ được tạo ra thông qua Apache CXF có thể được triển khai với Jetty, Spring, hoặc container Tomcat, có một hiệu suất cao, và có JBI và Java EE tích hợp sẵn là tốt.

Điều gì là mới trong phiên bản 3.1.0:

* CXF 3.1 không còn hỗ trợ Java 6. Bạn phải sử dụng Java 7 hoặc Java 8.
* JAX-WS / Simple lối ServerFactoryBean sẽ tự động gọi thiết lập lại vào cuối create () gọi. Điều này cho phép các nguồn lực phải được làm sạch và thu gom rác thải sớm. Tuy nhiên, nó cũng ngăn ngừa nhiều cuộc gọi để tạo ra () từ việc chia sẻ các ServerInfo cùng / EndpointInfo / etc ... các đối tượng như họ sẽ làm chúng ta 3.0.x. chia sẻ rằng đã gây ra nhiều vấn đề trong quá khứ do chia tài sản (như cache token) được lưu trữ trên các đối tượng để các hành vi mới là hơn & quot; đúng & quot ;, nhưng nó khác so với phiên bản trước đó để chăm sóc phải được thực hiện trong khi nâng cấp .
* Các Karaf tập features.xml cho CXF 3.1 sẽ không còn cài đặt vào mùa xuân hoặc mùa xuân-dm khi cài đặt & quot; CXF & quot; đặc tính. Nếu bạn cần mùa xuân / mùa xuân-dm, bạn sẽ cần phải cài đặt những tính năng trước khi cài đặt các tính năng CXF.

Apache Axis2, Sun/Oracle Metro và Apache CXF — đều cung cấp một mức hỗ trợ khá cao cho các tiêu chuẩn bảo mật, gồm cả hoạt động bảo mật lẫn cách các chồng được cấu hình với các tham số bảo mật thời gian chạy ra sao.

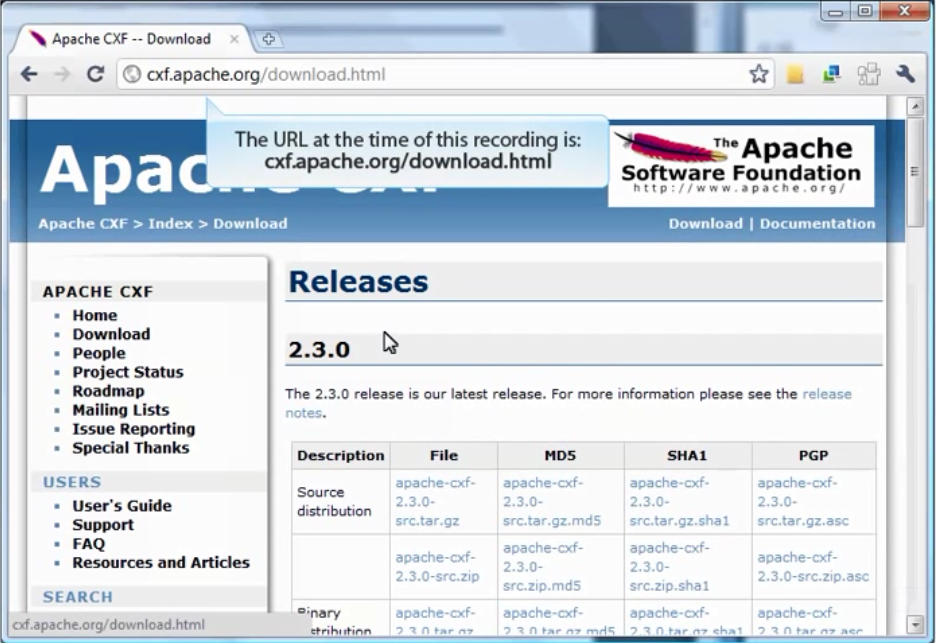
Các vấn đề của CXF:

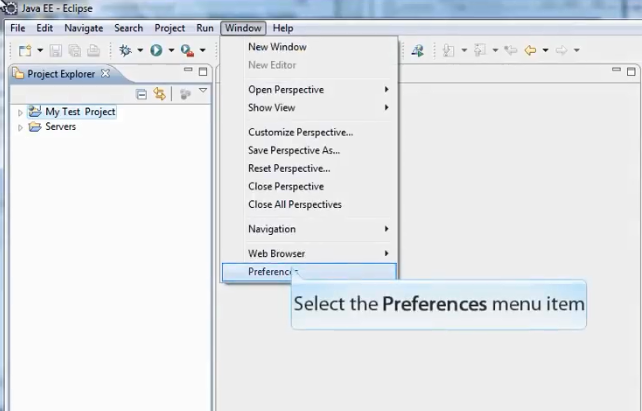
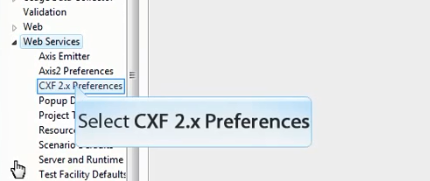


