

*Trường Đại học Sư phạm thành phố Hồ Chí Minh *

**BÁO CÁO ĐỒ ÁN**

**Cơ sở dữ liệu nâng cao || Giáo viên : Lương Trần Hy Hiến**

**Nội dung đề tài : *CouchDB***

******

**Danh sách nhóm :**

***Nguyễn Hoàng Long – 42.01.104.238***

***Đỗ Minh Đức – 42.01.104.033***

# Mục lục

[CHƯƠNG 1 : TÌM HIỂU VỀ CƠ SỞ DỮ LIỆU NOSQL 1](#_Toc528688425)

[1.1 Khái niệm NOSQL 1](#_Toc528688426)

[1.2 Ưu nhược điểm của CSDL NOSQL 2](#_Toc528688427)

[1.3 So sánh NOSQL với cơ sở dữ liệu quan hệ 3](#_Toc528688428)

[1.4 Phân Loại NOSQL 4](#_Toc528688429)

[CHƯƠNG 2 : CouchDB 5](#_Toc528688430)

[2.1 Khái niệm CouchDB 5](#_Toc528688431)

[2.2 Tại sao nên chọn CouchDB ? 6](#_Toc528688432)

[2.3 Mô hình dữ liệu 6](#_Toc528688433)

[2.4 Các tính năng trong CouchDB 7](#_Toc528688434)

[2.4.1 Map / Reduce 7](#_Toc528688434)

[2.4.2 Lưu trữ dạng document 8](#_Toc528688434)

[2.4.3 Các thuộc tính trong ACID 8](#_Toc528688434)

[2.4.3.1 Khả năng nén (Compaction) 8](#_Toc528688434)

[2.4.3.2 View 8](#_Toc528688434)

[2.5 Replication (Nhân bản) 9](#_Toc528688435)

[2.6 CouchDB vs MongoDB 9](#_Toc528688436)

[2.7 Cài đặt CouchDB 10](#_Toc528688437)

[TÀI LIỆU THAM KHẢO 13](#_Toc528688438)

[cHÚ THÍCH 13](#_Toc528688438)

# CHƯƠNG 1 : TÌM HIỂU VỀ CƠ SỞ DỮ LIỆU NOSQL

* 1. **Khái niệm NOSQL**

NoSQL là một khái niệm chỉ về một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ. (RDBMS). RDBMS vốn tồn tại khá nhiều nhược điểm như có hiệu năng không tốt nếu kết nối dữ liệu nhiều bảng lại hay khi dữ liệu trong một bảng là rất lớn.

NoSQL là cơ sở dữ liệu không quan hệ, ràng buộc giữa các Collection (hay còn gọi là bảng trong cơ sở dữ liệu bình thường). Tức là giữa các Collection (Bảng) sẽ không có khóa chính, khóa ngoại như trong cơ sở dữ liệu bình thường. Việc này sẽ giảm bớt rối khi cơ sở dữ liệu chúng ta có nhiều bảng. NoSQL đặc biệt nhấn mạnh đến mô hình lưu trữ cặp giá trị - khóa và hệ thống lưu trữ phân tán.

NoSQL ra đời năm 1998 bởi Carlo Strozzi khi ông lập mới một hệ cơ sở dữ liệu quan hệ mã nguồn mở nhanh và nhẹ không liên quan đến SQL Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL khi Johan Oskarsson của Last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: phân tán (distributed) + không ràng buộc (non-relational).

**Đặc điểm**

* NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin – Mô hình phân tán dưới sự kiểm soát phần mềm
* Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau
* Một truy vấn sẽ được gửi tới nhiều máy cùng lúc, do đó khi một máy nào đó không phục vụ được sẽ không ảnh hưởng lắm đến chất lượng trả về kết quả
* Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu
* Tính nhất quán không theo thời gian thực: Sau mỗi thay đổi CSDL, không cần tác động ngay đến tất cả các CSDL liên quan mà được lan truyền theo thời gian.
  1. **Ưu nhược điểm của CSDL NOSQL**

