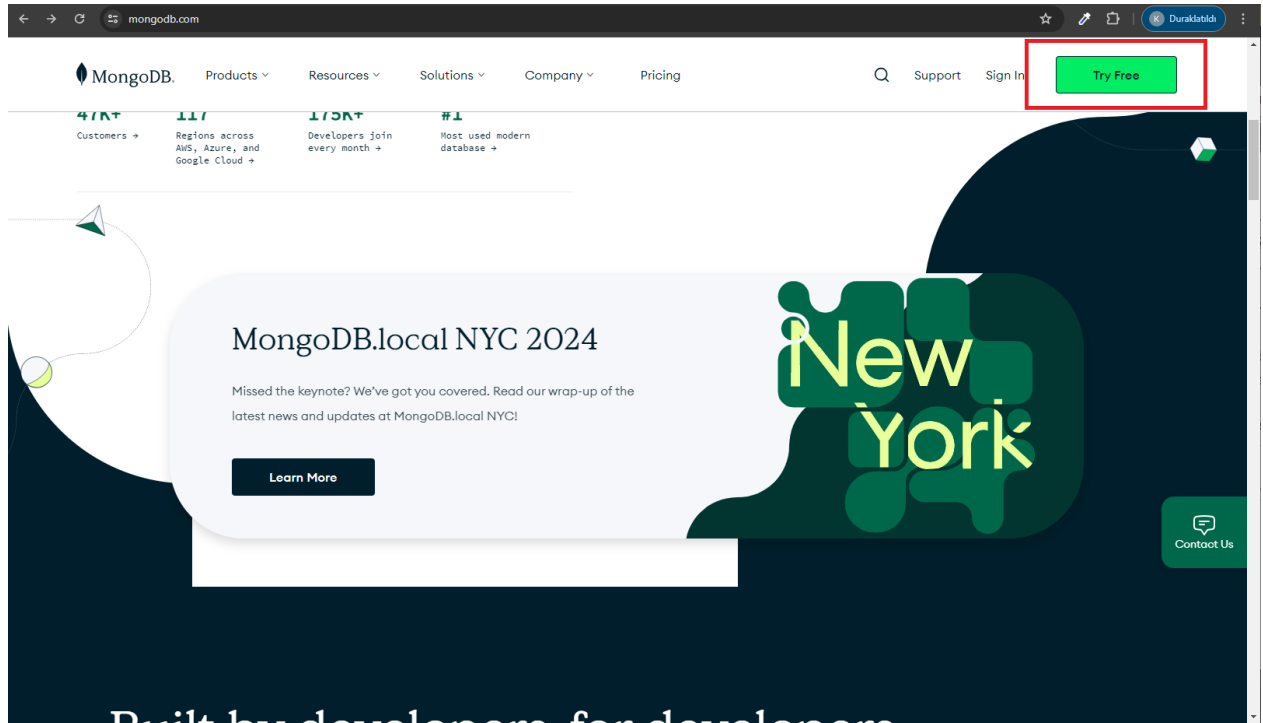


Kadir ÇAPKIN 21360859023

NodeJS ile Web programlama Raporu Hafta – 11

Task Manager -1

Bugün MongoDB girişi hakkında bilgiler verilecek. MongoDB, NoSQL bir veri tabanıdır. NoSQL veri tabanları ilişkisel veri tabanlarının alternatifi olarak kullanılır. Yapısal bir veri tabanı değildir. Veriler JSON benzeri belgeler halinde tutulur. Yapısal veri tabanında olduğu gibi satırlar ve sütunlardan ziyade field denilen alanlar mevcuttur. Bu sebeple veri yapılarının sık sık değiştiği veya karmaşık ilişkilerin bulunduğu durumlarda kullanımı avantajlıdır. MongoDB esnek yapısı ve farklı özellikleri sayesinde belirli projelerde işlevsel olabilirken, belirli projelerde ise diğer yapısal veri tabanlarını kullanmak daha işlevseldir. MongoDB diğer veri tabanlarındaki gibi belirli bir dosyayı yükleyip bilgisayarımıza kurabileceğimiz bir uygulama değildir. Cloud tabanlı çalışır. Bu sebeple oluşturduğumuz veri tabanlarını herhangi bir makinede kolay bir şekilde görüntüleyebiliriz. Kullanımına geçmek gerekirse Cloud tabanlı olduğu için MongoDB Atlas üzerinden çalıştırmamız gerekiyor. Google MongoDB yazarak <https://www.mongodb.com/> sitesine giriyoruz. Bu sitede öncelikle kaydolmamız gerekiyor.



