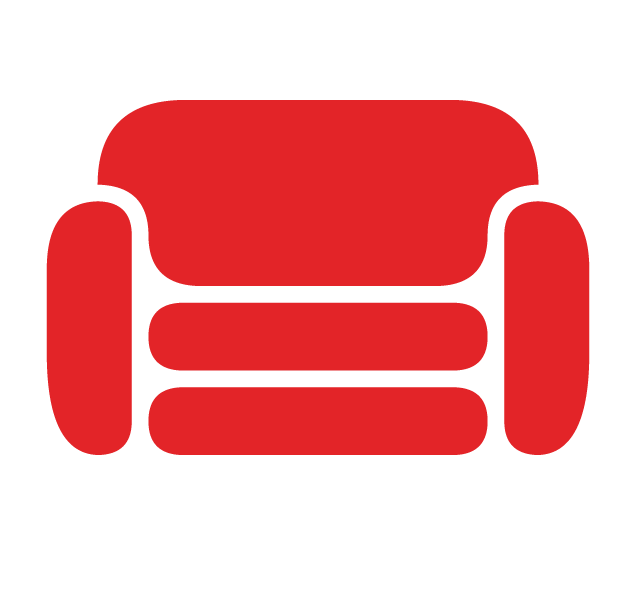
**TRƯỜNG ĐẠI HỌC SƯ PHẠM TPHCM**

**Khoa Công nghệ thông tin**

**Môn:** Cơ sở dữ liệu nâng cao

**Đề tài:**

** TÌM HIỂU COUCHDB**



**Giáo viên hướng dẫn**: Th.s Lương Trần Hy Hiến

**Sinh viên thực hiện:** Nhóm 20

Phan Tấn Tài - 42.01.104.140

Trần Phương Nguyên - 42.01.104.095

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn thầy Lương Trần Hy Hiến đã tận tình giảng dạy, trang bị, cung cấp cho chúng em những kiến thức nền tảng, chuyên môn cần thiết giúp chúng em hiểu sâu hơn về bộ môn Cơ sở dữ liệu nâng cao.

Mặc dù đã cố gắng hoàn thành tốt công việc được phân công trong phạm vi và khả năng cho phép nhưng chắc chắn không tránh khỏi những thiếu sót. Rất mong nhận được sự góp ý của thầy để đề tài được hoàn thiện hơn, đồng thời bổ sung vốn kinh nghiệm cho chúng em trên con đường sắp tới.

Xin chân thành cảm ơn!

Sinh viên thực hiện

Nhóm 20

**MỤC LỤC**

**Nội dung**

**Trang**

1. TỔNG QUAN VỀ COUCHDB 3

[1.1 Khái niệm 4](#_Toc500100620)

[1.2 Một số ví dụng về cơ sở dữ liệu 5](#_Toc500100621)

[**2. TÌM HIỂU VỀ COUCHDB** 9](#_Toc500100624)

[2.1. Khái niệm 9](#_Toc500100625)

[2.2. Các cài đặt đòi hỏi 9](#_Toc500100629)

[2.3 Cài đặt 9](#_Toc500100633)

[2.4. Cách sử dụng 14](#_Toc500100638)

[**3. ỨNG DỤNG** 21](#_Toc500100642)

[TÀI LIỆU THAM KHẢO 32](#_Toc500100668)

# **TỔNG QUAN NOSQL**

## **Khái niệm**

* NoSQL là gì?

NoSQL không Chỉ là SQL, ngụ ý rằng khi thiết kế một giải pháp phần mềm hoặc sản phẩm, có nhiều hơn một cơ chế lưu trữ có thể được sử dụng dựa trên các nhu cầu. NoSQL không có một định nghĩa rõ ràng mà chúng ta có thể hiểu là mô hình cơ sở dữ liệu mới có các đặc điểm như sau:

* Không sử dụng các mô hình quan hệ
* Chạy tốt trên các server
* Chủ yếu là mã nguồn mở
* Tối thiểu hóa lược đồ

Tại sao lại là NoSQL?

Chúng ta sẽ gặp vấn đề khó khăn khi có sự không tương ứng giữa cấu trúc dữ liệu quan hệ và cấu trúc dữ liệu được lưu trong bộ nhớ hệ thống. Tuy nhiên, sử dụng NoSQL cho phép chúng ta tiếp tục phát triển mà không cần chuyển đổi cấu trúc lưu trong bộ nhớ với cấu trúc dữ liệu quan hệ.