**Ưu điểm**

1. Là mã nguồn mở: không phải mất chi phí và có xu hướng tin cậy, an ninh và nhanh hơn để triển khai so với các hệ quản trị cơ sở dữ liệu độc quyền.
2. Linh hoạt trong việc mở rộng và phát triển: Về phía công ty quản lý thì giúp dễ dàng mở rộng máy chủ khi dữ liệu càng ngày càng lớn hoặc lượng truy cập, tải dữ liệu quá lớn, thay vì thuê một máy chủ lớn hơn để thế máy chủ trước thì công ty chỉ cần thuê thêm một máy chủ khác. Về phía người lập trình thì giúp dễ dàng thêm Collection (Bảng) hoặc cặp dữ liệu (Cột). Dễ dàng trong việc thống kê, truy vấn nhanh.
3. Áp dụng được công nghệ điện toán đám mây: dễ dàng mở rộng phạm vi được theo yêu cầu có sử dụng một dịch vụ như là Amazon EC2. Giống như tất cả công nghệ đám mây, EC2 dựa vào ảo hóa. Liên kết yếu của ảo hóa là sự thực thi của I/O, với bộ nhớ và CPU các các kết nối mạnh. NoSQL lưu trữ dữ liệu thường được mở rộng phạm vi theo chiều ngang tận dụng được sự cung cấp mềm dẻo của đám mây. Giúp mở rộng dữ liệu dễ dàng hơn. iv. Được các hãng lớn sử dụng như Amazon, BBC, Facebook và Google. Làm nền tảng, cơ sở cho các công ty lớn áp dụng cũng như có sự tin cậy cao về hệ quản trị NoSQL.

**Nhược điểm**

1. Hỗ trợ không đồng đều cho các doanh nghiệp. Các doanh nghiệp vừa được sự hỗ trợ tốt nhất từ nhà cung cấp RMBMS (vì được phát triển trước NoSQL một thời gian khá dài) còn các doanh nghiệp nhỏ thì thường sử dụng các mã nguồn mở thì không được sự hỗ trợ tốt nhất.
2. Chưa được sử dụng rộng rãi vì NoSQL vẫn chưa nhận được sự tin cậy với nhiều doanh nghiệp, một phần dữ liệu đã được xây dựng từ lâu, nên việc chuyển đổi cũng là vấn đề khó với nhiều doanh nghiệp và chưa được hỗ trợ tốt về chức năng cũng như sự ổn định như RMBMS.
3. Còn mới lạ với một số lập trình viên. Chưa được sử dụng để đào tạo rộng rãi. Chưa có tool hỗ trợ giao diện tương tác cũng như các phương thức tốt nhất. Dẫn đến hạn chế về tri thức nghiệp vụ.
4. Chia sẻ dữ liệu chưa theo một tiêu chuẩn chung. Mỗi CSDL NoSQL có các giao diện lập trình ứng dụng API riêng của mình. Sự thiếu hụt các tiêu chuẩn có nghĩa là nó không có khả năng để chuyển một cách đơn giản từ một nhà cung cấp này sang một nhà cung cấp khác.
   1. **So sánh NOSQL với cơ sở dữ liệu quan hệ**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí đánh giá** | **RDBMS** | **NoSQL** |
| Đánh chỉ mục lượng lớn dữ liệu, phân trang hoặc phân phối luồng dữ liệu media | Có nhiều điểm yếu kém | Thực hiện khá tốt |
| Mô hình dữ liệu | Table | Key/Value, Column-Based-Tabular |
| Yêu cầu phần cứng cho dữ liệu lớn | Yêu cầu phần cứng máy chủ cao | Không nhất thiết phần cứng phải cao, có thể mở rộng máy chủ ở nhiều nơi với tài nguyên phần cứng thấp |
| Cách lấy dữ liệu từ client | Thông qua chuỗi kết nối | Thông qua API |
| Hiệu suất | Kém hơn NoSQL | Rất tốt  Bỏ qua SQL và các ràng buộc dữ liệu |
| Khả năng mở rộng | Hạn chế về lượng | Hỗ trợ lượng lớn các node |
| Phù hợp với công nghệ điện toán đám mây | Không | Có |
| Hiệu suất đọc ghi | Kém | Tốt |
| Thay đổi số node trong hệ thống | Phải shutdown cả hệ thống, thay đổi phức tạp | Không cần shutdown cả hệ thống, thay đổi đơn giản, không ảnh hưởng đển hệ thống |
| Hỗ trợ lập trình hướng đối tượng | Kém | Tốt |

* 1. **Phân Loại NOSQL**

**Key–value data stores**: Dữ liệu lưu dưới dạng cặp key – value. Giá trị được truy xuất

thông qua key.

