# Final Project Report (Reuters)

Table des matières

[Final Project Report (Reuters) 1](#_Toc509745439)

[Introduction: 2](#_Toc509745440)

[Technology Stack chosen: 2](#_Toc509745441)

[Usage 2](#_Toc509745442)

[Importation of data 3](#_Toc509745443)

[Contents 3](#_Toc509745444)

[Search page 3](#_Toc509745445)

[Analysis page 4](#_Toc509745446)

## Introduction:

This project is about creating an application on the Reuters NoSQL database.

This application has to fulfill the following requirements:

* Connection features;
* Dataset Importation;
* Minimum 3 different interactions in order to show the result of queries (preformatted queries)
* A form to try queries (maybe with params) on the dataset

## Technology Stack chosen:

We have chosen Elasticsearch because the possibility to visualize on Kibana would inspire us to create our own chart on our application.

Moreover, we have chosen to create a web application with Node.js and React.js. Indeed this semester, we have learnt how to code in React.js.

It is a good way to link our Web application course in which we code in JavaScript to this course with Elasticsearch. Furthermore, Elasticsearch gives the opportunity to use their JavaScript API for Elasticsearch.

## Usage

* Initial step:

If you do not have node.js installed in your computer, download it on <https://nodejs.org/en/>

If you do not have npm installed in your computer, download it on <https://www.npmjs.com/get-npm>

* First step:

Launch the Elasticsearch.bat file.

* Second step:

With the command prompt, use the command line ‘npm install’ into:

* The server folder.
* The client folder.
* Third step:

With a command prompt use the command line:

* ‘node index.js’ into the server folder.

Then on a webpage, go to the <http://localhost:3001/article/populate> link. It will import all the data and do the mappings.

* Fourth step:

With a command prompt use the command line:

* ‘ npm start’ into the client folder.
* Fifth step:

The web application is now launching.

## Importation of data

The data is imported using the [reuters\_elastic.json](https://github.com/vidjul/NoSQLProject/blob/master/server/reuters_elastic.json) file. It is imported automatically when the application is launched.

## Contents

There are two pages:

* Search page
* Analysis page

### 

### Search page

In this page, you can do simple queries by filling the following parameters:

* Date.
* Title.
* Body.
* Places.
* Topics.
* People.

For each parameter, you can choose if the field has to be into a **should** or **must** bracket.

Here is an example:

Depending on the length of the article, the label of an article can be green (small article), blue (average size article) or red (long article).

If you click on one of those buttons, you can get details about that article:



### Analysis page

Our aim is to show you that we can plot graphs using queries from Elasticsearch (visualisation like in Kibana).

In this page, you have two graphs. When you select a country, it would count the number of articles group by the people column (for the first graph) and by the topics (for the second graph) if there are articles. Depending on the number of articles, there is a host of colours.

Here is an example with the country UK:



Here is an example with the country Iceland: