

Apache CouchDB Tutorial

Apache CouchDB Tutorial

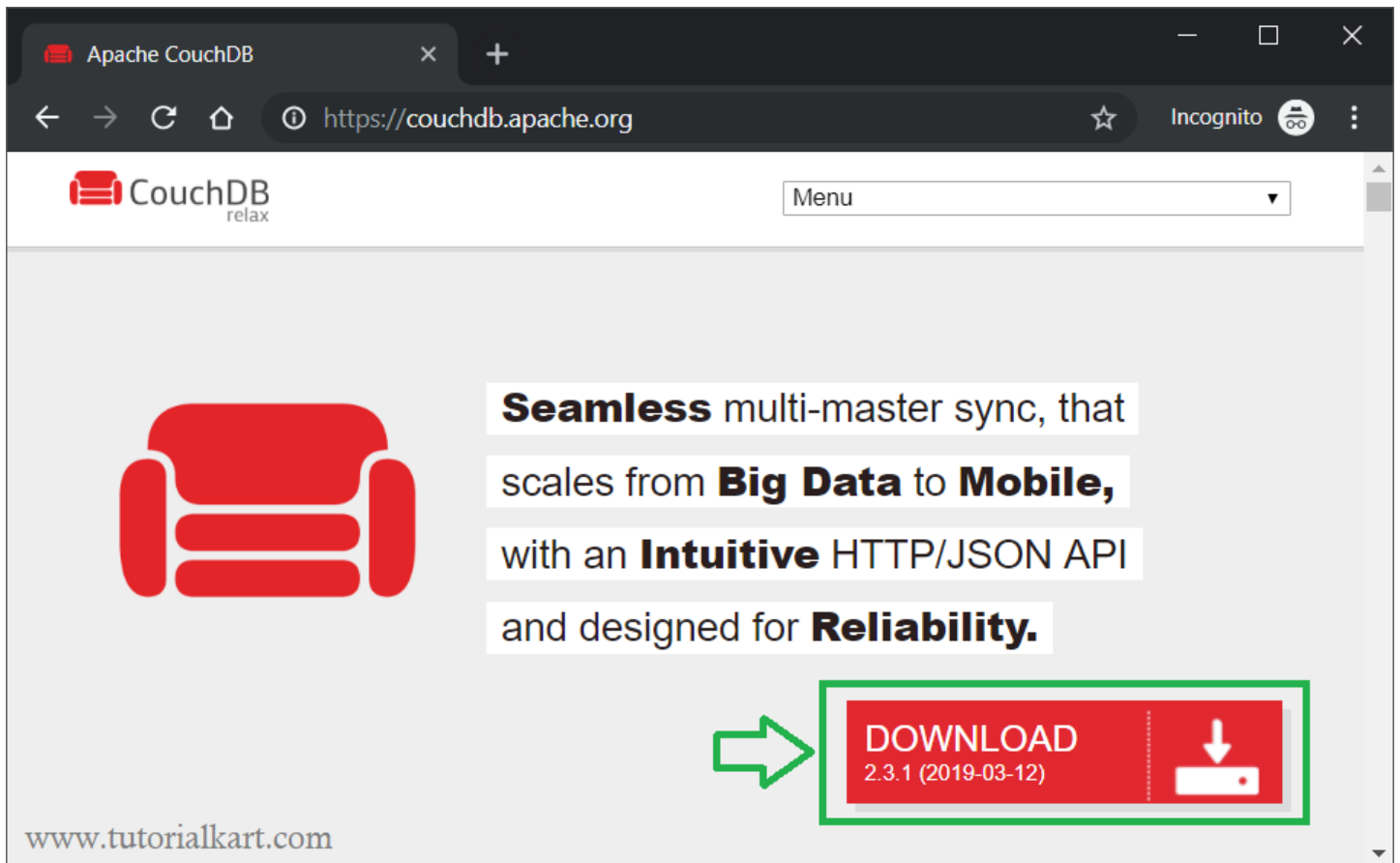
Welcome to CouchDB Tutorial. In this CouchDB Tutorial, we will learn how to install CouchDB, create database in CouchDB, create documents in a database, replication between CouchDBs, configure databases, and many other concepts.

What is CouchDB?