Try Free butonuna tıkladığımızda bizi kayıt sayfasına yönlendiriyor. Kişisel bilgilerimizi girdikten sonra mailimize gelen onay linkine tıklayarak üyeliğimizi onayladığımızda karşımıza MongoDB'yi neden kullandığımız, amacımız ve hangi dille beraber kullanacağımız

hakkında belirli sorular geliyor. Bu soruları kendimize göre yanıtladıktan sonra karşımıza şu ekran çıkıyor.

Deploy your cluster

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

M10

\$0.09/hour

For production applications with sophisticated workload requirements.

STORAGE

10 GB

RAM

2 GB

vCPU

2 vCPUs

Serverless

For application development and testing, or workloads with variable traffic.

STORAGE

Up to 1 TB

RAM

Auto-scale

vCPU

Auto-scale

M0

Free

For learning and exploring MongoDB in a cloud environment.

STORAGE

612 MB

RAM

Shared

vCPU

Shared

✓ Free forever!

Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Name

You cannot change the name once the cluster is created.

☒ Automate security setup ⓘ

☒ Preload sample dataset ⓘ

Provider

Region

Bahrain (me-south-1) ★

★ Recommended ⓘ

Low carbon emissions ⓘ

Burada ücretsiz seçeneği seçtikten sonra kendi seçtiğimiz bir cluster name giriyoruz. Cluster veri tabanı hizmetlerinin sağlandığı bir altyapıdır ve veri tabanlarını barındırmak için kullanılır. Kendi özel veri tabanlarımızı bu cluster altında oluşturacağız. Hemen altındaki bulunan 2.seçeneği preload sample dataset seçeneğini eğer seçersek bize hazır bir veri seti veriyor. Bu şekilde bilgilerimizi girdikten sonra create deployment seçeneğine tıkladığımızda karşımıza şu ekran geliyor.

BURSA TEKNİK ÜNİVERSİTESİ > PROJECT 0

Overview

Clusters

Create cluster ...

Cluster0

Connect

Edit configuration

Data Size: 27.02 MB

Browse collections →

View monitoring →

+ Add Tag

Connect seçeneğine tıkladığımızda bize bir database user oluşturmamızı istiyor.

Connect to Cluster0

1 Set up connection security 2 Choose a connection method 3 Connect

You need to secure your MongoDB Atlas cluster before you can use it. Set which users and IP addresses can access your cluster now. [Read more](#)

1. Add a connection IP address

✓ Your current IP address (176.54.255.191) has been added to enable local connectivity. Add another later in [Network Access](#).

2. Create a database user

This first user will have [atlasAdmin](#) permissions for this project.

i You'll need your database user's credentials in the next step. Copy the database user password.

Username

Password [HIDE](#) [Copy](#)

[Create Database User](#)

[Close](#) [Choose a connection method](#)

Kadir Çapkın (kadir55442@hotmail.com) oturum açtı

Burada username ve password bilgilerimizi girdikten sonra kaybetmemiz için bir yere kaydedebiliriz. Tüm bu işlemleri yaptıktan sonra vscode açıp Task-Manager Adlı yeni bir klasör altında mongodb.js adlı dosyamızı oluşturuyoruz. Ardından ilk yapmamız gereken şey terminalde `npm init -y` yazarak package.json dosyamıza dahil etmek ve ikinci yapmamız gereken şey ise `npm install mongodb` yazarak mongodb modülünü projemize dahil etmektir. Bunları yaptıktan sonra mongodb.js dosyamızın veri tabanı bağlantısını gerçekleştirmek için siteye dönüp

Clusters

Find a database deployment...

Edit

✓ Sample dataset successfully loaded. Access it in Collections or by connecting with the MongoDB Shell.

Cluster0

Connect

View Monitoring

Browse Collections

...

Visualize Your Data

Build dashboards and charts, and embed them in your apps with MongoDB Charts.

Dismiss

Explore Charts

R 0
W 0
Last 6 minutes
40.4/s

Connections 0
Last 11 minutes
4.0

In 304.4 KB/s
Out 0.0 B/s
Last 6 minutes
304.4 KB/s

VERSION

7.0.8

REGION

AWS / Bahrain (me-south-1)

CLUSTER TIER

M0 Sandbox (General)

TYPE

Replica Set - 3 nodes

BACKUPS

Inactive

LINKED APP SERVICES

None Linked

ATLAS SQL

Connect

ATLAS SEARCH

Create Index

+ Add Tag

Connect seçeneğine tıkladıktan sonra karşımıza şu ekran geliyor.

