Telerik Academy

MongoDB and using MongoDB with .NET

Welcome to the JSON-stores world

Databases

Telerik Software Academy
http://academy.telerik.com

Telerik Academy

Table of Contents

- MongoDB overview
  - Installation and drivers
  - Structure and documents
- Hosting locally MongoDB
- DB Viewers
  - Console CLI, UMongo, Mongo VUE
- Queries

Telerik Academy

MongoDB

MongoDB

- MongoDB is an open-source document store database
  - Save JSON-style objects with dynamic schemas
  - Support for indices
  - Has document-based queries
    - CRUD operations

Using MongoDB

- Download MongoDB from the official web site:
  - https://www.mongodb.org/downloads
  - Installers for all major platforms
- When installed, MongoDB needs a driver to be usable with a specific platform
  - One for .NET, another for Node.JS, etc…
- To install the MongoDB driver for .NET run in Visual Studio's Package Manager Console

```
Install-Package mongocsharpdriver
```

Telerik Academy

Installing MongoDB

Live Demo

Telerik Academy

Starting MongoDB Instance

- Once installed, the MongoDB must be started
  - Go to installation folder and run `mongod`

```
$ cd path/to/mondodb/installation/folder
$ mondgod
```

  - Or add the path to mongod.exe to the %PATH%
    - And simply run

```
$ mondgod
```

- When run, the MongoDB can be used from .NET, Node.js, etc…

Telerik Academy

Connecting to MongoDB with .NET and C#

- The C# driver for MongoDB provides a .NET SDK for working with MongoDB
  - Init a MongoClient with a connection string

```
MongoClient client = new MongoClient("mongodb://localhost);
```

  - Host the MongoDB instance

```
MongoServer server = client.GetServer();
```

  - Connect to a database
    - If the db does not exists, it will be createad

```
MongoDatabase db = server.GetDatabase("logger");
```

    - Work with the database

Telerik Academy

Using the MongoDB with .NET and C#

Live Demo

Telerik Academy

MongoDB Viewers

Telerik Academy

MongoDB Viewers

- MongoDB is an open-source DB system
  - So there are many available viewers
- Some, but not all are:
  - MongoDB CLI
    - Comes with installation of MongoDB
    - Execute queries inside the CMD/Terminal
  - MongoVUE & UMongo
    - Provide UI to view, edit are remove DB documents

Telerik Academy

MongoDB Viewers

Live Demo

Telerik Academy

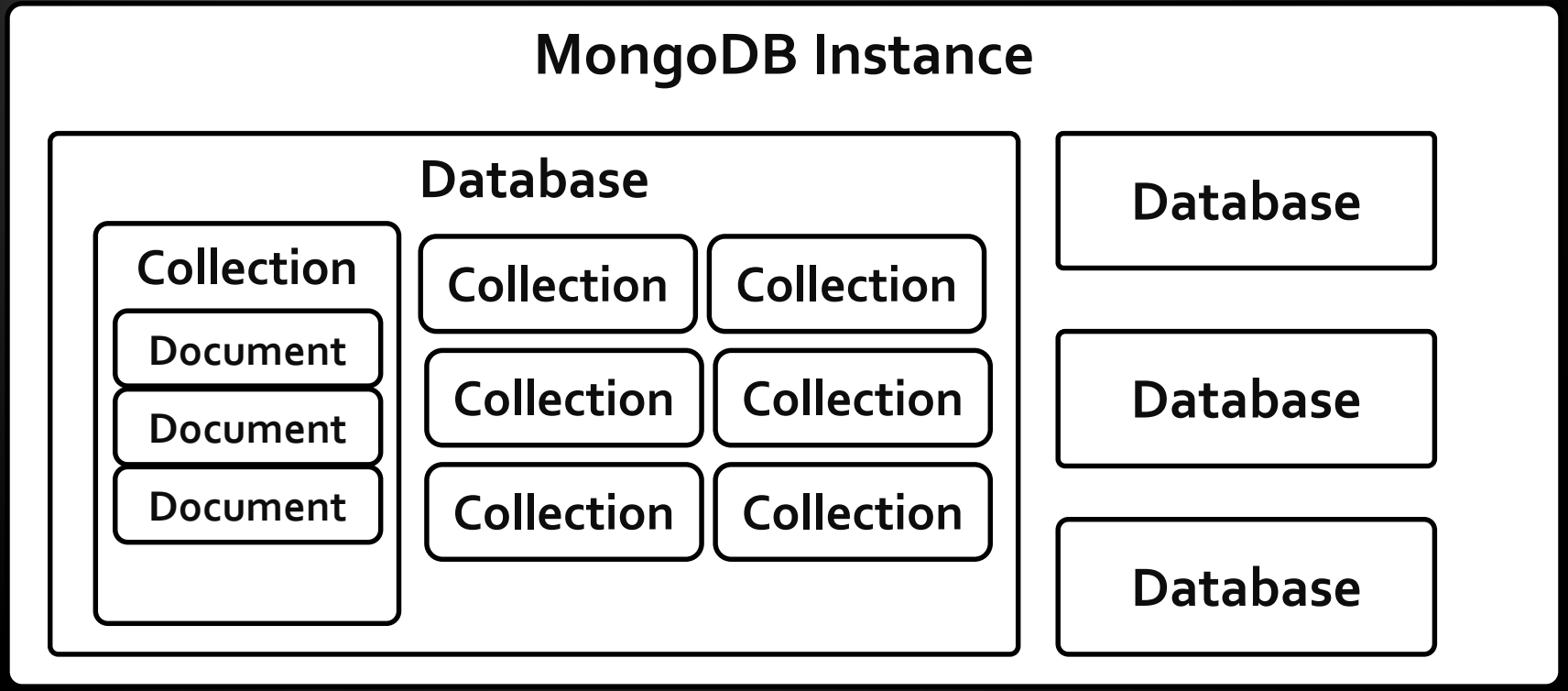
MongoDB Structure

How is the data structured in MongoDB

Telerik Academy

MongoDB Structure

- A MongoDB instance can have many databases
  - A database can have many collections
    - A collection can have many documents



