

Paweł Rajba

[pawel@cs.uni.wroc.pl](mailto:pawel@cs.uni.wroc.pl)

<http://pawel.ii.uni.wroc.pl/>

# MongoDB

# Agenda

- NoSQL introduction
- MongoDB
  - Getting started
  - Shell
  - Storage
  - Structure
  - Manipulating data
  - Querying data
  - Replication
  - Client application

# NoSQL Introduction

- New trend in a database world
- NoSQL stands for „Not only SQL“
- Different mindset
  - No joins
  - No complex transactions
- Easily scalable
- ACID in a node, BASE outside the node

# Why?

- Simple flexible structure (no schemas)
- Easier scalable (mostly horizontally)
- Performance (especially for write and delete)
- Indexing
- Easy replication and failover support
- ... and many others

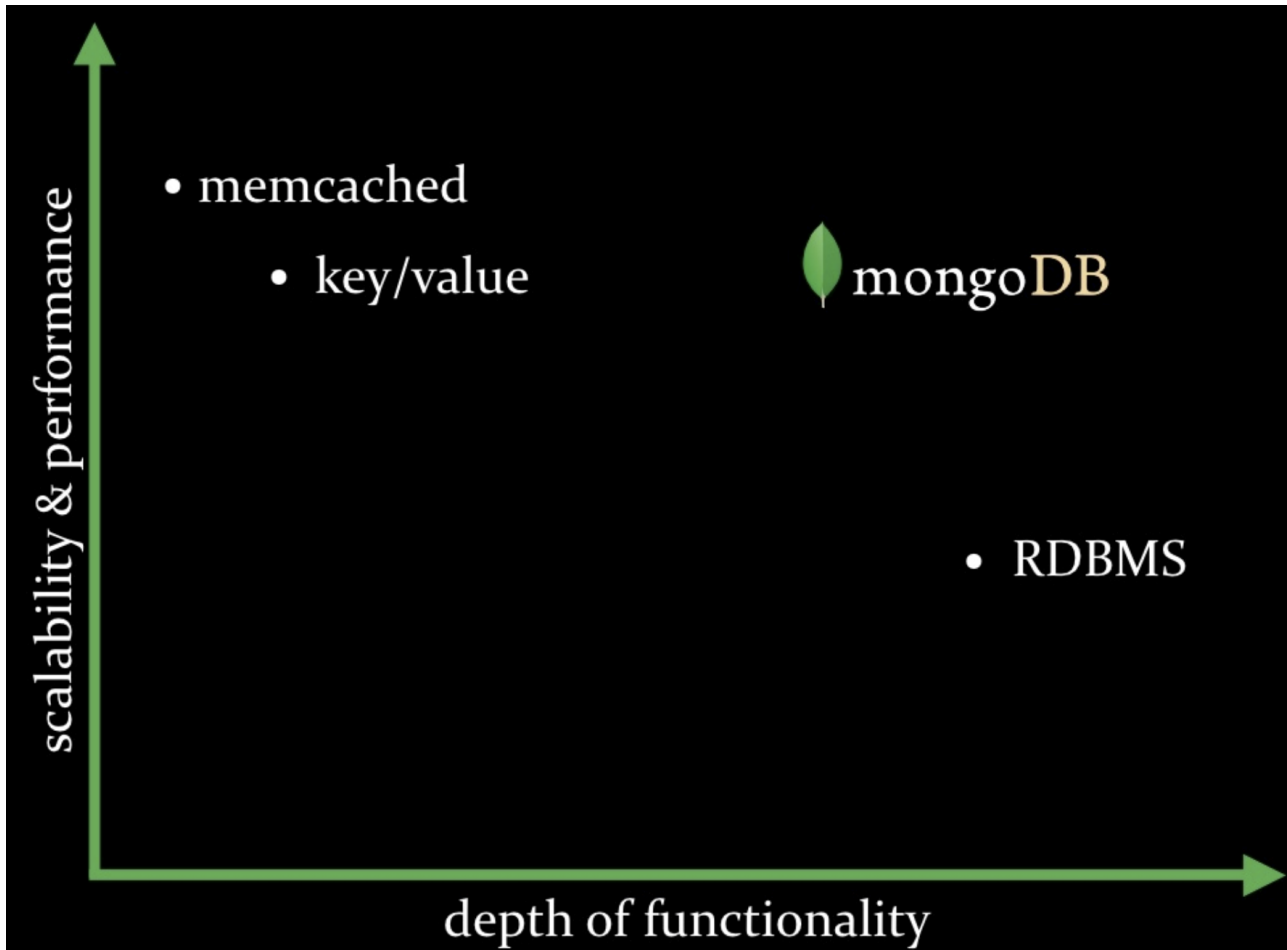
# Types of NoSQL

- Key Value Store
  - Asdf
- Document Store
  - MongoDB
- Column Store
  - Cassandra
- Graph Database
  - Neo4J

# MongoDB

- NoSQL DB
- Open source
- Document DB
- No schema mandatory
- Indexing
- Highly scalable
- MapReduce
- Easy Replication and sharding

# MongoDB position



# Getting started

- Web site: <https://www.mongodb.com/>
- Download: <https://www.mongodb.com/download-center>
- Different binaries: [https://www.mongodb.org/dl/win32/x86\\_64-2008plus-ssl](https://www.mongodb.org/dl/win32/x86_64-2008plus-ssl)
- We can download latest stable version:
















name	modified	size	md5	sig	sha1	sha256
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-v3.4-latest-signed.msi</a>	2018-06-08 19:41:25	159330816	<a href="#">md5</a>		<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-v3.4-latest.zip</a>	2018-06-08 19:41:20	189116760	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-v3.4-latest.zip</a>	2018-06-08 19:41:15	247077361	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-v3.6-latest.zip</a>	