* Ví dụ : Redis, Dynomite, Project Voldemort.
* Thường cho: Content caching Applications
* Ưu điểm: Tìm kiếm rất nhanh
* Nhược điểm: Lưu dữ liệu không theo khuôn dạng (schema) nhất định

**Column-based – Tabular**: Cơ sở dữ liệu tổ chức dưới dạng các bảng. Gần giống với mô hình RDBMS. Tuy nhiên, Chúng lưu dữ liệu bởi các cột chứ không phải bằng các dòng. Nó khá thích hợp với để hiển thị bằng các phần mềm quản lý kho dữ liệu

* Ví dụ : Apache Hbase, Apache Cassandra, Hypertable
* Thường cho: các hệ phân tán file
* Ưu điểm: Tìm kiếm nhanh, Phân tán dữ liệu tốt
* Nhược điểm: Hỗ trợ được với rất ít phần mềm

**Document-based**: Dữ liệu (bán cấu trúc hay semi-structured) được lưu trữ và tổ chức

dưới dạng một tập hợp các document. Các document này linh hoạt, mỗi document có một tập nhiều trường.

* Ví dụ : Apache CouchDB và MongoDB
* Thường cho: Web applications
* Ưu điểm: Dùng khi dữ liệu nguồn không được mô tả đầy đủ
* Nhược điểm: Hiệu năng truy vấn, Không có cú pháp chuẩn cho câu truy vấn dữ liệu

**Graph-based data-stores**: Những CSDL này áp dụng lý thuyết đồ thị trong khoa học máy tính để lưu trữ và truy xuất dữ liệu. Chúng tập trung vào tính rời rạc giữa các phần dữ liệu. Các phần tử đơn vị dữ liệu được biểu thị như một nút và liên kết với các thành phần khác bằng các cạnh.

* Ví dụ : Neo4j, InfiniteGraph, DEX
* Thường cho: Social networking, Hệ trợ giúp
* Ưu điểm: Ứng dụng các thuật toán trên đồ thị như Đường đi ngắn nhất, liên thông,…
* Nhược điểm: Phải duyệt nội bộ đồ thị, để trả lời lại các truy vấn. Không dễ để phân tán

**CHƯƠNG 2 : CouchDB**

## 2.1 Khái niệm và sơ lươc về CouchDB

CouchDB là 1 cơ sở dữ liệu dạng NoSQL mã nguồn mở, tập trung vào tính dễ sử dụng và có kiến ​​trúc có thể mở rộng, lưu trữ dữ liệu dạng document/JSON, JavaScript làm ngôn ngữ truy vấn để chuyển đổi Document, giao thức HTTP cho API để truy cập Document.



Thuật ngữ "Couch" là từ viết tắt của "Cluster Of Unreliable Commodity Hardware".

CouchDB được phát hành lần đầu tiên vào năm 2005 viết bằng C++. Tháng 4 năm 2008, trở thành dự án của Apache Software Foundation\*, chuyển sang nền tảng Erlang/OTP\* để nhấn mạnh đến khả năng chịu lỗi.

Không giống như một cơ sở dữ liệu quan hệ, một cơ sở dữ liệu CouchDB không lưu trữ dữ liệu và các mối quan hệ trong các bảng. Thay vào đó, mỗi database là một tập hợp các document độc lập. Mỗi document duy trì dữ liệu riêng của mình và lược đồ khép kín. Một ứng dụng có thể truy cập nhiều database, chẳng hạn như một database được lưu trữ trên điện thoại di động của người dùng và một database khác trên một máy chủ. Một document siêu dữ liệu chứa thông tin sửa đổi, làm cho nó có thể hợp nhất bất kỳ sự khác biệt nào có thể xảy ra trong khi cơ sở dữ liệu bị ngắt kết nối.