Connect to Cluster0

X

1
Set up connection security

2
Choose a connection method

3
Connect

Connect to your application



Drivers

Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)

>

Access your data through tools



Compass

Explore, modify, and visualize your data with MongoDB's GUI

>



Shell

Quickly add & update data using MongoDB's Javascript command-line interface

>



MongoDB for VS Code

Work with your data in MongoDB directly from your VS Code environment

>



Atlas SQL

Easily connect SQL tools to Atlas for data analysis and visualization

>

Go Back

Close

Drivers seçeneğine tıkladıktan sonra karşımıza gelen ekranda bize bir bağlantı kodu veriyor.

3. Add your connection string into your application code

 View full code sample

```
const { MongoClient, ServerApiVersion } = require('mongodb');
const uri = "mongodb+srv://kadircapkin:<password>@cluster0.pguczr6.mongodb.net/?retryW

// Create a MongoClient with a MongoClientOptions object to set the Stable API version
const client = new MongoClient(uri, {
  serverApi: {
    version: ServerApiVersion.v1,
    strict: true,
    deprecationErrors: true,
  }
});

async function run() {
  try {
    // Connect the client to the server (optional starting in v4.7)
    await client.connect();
    // Send a ping to confirm a successful connection
    await client.db("admin").command({ ping: 1 });
    console.log("Pinged your deployment. You successfully connected to MongoDB!");
  } finally {
    // Ensures that the client will close when you finish/error
    await client.close();
  }
}
run().catch(console.dir);
```

Bu bağlantı kodunda bulunan uri değişkenindeki kadircapkin:<password> kısmındaki <password> kısmını silip kendi belirlediğimiz cluster şifremizi yazıyoruz. Ardından bu kodu alıp mongodb.js dosyamıza yapıştırıyoruz. Yapıştırdıktan sonra bir önceki cümlede söylediğim gibi password kısmına kendi şifremizi yazmayı unutmuyoruz. Yoksa authentication hatası verecektir. Dosyamızı kaydettikten sonra terminale node mongodb.js yazarak uygulamamızı çalıştırıyoruz. Karşımıza bu şekilde bir görüntü çıkıyor.

```
JS mongodb.js M x .env M
Task-Manager > JS mongodb.js > run
1 const { MongoClient, ServerApiVersion } = require('mongodb');
2 require('dotenv').config();
3 const password = process.env.MONGODB_PASSWORD;
4 const uri = `mongodb+srv://kadircaPKinn:${password}@cluster0.pguczr6.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0`;
5
6 // Create a MongoClient with a MongoClientOptions object to set the Stable API version
7 const client = new MongoClient(uri, {
8   serverApi: {
9     version: ServerApiVersion.v1,
10    strict: true,
11    deprecationErrors: true,
12  }
13 });
14
15 async function run() {
16   try {
17     // Connect the client to the server (optional starting in v4.7)
18     await client.connect();
19     // Send a ping to confirm a successful connection
20     await client.db("admin").command({ ping: 1 });
21     console.log("Pinged your deployment. You successfully connected to MongoDB!");
22   } finally {
23     // Ensures that the client will close when you finish/error
24     await client.close();
25   }
26 }
27 run().catch(console.dir);
28
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\Task-Manager> node mongodb.js
Pinged your deployment. You successfully connected to MongoDB!
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\Task-Manager>
```

Burada yazdığımın farklı olarak kendi MongoDB şifremini paylaşmamak için .env dosyası altında yazıp modülü uygulamaya dahil edip eriştim. Şimdi oluşturmak istediğimiz veri tabanı ismini bir değişkende saklıyoruz. Daha sonra client üzerinden parametre olarak oluşturduğumuz db ismini verip db.collection('users').insertOne dediğimizde oluşturduğumuz veri tabanı altındaki users collection'a tek bir veri eklemiş oluyoruz.

```
const db = client.db(databaseName)
await db.collection('users').insertOne({ // tek veri ekleme
  name: "Mert",
  age : 18
}); // tek kayıt*/
```

Bu kodu çalıştırdıktan sonra sitemize dönüp cluster altındaki browse collection butonuna tıklarsak karşımıza şu ekran çıkacaktır.

Clusters

[Edit Config](#)[+ Create](#)**Cluster0**[Connect](#)[View Monitoring](#)[Browse Collections](#)

...

Visualize Your Data

Build dashboards and charts, and embed them in your apps with MongoDB Charts.

