XML DOM

GV Lương Hán Cơ
Khoa CNTT – ĐH KHTN

Nội dung

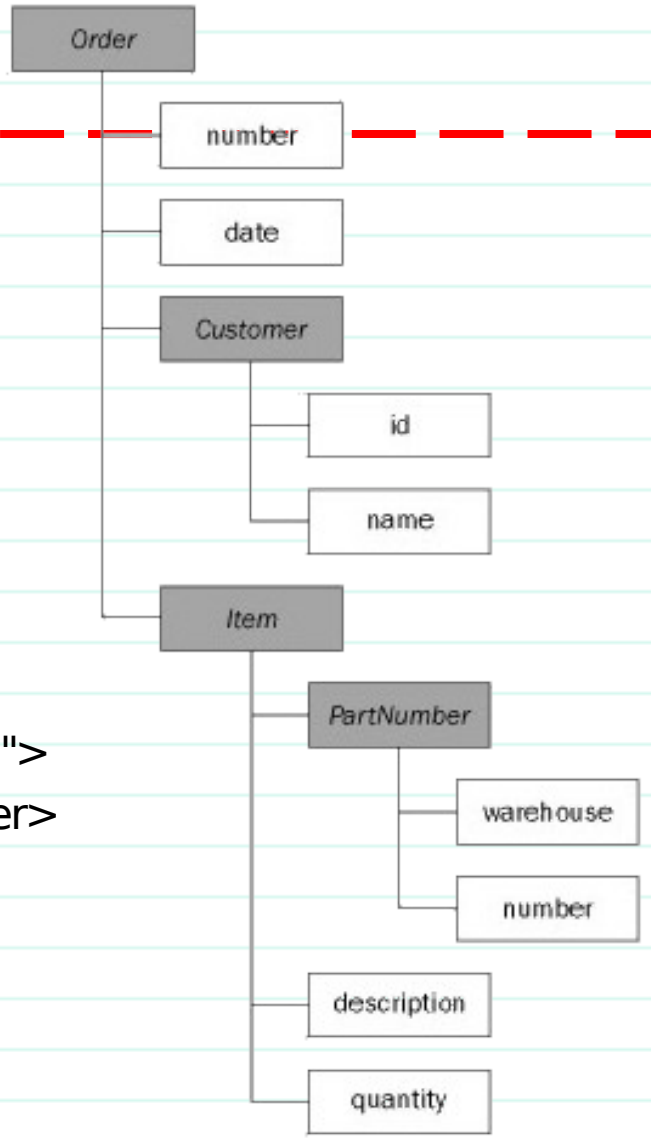
- Giới thiệu
- Lấy dữ liệu từ element
- Lấy dữ liệu từ attribute
- Thay đổi tài liệu XML

Giới thiệu

- Ứng với mỗi cấu trúc tài liệu XML phải có một cách thức để truy xuất ➔ khó phát triển ứng dụng
- Cần xây dựng một thư viện tổng quát sử dụng cho mọi tài liệu.
- DOM (Document Object Model) được sử dụng để thao tác lên tất cả các tài liệu XML.
- DOM có nhiều phiên bản: DOM Level 1, **DOM Level 2**, DOM Level 3 (draft).
- **http://www.w3.org/TR/DOM-Level-2/**

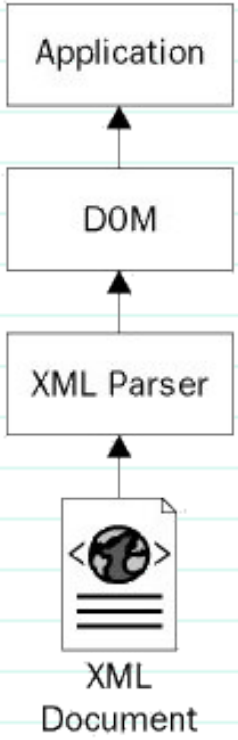
Ví dụ

```
<?xml version="1.0"?>
<order number="312597">
        <date>2000/1/1</date>
        <customer id="216A">Company A</customer>
        <item>
                <part-number warehouse="Warehouse 11">
                        E16-25A</part-number>
                <description>Production-Class
                        Widget</description>
                <quantity>16</quantity>
        </item>
</order>
```



Giới thiệu

- XML DOM đóng vai trò như lớp giao tiếp ở giữa của chương trình ứng dụng và XML parser.



Tạo đối tượng DOM

- Java script:

```
var oDOM;
oDOM = new ActiveXObject("MSXML.DOMDocument");
oDOM.async = false;
oDOM.load("domnode.xml");
```

Sử dụng DOM để lấy dữ liệu

- Toàn bộ tài liệu XML được chuyển đổi sang mô hình đối tượng.
- Element được chuyển thành node
- Element Content tương ứng một node
- Mỗi attribute tương ứng một node
- Tùy theo từng loại node sẽ có cách lấy dữ liệu tương ứng.
- Sau khi mở tài liệu XML, toàn bộ tài liệu được thành một node. Để truy xuất đến root Node có thể dùng thuộc tính documentElement hay là firstChild.