CouchDB thực hiện một dạng  Multiversion concurrency control (MCC or MVCC) – kiểm soát đồng thời đa nhiệm – để nó không khóa tệp cơ sở dữ liệu trong khi viết, giải quyết xung đột đến việc hợp nhất dữ liệu khi có nhiều user truy cập thay đổi thông tin.

Các công ty sử dụng CouchDB : Talend SA, Akamai Technologies, Hothead Games, GenCorp Technologies, Vivint Solar,…

## 2.2 Tại sao nên chọn CouchDB

* CouchDB có API dạng RESTFul\* giúp cho việc giao tiếp với cơ sở dữ liệu được đơn giản.
* Các RESTFul API rất trực quan và dễ thao tác.
* Dữ liệu được lưu dưới cấu trúc document rất mềm dẻo, chúng ta không cần phải lo lắng về cấu trúc dữ liệu.
* Map/reduce giúp việc lọc, tìm, tổng hợp dữ liệu dễ hơn bao giờ hết.
* Nhân bản / đồng bộ là sức mạnh đặc biệt của CouchDB mà hiếm database nào có.
* Miễn phí

## 2.3 Mô hình dữ liệu

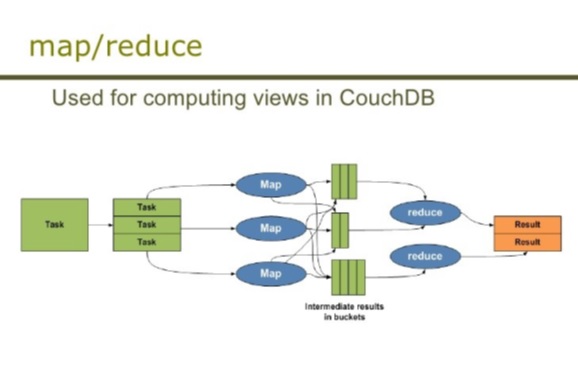


* Nó là một kiến trúc 2 tầng
* CouchDB sử dụng HTTP làm giao diện lập trình chính và JSON để lưu trữ dữ liệu.
* Vì chúng ta có JavaScript ở phía Client, hiệu suất sẽ tăng lên do khả năng tương thích của JSON với JavaScript.
* CouchDB tương thích với các nền tảng như Windows, Linux, Mac, iOS, Android.
* CouchDB là phù hợp cho ứng dụng Client, các ứng dụng web.
* Mỗi database là 1 danh sách các document độc lập.
* Document bao gồm dữ liệu người dùng thao tác lẫn thông tin về phiên bản của dữ liệu để tiện việc merge dữ liệu.
* CouchDB sử dụng cơ chế phiên bản hoá dữ liệu để tránh tình trạng khoá dữ liệu khi đang ghi.

**2.4 Các tính năng trong CouchDB**

### 2.4.1 Map/Reduce

* Map/Reduce bao gồm:
* **Map**(): thủ tục (method) thực hiện lọc và sắp xếp (chẳng hạn như sắp xếp học sinh theo tên)
* **Reduce**()**:** mothod thực hiện thao tác tóm tắt (chẳng hạn như đếm số lượng sinh viên, tần suất tên).
* Các hàm này cung cấp sự linh hoạt tuyệt vời vì chúng có thể thích ứng với các biến trong cấu trúc document và các chỉ mục cho mỗi document có thể được tính toán độc lập và song song. Sự kết hợp giữa *map* và *reduce* được gọi là *view* trong thuật ngữ CouchDB.



### 2.4.2 Lưu trữ dạng document

CouchDB là một NoSQL database dạng document. Document là một đơn vị dữ liệu (giống như 1 object của Javascript), mỗi field có một tên riêng không trùng nhau, chứa các loại dữ liệu như chữ, số, Boolean, danh sách… Không có bất kì giới hạn nào về dung lượng text hay số field trong 1 doucment.

CouchDB cung cấp 1 RESTFul API cho việc đọc và ghi (thêm, sửa, xoá) document.

Sau đây là 1 ví dụ về 1 document :

{

"title": "Macbook",

"price": 1500,

"SKU": "abcd1234"

}

### 2.4.3 Các thuộc tính ACID

Khi dữ liệu được ghi xuống ổ cứng thì nó sẽ không bị ghi đè. Bất kì thay đổi nào (thêm, sửa, xoá) đều theo chuẩn Atomic, có nghĩa là dữ liệu sẽ được lưu lại toàn diện hoặc không được lưu lại. Database không bao giờ thêm hay sửa một phần dữ liệu.