Sự gia tăng của các trang web cũng tạo ra một sự thay đổi quan trọng trong việc lưu trữ dữ liệu như sự cần thiết để hỗ trợ khối lượng dữ liệu lớn bằng cách chạy trên các cụm. Tuy nhiên, cơ sở dữ liệu quan hệ không được thiết kế để hoạt động hiệu quả trên các cụm. Ví dụ, các nhu cầu lưu trữ dữ liệu của một ứng dụng ERP là rất nhiều khác biệt so với nhu cầu lưu trữ dữ liệu của một Facebook hoặc một Etsy.

* Một số mô hình trong NoSQL:
* Mô hình dữ liệu tổng hợp
* Mô hình phân tán

## **Một số ví dụ về cơ sở dữ liệu**

* Key-value data stores: Dữ liệu lưu dưới dạng cặp key-value. Giá trị được xuất ra thông qua key.
* VD: Redis, Dynomite, Project Voldemort.
* Thường cho: Content caching Applications
* Ưu điểm: Tìm kiếm rất nhanh
* Nhược điểm: Lưu trữ dữ liệu không theo khuôn dạng nhất định.
* Column-based-Tabular: Cơ sở dữ liệu tổ chức dưới dạng các bảng. Gần giống mô hình RDBMS. Tuy nhiên, chúng lưu dữ liệu bởi các cột chứ không phải bằng các dòng. Nó khá thích hợp để hiển thị bằng các phần mền quản lý kho dữ liệu.
* VD: 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 ít phần mềm
* Document-based: Dữ liệu (bán cấu trúc hay semi) 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.
* VD: 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 xuất, không có cứu pháp chuẩn cho câu truy vấn dữ liệu.
* Grap-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.
* VD: Neo4j, InfiniteGraph, DEX
* Thường cho: Social networking, Hệ trợ giúp
* Ưu điểm: Ứng dụng có thuật toán trên đồ thị như đường đi ngắn nhất, không dễ để phân tán.

# **COUCHDB**

## **2.1. Khái niệm**

**CouchDB** là một cơ sở dữ liệu dạng NoSQL mã nguồn mở database lưu trữ dữ liệu dạng document/JSON.

**CouchDB** được thiết kế nhắm tới tính dễ sử dụng và khắc phục cho môi trường web.

Tại sao cần sử dụng **COUCHDB**?

CouchDB có API dạng RESTFul giúp cho việc giao tiếp với cở 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.

* Mô hình dữ liệu:
* Database là cấu trúc dữ liệu lớn nhất của CouchDB.
* 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 hóa dữ liệu để tránh tình trạng khóa dữ liệu khi đang chạy.
* Lưu trữ dạng document:
* CouchDB là một NOSQL database dạng document. Document là một đơn vị dữ liệu (giống như một object của Javascript), mỗi field có một tên riêng không trùng nhau, chứa các loại dữ liệu như chữ, số, boolear, danh sách,... Không có bất kì giới hạn nào về dung lượng text hay số field trong 1 document.
* CouchDB cung cấp 1 RESTFul API cho việc đọc và ghi (thêm, xóa, sửa) document.
* Ví dụ về 1 document:

{

"title": "Macbook",

"price": 1500,

"SKU": "abcd1234"

}

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, xóa, sửa) 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.
* 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.
* Views:
* Dữ liệu trong CouchDB được lưu trữ trong các document. Bạn có thể tưởng thượng như 1 database là 1 table và 1 document là 1 row. Khi chúng ta muốn trình bày dữ liệu bằng nhiều góc nhìn thì chúng ta cần một 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 một 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 tùy ý vào nhu cầu trình bày dữ liệu.