2018-06-08 18:58:40	248209864	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-v3.6-latest-signed.msi</a>	2018-06-08 18:58:34	192792064	<a href="#">md5</a>		<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-v3.6-latest.zip</a>	2018-06-08 18:58:26	306956817	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-4.0.0-rc4.zip</a>	2018-06-07 15:25:38	177274152	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc4-signed.msi</a>	2018-06-07 15:25:28	194460160	<a href="#">md5</a>		<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc4.zip</a>	2018-06-07 15:25:19	228693792	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-4.0.0-rc3.zip</a>	2018-06-06 15:12:42	177259827	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc3-signed.msi</a>	2018-06-06 15:12:37	194444288	<a href="#">md5</a>		<a href="#">sha1</a>	<a href="#">sha256</a>
<a href="#">win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc3.zip</a>	2018-06-06 15:12:30	228679154	<a href="#">md5</a>	<a href="#">sig</a>	<a href="#">sha1</a>	<a href="#">sha256</a>



# Getting started

- Files we get in archive
- Default
  - data folder: \Data\Db
  - port: 27017
- Options:
  - `mongod --help`
- Simplest usage:
  - `mongod`
  - `mongo`
- Better usage:
  - `mongod --dbpath c:\Data\Databases\MongoDB`
  - `mongo`



-  `bsondump`
-  `mongo`
-  `mongod`
-  `mongodump`
-  `mongoexport`
-  `mongofiles`
-  `mongoimport`
-  `mongooplog`
-  `mongoperf`
-  `mongorestore`
-  `mongos`
-  `mongostat`
-  `mongotop`

# Getting started

- Better usage:

- `mongod --dbpath c:\Data\Databases\MongoDB\db`
- `mongo`

- Best usage:

- `mongod -f c:\Data\Databases\MongoDB\mongod.conf`

- Config file:

```
1 dbpath=C:\Data\Databases\MongoDB\db  
2 logpath=C:\Data\Databases\MongoDB\mongo-server.log  
3 verbose=vvvv
```