[Dismiss](#)[Explore Charts](#)

R 0.06
W 0.003
Last 41 minutes



Connections 12.0
Last 41 minutes



In 197.7 B/s
Out 1.5 KB/s
Last 41 minutes



VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES
7.0.8	AWS / Bahrain (me-south-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked

[+ Add Tag](#)

Atlas

BURSA TEKNİK ÜNİVERSİTESİ > PROJECT 0 > DATABASES

Cluster0

VERSION: 7.0.8, REGION: AWS Bahrain (me-south-1)

Overview, Real Time, Metrics, Collections, Atlas Search, Performance Advisor, Online Archive, Cmd Line Tools

DATABASES: 2, COLLECTIONS: 7

+ Create Database

Search Namespaces

sample_mflix, task-manager, users

task-manager.users

STORAGE SIZE: 20KB, LOGICAL DATA SIZE: 46B, TOTAL DOCUMENTS: 1, INDEXES TOTAL SIZE: 20KB

Find, Indexes, Schema Anti-Patterns, Aggregation, Search Indexes

Generate queries from natural language in ComposeIt

INSERT DOCUMENT

Filter: Type a query: { field: 'value' }

Reset, Apply, Options

QUERY RESULTS: 1-1 OF 1

```
{
  "_id": "ObjectId('663bc9a5ca0de1e1e32')",
  "name": "Heri",
  "age": 18
}
```

Burada görüldüğü gibi sample db'si başta bize sorduğu veri setidir. Hemen altındaki task-manager ise benim oluşturmuş olduğum veri tabanıdır. Users'a tıkladığımızda ise az

önce eklediğimiz veriyi bize gösterir. Sadece tekli veri eklemek yerine çoklu veriler de ekleyebiliriz. Bunun için ise şu kodu yazıyoruz.

```
await db.collection('users').insertMany([ // çoklu veri ekleme
  {
    name: "test1",
    age: 20
  }, {
    name: "test2",
    age: 30
  }
])
```

Bu sayede users collection'ı altında iki tane test verisi ekledik. Tekrar kodu çalıştırıp sitemize dönersek eklendiğini görebiliriz.

task-manager.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 140B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' } Reset Apply Options

QUERY RESULTS: 1-3 OF 3

```
{
  "_id": ObjectId('663bc93a51caa6de1e1e8e32'),
  "name": "Mert",
  "age": 18
}
```

```
{
  "_id": ObjectId('663bcb3ccbd455b34efcb701'),
  "name": "test1",
  "age": 20
}
```

```
{
  "_id": ObjectId('663bcb3ccbd455b34efcb702'),
  "name": "test2",
  "age": 30
}
```

Şimdi veri tabanındaki veriyi mevcut uygulamamıza getirmek içinse şu kodu kullanıyoruz.

```
const db = client.db(databaseName)

await db.collection('users').findOne({name: "Mert"}).then((data) => {
  console.log(data.age);
}).catch(error => {
  console.log(error);
}) // Tek bir değer döndürür.*/
```

Burada hangi collection üzerinden veri çekmek istediğimizi yazıyoruz ardından findOne'a parametre olarak bir nesne gönderiyoruz. Eğer varsa bize bir data döndürüyor. Eğer birden

fazla aynı koşulu sağlayan veri varsa ilk bulduğunu döndürüyor. Bunu yaptıktan sonra console üzerinde gelen data nesnesinin age attribute'ına ulaşıyoruz ve konsola yazdırıyoruz.

```
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\Task-Manager> node mongodb.js
Pinged your deployment. You successfully connected to MongoDB!
18
```

Bu veriler eklenirken diğer veri tabanlarında olduğu gibi id'ler tanımlamadık. Bu id'leri MongoDB kendisi tanımlıyor. Bu id'lere ve özelliklerine erişmek içinse eklediğimiz MongoDB modülüne ObjectId sınıfını ekliyoruz.

