

LAB 3 : XML NAMESPACE

Sau khi thực hành xong bài này, học viên có khả năng:

- Hiểu được những nhược điểm của DTD, những ưu điểm của XML Schema so với DTD.
- Nắm được cấu trúc của một tài liệu XML Schema.
- Biết cách tạo file xml schema và dùng nó để validation một tài liệu XML.

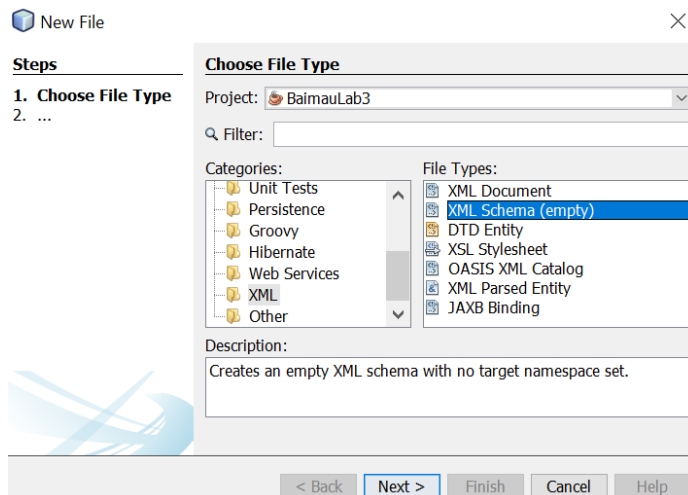
Bài mẫu : Minh họa sau là tập tin xml chứa thông tin về quản lý mượn sách tại một thư viện, phân biệt thẻ <tên>, thuộc tính id, thẻ <địa-chỉ> bằng các namespace.

File baimau-module-2.xml có nội dung sau :