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Khi CouchDB tạo ra 1 khung nhìn, nó lưu kết quả của khung nhìn trên đĩa trong cấu trúc dữ liệu B\_Tree, rất hiệu quả để truy cập. | Không có độc quyền đọc từng tài liệu |
| CouchDB sẽ chỉ cần cập nhật chế độ xem cho các tài liệu đã thay đổi / được thêm vào. Nó sẽ từng bước cập nhật chỉ mục của khung nhìn, vì vậy nó không phải bắt đầu từ đầu mỗi lần.  ->lượt xem đặc biệt lý tưởng cho các tập tin lớn. | Không khôi phục mật khẩu |
| Sử dụng API chế độ xem, bạn có thể chỉ định phạm vi khóa khi thực hiện truy vấn của mình, chỉ tìm nạp dữ liệu bạn muốn. | Không di chuyển cơ sở dữ liệu |
| Lượt xem rất mạnh mẽ và cung cấp một số lượng lớn tính linh hoạt. |  |

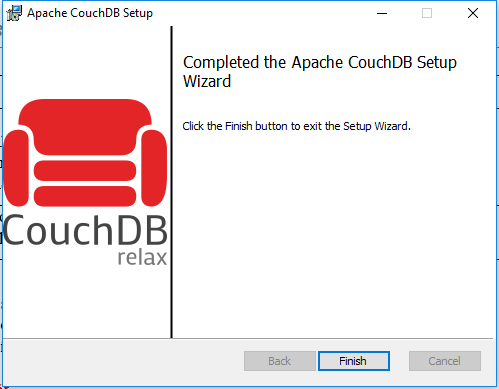
**2.2. Các cài đặt đòi hỏi – Yêu cầu hệ thống**

Cài đặt CouchDB**:** [**http://couchdb.apache.org/**](http://couchdb.apache.org/)

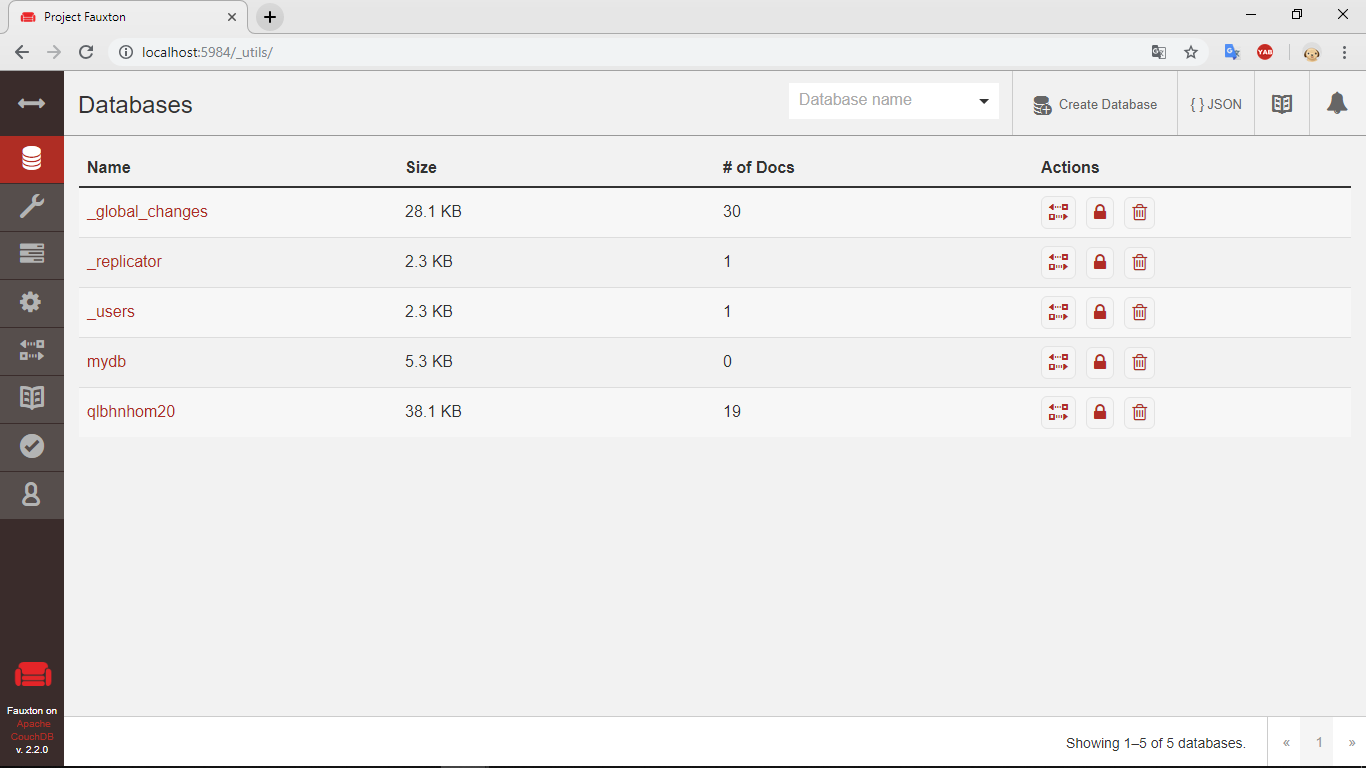
Hỗ trợ Windows 64bits, MacOS (10.10+), Debian, Ubuntu, RHEL, CentOS.