# Getting started (Azure)

The screenshot displays the Microsoft Azure portal interface for creating a new Azure Cosmos DB account. The left sidebar shows the navigation menu with 'Azure Cosmos DB' highlighted in red. The main content area is divided into two panes. The left pane shows a list of existing Cosmos DB accounts with columns for 'NAZWA' and a search filter. The right pane is the 'New account' configuration form, which includes the following fields and options:

- ID:** A text input field with the placeholder 'Enter account ID' and a dropdown menu showing 'documents.azure.com'.
- API:** A dropdown menu set to 'MongoDB'.
- Subskrypcja:** A dropdown menu set to 'Visual Studio Enterprise'.
- Resource Group:** Radio buttons for 'Utwórz nowy' (selected) and 'Użyj istniejącego', with a dropdown menu set to 'Group'.
- Lokalizacja:** A dropdown menu set to 'Europa Zachodnia'.
- Optional Settings:** Checkboxes for 'Enable geo-redundancy' and 'Enable Multi Master', both currently unchecked.
- Multi Master Preview:** A section with a 'Sign up to preview today' link and a right-pointing arrow.
- Virtual networks:** A section with a 'Configure virtual networks' link and two buttons: 'Disabled' and 'Enabled'.

# Getting started (Azure)

The screenshot displays the MongoDB Data Explorer interface. The top bar shows the account name 'mongodatabase - Data Explorer' and 'Azure Cosmos DB account'. The left sidebar contains navigation options: Overview, Dziennik aktywności, Kontrola dostępu (IAM), Tagi, Diagnostowanie i rozwiązywanie..., Quick start, and Data Explorer. The main area is divided into three sections: a left-hand navigation tree for the 'MONGODB API' showing a 'library' collection with a 'books' sub-collection containing 'Documents', 'Scale & Settings', 'Stored Procedures', 'User Defined Functions', and 'Triggers'; a central document viewer showing a list of document IDs with the first one, '5b1803cefa26ae03c0b9e36c', selected; and a right-hand pane displaying the JSON structure of the selected document: 

```
1 {
2   "_id" : ObjectId("5b1803cefa26ae03c0b9e36c"),
3   "title" : "Mistrz i Małgorzata",
4   "author" : "Bułhakow"
5 }
```

# Shell

- mongo.exe
- JavaScript interpreter
- Multiline input is ok

```
> var hello = function() {  
... print("Hello World!");  
... }  
> hello();  
Hello World!
```

- Run external script
  - mongo C:\Data\Databases\MongoDB\booksCount.js
  - > load('C:/Data/Databases/MongoDB/booksCount.js')
- Non-interactive mode
  - mongo localhost/admin --eval  
"printjson(db.runCommand({logRotate:1}))"

# Storage

- Mongo talk with JSONs
  - JSONs can be easily imported and queried
- Documents stored in BSON
  - <http://bsonspec.org/>

BSON { 01010100  
11101011  
10101110  
01010101 }

BSON [*bee · sahn*], short for Binary JSON, is a binary-encoded serialization of JSON-like documents. Like JSON, BSON supports the embedding of documents and arrays within other documents and arrays. BSON also contains extensions that allow representation of data types that are not part of the JSON spec. For example, BSON has a Date type and a BinData type.

BSON can be compared to binary interchange formats, like Protocol Buffers. BSON is more "schema-less" than Protocol Buffers, which can give it an advantage in flexibility but also a slight disadvantage in space efficiency (BSON has overhead for field names within the serialized data).

BSON was designed to have the following three characteristics:

1 **Lightweight**

Keeping spatial overhead to a minimum is important for any data representation format, especially when used over the network.

2 **Traversable**

BSON is designed to be traversed easily. This is a vital property in its role as the primary data representation for MongoDB.