- CouchDB is a NoSQL Database that uses JSON for documents.
- CouchDB uses JavaScript for MapReduce indexes.
- CouchDB uses HTTP for the REST API.

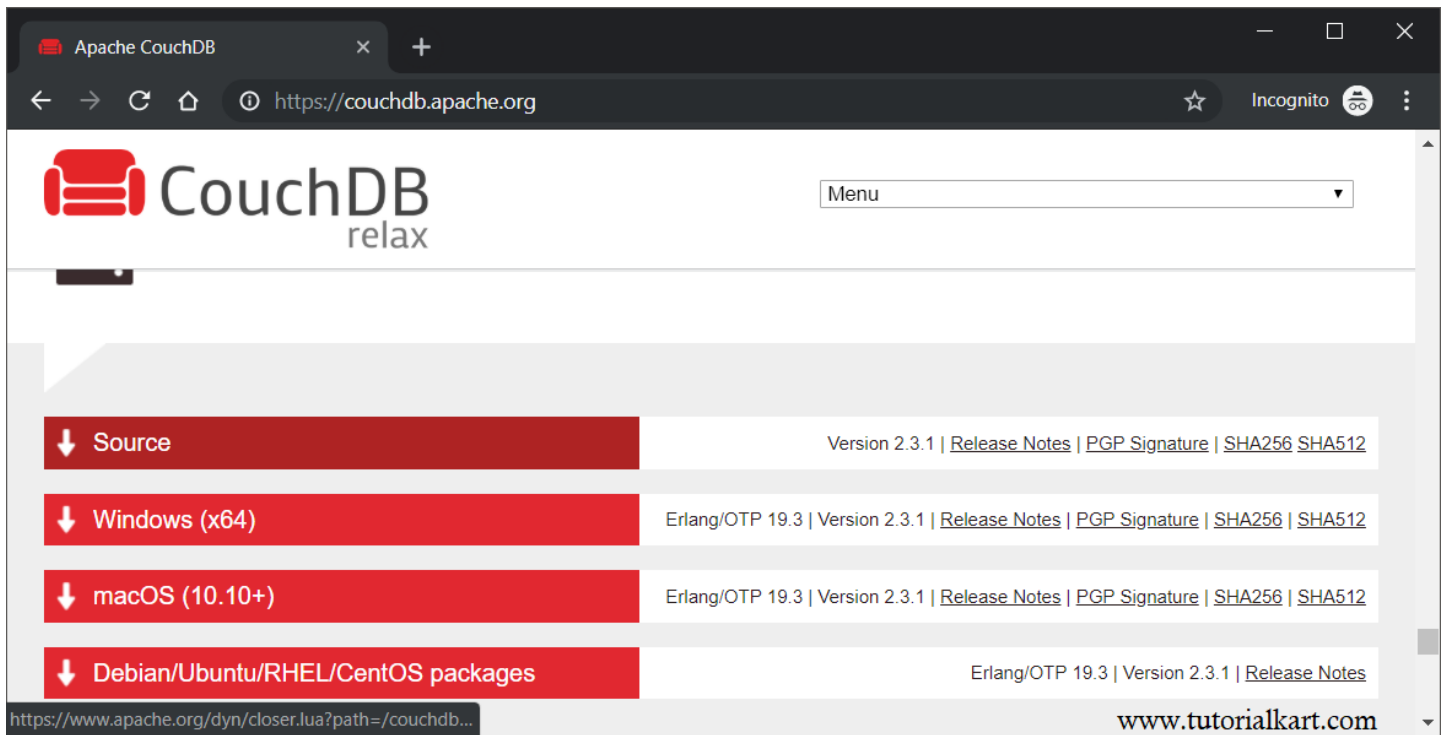
CouchDB Installation

To install CouchDB, visit [<https://couchdb.apache.org/>] and click on the download button as shown below.



The screenshot shows a web browser window with the URL <https://couchdb.apache.org/>. The page features the CouchDB logo (a red sofa) and the tagline "relax". The main content area contains the text: "Seamless multi-master sync, that scales from **Big Data** to **Mobile**, with an **Intuitive** HTTP/JSON API and designed for **Reliability**." Below this text is a red button labeled "DOWNLOAD 2.3.1 (2019-03-12)" with a download icon. A green arrow points to the button, which is also enclosed in a green rectangular box. The footer of the page displays "www.tutorialkart.com".