**2.3 Cài đặt**

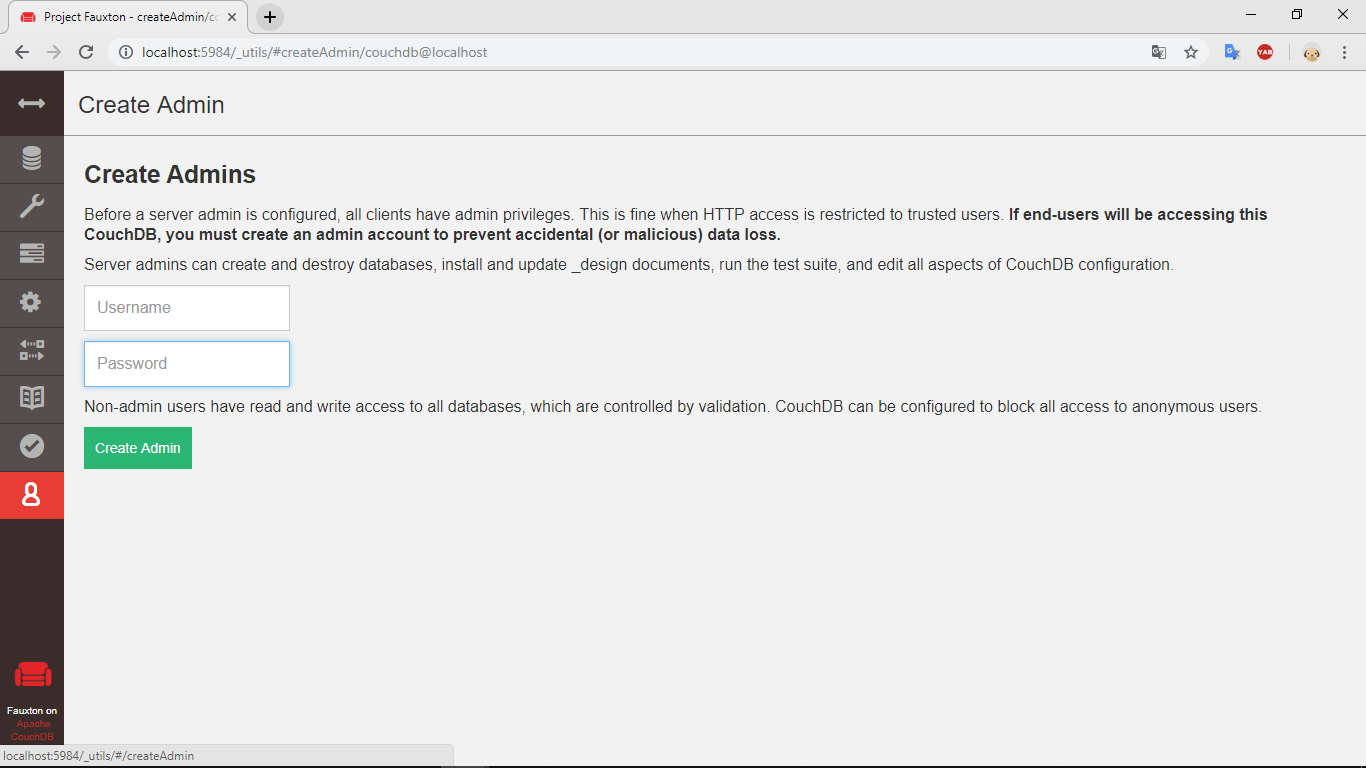
* Cài CouchDB.



* Sau khi đã cài xong. Vào [http://localhost:5984/\_utils/](%20http:/localhost:5984/_utils/) xem giao diện của CouchDB.