3 **Efficient**

Encoding data to BSON and decoding from BSON can be performed very quickly in most languages due to the use of C data types.

specification

implementations

FAQ

discussion

# Structure

- Structure hierarchy
  - Instance → Databases → Collections → Documents
- Collection optionally may have a schema
  - <https://docs.mongodb.com/manual/core/schema-validation/>
- Rule: every document must have a key
  - `_id`
    - Present in all documents
    - Unique across collection
    - Any type (except array)

# Manipulating data

- Basic
  - show dbs
  - use library
  - db
  - db.books.save({\_id:1, author:"Bułhakow", title:"Mistrz i Małgorzata"})
  - db.books.save({\_id:2, author:"Golden", title:"Wyznania gejszy"})
  - db.books.save({\_id:3, author:"Golding", title:"Władca much"})
  - db.books.find()
- ObjectId
  - db.books.save({author:"Rowling", title:"Harry Potter"})
  - ObjectId()
  - ObjectId().getTimestamp()
- Insert command
  - db.books.save({\_id:3, author:"Orwell", title:"Folwark zwierzęcy"})
  - db.books.save({\_id:3, author:"Golding", title:"Władca much"})
  - db.books.insert({\_id:3, author:"Orwell", title:"Folwark zwierzęcy"})
  - db.books.insert({\_id:4, author:"Orwell", title:"Folwark zwierzęcy", rating:8})
- More
  - <https://docs.mongodb.com/manual/tutorial/insert-documents/>



# Manipulating data

- Update problem with save
  - `> var b = db.books.findOne({_id:4})`
  - `> b.rating = b.rating+1;`
    - here someone else gets the book and modify rating
  - `> db.books.save(b);`
  
  - `> db.books.save({_id:4, author:"Orwell", title:"Folwark zwierzęcy", rokWydania:1945})`
  - -- and then
  - `> db.books.save(b);`

# Manipulating data

- Update command
  - `db.col.update(query, update, options)`
- Examples
  - `db.books.update({_id:4}, {$inc:{rating:1}});`
- Operators
  - `$inc{rating:1}`
  - `$set:{y:3}`
  - `$unset:{y:0}`
  - `$rename:{'rko': 'rok'}`
- More
  - <https://docs.mongodb.com/manual/tutorial/update-documents/>

# Manipulating data

- Delete

- `db.books.deleteMany({})`
- `db.inventory.deleteOne( { _id: 3 } )`

- More

- <https://docs.mongodb.com/manual/tutorial/remove-documents/>

# Querying data

- Query

- `db.col.find(query, projection)`
  - Projection: `{field:0|1, field:0|1, ...}` (all 0 or all 1)

- More

- <https://docs.mongodb.com/manual/tutorial/query-documents/>

- Extend our data

```
db.books.save({_id:4, author:"Orwell", title:"Folwark zwierzęcy", year:1945, rating:8, location:{room:4,segment:2}, catalogue:
```

```
[
  {number:"A01", available:true},
  {number:"A02", available:false, rentDate:'2018-01-01'},
  {number:"A03", available:true},
]
})
```

```
db.books.save({_id:5, author:"Steinbeck", title:"Grona gniewu", rating:7, location:{room:4,segment:3}, catalogue:
```

```
[
  {number:"A04", available:false, rentDate:'2018-01-02'},
  {number:"A05", available:false, rentDate:'2018-01-03'},
]
})
```

# Querying data

- Basic queries
  - `> db.books.find({_id:4});`
  - `> db.books.find({_id:4}, {_id:1});`
  - `> db.books.find({_id:4}, {_id:0});`
  - `> db.books.find({_id: {$gt:2}})`
  - `> db.books.find({_id: {$not:{$gt:2}}})`
  - `> db.books.find({_id: {$in:[1,2]}})`
  - `> db.books.find({_id: {$nin:[1,2]}})`
  - `> db.books.find({author:/^Gold/});`
  - `> db.books.find().count()`

# Querying data

## ■ Nested documents

- `> db.books.find({"location.room":4});`
- `> db.books.find({"catalogue.available":true}, {_id:1});`
- `> db.books.find({"catalogue.available":false}, {_id:1});`