Hầu hết các cập nhật đều được serialized để đảm bảo tất cả người dùng có thể đọc document mà không bị chờ đợi hoặc gián đoạn.

#### 2.4.3.1 Khả năng nén (compaction)

Nén là 1 hành động giúp giải phóng dung lượng ổ cứng được sử dụng bằng cách xoá đi các dữ liệu không còn được sử dụng. Khi tiến hành nén dữ liệu ở 1 file thì 1 file mới với định dạng **.compaction** sẽ được tạo ra và dữ liệu sẽ được sao chép vào file mới này. Khi quá trình copy hoàn thành thì file cũ sẽ được xoá bỏ. Database vẫn online trong quá trình nén và các thao tác thay đổi / đọc dữ liệu vẫn diễn ra bình thường.

#### 2.4.3.2 View

Dữ liệu trong CouchDB được lưu trữ trong các document. Bạn có thể tưởng tượng như 1 database là 1 table và 1 document là 1 row. Khi. Khi chúng ta muốn trình bày dữ liệu bằng nhiều góc nhìn khác nhau thì chúng ta cần 1 phương pháp để filter, tổ chức để hiển thị kết quả cuối cùng.

Để giải quyết vấn đề này, CouchDB sử dụng mô hình View. View là 1 phương pháp tổng hợp dữ liệu trong các document ở 1 database. Các View được build động và không ảnh hưởng đến dữ liệu đã ghi của các document nên chúng ta có thể có bao nhiêu View tuỳ ý tuỳ vào nhu cầu trình bày dữ liệu.

## 2.5 Replication (Nhân bản)

Replication là một quá trình một chiều liên quan đến hai cơ sở dữ liệu (một nguồn và một đích).

Mục đích của bản sao là ở cuối quá trình, tất cả các tài liệu đang hoạt động trên cơ sở dữ liệu nguồn cũng nằm trong cơ sở dữ liệu đích và tất cả các tài liệu đã bị xóa trong cơ sở dữ liệu nguồn cũng bị xóa (nếu tồn tại) trên cơ sở dữ liệu đích.

Quá trình sao chép chỉ sao chép bản sửa đổi cuối cùng của một tài liệu, vì vậy tất cả các bản sửa đổi trước đó chỉ có trên cơ sở dữ liệu nguồn không được sao chép vào cơ sở dữ liệu đích