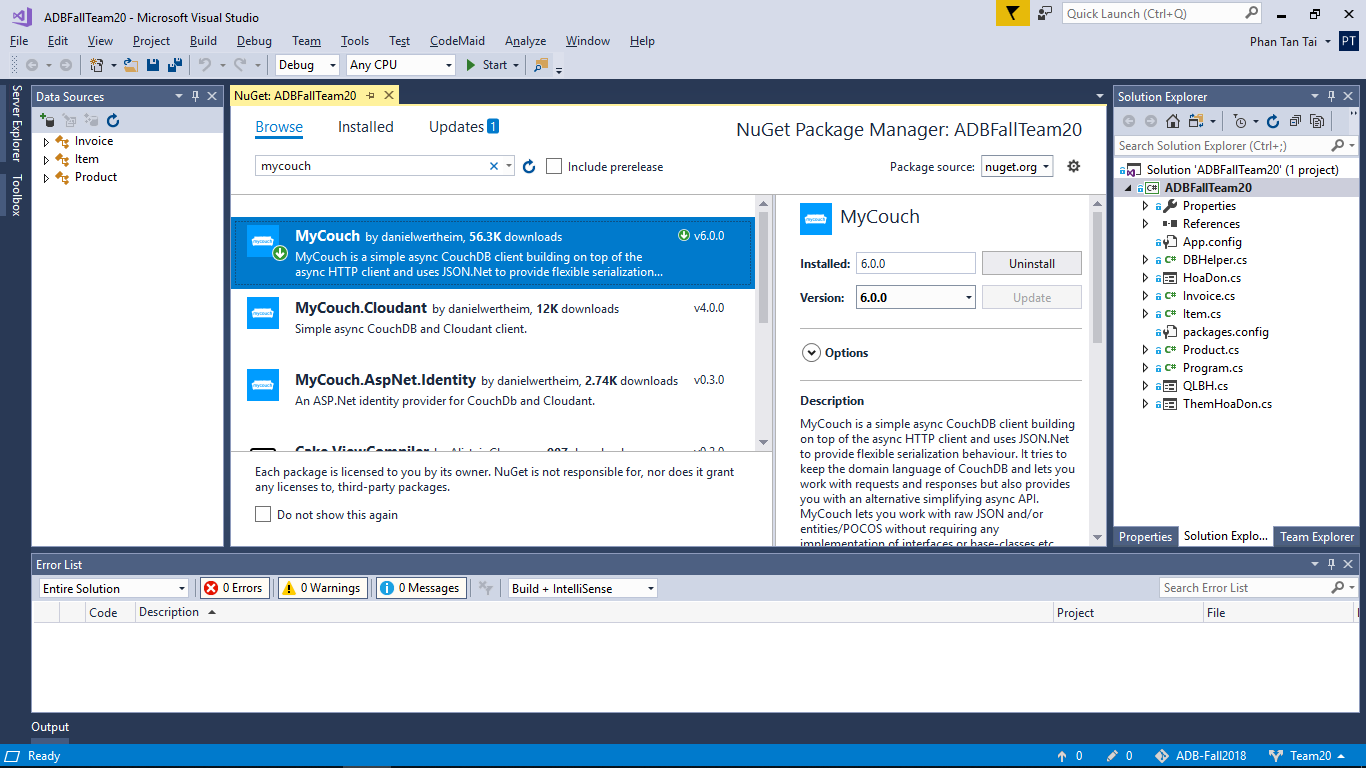


* Tạo tài khoản Admin để đăng nhập cho CouchDB.



* Sử dụng phương thức http để giao tiếp với CouchDB.
* Hướng dẫn sử dụng C# kết nối với CouchDB:

Thêm NutGet MyCouch của Daniel Wertheim



Kết nối từ client:



Thay **admin:admin** thành tài khoản đã tạo. dbname là tên database.

