MongoDB 全文检索 MongoDB 管理工具: Rockmongo

MongoDB 正则表达式

正则表达式是使用单个字符串来描述、匹配一系列符合某个句法规则的字符串。

许多程序设计语言都支持利用正则表达式进行字符串操作。

MongoDB 使用 **$regex** 操作符来设置匹配字符串的正则表达式。

MongoDB使用PCRE (Perl Compatible Regular Expression) 作为正则表达式语言。

不同于全文检索，我们使用正则表达式不需要做任何配置。

考虑以下 **posts** 集合的文档结构，该文档包含了文章内容和标签：

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

使用正则表达式

以下命令使用正则表达式查找包含 runoob 字符串的文章：

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

以上查询也可以写为：

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

不区分大小写的正则表达式

如果检索需要不区分大小写，我们可以设置 $options 为 $i。

以下命令将查找不区分大小写的字符串 runoob：

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

集合中会返回所有包含字符串 runoob 的数据，且不区分大小写：

```
{
   "_id" : ObjectId("53493d37d852429c10000004"),
   "post_text" : "hey! this is my post on  runoob",
```

```
    "tags" : [ "runoob" ]
}
```

数组元素使用正则表达式

我们还可以在数组字段中使用正则表达式来查找内容。 这在标签的实现上非常有用，如果你需要查找包含以 run 开头的标签数据(ru 或 run 或 runoob)， 你可以使用以下代码：

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

优化正则表达式查询

- 如果你的文档中字段设置了索引，那么使用索引相比于正则表达式匹配查找所有的数据查询速度更快。
- 如果正则表达式是前缀表达式，所有匹配的数据将以指定的前缀字符串为开始。例如： 如果正则表达式为 **^tut** ，查询语句将查找以 tut 为开头的字符串。

这里面使用正则表达式有两点需要注意：

正则表达式中使用变量。一定要使用eval将组合的字符串进行转换，不能直接将字符串拼接后传入给表达式。否则没有报错信息，只是结果为空！实例如下：

```
var name=eval("/" + 变量值key +"/i");
```

以下是模糊查询包含title关键词, 且不区分大小写:

```
title:eval("/"+title+"/i")    // 等同于 title:{$regex:title,$Option:"$i"}
```

← MongoDB 全文检索 MongoDB 管理工具: Rockmongo →

- {<field>: /pattern/<options>}

  $regex与正则表达式对象的区别:

- 在$in操作符中只能使用正则表达式对象，例如:{name:{$in:[/^joe/i,/^jack/}}
- 在使用隐式的$and操作符中，只能使用$regex，例如:{name:{$regex:/^jo/i, $nin:['john']}}
- 当option选项中包含X或S选项时，只能使用$regex，例如:{name:{$regex:/m.*line/,$options:"si"}}

  $regex操作符的使用

  $regex操作符中的option选项可以改变正则匹配的默认行为，它包括i, m, x以及S四个选项，其含义如下

- i 忽略大小写，{<field>{$regex/pattern/i}}，设置i选项后，模式中的字母会进行大小写不敏感匹配。
- m 多行匹配模式，{<field>{$regex/pattern/,$options:'m'}，m选项会更改^和$元字符的默认行为，分别使用与行的开头和结尾匹配，而不是与输入字符串的开头和结尾匹配。
- x 忽略非转义的空白字符，{<field>:{$regex:/pattern/,$options:'m'}，设置x选项后，正则表达式中的非转义的空白字符将被忽略，同时井号(#)被解释为注释的开头注，只能显式位于option选项中。
- s 单行匹配模式{<field>:{$regex:/pattern/,$options:'s'}，设置s选项后，会改变模式中的点号(.)元字符的默认行为，它会匹配所有字符，包括换行符(\n)，只能显式位于option选项中。

  使用$regex操作符时，需要注意下面几个问题:

- i，m，x，s可以组合使用，例如:{name:{$regex:/j*k/,$options:"si"}}
- 在设置索弓}的字段上进行正则匹配可以提高查询速度，而且当正则表达式使用的是前缀表达式时，查询速度会进一步提高，例如:{name:{$regex: /^joe/}

牛牛 1年前 (2018-01-03)