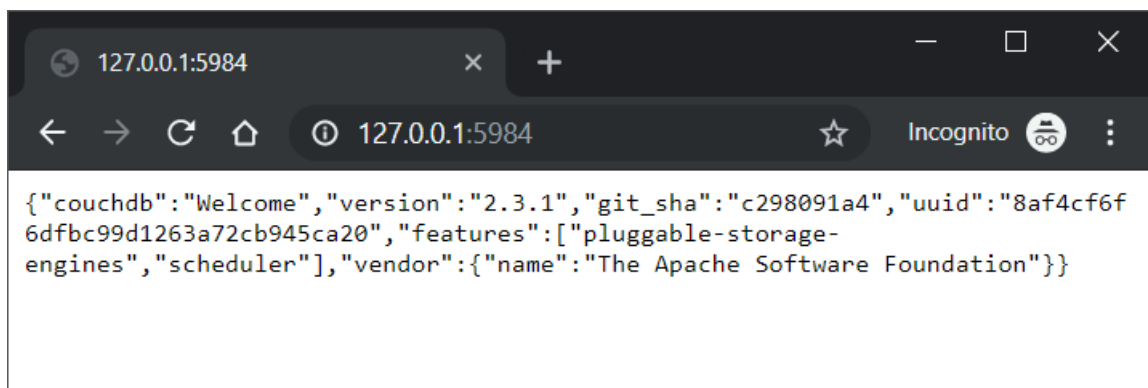
When you click on the download button, it scrolls to the section, where based on your Operating System, you can download the installer.



In this tutorial, we have downloaded for Windows (x64), and it should not make any difference if you download for macOS or Debian/Ubuntu/RHEL/CentOS.

Double click the downloaded installer and follow through the steps.

Once the installation is complete, you can check if CouchDB is installed successfully by requesting the URL `http://127.0.0.1:5984/` in your browser.



This is CouchDB saying welcome to you, along with information about CouchDB version, GIT hash, UUID, features and vendor.

Fauxton

Fauxton is a web based interface built into CouchDB. You can do actions like creating and deleting databases, CRUD operations on documents, user management, running MapReduce on indexes, replication between CouchDB instances.

You can access CouchDB through Fauxton available at the URL `http://127.0.0.1:5984/_utils/`.

Project Fauxton - _all_dbs

127.0.0.1:5984/_utils/#/_all_dbs

Incognito

Databases

Create Database {}JSON

Name	Size	# of Docs	Actions
------	------	-----------	---------

Showing 1-0 of 0 databases. Databases per page 20 « 1 »

