

MONGODB - REGULAR EXPRESSION

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

Copyright © tutorialspoint.com

Regular Expressions are frequently used in all languages to search for a pattern or word in any string. MongoDB also provides functionality of regular expression for string pattern matching using the **\$regex** operator. MongoDB uses PCRE *PerlCompatibleRegularExpression* as regular expression language.

Unlike text search, we do not need to do any configuration or command to use regular expressions.

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

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

Using regex Expression

The following regex query searches for all the posts containing string **tutorialspoint** in it:

```
>db.posts.find({post_text:{$regex:"tutorialspoint"}})
```

The same query can also be written as:

```
>db.posts.find({post_text:/tutorialspoint/})
```

Using regex Expression with case insensitive

To make the search case insensitive, we use the **\$options** parameter with value **\$i**. The following command will look for strings having word **tutorialspoint**, irrespective of smaller or capital case:

```
>db.posts.find({post_text:{$regex:"tutorialspoint", $options:"$i"}})
```

One of the results retuned from this query is following document which contains word **tutorialspoint** in different cases:

```
{  
  "_id" : ObjectId("53493d37d852429c10000004"),  
  "post_text" : "hey! this is my post on TutorialsPoint",  
  "tags" : [ "tutorialspoint" ]  
}
```

Using regex for array elements:

We can also use the concept of regex on array field. This is particularly very important when we implement the functionality of tags. So, if you want to search for all the posts having tags beginning from word **tutorial** either **tutorial** or **tutorialspoint** or **tutorialphp**, you can use the following code:

```
>db.posts.find({tags:{$regex:"tutorial"}})
```

Optimizing Regular Expression Queries:

- If the document fields are **indexed**, the query will use make use of indexed values to match the regular expression. This makes the search very fast as compared to the regular expression scanning the whole collection.
- If the regular expression is a **prefix expression**, all the matches are meant to start with a certain string characters. For e.g., if the regex expression is `^tut`, then the query has to ~~search for only those strings that begin with tut.~~

>Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js