HTTP -------🡪 request -> web server

* SOAP stands for **S**imple **O**bject **A**ccess **P**rotocol
* SOAP is an application communication protocol
* SOAP is a format for sending and receiving messages
* SOAP is platform independent
* SOAP is based on XML
* The best way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this.
* SOAP provides a way to communicate between applications running on different operating systems, with different technologies and programming languages.

## SOAP – Simple Object Access Protocol

* Một tiêu chuẩn của W3C,  là giao thức sử dụng XML để định nghĩa dữ liệu dạng thuần văn bản (plain text) thông qua HTTP. SOAP là cách mà  Web Service sử dụng để truyền tải dữ liệu. Vì dựa trên XML nên SOAP là một giao thức không phụ thuộc platform cũng như bất kì ngôn ngữ lập trình nào.
* Một thông điệp SOAP được chia thành hai phần là header và body. Phần header chỉ ra địa chỉ Web Service, host, Content-Type, Content-Length tương tự như một thông điệp HTTP.

SOAP use XML, HTTP Post

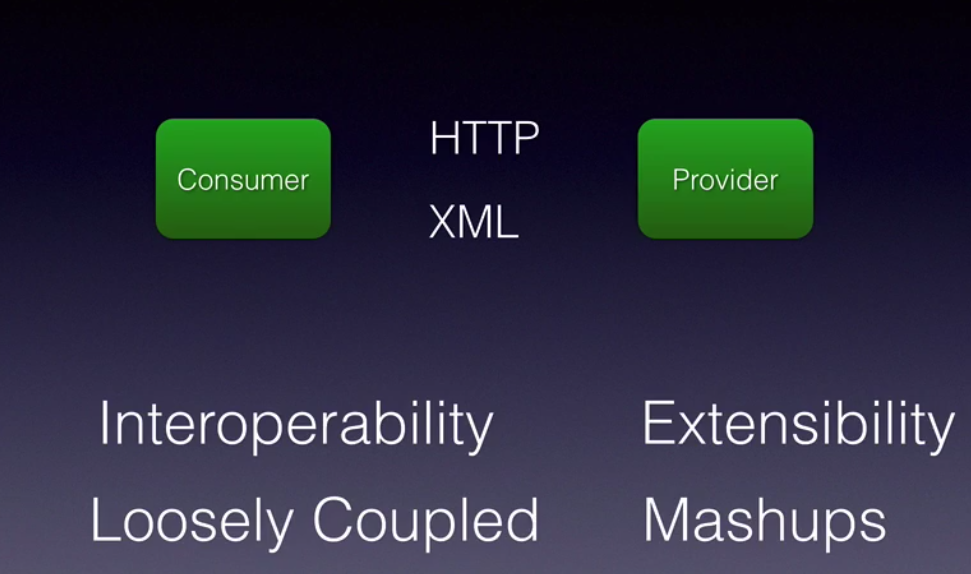
REST là viết tắt của cụm từ Representational State Transfer (đôi khi còn được viết là ReST) là một kiểu kiến trúc được sử dụng trong việc giao tiếp giữa các máy tính (máy tính cá nhân và máy chủ của trang web) trong việc quản lý các tài nguyên trên internet. REST được sử dụng rất nhiều trong việc phát triển các ứng dụng Web Services sử dụng giao thức HTTP trong giao tiếp thông qua mạng internet. Các ứng dụng sử dụng kiến trúc REST này thì sẽ được gọi là ứng dụng phát triển theo kiểu RESTful.

HTTP là một giao thức truyền dữ liệu. Nói đơn giản như là một là cách thức, quy định, quy ước để 2 bên có thể hiểu nhau, ở đây là client và server.  
Restful là một phong cách /chuẩn kiến trúc tổ chức và xây dựng web service.  
HTTP và Restful là hai thứ không thể so sánh với nhau, Restful thường được xây dựng dựa trên giao thức HTTP, nhưng không nhất thiết phải xây dựng trên HTTP.

Bài viết này chỉ ra rằng khi REST được biết đến nhiều hơn thì việc cụ thể hóa một Web service REST sẽ tuân thủ theo bốn nguyên tắc thiết kế cơ bản sau:

* Sử dụng các phương thức HTTP một cách rõ ràng
* Phi trạng thái
* Hiển thị cấu trúc thư mục như URls
* Chuyển đổi JavaScript Object Notation (JSON) và XML hoặc cả hai.





