

MongoDB Project Fields – Restrict Fields to Return

MongoDB Project Fields

MongoDB Project Fields – In this [MongoDB Tutorial](#), we shall learn to restrict only those fields to return in the result of a query.

In [MongoDB Query Documents](#) tutorial, we have learnt to filter [MongoDB Documents](#) based on a criteria, in which all the fields are returned in the result. But it is not always the case that we need all the fields in the result. We might be interested only on certain fields of the queried documents.

To limit the fields, we can make use of `find()` method. We can provide the required fields in a projection document and provide this projection document as second argument of `find()` method.

Syntax of `find()` method

Following is the syntax of `find()` method to accept projection document

```
db.collection.find(query_
```

```
db.collection.find(query_document, projection_document)
```

where `projection_document` is

```
{field1:projection_valu
```

```
{field1:projection_value, field1:projection_value, ..}
```

and `projection_value` can take 1 or 0.

Projection_value	Description
1	Include the field:value in Result
0	Do not include the field:value in Result

Example – MongoDB Projection

Following is an example to limit fields in result using projection document.

1. Select Database.

```
> use tutorialkart
switched to db
```

```
> use tutorialkart
switched to db tutorialkart
```

2. Prepare projection document.

```
> projection_doc=
```

```
{"name":1,"place":1,_id:0}
```

```
> projection_doc={"name":1,"place":1,_id:0}
{"name" : 1, "place" : 1, "_id" : 0 }
```

When a projection document is provided, by default projection value of `_id` is 1. Change it to 0 if you do not require `_id` in the result. For rest of the fields you may set the projection value to 1, only for those required fields.

3. Query with projection document provided as second argument to `find()` method.

```
>
db.people.find()
```

```
> db.people.find({},projection_doc)
{"name" : "Midhuna", "place" : "Amaravati" }
{"name" : "Akhil", "place" : "New York" }
{"name" : "Honey" }
{"name" : "Manju", "place" : "Vizag" }
{"name" : "Bharat", "place" : "New York" }
{"name" : "Arya" }
```

The result contains only those fields with respect to projection document.

Conclusion :

In this [MongoDB Tutorial](#), we have learnt to limit the fields in the result using projection document as second argument to `find()` method.

MongoDB Tutorial

- ▮ [MongoDB Tutorial](#)
- ▮ [Install MongoDB on Ubuntu](#)
- ▮ [Start MongoDB Server](#)
- ▮ [MongoDB Shell](#)
- ▮ [Check MongoDB Version](#)
- ▮ [MongoDB Server Port Number - Default Value and How to Change it](#)
- ▮ [MongoDB Script](#)

↳ MongoDB Database

↳ MongoDB Create Database

↳ MongoDB Delete Database

Collections

↳ MongoDB Collection

↳ MongoDB Create Collection

↳ MongoDB Delete Collection

Documents

↳ MongoDB Document

↳ MongoDB Insert Document

↳ MongoDB Query Documents

↳ MongoDB Project Fields in Result

↳ MongoDB Update Document

↳ MongoDB Delete Document

↳ MongoDB Limit Documents

↳ MongoDB Skip Documents

↳ MongoDB Sort Documents

↳ MongoDB Setup Replica Set

↳ MongoDB Locks

MongoDB Concepts

↳ MongoDB Text Search

↳ MongoDB MapReduce

↳ MongoDB Backup - mongodump

MongoDB Queries

↳ MongoDB Date

MongoDB Queries

↳ MongoDB Date()

MongoDB Integration

MongoDB Java

↳ Connect to MongoDB from Java

MongoDB Python

↳ Connect to MongoDB from Python

MongoDB Kotlin

MongoDB Kotlin

- [Connect to MongoDB from Kotlin](#)

MongoDB Node.js

- [Node.js MongoDB](#)
- [Node.js MongoDB Connection](#)
- [Node.js MongoDB Create Database](#)
- [Node.js MongoDB Drop Database](#)
- [Node.js MongoDB Create Collection](#)
- [Node.js MongoDB Delete Collection](#)
- [Node.js MongoDB Insert Documents](#)
- [MongoError: failed to connect to server](#)

MongoDB Others

- [MongoDB Interview Questions](#)
- [Uninstall MongoDB from Ubuntu](#)

Useful Resources

- [How to Learn Programming](#)