## ■ Where

- `> db.books.find({$where: "this.author=='Golden' || this.title=='Władca much'"});`

## ■ Sorting

- `> db.books.find({}, {title:1}).sort({'catalogue.available':-1,title:1});`

## ■ Paging

- `> db.books.find({}, {_id:1}).sort({_id:1}).skip(2).limit(2);`

# Querying data

- Iterating cursor

- > var c = db.books.find({}, {title:1});

- > c.size()

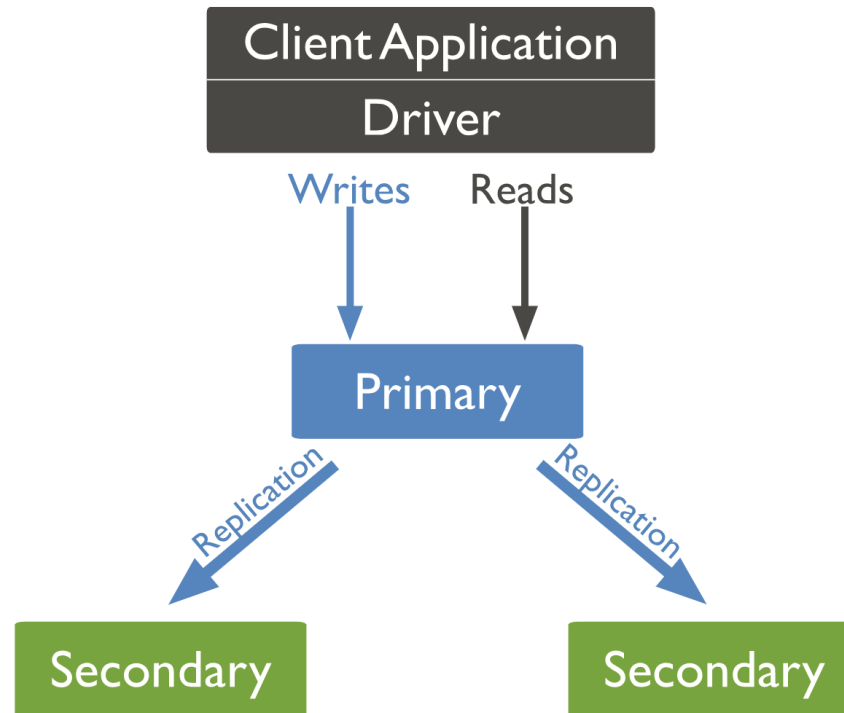
- > c.hasNext()

- > c.forEach(function(d){ print(d.title); })

- > c.hasNext()