<complexType> and <simpleType> both define types. Complex types can have element descendants and attributes while simple types can't.

Complex types can have simple or complex content. Types with <complexContent> can contain child elements while those with <simpleContent> can only contain characters.

. XML Schema được dùng để mô tả cấu trúc và các kiểu dữ liệu của một tài liệu XML thay thế cho chuẩn DTD ([Document Type Definition](http://www.w3schools.com/dtd/dtd_intro.asp)) trước đây

**Một namespace là một thành phần của chương trình C# nó giúp cho việc quản lý các class, interface và structures. Namespace tránh tình trạng các tên trùng lặp giữ các class , interface,structures trong C#. Bạn có thể tạo nhiều namespace trong một namespace.**

elementFormDefault=*"qualified" 🡪 phải có namespaces prefix*

Website theo chuẩn [W3C](http://sieuthiwebsitedep.com/Kien-thuc-website/chuan-web-w3c-thiet-ke-website-theo-chuan-w3c-chuan-w3c-la-gi.html) sẽ tuân thủ 4 giai đoạn:  
    + Phác thảo (Working Draft)  
    + Chỉnh sửa cuối cùng (Last Call)  
    + Trình chuẩn ( Proposed Recommendation)  
    + Chuẩn đủ tư cách và ứng xử ( Candidate Recommendation)

 This gives you four style/use models:

1. RPC/encoded
2. RPC/literal
3. Document/encoded
4. Document/literal

* **Dependency Inversion**: Đây là một nguyên lý để thiết kế và viết code.
* **Inversion of Control**: Đây là một [design pattern](https://toidicodedao.com/2016/03/01/nhap-mon-design-pattern-phong-cach-kiem-hiep/) được tạo ra để code có thể tuân thủ nguyên lý Dependency Inversion. Có nhiều cách hiện thực pattern này: ServiceLocator, Event, [Delegate](https://toidicodedao.com/2015/02/10/series-c-hay-ho-callback-trong-c-delegate-action-predicate-func/), … Dependency Injection là một trong các cách đó.
* **Dependency Injection**: Đây là một cách để hiện thực Inversion of Control Pattern (Có thể coi nó là một design pattern riêng cũng được). Các **module phụ thuộc (dependency) sẽ được inject** vào module cấp cao.

### ****Các dạng DI****

Có 3 dạng Dependency Injection:

1. **Constructor Injection**: Các dependency sẽ được container **truyền vào (inject vào)** 1 class thông qua constructor của class đó. Đây là cách thông dụng nhất.
2. **Setter Injection**: Các dependency sẽ được truyền vào 1 class thông qua các hàm Setter.
3. **Interface Injection**: Class cần inject sẽ implement 1 interface. Interface này chứa 1 hàm tên Inject. Container sẽ injection dependency vào 1 class thông qua việc gọi hàm Inject của interface đó. Đây là cách rườm rà và ít được sử dụng nhất.

Java API for XML Web Services (JAX-WS) is a technology for building web services and clients that communicate using XML. JAX-WS allows developers to write message-oriented as well as Remote Procedure Call-oriented (RPC-oriented) web services.

In JAX-WS, a web service operation invocation is represented by an XML-based protocol, such as SOAP. The SOAP specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses. These calls and responses are transmitted as SOAP messages (XML files) over HTTP.

|  |  |
| --- | --- |
| **Cookie** | **Session** |
| Cookie được lưu trữ trên trình duyệt của người dùng. | Số phiên không được lưu trữ trong trình duyệt của người dùng. |
| Dữ liệu cookie được lưu trữ ở phía máy khách. | Dữ liệu session được lưu trữ ở phía máy chủ. |
| Dữ liệu cookie dễ dàng sửa đổi khi chúng được lưu trữ ở phía khách hàng. | Dữ liệu phiên không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. |
| Dữ liệu cookie có sẵn trong trình duyệt của chúng ta đến khi hết hạn. | Dữ liệu phiên có sẵn cho trình duyệt chạy. Sau khi đóng trình duyệt sẽ mất thông tin phiên. |

Chương trình Java sử dụng RAM máy tính để lưu các biến và đối tượng. Trong bộ nhớ Java chia làm 2 phân vùng chính là bộ nhớ HEAP và bộ nhớ STACK. HEAP lưu trữ các đối tượng, còn STACK lưu trữ các biến, hàm, đối số, tham chiếu.