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- thông tin về mượn sách, sử dụng namespace mặc định: http://www.quan-1.com
3 và 5 namespaces không mặc định
4 -->
5 <thu-vien xmlns="http://www.quan-1.com"
6 xmlns:q3="http://www.quan-3.com"
7 xmlns:q5="http://www.quan-5.com"
8 xmlns:q10="http://www.quan-10.com"
9 xmlns:sach="http://www.sach.com"
10 xmlns:nguoi="http://www.person.com">
11
12 <phieu-muon>
13 <doc-gia nguoi:id="12A98"> <!-- override default namespace cho thuộc tính -->
14 <nguoi:ten>Hoang</nguoi:ten>
15 <q3:dia-chi>123 Ba Hat</q3:dia-chi><!-- override namespace cho thẻ -->
16 <dien-thoai>(08)723104</dien-thoai>
17 </doc-gia>
18 <sach-muon sach:id="IT-123" sach:tinhh-trang="tot">
19 <sach:ten>"Object Oriented Programming"</sach:ten>
20 <nguoi:tac-gia>J. Neilson</nguoi:tac-gia>
21 </sach-muon>
22 </phieu-muon>
23 <!-- bạn tự thêm các phiếu mượn khác, sử dụng thêm các namespace khác -->
24 </thu-vien>
  
```

View trên browser



Sử dụng NetBean để tạo file xml schema

Bài tập 1: Global Express ở Mỹ là nhà cung cấp hàng đầu về dịch vụ vận chuyển nhanh bằng đường hàng không và bảo hiểm kho (hậu cần-logistics). Công ty này sở hữu một mạng lưới rộng khắp 20 địa điểm. Mỗi văn phòng của công ty duy trì một bộ lưu trữ các thư vận chuyển trên cơ sở từng ngày. Các thư này là các bản viết trên giấy (hard copy) nên có khả năng (prone) bị mất mát. **Để giải quyết vấn đề này, công ty quyết định lưu trữ thông tin vào máy tính để dữ liệu có thể truyền đi theo dạng XML.**

Hãy tạo tập tin Order.xml theo cấu trúc gồm các thẻ và các thuộc tính trong đề

Create an XML file called "Order.xml" using the following element names. Follow the specifications provided in the data given below.

- Invoice
- Order: (Should have an order number attribute)
 - Customer Name
 - Address
 - Order id
 - Total Price
- Cod:
 - Shipping Address
 - Shipping method (air or ground)
 - Shipping total
- Account: (State if the account is active or not. The default should be active.)
 - Customer ID
 - Invoice period
- Shipping info
 - Ship date
 - Billing Address (This element should be optional)
- Description
 - Description should be one of the product

Keeping into considerations that there are five regional offices and five different set of data is to be merged at the Head office, duplication of data is a huge possibility. For example, Customer Name in New York can match a customer name in a Chicago. To differentiate the data when it is merged, use namespaces to declare elements. Use entity declarations wherever necessary in the XML document.

Khi xem xét 5 văn phòng và 5 tập dữ liệu, dữ liệu trùng lặp là rất lớn, chẳng hạn như tên khách hàng ở New York có thể trùng tên khách hàng ở Chicago. Do vậy, các namespaces phải được dùng để khai báo các thẻ nhờ đó phân biệt các dữ liệu trùng lặp này. Hãy dùng các khai báo entity khi cần thiết trong tài liệu XML.

Nhận xét

- ☐ Chúng ta sẽ ấn định các namespace tùy ý để phân biệt dữ liệu.
- ☐ Namespace mặc định không có prefix.
- ☐ Namespace không mặc định phải có prefix.
- ☐ Thay thế (override) namespace mặc định cho 1 thẻ bằng cách chỉ định xmlns ngay trong thẻ đó.
- ☐ Trong bài tập này, chỉ cần kiểm tra well-formed, không kiểm tra validation.
- ☐ Namespace là các chuỗi ký tự nên theo dạng http://... và không được dùng kiểm tra validation.

Gợi ý

Chuẩn bị vài tên namespace (tham khảo bài mẫu)

Thẻ invoice gồm các thẻ con **order, cod, account, shipping-info, description**

Thẻ order gồm thuộc tính **order-no** và 4 thẻ con như đề bài.

Thẻ code gồm 3 thẻ con như đề bài

Thẻ account có thuộc tính **state** và 2 thẻ con như đề bài

Thẻ shipping-info gồm 2 thẻ con như đầu bài

Thẻ description chỉ chứa chữ.

Khi nhập dữ liệu vào file xml, **tùy ý ấn định namespace** cho thẻ hoặc thuộc tính theo các tên prefix của các namespace đã được chuẩn bị trước (tham khảo bài mẫu).

Bài tập 2 :

Soạn thảo XML, DTD, XML Schema bằng NetBeans.

Lưu ý, có nhiều cách viết XML Schema từ một phác thảo tài liệu XML cho trước.

- Liên kết tài liệu XML với DTD hoặc XML Schema dùng để validation nó. Kiểm tra "ngữ pháp" tài liệu XML đó.

Hướng dẫn thực hành XML AND JSON

- Trong NetBeans, từ XML Schema sinh tài liệu XML mẫu.
- Đọc hiểu XML Schema và DTD của tài liệu XML sau:

employee.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<humanres>
  <department name="tech">
    <employee id="E100">
      <name role="lead">Karl Marx</name>
      <mobile>0908493473</mobile>
    </employee>
    <employee id="E200">
      <name>Vladimir Putin</name>
      <mobile>0918995899</mobile>
    </employee>
  </department>
  <department name="sales">
    <employee id="E300">
      <name role="lead">Fidel Castro</name>
      <mobile>0903759412</mobile>
    </employee>
  </department>
</humanres>
```

Employee.dtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT humanres (department*)>
<!ELEMENT department (employee+)>
<!ATTLIST department name CDATA #REQUIRED>
<!ELEMENT employee (name,mobile*)>
<!ATTLIST employee id ID #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name role CDATA #IMPLIED>
<!ELEMENT mobile (#PCDATA)>
```

Employee.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="humanres">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="department" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="department">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="employee" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="employee">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
        <xs:element name="mobile" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="name">
    <xs:complexType mixed="true">
      <xs:attribute name="role" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```