CouchDB
Fauxton on Apache CouchDB v. 2.3.1

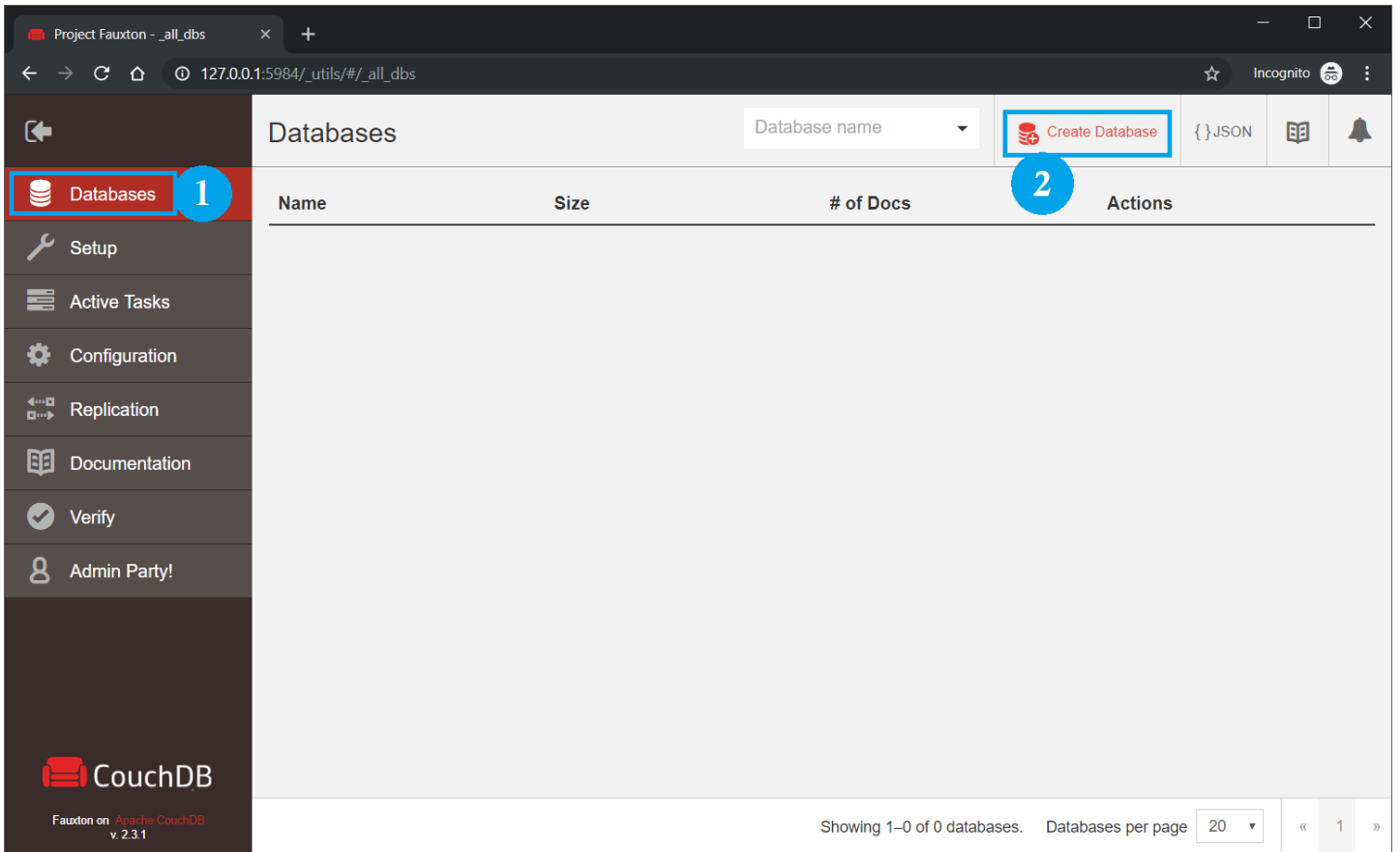
www.tutorialkart.com

Here you can access the following tabs in the left menu.

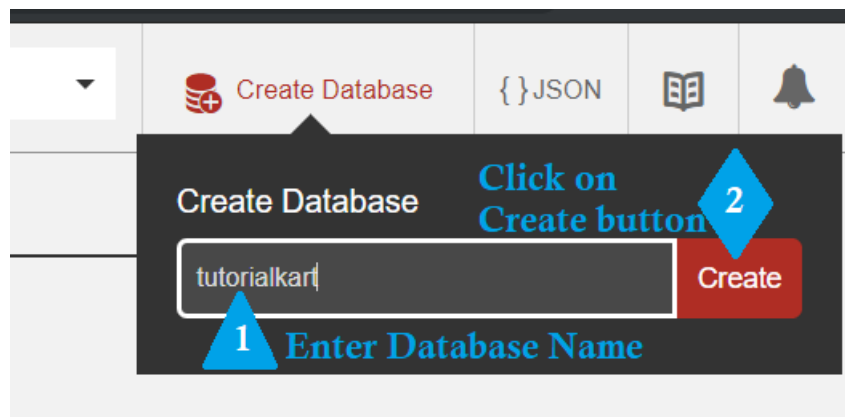
- All Databases
- Setup
- Active Tasks
- Configuration
- Replication
- Documentation
- Verify

Create Database in CouchDB

To create a CouchDB Database, click on Databases tab in the left menu and then click on Create Database.



After you click on the 'Crete Database' button, a pop up appears as shown below.



Enter the database name you like to create and click on **Create** button.

