

# C# MongoDB CRUD Basics

October 3, 2023

---

what are some simple c# mongodb CRUD commands

---

In C#, you can interact with MongoDB using the MongoDB C# driver. Here are some simple CRUD (Create, Read, Update, Delete) commands using this driver:

## Install the MongoDB C# Driver:

Before you start, make sure you have installed the MongoDB C# driver using NuGet Package Manager. You can do this by running the following command in your Visual Studio Package Manager Console:

```
csharp
```

 Copy code

```
Install-Package MongoDB.Driver
```

## Create a MongoDB Client:

To interact with a MongoDB database, you need to create a MongoClient. You should typically do this once for your application:

```
csharp
```

 Copy code

```
using MongoDB.Driver; var client = new MongoClient("mongodb://localhost:27017");
```

## Select a Database:

You can select a specific database from the MongoClient:

```
csharp
```

 Copy code

```
var database = client.GetDatabase("your-database-name");
```

## CRUD Operations

Here are some simple CRUD operations:

#### Create (Insert) Document:

csharp

 Copy code

```
using MongoDB.Bson; using MongoDB.Driver; var collection =  
database.GetCollection<BsonDocument>("your-collection-name"); var document = new  
BsonDocument { { "name", "John" }, { "age", 30 }, { "city", "New York" } };  
collection.InsertOne(document);
```

#### Read (Find) Document:

csharp

 Copy code

```
var filter = Builders<BsonDocument>.Filter.Eq("name", "John"); var result =  
collection.Find(filter).ToList(); foreach (var doc in result) {  
Console.WriteLine(doc); }
```

#### Update Document:

csharp

 Copy code

```
var filter = Builders<BsonDocument>.Filter.Eq("name", "John"); var update =  
Builders<BsonDocument>.Update.Set("age", 31); collection.UpdateOne(filter,  
update);
```

#### Delete Document:

csharp

 Copy code

```
var filter = Builders<BsonDocument>.Filter.Eq("name", "John");  
collection.DeleteOne(filter);
```

These are basic examples of CRUD operations using the MongoDB C# driver. Remember to handle exceptions and errors appropriately in your application for robust database interaction. Additionally, you should use strongly-typed classes instead of `BsonDocument` for better code