Thêm document mới:

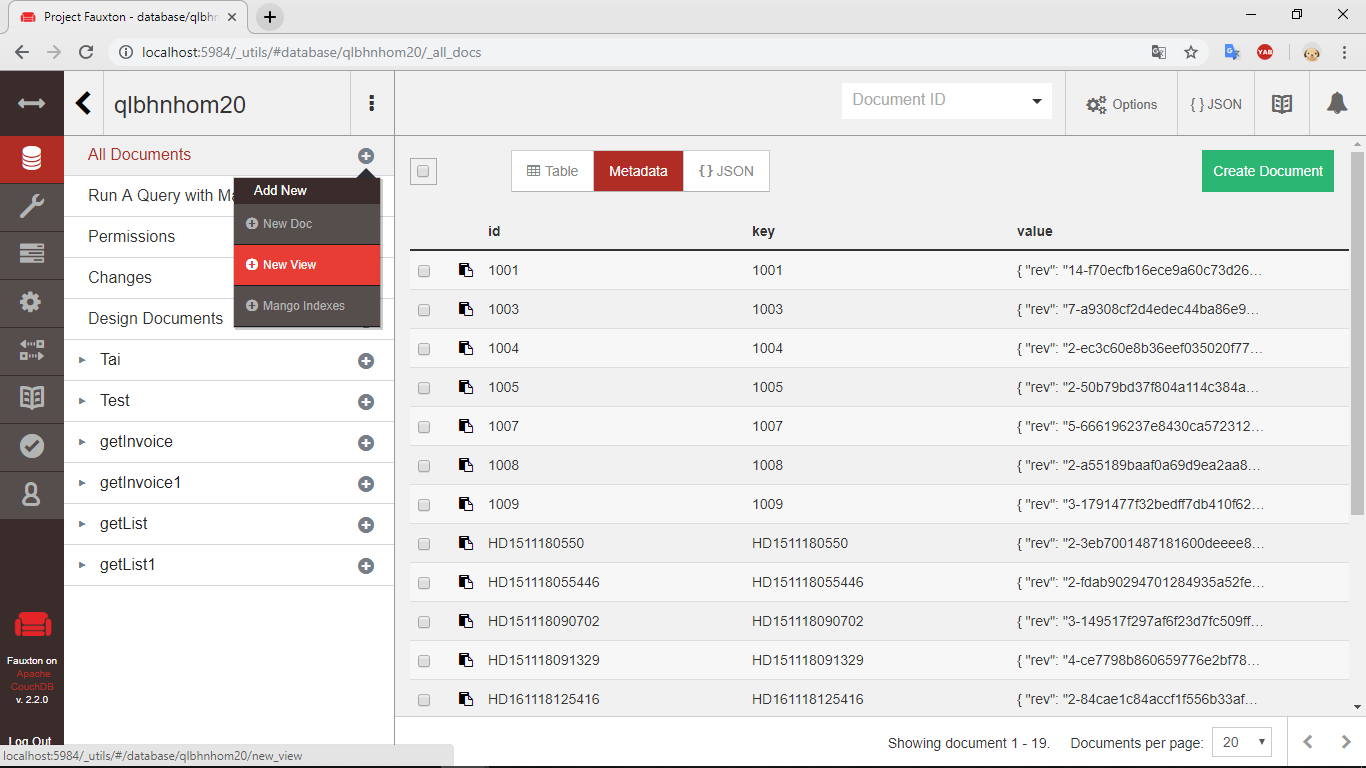
* Client.Documents.PostAsync(json); -Thêm 1 document kiểu json.
* Client.Entities.PostAsync(myEntity); - thêm 1 document là một thực thể.
* Nếu document có \_id và \_rev đã tồn tại thì document sẽ được cập nhật lại.

Xoá document.

