

מבוא לשיעור הבא

Mongoose

TomerBu

מה Mongoose מאפשרת:

מערכת לניהול דטה-בייס

הגדרה מראש של סכמה

תכנות מונחה עצמים

מתודות פשוטות לשמירה/עדכון/מחיקה/קריאה

בשקפים הבאים:

(1) התחברות

(2) יצירת סכמה

(3) יצירת מודל

(4) יצירת מופע

(5) שמירה של אובייקט בדטה-בייס

התקנה:

```
npm i mongoose
```



Nearly Perfected Mystique

npm



Search packages

mongoose 

תמיכה מובנית בTypescript

8.0.0 • **Public** • Published 2 days ago



Readme



Code

Beta

mongoose

elegant **mongodb** object modeling for **node.js**

read the docs

discover plugins



Version 8.0.0



Let's face it, **writing MongoDB validation, casting and business logic boilerplate is a drag**. That's why we wrote Mongoose.

כל 5 השלבים יחד:

```
const mongoose = require('mongoose');
mongoose.connect('mongodb://127.0.0.1:27017/test');

const Cat = mongoose.model('Cat', { name: String });

const kitty = new Cat({ name: 'Zildjian' });
kitty.save().then(() => console.log('meow'));
```

נרצה לעבוד עם מונגוס - כולל ולידציות וסכמה.

התחברות לשרת:

כתובת

פורט

שם הדטה-בייס

```
export const connect = async () => {  
  await mongoose.connect("mongodb://127.0.0.1:27017/test");  
  console.log("Database Connected");  
};
```

דוגמא לסכמה בMongoose

```
import mongoose from 'mongoose';
const { Schema } = mongoose;

const blogSchema = new Schema({
  → title: String, // String is shorthand for {type: String}
  author: String,
  body: String,
  comments: [{ body: String, date: Date }],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs: Number
  }
});
```

התחברות, סכמה ומודל

```
import mongoose, { Schema } from "mongoose";
```

```
// connection:
```

```
export const connect = async () => {  
  await mongoose.connect("mongodb://127.0.0.1:27017/test");  
  console.log("Database Connected");  
};
```

התחברות

1

```
// Schema Definition:
```

```
export const personSchema = new Schema({  
  idNumber: Number,  
  firstName: String,  
  lastName: String,  
  image: String, ←  
  address: String,  
  phone: String,  
});
```

הגדרת סכמה
(מבנה)

2

הטיפוסים באות גדולה

```
// Model Definition ~ Person class:
```

```
//Mongoose generates a class with CRUD methods
```

```
export const Person = model("Person", personSchema);
```

מודל

3

הספרייה יוצרת עבורנו מחלקה עם מתודות CRUD

בפרוייקט - נפצל את הconnection לקובץ נפרד, את הסכמה לקובץ נפרד ואת המודל לקובץ נפרד

יצירת מופע של המודל ושמירה

4

```
// router POST /api/v1/people {body}
//example person instance:
const mike = new Person({
  firstName : "Mike",
  lastName: "Doe",
  address: "Sesammy street",
  idNumber: 123,
  image: "https://picsum.photos/200/300",
  phone: "0505045002"
})
```

יצירת מופע

```
mike.save().then(()=>{
  console.log("Saved");
}).catch(e=>{
  console.log(e);
})
```

שמירה

5

קוד שיופיע בRouter בתגובה לבקשת POST להוספת Person חדש לדטה-בייס

כל הקוד ביחד

```
import mongoose, { Schema, model } from "mongoose";

// connection:
export const connect = async () => {
  await mongoose.connect("mongodb://localhost:27017/test");
  console.log("Database Connected");
};

// Schema Definition:
export const personSchema = new Schema({
  idNumber: Number,
  firstName: String,
  lastName: String,
  image: String,
  address: String,
  phone: String,
});

// Model Definition ~ Person class:
//Mongoose generates a class with CRUD methods
export const Person = model("Person", personSchema);

connect().then(() => {
  // router POST /api/v1/people {body}
  //example person instance:
  const mike = new Person({
    firstName: "Mike",
    lastName: "Doe",
    address: "Sesammy street",
    idNumber: 123,
    image: "https://picsum.photos/200/300",
    phone: "0505045002",
  });

  mike
    .save()
    .then(() => {
      console.log("Saved");
    })
    .catch((e) => {
      console.log(e);
    });
});
```