

método pretty () no MongoDB

Publicado por [admin](#) em 14 de dezembro de 2018

método pretty () no MongoDB:

O método `pretty ()` é usado principalmente para exibir o resultado em um formato mais fácil de ler. Neste tutorial, mostrarei como usar o método `pretty ()` e como os dados são exibidos com e sem o uso de `pretty ()` e como são os dados. A sintaxe do método `pretty ()` é a seguinte:

```
01. db.collection. find ( ) . bonita ( )
```

Se chamarmos o método `find` em uma coleção, ele mostrará todos os documentos disponíveis nessa coleção. Opcionalmente, podemos passar uma `consulta` para esse método.

Exemplo de uso:

Vamos tentar entender esse método com um exemplo. Abra seu Mongo-shell e crie uma coleção com poucos documentos, como abaixo:

```
01. db.list. insertMany ( [  
02.   { _id: 1 , nome: "Alex" , idade: 16 anos , notas: { física: 90 , química: 85 , matemática: 59  
    } } ,  
03.   { _id: 2 , nome: "Albert" , idade: 17 anos , notas: { física: 92 , química: 87 , matemática:  
    88 } } ,  
04.   { _id: 3 , nome: "Bob" , idade: 16 anos , notas: { física: 88 , química: 81 , matemática: 66  
    } } ,  
05.   { _id: 4 , nome: "John" , idade: 17 anos , notas: { física: 78 , química: 55 , matemática: 63  
    } }  
06. ] )
```

Ele irá inserir esses quatro documentos na `lista de coleta`. Agora, chame `db.list.find ()` para imprimir todos os documentos nele. Ele imprimirá a saída abaixo:

```
01. { "_id" : 1 , "nome" : "Alex" , "idade" : 16 , "marcas" : { "física" : 90 , "química" : 85 ,  
    "matemática"  
02.   : 59 } }  
03. { "_id" : 2 , "nome" : "Albert" , "idade" : 17 , "marcas" : { "física" : 92 , "química" : 87  
    , "matemática  
04.   ": 88 } }  
05. { "_id" : 3 , "nome" : "Bob" , "idade" : 16 , "marcas" : { "física" : 88 , "química" : 8  
    "matemática" :  
06.   66 } }
```



Code vs Color



```
> db.list.insertMany([
... { _id : 1, name : "Alex", age : 16, marks : { physics : 90, chemistry : 85, mathematics : 59 } },
... { _id : 2, name : "Albert", age : 17, marks : { physics : 92, chemistry : 87, mathematics : 88 } },
... { _id : 3, name : "Bob", age : 16, marks : { physics : 88, chemistry : 81, mathematics : 66 } },
... { _id : 4, name : "John", age : 17, marks : { physics : 78, chemistry : 55, mathematics : 63 } }
... ])
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
> db.list.find()
{ "_id" : 1, "name" : "Alex", "age" : 16, "marks" : { "physics" : 90, "chemistry" : 85, "mathematics" : 59 } }
{ "_id" : 2, "name" : "Albert", "age" : 17, "marks" : { "physics" : 92, "chemistry" : 87, "mathematics" : 88 } }
{ "_id" : 3, "name" : "Bob", "age" : 16, "marks" : { "physics" : 88, "chemistry" : 81, "mathematics" : 66 } }
{ "_id" : 4, "name" : "John", "age" : 17, "marks" : { "physics" : 78, "chemistry" : 55, "mathematics" : 63 } }
>
```

codevscolor.com

Temos apenas quatro elementos aqui. Se o tamanho dos dados for muito grande, torna-se difícil de ler. Se você chamar o método `pretty()`, ele imprimirá o resultado como abaixo:

```
01.  {
02.      "_id" : 1 ,
03.      "nome" : "Alex" ,
04.      "idade" : 16 ,
05.      "marks" : {
06.          "física" : 90 ,
07.          "química" : 85 ,
08.          "matemática" : 59
09.      }
10.  }
11.  {
12.      "_id" : 2 ,
13.      "nome" : "Albert" ,
14.      "idade" : 17 ,
15.      "marks" : {
16.          "física" : 92 ,
17.          "química" : 87 ,
18.          "matemática" : 88
19.      }
20.  }
21.  {
22.      "_id" : 3 ,
```



Code vs Color



```
26.         "física" : 80 ,
27.         "química" : 81 ,
28.         "matemática" : 66
29.     }
30. }
31. {
32.     "_id" : 4 ,
33.     "name" : "John" ,
34.     "idade" : 17 ,
35.     "marks" : {
36.         "física" : 78 ,
37.         "química" : 55 ,
38.         "matemática" : 63
39.     }
40. }
```

```
> db.list.find().pretty()
{
  "_id" : 1,
  "name" : "Alex",
  "age" : 16,
  "marks" : {
    "physics" : 90,
    "chemistry" : 85,
    "mathematics" : 59
  }
}
{
  "_id" : 2,
  "name" : "Albert",
  "age" : 17,
  "marks" : {
    "physics" : 92,
    "chemistry" : 87,
    "mathematics" : 88
  }
}
{
  "_id" : 3,
  "name" : "Bob",
  "age" : 16,
  "marks" : {
    "physics" : 88,
    "chemistry" : 81,
    "mathematics" : 66
  }
}
{
  "_id" : 4,
  "name" : "John",
  "age" : 17,
  "marks" : {
    "physics" : 78,
    "chemistry" : 55,
    "mathematics" : 63
  }
}
>
```

codevscolor.com



Code vs Color



saída abaixo:

```
01.  {
02.      "_id" : 1 ,
03.      "nome" : "Alex" ,
04.      "idade" : 16 ,
05.      "marks" : {
06.          "física" : 90 ,
07.          "química" : 85 ,
08.          "matemática" : 59
09.      }
10.  }
11.  {
12.      "_id" : 3 ,
13.      "nome" : "Bob" ,
14.      "idade" : 16 ,
15.      "marks" : {
16.          "física" : 88 ,
17.          "química" : 81 ,
18.          "matemática" : 66
19.      }
20.  }
```



Code vs Color



```
{
  "_id" : 1,
  "name" : "Alex",
  "age" : 16,
  "marks" : {
    "physics" : 90,
    "chemistry" : 85,
    "mathematics" : 59
  }
}
{
  "_id" : 3,
  "name" : "Bob",
  "age" : 16,
  "marks" : {
    "physics" : 88,
    "chemistry" : 81,
    "mathematics" : 66
  }
}
> |
```

codevscolor.com

Conclusão:

pretty () é útil em muitos casos. Se nosso banco de dados for enorme, esse método realmente será útil. Siga os exemplos acima e solte um comentário abaixo, se você tiver alguma dúvida.

Você pode gostar também :

Verifique também

[MongoDB: Inserir um único documento em uma coleção](#)

[Inserir vários documentos em uma coleção do MongoDB usando insertMany](#)

[Como verificar o tipo de um campo no shell Mongo](#)

Categorias: **MONGODB**