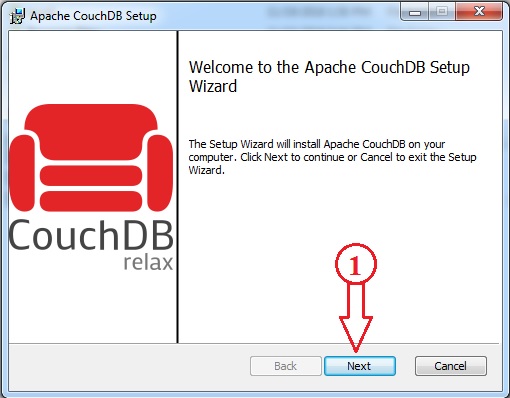


## 2.6 CouchDB vs MongoDB

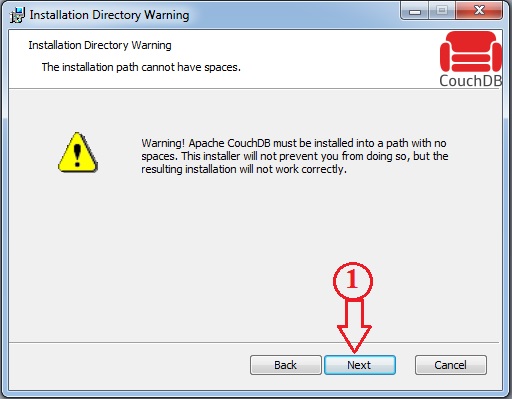


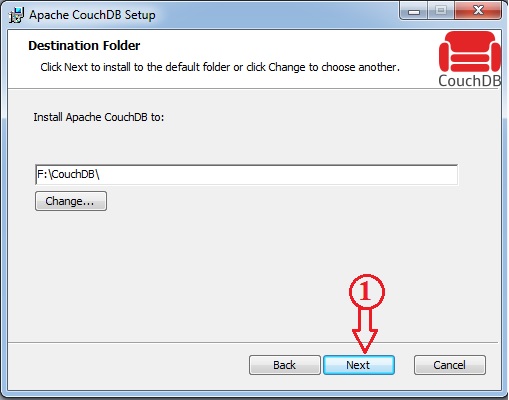
## 2.7 Cài đặt CouchDB

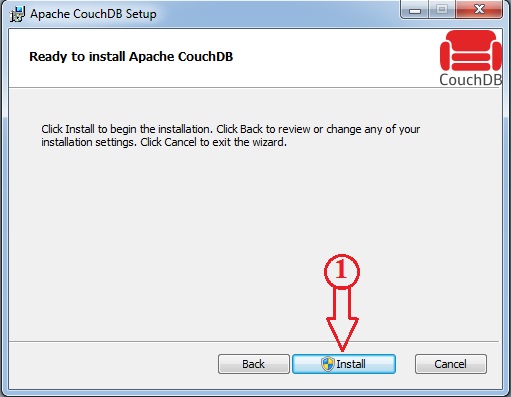
Vào trang <http://couchdb.apache.org/> để download về và Install theo hướng dẫn hình bên dưới :

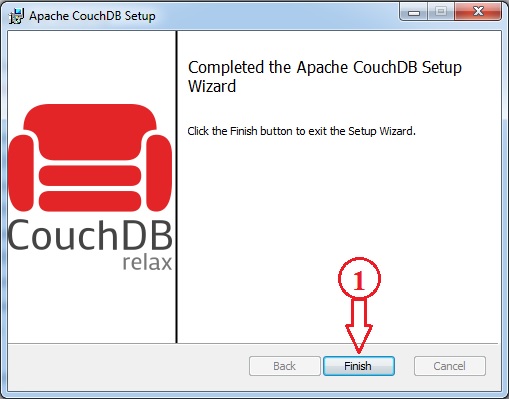












# TÀI LIỆU THAM KHẢO

<https://tbson.info/db-couchdb-cac-khai-niem-co-ban/>

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

<https://www.tutorialspoint.com/couchdb/index.htm>

# CHÚ THÍCH

Erlang là một ngôn ngữ lập trình được sử dụng để xây dựng các hệ thống thời gian thực mềm có thể mở rộng quy mô lớn với các yêu cầu về tính khả dụng cao. Một số sử dụng của nó là trong viễn thông, ngân hàng, thương mại điện tử, điện thoại máy tính và tin nhắn tức thời. Hệ thống thời gian chạy của Erlang đã tích hợp sẵn hỗ trợ cho sự đồng thời, phân phối và khả năng chịu lỗi.

OTP là tập hợp các thư viện Erlang và các nguyên tắc thiết kế cung cấp phần giữa để phát triển các hệ thống này. Nó bao gồm cơ sở dữ liệu phân tán riêng của mình, các ứng dụng để giao diện với các ngôn ngữ khác, gỡ lỗi và phát hành các công cụ xử lý.

Apache Software Foundation ( ASF ) là một tập đoàn Mỹ phi lợi nhuận để hỗ trợ các dự án phần mềm Apache, bao gồm các [máy chủ Apache HTTP](https://en.wikipedia.org/wiki/Apache_HTTP_Server) . ASF được thành lập vào ngày 25 tháng 3 năm 1999.

REST viết tắt cho REpresentational State Transfer Khái niệm về REST lần đầu tiên được giới thiệu vào năm 2000 trong luận văn Tiến sĩ của Roy Fielding (đồng sáng lập giao thức HTTP) Là 1 kiểu kiến trúc lập trình, định nghĩa các quy tắc để thiết kế web service chú trọng vào tài nguyên Mọi thứ trong REST đều được coi là tài nguyên và được định danh thông qua URI, và có thể được biểu diễn thông qua dạng văn bản, XML, JSON .... RESTful là những ứng dụng mà có sử dụng kiến trúc REST.