Telerik Academy

MongoDB Documents

- Documents in MongoDB are JSON objects

Name: Databases

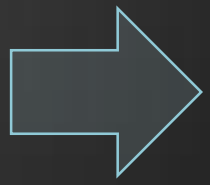
Students: 250

Titles:

- Intro to SQL
- NoSQL databases
- Entity Framework

Evaluation:

- Homework: 10
- Exam: 60
- Teamwork: 20
- Attendance: 10



```
{"name": "Databases",
 "students": 250,
 "titles": [
    "Intro to SQL",
    "NoSQL databases",
    "Entity Framework"
 ],
 "evaluation": {
    "homework": 10,
    "exam": 60,
    "teamwork": 20,
    "attendance": 10
}
```

Telerik Academy

MongoDB Documents (2)

- MongoDB documents many types of values
  - Numbers, Booleans

```
17, 3.14, true/false
```

  - Strings, ObjectIds

```
"Doncho Minkov"
```

  - Dates

```
2014-09-01T14:58:48.126Z
```

  - Objects
    - Also know as nested documents

```
{
 "username": "Minkov",
 "accessLvl": 2
}
```

  - Arrays

```
['bananas', 'oranges']
```

Telerik Academy

.NET Driver

Telerik Academy

The .NET MongoDB SDK

- The .NET driver provides a SDK to work with MongoDB databases
  - Supports CRUD operations
    - create, read, update and delete
  - Can create indexes
  - Provides LINQ-like functionality
    - That is executed over the database using the native query commands
  - Does not have transactions
    - Still has atomicity on single document

Telerik Academy

CRUD Operations with MongoDB and .NET

Insert, Remove, Update, Get

Telerik Academy

CRUD Over MongoDB with .NET

- A collection must be created/fetched
  - Before performing any operations

```
var logsCollection = db.GetCollection<Log>("logs");
```

  - All operations are done over this collection:
    - Fetch documents

```
logsCollection.FindAll();
logs.FindOneById(id);
```

    - Update

```
logsCollection.Update(
            query, update);
```

    - Insert

```
logsCollection.Insert(log);
```

    - Remove

```
logs.Remove(removeQuery);
```

Telerik Academy

CRUD Operations with MongoDB and .NET

Live Demo

Telerik Academy

MongoDB Queries

Perform Operations over the Database

Telerik Academy

MongoDB Queries

- MongoDB supports document-based queries
  - They are executed on the server
  - They are the equivalent of SQL for RDBMS
- Queries can be made in two ways:
  - Using native-like MongoDB queries
  - Using LINQ-like queries

Telerik Academy

Native-like MongoDB Queries

- Find queries are created using similar to the native MongoDB document queries:
  - Find recent logs:

```
IMongoQuery findNewLogsQuery =
          Query.And(Query.GT("LogDate", date));
var logs = logsCollection.Find(findNewLogsQuery);
```

  - Remove logs, older than a day:

```
IMongoQuery findOldLogsQuery = Query.And(
              Query.LT("LogDate", Date.now.addDays(-1)));
logsCollection.Remove(findOldLogsQuery);
```

Telerik Academy

Native-like MongoDB Queries

Live Demo

LINQ-like Queries

- LINQ-like Queries:

```
var findRecentBugsQuery =
  Query<Log>.Where(log => log.LogType.Type == "bug" &&
                          log.LogDate > date);
var logs = logsCollection.Find(findRecentBugsQuery)
                .Select(log => log.Text);
logsCollection.Update
```

```
var findOldPendingBugsQuery = Query<Log>.Where(
  log => log.LogDate < DateTime.Now.AddDays(-10) &&
         log.LogType.Type == "bug" &&
         log.LogType.State == "pending");
```

Telerik Academy

LINQ-like MongoDB Queries

Live Demo

Telerik Academy

LINQ-to-MongoDB

- The MongoDB C# driver supports part of LINQ over MongoDB collections
  - Only LINQ queries that can be translated to an equivalent MongoDB query are supported
    - Others throw runtime exception and the error message will indicate which part of the query wasn't supported

```
var logs =
  from log in db.GetCollection<Log>("Logs").AsQueryable<Log>()
  where log.LogType.Type == "ticket" &&
        log.LogType.State == "pending"
select log;
```

Telerik Academy

LINQ-to-MongoDB

Live Demo

Telerik Academy

MongoDB and MongoDB with .NEt

http://academy.telerik.com

Telerik Academy

Homework

1. Research how to create a MongoDB database in Mongolab(http://mongolab.com)
2. Create a chat database in MongoLab
   - The database should keep messages
   - Each message has a text, date and an embedded field user
   - Users have username

Telerik Academy

Homework (2)

3. Create a Chat client, using the Chat MongoDB database
   - When the client starts, it asks for username
     - Without password
   - Logged-in users can see
     - All posts, since they have logged in
     - History of all posts
   - Logged-in users can post message
   - Create a simple WPF application for the client