MONGODB - TEXT SEARCH

http://www.tutorialspoint.com/mongodb/mongodb_text_search.htm

Starting from version 2.4, MongoDB started supporting text indexes to search inside string content. The **Text Search** uses stemming techniques to look for specified words in the string fields by dropping stemming stop words like a, an, the, etc. At present, MongoDB supports around 15 languages.

Enabling Text Search:

Initially Text Search was an experimental feature but starting from version 2.6, the configuration is enabled by default. But if you are using previous version of MongoDB, you have to enable text search with following code:

```
>db.adminCommand({setParameter:true,textSearchEnabled:true})
```

Creating Text Index:

Consider the following document under **posts** collection containing the post text and its tags:

```
{
   "post_text": "enjoy the mongodb articles on tutorialspoint",
   "tags": [
      "mongodb",
      "tutorialspoint"
   ]
}
```

We will create a text index on post_text field so that we can search inside our posts' text:

```
>db.posts.ensureIndex({post_text:"text"})
```

Using Text Index:

Now that we have created the text index on post_text field, we will search for all the posts that have word **tutorialspoint** in their text.

```
>db.posts.find({$text:{$search:"tutorialspoint"}})
```

The above command returned the following result documents having tutorialspoint word in thier post text:

```
{
   "_id" : ObjectId("53493d14d852429c10000002"),
   "post_text" : "enjoy the mongodb articles on tutorialspoint",
   "tags" : [ "mongodb", "tutorialspoint" ]
}
{
   "_id" : ObjectId("53493d1fd852429c10000003"),
   "post_text" : "writing tutorials on mongodb",
   "tags" : [ "mongodb", "tutorial" ]
}
```

If you are using old versions of MongoDB, you have to use the following command:

```
>db.posts.runCommand("text",{search:" tutorialspoint "})
```

Using Text Search highly improves the search efficiency as compared to normal search.

Deleting Text Index:

To delete an existing text index, first find the name of index using following query:

```
>db.posts.getIndexes()
```

After getting the name of your index from above query, run the following command. Here, **post_text_text** is the name of the index.

```
>db.posts.dropIndex("post_text_text")
```