The screenshot displays the TutorialKart web interface for a CouchDB database named 'tutorialkart'. The browser's address bar shows the URL '127.0.0.1:5984/_utils/#database/tutorialkart/_all_docs'. The interface is divided into a sidebar on the left and a main content area on the right. The sidebar contains navigation links: Databases (highlighted), Setup, Active Tasks, Configuration, Replication, Documentation, Verify, and Admin Party. The main content area shows a large couch icon and the text 'No Documents Found'. A green 'Create Document' button is located in the top right corner. The footer of the interface includes the CouchDB logo and the text 'Fauxton on Apache CouchDB v. 2.3.1'. At the bottom right, there is a 'Documents per page:' dropdown menu set to '20' and navigation arrows.

Now that a Database is created. We shall look into Documents inside Database.

Create a Document in CouchDB Database

To create a document in database, click on the **Create Document** button.

Project Fauxton - database/tutorialkart

127.0.0.1:5984/_utils/#database/tutorialkart/_all_docs

Incognito

tutorialkart

Databases All Documents +

Run A Query with Mango

Permissions

Changes

Design Documents +

Create Document

No Documents Found

www.tutorialkart.com

Documents per page: 20

CouchDB

Fauxton on Apache CouchDB v.2.3.1

Now you will see a JSON document as shown below, with `_id` field pre-populated.

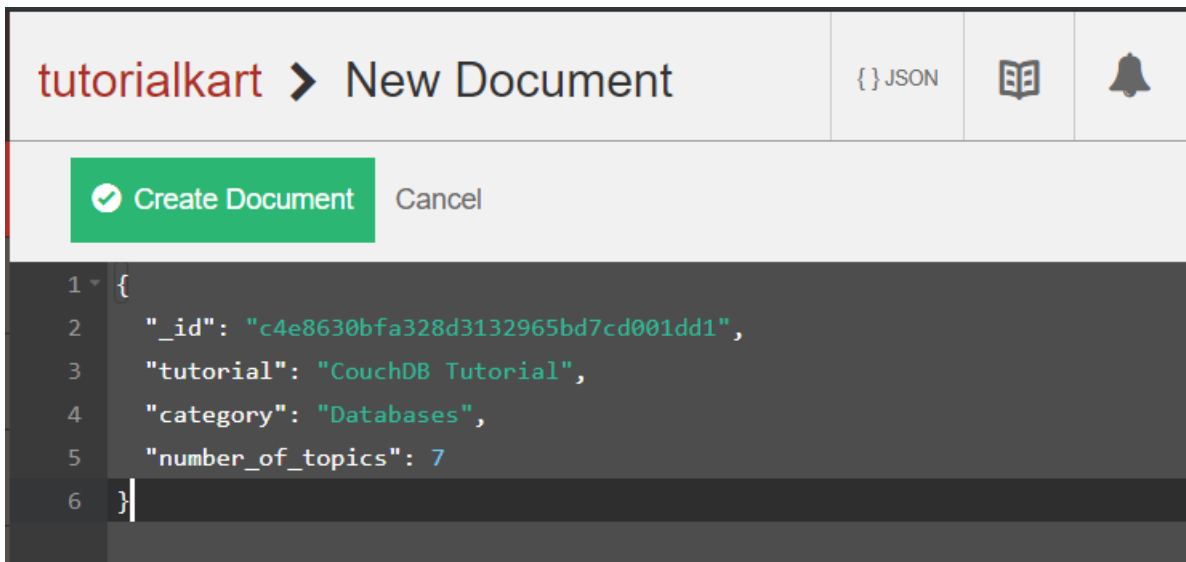
tutorialkart > New Document

{ } JSON

Create Document Cancel

```
1 {
2   "_id": "c4e8630bfa328d3132965bd7cd001dd1"
3 }
```

You may keep the `_id` as is, or you can change. You can add more fields to the JSON document. And click on **Create Document** button.



View documents of CouchDB Database

You can view the documents of CouchDB Database in three views.

1. Table
2. Metadata
3. JSON

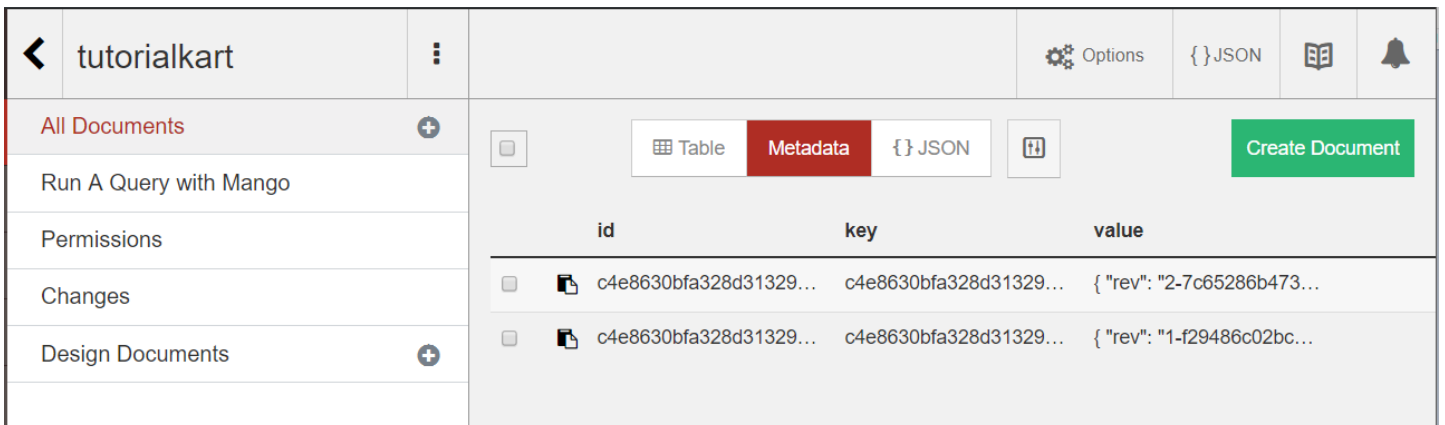
Table

The table view contains all the key values across documents as column names and their corresponding values for each document as row.

	_id	category	number_of_to	tutorial
<input type="checkbox"/>	c4e8630bfa328...	Databases	7	CouchDB Tutorial
<input type="checkbox"/>	c4e8630bfa328...	Databases	34	MongoDB Tutorial

Metadata

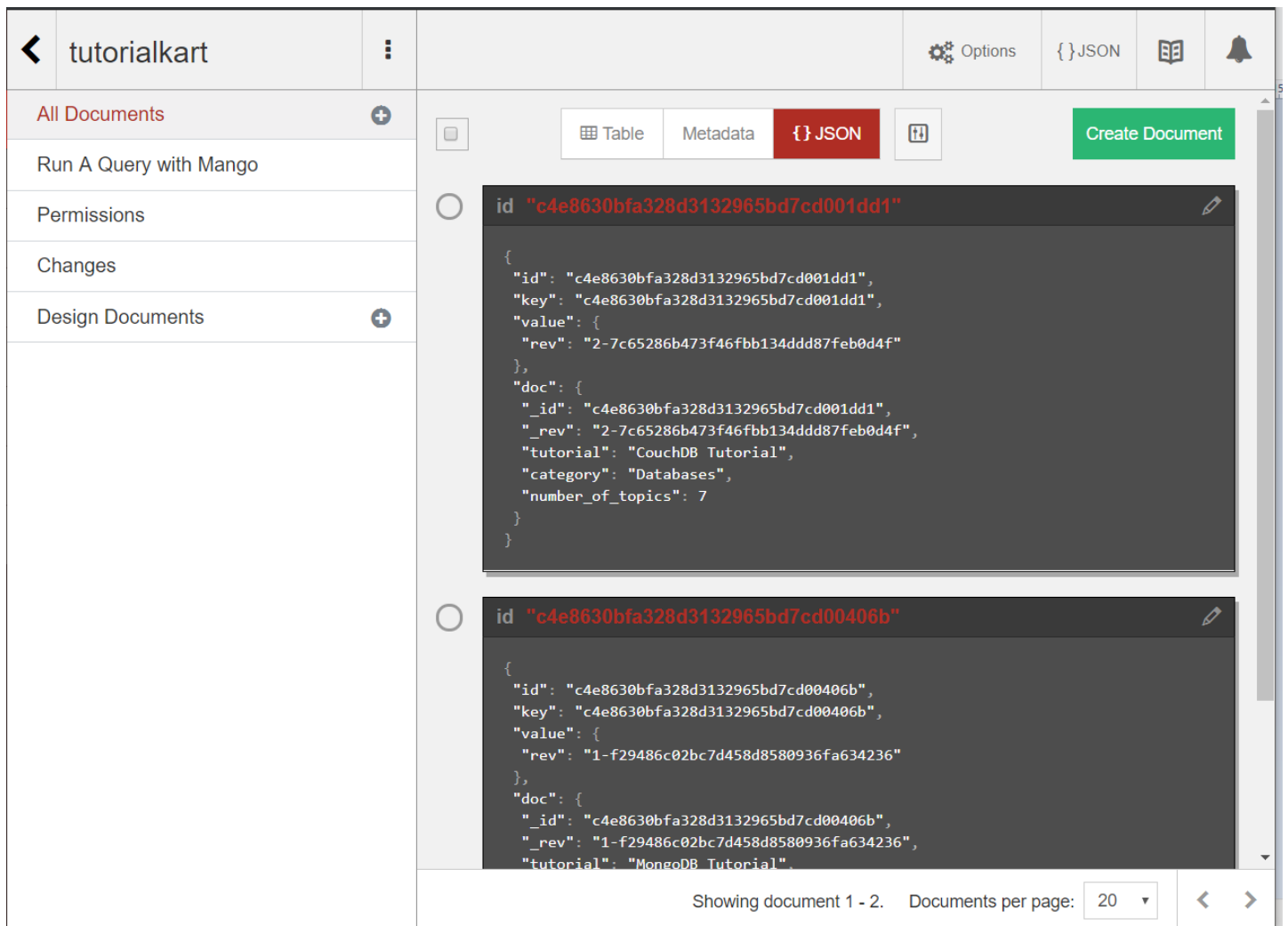
The Metadata view contains id, key and value as columns.