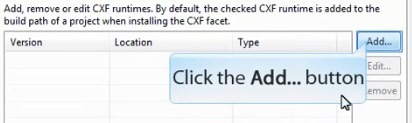
* Tính chính xác: CXF chỉ có các vấn đề đã biết tương đối nhỏ và trả lời nhanh với các vấn đề và chu kỳ phát hành nhanh có nghĩa là các vấn đề thường được sửa chữa ngay sau khi người ta tìm ra chúng.
* Tính đầy đủ: hỗ trợ cho tất cả các tiêu chuẩn bảo mật.
* Hiệu năng: Metro nhanh hơn một chút so với CXF về tổng thể.
* Tính dễ sử dụng: CXF cũng hỗ trợ các tài liệu tham khảo chính sách bên ngoài trong WSDL, một tính năng quan trọng đối với các tổ chức muốn chuẩn hóa các chính sách bảo mật trên toàn doanh nghiệp.

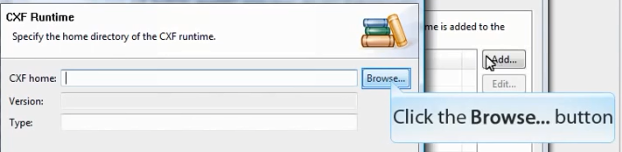
d. Configuration Api apache CXF:

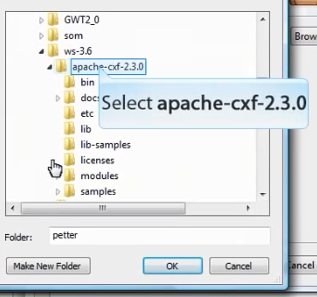
* Bước 1: Download Apache CXF từ:
  + <http://cxf.apache.org/download.html>



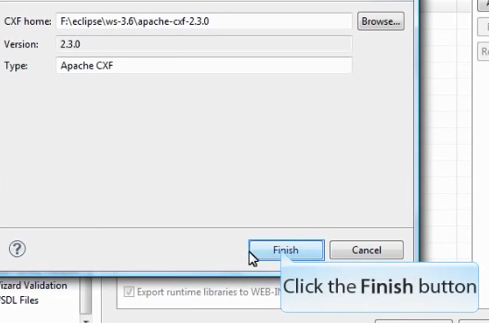
* Bước 2.1: Cấu hình Apche CXF trên Eclipse:
  + Mở Eclipse>Window>Preferences.
  + Trong cửa sồ Preferences chọn Web Service>CXF 2x Preferences. 
  + Chọn Add:

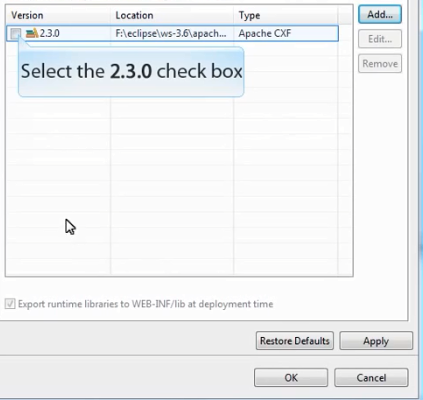
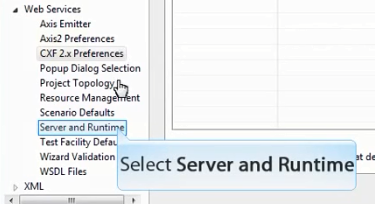
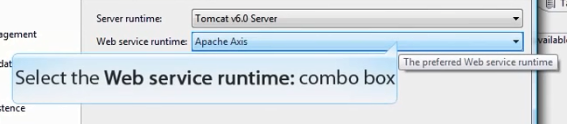


* + Bấm Browse chọn tới đường dẫn đã tải file CXF về.
  + Chọn thư muc CXF và bấm OK



* + Xong chọn Finish để add CXF vào.



* + Chọn CXF và bấm OK
  + Chọn Window>Preferences, trong cửa sồ Preferences chọn Web Service>Server and Runtime:
  + Chọn Web Sevice runtime: combo box: chọn Apche CXF 2x>Ok.

1. **JQUERY**
   1. Giới thiệu

Là thư viện mở của JavaScript tạo ra các hiệu ứng có thể tương tác trực tiếp với người đọc một cách nhanh chóng và dễ dàng hơn rất nhiều là sử dụng thuần JavaScript.