# Replication

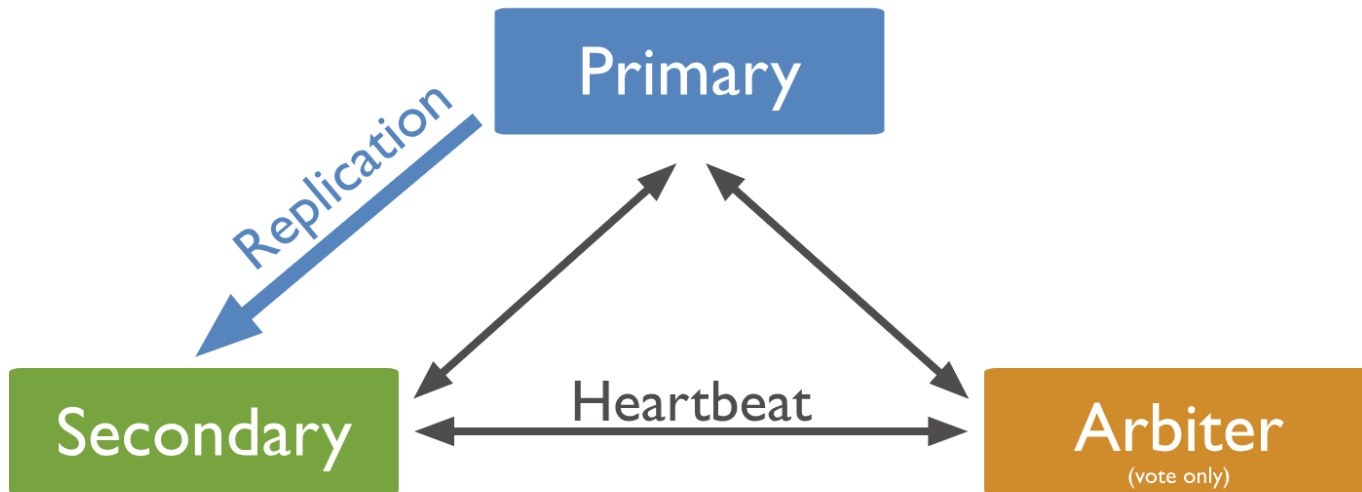
- Replica Set concept





# Replication

- Role of arbiter
  - Doesn't have data, can be weak node
  - Support voting for primary in case there are even number of nodes



# Replication

## ■ DEMO

### ■ Create folders

- c:\Data\Databases\MongoDB\db1
- c:\Data\Databases\MongoDB\db2
- c:\Data\Databases\MongoDB\db3

### ■ Run 3 instances

- start "A" mongod --dbpath c:\Data\Databases\MongoDB\db1 --port 10000 --replSet "demo"
- start "B" mongod --dbpath c:\Data\Databases\MongoDB\db2 --port 20000 --replSet "demo"
- start "C" mongod --dbpath c:\Data\Databases\MongoDB\db3 --port 30000 --replSet "demo"

### ■ Run shell

- mongo --port 10000

# Replication

## ■ DEMO

### ■ Create a configuration object

(more <https://docs.mongodb.com/manual/reference/replica-configuration/>)

- `var rsConfig={ _id: "demo", members: [{_id: 0, host: 'localhost:10000', priority: 10}, {_id: 1, host: 'localhost:20000'}, {_id: 2, host: 'localhost:30000', arbiterOnly: true}]};`

### ■ Let's take a look

- `rsConfig`

### ■ Initiate a cluster

- `rs.initiate(rsConfig)`

# Replication

- DEMO
  - Save something
    - `db.books.save({_id:1, title:"Mistrz i Małgorzata"})`
    - `db.books.find()`
  - Let's check second server
    - `mongo --port=20000`
    - `db.books.save({_id:2, title:"Wyznania gejszy"})`
    - `db.books.find();`
    - `db.setSlaveOk();`
    - `db.books.find();`

# Replication

- DEMO
  - Let's check replication
    - Kill PRIMARY
    - Check SECONDARY
  - Resurrect PRIMARY
    - `start "A" mongod --dbpath c:\Data\Databases\MongoDB\db1 --port 10000 --replSet "demo"`
  - Check again

# Client application

The screenshot shows the NuGet Package Manager interface for 'MongoDB.Driver'. The search results list several packages, with 'MongoDB.Driver' by MongoDB, Inc. (3,47M downloads) being the primary focus. The package details for 'MongoDB.Driver' show the latest stable version (2.6.1) and an 'Install' button. The 'Options' section is expanded, and the 'Description' section provides details about the package, including its version, author, license, and dependencies.

**MongoDB.Driver** by MongoDB, Inc., 3,47M downloads  
Official .NET driver for MongoDB.

**DCouple.Mongo** by <https://github.com/myles-mcdonnell>, 12.8K downloads  
Enables unit testing of applications that use MongoDB official driver.