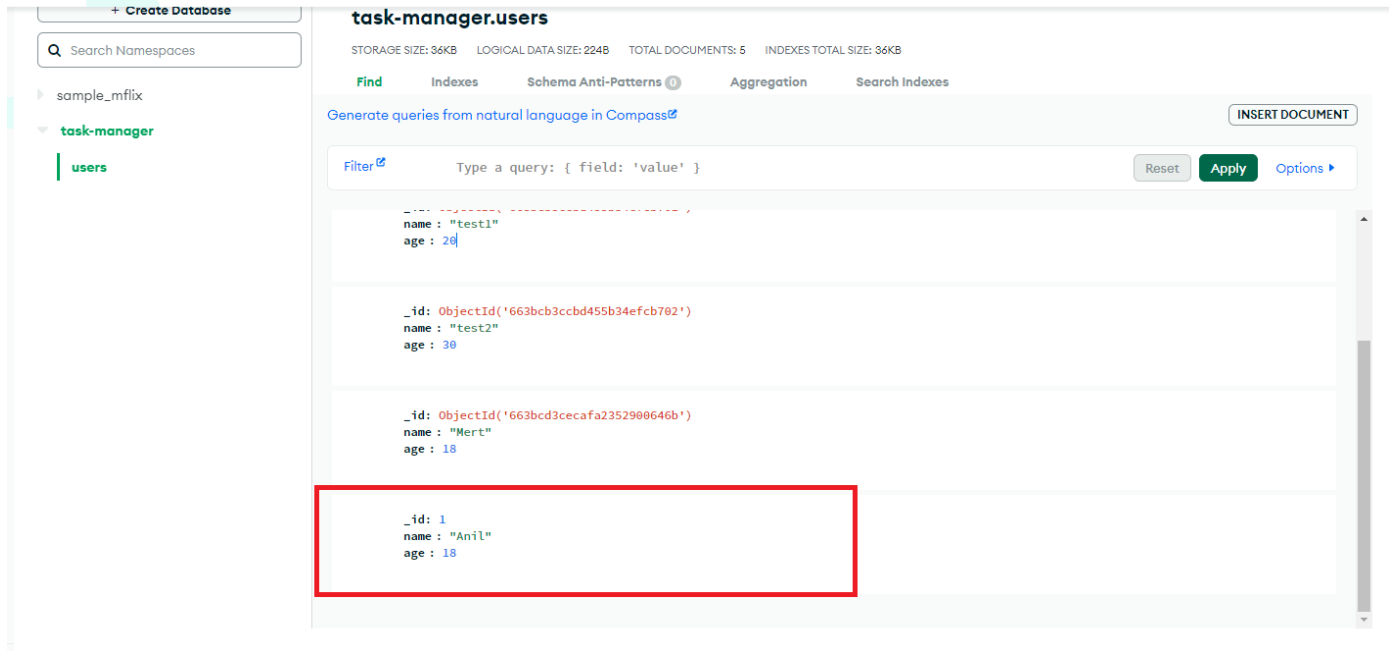
```
const id = new ObjectId();
console.log(id)
console.log(id.getTimestamp()) // tarih ve zaman bilgisi
console.log(id.id.length)
console.log(id.toHexString.length)

await db.collection('users').insertOne({
  _id:1,
  name: "Anil",
  age : 18
});
```

Bu sınıfı ekledikten sonra bu sınıf üzerinden bir nesne oluşturuyoruz ve bu nesnelerin belirli attribute değerlerine ulaşabiliyoruz. Aynı zamanda nesnemizi oluştururken de default bir id oluşturmak yerine kendimiz şu an test amaçlı 1 verebiliriz. Kodu çalıştırdığımızda karşımıza şu çıktı gelir.

```
PS C:\Users\kadir\Desktop\Programlama\NodeJS\Okul\Task-Manager> node mongodb.js
Pinged your deployment. You successfully connected to MongoDB!
new ObjectId('663bcd7363ae2776cefe5ab2')
2024-05-08T19:07:31.000Z
12
0
```

Sitemize dönüp eklenmiş mi diye kontrol edersek karşımıza şu çıktı gelir.



Görüldüğü gibi kendi elimizle de id bilgisini girebiliyoruz. Tüm bunları yaptıktan sonra Task-Manager uygulamamız için task collection'ı oluşturup altına görevler verilerimizi ekleyelim.

```
await db.collection('tasks').insertMany([{\n  description:"Odani temizle",\n  completed:true\n},{\n  description:"Cicekleri sula",\n  completed:false\n},{\n  description:"Odevlerini yap",\n  completed:false\n}])
```

Kodu çalıştırıp siteye dönüp kontrol edersek karşımıza gelecek ekran görüntüsü şu şekildedir.

Navigation: Data Services, App Services, Charts

task-manager.tasks

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 197B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

INSERT DOCUMENT

Filter Type a query: { field: 'value' } Reset Apply Options

QUERY RESULTS: 1-3 OF 3

<pre>_id: ObjectId('663bcea8f512b674d68c7615') description: "Odani temizle" completed: true</pre>
<pre>_id: ObjectId('663bcea8f513b674d68c7616') description: "Çiçekleri sula" completed: false</pre>
<pre>_id: ObjectId('663bcea8f513b674d68c7617') description: "Odevlerini yap" completed: false</pre>

Görüldüğü gibi girilen görev verileri başarılı bir şekilde eklenmiş.