**Các đặc trưng cơ bản của Jquery:**

- Thao tác HTML/DOM

- Thao tác CSS

- Phương thức sự kiện HTML

- Hiệu ứng và chuyển động

- Ajax

**Jquery dùng để làm gì?**

- Cải thiện hiệu suất của ứng dụng

- Tượng thích hầu hết các trình duyệt

- Jquery giúp thực hiện UI mà không phải viết nhiều dòng code

- Với Jquẻy bạn có thể tạo ra các hiệu ứng đẹp mắt cho website

**Sử dụng cục bộ:**

-Vào trang chủ jquery. com để tải phiên bản jquery mới nhất

- Chèn thư viện jquery vào trang như sau:

Ví dụ:

**Sử dụng thư viện có sẵn:**

- Chèn đoạn Script online có sẵn vào để khai báo thư viện Jquery:

Ví dụ:

**Có 2 cách để thực thi code:**

**Ví dụ:**

**2. JQuery Selectors**

Thư viện jQuery tận dụng kiến thức và thế mạnh của CSS Selector để cho phép bạn nhanh chóng và dễ dàng truy cập nhiều phần tử hoặc nhóm các phần tử trong DOM (Document Object Model).

**Hàm $()**

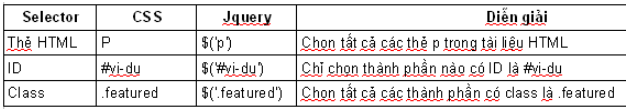
Cho dù bạn sử dụng Selector nào đi chăng nữa trong jQuery, bạn luôn bắt đầu bằng một dấu dollar ($) và một đôi ngoặc đơn như: $(). Tất cả những gì có thể được sử dụng trong CSS cũng có thể được lồng vào dấu ngoặc kép (”) và đặt vào trong hai dấu ngoặc đơn, cho phép chúng ta áp dụng các phương pháp jQuery cho tập hợp các phần tử phù hợp.

Ba thành phần quan trọng nhất của jQuery Selector là tên thẻ HTML, ID và Class. Bạn có thể chỉ sử dụng nó hoặc kết hợp với những Selector khác để chọn.

- Để chọn một hoặc nhiều Element. Element nào được chọn thì chúng ta có thể thực hiện các thao tác jquery lên nó.

* Thẻ HTML
  + $(“p”): chọn tất cả các thẻ p
  + $(“a”): Chọn tất cả các thẻ a
* Theo ID
  + $(“#content”): Chọn thẻ có ID= content
* Theo Class
  + $(“.box”): Chọn tất cả các thẻ có class=“box”

**Ví dụ:**



**Tạo host jquery và import file first-jquery vào:**

**Ajax**

Ajax là gì?

-Ajax là từ viết tắt của Asynchronous JavaScript and XML

-JavaScript bao gồm các tính năng gửi yêu cầu http không đồn bộ sử dụng đối tượng XMLHttpRequest.

-Ajax sử dụng khả năng này của JavaScript để gửi yêu cầu về máy chủ và nhận được dữ liệu xml từ máy chủ sau khi xử lý (cũng có các định khác) và cập nhật các phần của trang web (sử dụng JavaScript) mà không cần tải lại hoặc làm mới toàn bộ trang web.

**Hình:**

**Ưu điểm của Ajax:**

- Hổ trợ đa trình duyệt

- Phương pháp đơn giản dễ sử dụng

- Khả năng gửi GET và POST yêu cầu

- Khả năng để tải Json, XML, HTML hoặc Script

**Một số phương thức Ajax Jquery**

- Ajax():Gửi yêu cầu http không đồng bộ với máy chủ

- load(): Gửi yêu cầu http nạp html hay văn bản nội dung từ máy chủ và thêm chúng vào DOM yếu tố

-get(**):** Gửi http GET yêu cầu để tải dữ liệu từ máy chủ.

- post(): Gửi yêu cầu POST http nộp hoặc tải dữ liệu đến máy chủ.

- getJSON(): Gửi http GET yêu cầu để tải JSON mã hoá dữ liệu từ máy chủ.

- getScript(): Gửi http GET yêu cầu để tải các tập tin JavaScript từ máy chủ và sau đó thực hiện nó

**Phương thức jquery ajax():**

- Phương thức ajax() lấy dữ liệu từ server và trả dữ liệu cho phần tử được chọn.

