
Introducción al desarrollo web entorno servidor

Gabriel Rodríguez Flores

February 13, 2024

- Instalación del módulo mongoose en NodeJS
- Creación de schemas y documentos con NodeJS
- Añadir a la estructura del proyecto

Contents

1	Teoría	3
1.1	Instalación	3
1.2	Conexión	3
1.3	Schemas	3
1.3.1	Nested Schemas	4
1.4	CRUD	4
1.5	Find	4
1.6	Insert	4
1.7	Update	5
1.7.1	Remove	5
1.8	Avanzados	5
2	Ejemplos	5
3	Ejercicios	5
4	Entregables	5
4.1	En clase	5
4.2	Tarea	5

1 Teoría

- Ejemplo integración sencillo
- Recomendaciones de rendimiento

1.1 Instalación

```
npm i mongoose
```

1.2 Conexión

```
const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/mydb', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

const db = mongoose.connection;

db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', () => {
  console.log('Connected to MongoDB');
});
```

1.3 Schemas

```
const mongoose = require('mongoose');

const { Schema } = mongoose;

const userSchema = new Schema({
  name: String,
  email: String,
  age: Number,
});

const User = mongoose.model('User', userSchema);

module.exports = User;
```

1.3.1 Nested Schemas

```
const mongoose = require('mongoose');

const { Schema } = mongoose;

const addressSchema = new Schema({
  street: String,
  city: String,
  state: String,
  country: String,
  postalCode: String,
});

const userSchema = new Schema({
  name: String,
  email: String,
  age: Number,
  address: addressSchema,
});

const User = mongoose.model('User', userSchema);

module.exports = User;
```

1.4 CRUD

1.5 Find

```
const user = await User.find({ name: 'John Doe' });
```

1.6 Insert

```
const User = require('./models/user');

const user = new User({
  name: 'John Doe',
  email: '',
  age: 30,
});

user.save();
```

1.7 Update

```
User.updateOne({ name: 'John Doe' }, { age: 31 });  
User.updateMany({ name: 'John Doe' }, { age: 31 });
```

```
const user = await User.find({ name: 'John Doe' });  
user.age = 31;  
user.save();
```

1.7.1 Remove

```
User.deleteOne({ name: 'John Doe' });  
User.deleteMany({ name: 'John Doe' });
```

```
const user = await User.find({ name: 'John Doe' });  
user.delete();
```

1.8 Avanzados

- Optimización y eficiencia

2 Ejemplos

3 Ejercicios

4 Entregables

4.1 En clase

4.2 Tarea