JSON



The JSON view shows every document in JSON format.

You can see here clearly what each document is comprised of, other than the value we provide while creating a document. The meta information consists of `id`, `key` and `value` fields. The `doc` field is the actual document we provide when we created a document in this database.



Update CouchDB Document



To update CouchDB Document, from any of the View (Table, Metadata, or JSON), click on the document you would like to edit.

tutorialkart > c4e8630bfa328d3132965bd7cd001dd1 {JSON}  

Save Changes Cancel

```
1 {
2   "_id": "c4e8630bfa328d3132965bd7cd001dd1",
3   "_rev": "2-7c65286b473f46fbb134ddd87feb0d4f",
4   "tutorial": "CouchDB Tutorial",
5   "category": "Databases",
6   "number_of_topics": 7
7 }
```

You can edit any of the fields. We will change `tutorial` field to `Apache CouchDB Tutorial` and `number_of_topics` to `9`. Make the changes to fields and click on **Save Changes** button.

tutorialkart > c4e8630bfa328d3132965bd7cd001dd1 {JSON}  


Save Changes Cancel

```
1 {
2   "_id": "c4e8630bfa328d3132965bd7cd001dd1",
3   "_rev": "2-7c65286b473f46fbb134ddd87feb0d4f",
4   "tutorial": "Apache CouchDB Tutorial",
5   "category": "Databases",
6   "number_of_topics": 9
7 }
```

2 **1 Update values**

Once you click on **Save Changes**, a message `Saving document.` is displayed.

Project Fauxton - database/tutor... +

127.0.0.1:5984/_utils/#database/tutorialkart/_all_docs ☆ Incognito 

i Saving document. ×

You can also add new fields to the document.

Accessing CouchDB Database through REST API

You can access CouchDB Database through REST API using HTTP requests like GET, PUT, etc., for operations like view, update and delete operations.

View Document – GET request

To get Document, send a GET request to the document URL

http://hostname_or_IP:Port/databasename/document_id.

If you are using Postman, send GET request with the following URL:

```
http://127.0.0.1:5984
/tutorialkart/c4e8630bf
http://127.0.0.1:5984/tutorialkart/c4e8630bfa328d3132965bd7cd001dd1
```



Update Document – PUT request

To update Document, send a PUT request with the document URL (http://hostname/databasename/document_id/) and pass the JSON data for update in the body.