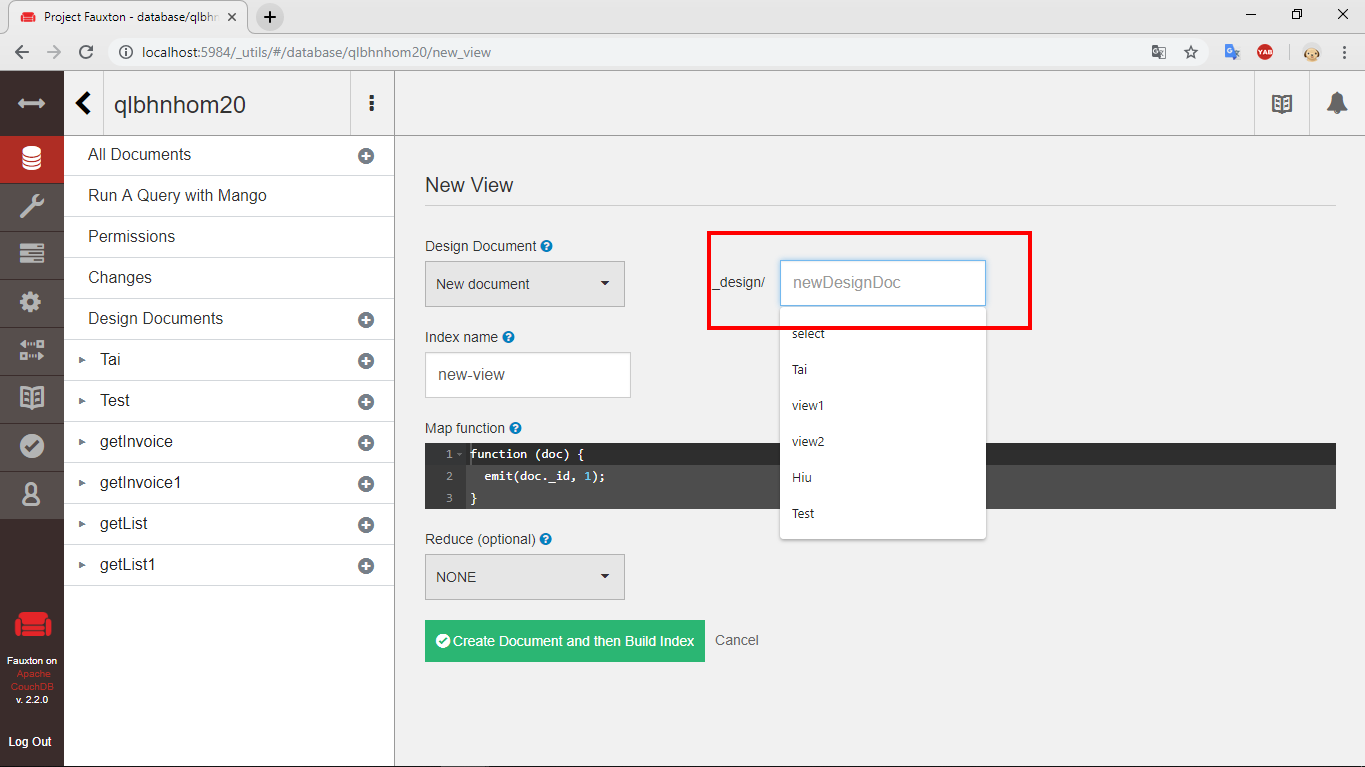
* client.Entities.DeleteAsync(myEntity);
* client.Documents.DeleteAsync(\_id,\_rev) : truyền vào \_id và \_rev của document cần xoá.

Tạo view.

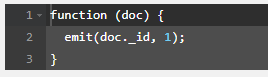
* Vào database, chọn New View



* Đặt tên View



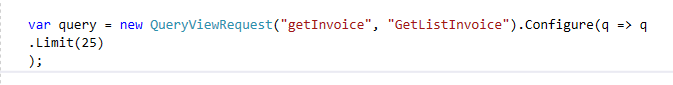
* Dùng function để truy vấn.



Hàm này sẽ duyệt toàn bộ document trong database.

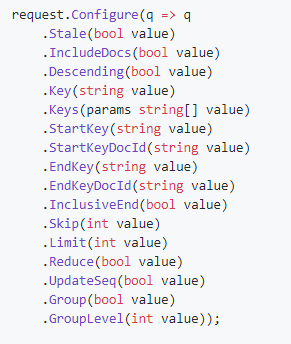
emit: sẽ trả các document thoả điều kiện ta xét. emit(key, value);

* Gọi View.



QueryViewRequest(Tên View, Tên Hàm).

Một số Configure:



* Key(value) : trả về danh sách có key = value.
* StartKey(value) : trả về danh sách có key lớn hơn value.
* EndKey(value) : trả về danh sách có key nhỏ hơn value.
* Skip(value) : trả về danh sách bắt đầu từ dòng value.
* Limit(value) : trả về danh sách bị giới hạn value dòng.
* Descending : sắp xếp giảm theo key.

# **TÀI LIỆU THAM KHẢO**

<https://github.com/danielwertheim/mycouch/wiki/documentation>

<http://docs.couchdb.org/en/stable/>