List gồm arayList,Linketlis, vetor

LinkedList là một class implement interface List, vì vậy cách sử dụng cũng tương tự ArrayList, tuy vậy hiệu suất tốt hơn nếu thường xuyên thêm và xoá dữ liệu trong list.

Vector is almost identical to ArrayList, and the difference is that Vector is synchronized

Sử dụng interface Comparable để override hàm compareTo để so sánh các đối tượng trong danh sách List để có thể sắp xếp các phần tử theo thuộc tính mong muốn.

Sử dụng Comparator để so sánh các đối tượng trong danh sách List để sắp xếp các phần tử theo thứ tự tặng dần hoặc giảm dần theo thuộc tính mong muốn.

Sử dụng interface Iterator để lặp các phần tử trong danh sách List, Set, hoặc Queue để có thể in các phần tử ra màn hình. Tượng tự vòng lắp for each.. itterrator dung remove cac phan tu khi lap iterator.hashnext  
Set bao gồm hashset,linkedhashset,TreeSet

Set là tập hợp các phần tử không trùng nhau. HashSet là một class implement của interface Set được sử dụng phổ biến trong Java.

LinkedHashSet là một class implement interface Set nhưng các phần tử giữ nguyên vị trí theo thứ tự nhập vào.

TreeSet là một class implement interface Set đảm bảo các phần tử là duy nhất và được sắp xếp theo thứ tự tăng dần.  
ueue là tập hớp các phần tử giống như xếp hàng. Phần tử nào đến trước thì đứng trước, phần tử nào thêm sau thì bên dưới.

Map sử dụng cặp key/value để thể hiện giá trị. Key là duy nhất và trong HashMap có thể null. Map có thể sử dụng để lưu các giá trị với kiểu dữ liệu tham chiếu.

TreeMap là một class implement interface Map trong bộ Java Collection. Bạn có thể dung TreeMap để sắp xếp các key trong Map theo thứ tự tăng dần của tự nhiên.

HashMap is non synchronized and not thread safe(can use multi-threading).On the other hand, HashTable is thread safe and synchronized.  
Hashmap allows one null key and any number of null values, while Hashtable do not allow null keys and null values in the HashTable object

Serializable giúp chuyển các đối tượng thành kiểu String để lưu xuống file hoặc truyền qua mạng. Việc này đảm bảo các đối tượng được lưu chính xác và khi chuyển đổi ngược lại chuẩn.

List - a list of values, something like a "resizable array"  
Set - a container that does not allow duplicates  
Map - a collection of key/value pairs

|  |  |
| --- | --- |
|  | HashMap  * It has pair values(keys,values) * NO duplication key values * unordered unsorted * it allows one null key and more than one null values  HashTable  * same as hash map * it does not allows null keys and null values  LinkedHashMap  * It is ordered version of map implementation * Based on linked list and hashing data structures  TreeMap  * Ordered and sortered version * based on hashing data structures |

1. Key-values Stores  
Examples: Aerospike, CouchDB, Dynamo, FairCom c-treeACE, FoundationDB, HyperDex, MemcacheDB, MUMPS, Oracle NoSQL Database, OrientDB, Redis, Riak, Berkeley DB  
  
2. Column Family Stores  
Examples: Hadoop/HBase – Apache, BigTable – Google, Cassandra - Facebook/Apache, Hypertable - Zvents Inc/Baidu, Cloudera, SciDB, Mnesia, Tablets,…  
…  
  
3. Document Databases  
Examples: Apache Jackrabbit, CouchDB, IBM Lotus Notes Storage Format (NSF), MongoDB, Terrastore, ThruDB, OrientDB, RavenDB,...  
  
  
4. Graph Databases  
Examples: Neo4J, Sones, AllegroGraph, Core Data, DEX, FlockDB, InfoGrid, OpenLink Virtuoso,...