URL

```
http://127.0.0.1:5984
/tutorialkart/c4e8630bf
http://127.0.0.1:5984/tutorialkart/c4e8630bfa328d3132965bd7cd001dd1/
```

Body

```
{
  "_rev": "3-729ec8f85148981fdf155cbc4d3e41fd",
  "tutorial": "CouchDB Tutorial",
  "category": "NoSQL Databases",
  "number_of_topics": 7
}
```

Get “_rev” for the document which you would like to edit and send it along with the fields with updated values.

The screenshot shows a REST client interface. At the top, the method is set to **PUT** and the URL is `http://127.0.0.1:5984/tutorialkart/c4e8630bfa328d3132965bd7cd001dd1/`. The **Body** tab is selected, showing a JSON request body:

```
{
  "_rev": "3-729ec8f85148981fdf155cbc4d3e41fd",
  "tutorial": "CouchDB Tutorial",
  "category": "NoSQL Databases",
  "number_of_topics": 7
}
```

 A green arrow points to this body with the text "Data sent in request". Below the request, the **Body** tab shows the response:

```
{
  "ok": true,
  "id": "c4e8630bfa328d3132965bd7cd001dd1",
  "rev": "4-3a0d4167a3ccbdf5a017b975798f145f"
}
```

 A green arrow points to this response with the text "Response". The status bar indicates "Status: 201 Created", "Time: 103ms", and "Size: 458 B".

In the response, `ok` field is `true`, which means the update is successful. Also the revision field `rev` is updated to `4-xxxxx` meaning fourth revision.

To verify the document update, you may send GET request with the database and document id in url.

The screenshot shows a REST client interface. At the top, the method is set to **GET** and the URL is `http://127.0.0.1:5984/tutorialkart/c4e8630bfa328d3132965bd7cd001dd1/`. The **Body** tab is selected, showing the message "This request does not have a body". Below, the **Body** tab shows the response:

```
{
  "_id": "c4e8630bfa328d3132965bd7cd001dd1",
  "_rev": "4-3a0d4167a3ccbdf5a017b975798f145f",
  "tutorial": "CouchDB Tutorial",
  "category": "NoSQL Databases",
  "number_of_topics": 7
}
```

 A green arrow points to this response with the text "Updated Document". The status bar indicates "Status: 200 OK", "Time: 16ms", and "Size: 447 B".

Delete Document

To delete document from CouchDB Database, send HTTP DELETE request with the url `http://hostname/database_name/document_id/` with parameter `ver`.

We made HTTP GET request in the above section where we updated the document. We will delete that document, hence use the `_id` and `_ver`.

http://127.0.0.1:5984

http://127.0.0.1:5984/tutorialkart/c4e8630bfa328d3132965bd7cd001dd1/?rev=4-3a0d4167a3ccbd5a017b975798f145f

The screenshot shows a REST client interface with a DELETE request to the URL `http://127.0.0.1:5984/tutorialkart/c4e8630bfa328d3132965bd7cd001dd1/?rev=4-3a0d4167a3ccbd5a017b975798f145f`. The response is a JSON object with the following structure:

```
{
  "ok": true,
  "id": "c4e8630bfa328d3132965bd7cd001dd1",
  "rev": "5-904478687b7954f194f6f965a273f8e0"
}
```

The response status is 200 OK, with a time of 207ms and a size of 374 B. The response is displayed in a Pretty JSON format.

In the response, we get ok field with true value. Therefore, the document is deleted. The response also contains the document id and revision number.

Concluding this CouchDB Tutorial

Summarizing what we have learned in this CouchDB Tutorial: What CouchDB is? How to install CouchDB, How to create a database, How to create Document in Database, and the CRUD operations using web interface and HTTP requests.

CouchDB Tutorial

- ↳ CouchDB Tutorial

CouchDB Database

- ↳ CouchDB - Create Database

- ↳ CouchDB - Delete Database

- ↳ CouchDB - Get Database List

CouchDB Document

- ↳ CouchDB - Create Document

- ↳ CouchDB - View Document

- ↳ CouchDB - Update Document

- ↳ CouchDB - Delete Document

CouchDB Tools

- ↳ CouchDB Replication

- ↳ CouchDB Configuration

- ↳ CouchDB - Configure a Cluster