

1.

Conexión con mongoDB

Mtro. Iván Azamar Palma
Revisión Académica

M 05 A 24

Scribe

Fecha Inicio: 30/04/24 - 02-05-24 Kevin Jimenez Leyva

Investigar 3 formas diferentes de conectarse con mongoDB. La cadena de conexión debe cambiar (localhost, Dominio, Atlas, nombre del servidor, etc)

• Local host: `uri = mongodb://localhost:27020`

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

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

• Servidor remoto por dominio:

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

```
mongoose.connect('mongodb://nombreDominio:  
puerto/nombreBaseDatos',{  
  useNewUrlParser: true,  
  useUnifiedTopology: true  
});
```


Kevin Jiménez Leiva

Actividad 2. Método Express

Mtro. Iván Azamar Palma

Revisión
Académica



16

05

24

Scribe

Fecha Inicio: 09/05/24

Fecha Entrega: 16/05/24

1. ¿Que realiza la siguiente línea: `a. use (b. urlencoded (extendet : false }));`;

- Esta línea utiliza `body - Parser` para manejar los datos codificados en URL. `{ extend false }` es una opción que indica que los valores que recibimos son de tipo `String` a `array`.

2. ¿Que hace esta línea: `a. use (b. json ());` ?

Esta configurando `body parser` para manejar datos formato `json`.

3. Modularidades y formas de use que puede usar express

1. App basicas: Crea instancia de una app y se definen rutas, middleware y configuraciones directamente con ellas.

2. Enrutadores: Permite modularizar y organizar las rutas de la app en archivos separados.

4. uso del metodo `use (c)`

1. Middleware incorporado: Sirve para analizar solicitudes `json` y `Express`

`app. use (express. json ());`
`app. use (express. urlencoded ({ extendet: false }));`

2. Middleware personalizado: Podemos escribir nuestro propio middleware para realizar tareas especificas.

```
app. use (function (req, res, next) {  
  console. log ('middleware personalizado');  
  next ();  
});
```

5. Cuantos metodos Get puede usar Express:
Recibir datos:

`req. query`: Permite acceder a los parametros de consulta en la URL

`req. params`: Utilizando para acceder a los parametros de la ruta en la URL

Probando Apis 16-05-24 Termino: 23-05-24

Token

[Handwritten signature]
23-05-24

Kevin Jiménez Lexva

4. =

Mongoose

Mtro. Iván Azamar Palma

Fecha Inicio: 19/05/24 Fecha termino: 21/05/24

Mongoose: Es una biblioteca de Node.js que proporciona una solución basada en esquemas para modelar los datos de MongoDB.

① → Guardar: Este método se utiliza para guardar un nuevo documento en la BD o para actualizar un documento existente si ya tiene un ID.

Ejemplo:

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

```
const schema = mongoose.Schema;
```

```
const userSchema = new Schema({
```

```
  name: String,
```

```
  name: String,
```

```
  email: String,
```

```
  age: Number
```

```
});
```

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

```
// Guardar en la BD
```

```
new User.save((err) => {
```

```
  if (err) return console.error(err);
```

```
  console.log('Usuario guardado exitosamente')
```

```
});
```


Actividad 5. Heroku

Tema 3

Fecha: Inicio: 21/05/24 Entrega: 23/05/24

¿Qué es heroku?

R= Es una plataforma como servicio (PaaS) que permite a los desarrolladores desplegar, gestionar y escalar aplicaciones. Fundada en 2007 y adquirida por Salesforce en 2010, Heroku es popular por su simplicidad y capacidad de soportar múltiples lenguajes de programación como Ruby, Node.js, Python, Java etc.

¿Para que sirve? Se utiliza para desplegar apps web y móviles. Ofrece una estructura gestionada que permite a los desarrolladores concentrarse en la estructura de código en lugar de gestión de servidores.