- Cú pháp:

$.ajax(url);

$.ajax(url,[option]);

-URL: mà bạn muốn lấy dữ liệu

-Option: là tham số tuỳ chọn

-Nếu có tham số thì định dạng tham số theo JSON

**Phương thức query get()**

- Phương thức get() gửi yêu cầu GET không đồng bộ đến máy chủ và truy xuất dữ liệu…

-Cú pháp:

$.get(Url, [data], [callback]);

-Url: mà bạn muốn lấy dữ liệu.

-Data: dữ liệu được gửi đến máy chủ với các yêu cầu như là một chuỗi truy vấn.

-Callback: hàm được thực hiện khi có yêu cầu thành công

- Lưu ý : data, callback là 2 tham số tuỳ chọn

**Ví dụ:**

**Phương thức jquery post()**

- Phương thức post() gửi yêu cầu POST không đồng bộ đến máy chủ và truy xuất dữ liệu.

- Cú pháp:

$.post(Url, [data], [callback], [type] );

-Url: mà bạn muốn lấy dữ liệu.

-Data: dữ liệu được gửi đến máy chủ với các yêu cầu như là một chuỗi truy vấn

-Callback: hàm được thực hiện khi có yêu cầu thành công

-Type: kiểu dữ liệu của các nội dung trả lời.

-Lưu ý: data, callback, type là các tham số tuỳ chọn

**Ví dụ:**

1. **JSON**
2. **Giới thiệu**

JSON viết tắt của JavaScript Object Notation nó là một chuẩn để định dạng dữ liệu,về mặt này,có thể so sánh JSON với XML YAML…Nhưng khi JSON đi với JAvaScript hoặc ActionScript thì nó có tính ưu việt hơn.

Khi sử dụng Json với JavaScript không cần phải có các bước phân tích phức tạp như đối với XML mà có thể truy vấn trực tiếp giá trị theo tên(khoá) được định nghĩa trong JSON.

Tập tin JSON có phần mở rộng là .json

JSON Cũng như XML, được xây dựng để mô tả dữ liệu, không có khả năng trình bày, không có khả năng tự chạy. Ta chỉ có thể sử dụng JSON để trao đổi dữ liệu, chứ ta không thể chạy JSON.

**Đặc điểm:**

- JSON có đặc điểm là dễ dàng để đọc và viết

- Là một định dạng có dung lượng nhẹ

- JSON là ngôn ngữ độc lập

- JSON “tự mô tả” cấu trúc và dễ hiểu

1. **Làm việc với JSON**

- JSON được sử dụng trong khi viết JavaScript dựa trên các ứng dụng bao gồm các phần mở rộng trình duyệt và các trang web.

- Định dạng JSON được sử dụng cho việc tuần tự và truyền dữ liệu có cấu trúc qua kết nối mạng.

- JSON chủ yếu được sử dụng để truyền dữ liệu giữa máy chủ và các ứng dụng web.

- Dịch vụ Web và các API sử dụng định dạng JSON để cung cấp dữ liệu

- JSON có thể được sử dụng với các ngôn ngữ lập trình hiện đại.

1. Làm thế nào sử dụng JSON

**Cú pháp Json**

Json là một tập hợp con của cú pháp JavaScript:

- Dữ liệu là key | value

- Dữ liệu được phân cách bằng dấu phẩy

- Dấu ngoặc nhọn: các đối tượng

- Dấu ngoặc vuông: mảng

Dữ liệu là key: value

- Một cặp key/ value bao gồm một tên trường (trong ngoặc kép), tiếp theo là một dấu hai chấm, tiếp theo là một giá trị.

**Ví dụ:**

**Trong đó giá trị có thể là**

-Số (số nguyên hoặc số thực)

- Chuỗi (trong ngoặc kép)

- Logic (đúng hoặc sai)

- Một mảng (trong ngoặc vuông)

- Một đối tượng ( trong dấu ngoặc nhọn)

- Null

**Json Object**

- Đối tượng Json được viết bên trong dấu ngoặc nhọn

- Cũng giống như JavaScript, các đối tượng JSON có thể chứa nhiều key:value

**Ví dụ:**

**Json Arrays:**

- Mảng JSON được viết bên trong dấu ngoặc vuông.

- Cũng giống như JavaScript, một mảng JSON có thể chứa nhiều đối tượng

**Ví dụ:**

**JSON File:**

JSON cho phép tạo file lưu trữ:

**Ví dụ:**

5. **REFERENCE**

<https://techmaster.vn/posts/33487/lap-trinh-web-front-end-vs-back-end-vs-full-stack>

<http://www.w3schools.com>

<http://freetuts.net/hoc-javascript/javascript-can-ban>

<http://www.ibm.com/developerworks/vn/library/12/j-jws19/>

<http://cxf.apache.org/>

<http://wiki.netbeans.org/ApacheCXFonNetBeans>