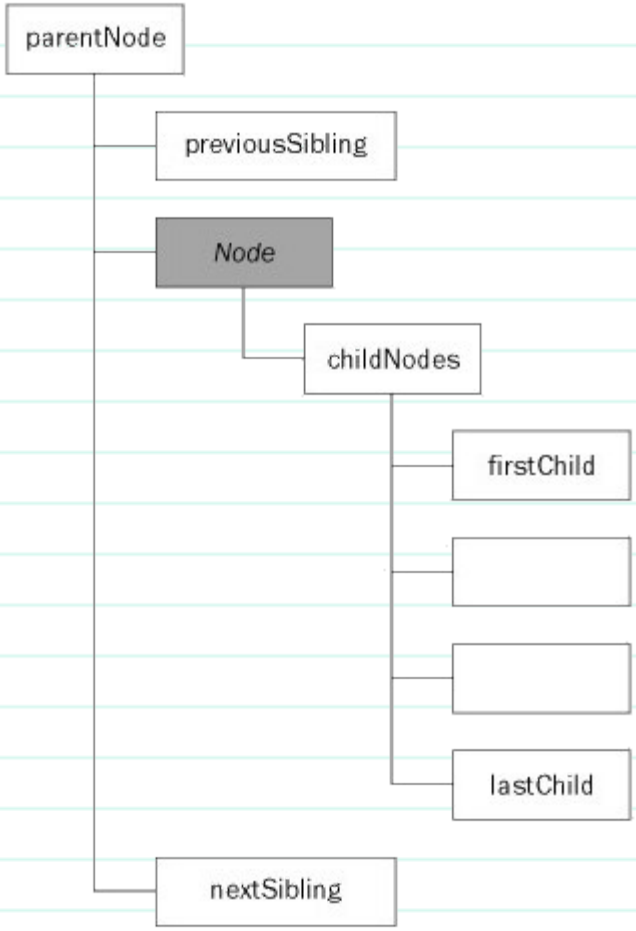
Sử dụng DOM để lấy dữ liệu

- Mỗi node đều có 3 thuộc tính: nodeType, nodeName, và nodeValue.
- Sử dụng thuộc tính nodeType để kiểm tra loại của node để xác định giá trị của 2 thuộc tính còn lại.

nodeType property constant	Giá trị	nodeName	nodeValue
ELEMENT_NODE	1	Tag name	NULL
ATTRIBUTE_NODE	2	Name of attribute	Value of attribute
TEXT_NODE	3	#text	Content of the text node
DOCUMENT_NODE	9	#document	NULL

Di chuyển giữa các node

- Sử dụng các thuộc tính: parentNode, firstChild, lastChild, previousSibling, và nextSibling để truy xuất Node xác định.
- childNodes trả về NodeList (danh sách các Node).
- Trường hợp Node hiện tại không có node con thì firstChild, lastChild trả về giá trị null. Còn childNodes sẽ trả về NodeList rỗng.
- Chỉ áp dụng đối với element, và element content.



Lấy dữ liệu từ attributes

- Sử dụng thuộc tính attributes
- Thuộc tính attributes chỉ có hiệu lực đối với node là loại element. Các trường hợp khác sẽ mang giá trị null.
- Thuộc tính attributes sẽ trả về đối tượng NamedNodeMap.
- Mỗi phần tử trong attributes là một đối tượng node với nodeType =ATTRIBUTE_NODE.

NodeList vs NamedNodeMap

	NodeList	NamedNodeMap
Nguồn:	childNodes getElementsByTagName(“..”) selectNodes(“…”)	attributes
Sử dụng:	đối với những node có tính thứ tự	không quan tâm đến thứ tự
Thuộc tính:	length	length
Phương thức:	item(…) chỉ số từ 0 đến length – 1	item(…) chỉ số từ 0 đến length – 1 getNamedItem(“…”)

Thay đổi tài liệu XML

- Tạo node
  - createNode: dùng để tạo mới tất cả các loại node
  - createElement: =createNode với loại ELEMENT_NODE
  - createTextNode: =createNode với loại TEXT_NODE
- Gắn node mới vào tài liệu XML
  - nodeCha.appendChild: thêm vào phần tử cuối cùng của danh sách childNodes của node cha.
  - nodeCha.insertBefore : thêm node mới vào trước phần tử nào đó trong childNodes của node cha.
- Tạo attribute
  - createAttribute: =createNode với loại ATTRIBUTE_NODE
- Thêm attribute vào tài liệu XML
  - node. setAttributeNode
  - node.attributes.setNamedItem
- Xóa node
  - node.removeChild