**Bsynchro.DataAccess.Mongo.Core** by DataAccess.Mongo.Core, 287 downloads  
Exposing methods to execute on a Mongo database

**Bsynchro.DataAccess.Mongo.Abstract** by DataAccess.Mongo.Abstract, 152 d  
Exposing different methods interfaces to execute on a Mongo database

**Graphene.Mongo** by Boban Jose, 454 downloads  
Mongo connectors for Graphene.

**Repository.Mongo.Cqrs** by usame.esendir, 808 downloads  
CQRS pattern based on repository pattern of MongoDB

**Cqrs.Mongo** by Chinchilla Software, 53,5K downloads  
Use MongoDB as the read store and data store in CQRS.NET

**Cqrs.Ninject.Mongo** by Chinchilla Software, 52,3K downloads  
Use Ninject as your IoC container of choice with MongoDB for CQRS.NET

**Preview Changes**

Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.

**MongoDBClient**

**Installing:**

- DnsClient.1.0.7
- Microsoft.NETCore.Targets.1.1.0
- Microsoft.Win32.Primitives.4.3.0
- Microsoft.Win32.Registry.4.0.0
- MongoDB.Bson.2.6.1
- MongoDB.Driver.2.6.1
- MongoDB.Driver.Core.2.6.1
- runtime.native.System.4.3.0
- runtime.native.System.Net.Http.4.0.1
- runtime.native.System.Security.Cryptography.4.0.1
- System.Buffers.4.3.0
- System.Collections.4.3.0
- System.Collections.Concurrent.4.3.0
- System.Collections.NonGeneric.4.0.1
- System.Collections.Specialized.4.0.1
- System.ComponentModel.4.0.1
- System.ComponentModel.Primitives.4.1.0
- System.ComponentModel.TypeConverter.4.1.0
- System.Diagnostics.Debug.4.3.0

Do not show this again

OK Cancel

**MongoDB.Driver**

Version: Latest stable 2.6.1

Install

**Options**

**Description**

Official .NET driver for MongoDB.

**Version:** 2.6.1

**Author(s):** MongoDB, Inc.

**License:** <http://www.apache.org/licenses/LICENSE-2.0>

**Date published:** Thursday, May 17, 2018 (5/17/2018)

**Project URL:** <http://www.mongodb.org/display/DOCS/CSharp+Language+Center>

**Report Abuse:** <https://www.nuget.org/packages/MongoDB.Driver/2.6.1/ReportAbuse>

**Tags:** mongo, mongodb, nosql

**Dependencies**

- .NETFramework,Version=v4.5**
  - MongoDB.Bson (>= 2.6.1)
  - MongoDB.Driver.Core (>= 2.6.1)
- .NETStandard,Version=v1.5**
  - MongoDB.Bson (>= 2.6.1)
  - NETStandard.Library (>= 1.6.1)
  - System.ComponentModel.TypeConverter (>= 4.1.0)
  - MongoDB.Driver.Core (>= 2.6.1)
  - System.Linq.Queryable (>= 4.0.1)

# Other interesting stuff

- Indexing
- Aggregation
  - <https://docs.mongodb.com/manual/aggregation/>
- Views
- MapReduce
- Capped Collections
- Geo

# References

- Introductions
  - <https://www.mongodb.com/nosql-explained>
  - <https://www.slideshare.net/Leesy/an-introduction-to-nosql-mongodb/>
  - <https://www.slideshare.net/mdirolf/introduction-to-mongodb>
  - <https://www.slideshare.net/mongodb>
  - <https://www.slideshare.net/drumwurzels/intro-to-mongodb/>
  - <https://www.toptal.com/database/the-definitive-guide-to-nosql-databases>
- Documentation
  - <https://docs.mongodb.com/manual/crud/>
  - <https://www.tutorialspoint.com/mongodb/index.htm>
- Client C# application
  - <https://docs.mongodb.com/ecosystem/drivers/csharp/>
  - <http://mongodb.github.io/mongo-csharp-driver/2.2/reference/driver/>
  - <https://blog.oz-code.com/how-to-mongodb-in-c-part-1/>
  - <https://code.visualstudio.com/docs/azure/mongodb>
- Cloud Hosting
  - <https://mlab.com/>