Backend Creation MYSQL

▼ Basic steps

1. Import Express and create server.

```
// Express is a nodejs framework
const express = require('express')
const app = express()

const PORT = 8080

// get request res.send data to the website
app.get('/',(req,res)=>{
  res.send("Hello World!");
}

// serveris listinging at port
app.listen(PORT,()=>{
  console.log(`Example app listening on port ${port}`)
});
```

2. Import Sequelize

```
const { Sequelize } = require ('sequelize');
// this will create a sequelize instance
```

3. Connect to database

```
// It will create an instance of sequelize class and we are giong to use
// object of it.
const sequelize = new Sequelize("database", "username", "password", {
  host: "localhost",
  dialect: "mysql",
});
```

4. Check the connection

```
// Use the .authenticate() function to test if the connection is OK:
async function  checkConnection(){
  try {
    await sequelize.authenticate();
    console.log("Connection has been established successfully.");
  } catch (error) {
    console.error("Unable to connect to the database:", error);
  }
}

checkConnection();
```

5. Define the model:

```
// Define the model by using .define method with sequelize object.
// It will create a table in database.
```

```
const Product = sequelizeObj.define('product',{
  id: {
    type: DataTypes.INTEGER,
    autoIncrement: true
  },
  Name : {
    type: DataTypes.STRING,
    allowNull: false
  }
})
```

6. Create table, Insert Data and show data

```
// Create table, insert and show  data in the table created.
async function addData(){
  try{

    // sync is necessary for creation of table. If it is true. It will drop table and create a new one
    await sequelizeObj.sync({force: true});

    // insert into tables created
    await userObject.create({id: 1, name: "Anish", email: "akjnv540@gmail.com"})

    // Show data
    let data = await userObject.findAll();
    console.log(data);


  }
  catch{
    console.log("something went wrong")
  }
}

addData();
```

▼ Create a project using Sequelize-Cli

1. Setup your project with npm init
2. install express ⇒ npm install express
3. install sequelize ⇒ npm install sequelize
4. install a dependency of sequelize **i.e. mysql2.** In our case we are using mysql database
    a. ⇒ npm install mysql2
5. npm install sequelize-cli
    a. tihs will provide some execution command. to use it
        i. **npx sequelize-cli init**
            1. This will create following folders

- `config`, contains config file, which tells CLI how to connect with database
- `models`, contains all models for your project
- `migrations`, contains all migration files
- `seeders`, contains all seed files

6. How to connect mysql database with our project

> 💡 MySql server runs on mysql server. Just like any other server it is responsible for handling the request coming to it. It also has a port no. and url associated with it. Port is 3306 by default. But due to ORM we don't need to bother about how mysql is working behind the scene. The ORM which we are using for MySql is Sequelize. It provides us a very simple way of connect with DBMS server by providing configuration of your server to Sequelize.

7. **npx sequelize-cli db: create**
    a. run this command if database is not created. It will create the database. In config.json we need to provide the database name.
8. **npx sequelize-cli model:generate —name <name> —attributes name:string,address:string**
    a. This will create a model in model folder.
        i. A model is a structure of your tables
    b. A migration in migration folder
        i. A migration file tells us what kind of changes it is going to do with your db. make changes to the migration to change the structure of the table.
9. **npx sequelize-cli db:migrate**
    a. This will create the actual table in database.
10. These command will create the tables in the database. But to add data to the table we need to import the **model.index.js** file. The file exports all the models.

```
const express = require ('express');
const app = express();
const port = 8080;

// Import customer model
const {Customer} = require("./models/index");

// Sending data to frontend.
app.get('/',(req, res)=> {
    res.send("<h1>Express server sending data to frontend</h1>")

})

app.listen(port,async ()=>{
    console.log(`Express server is listening to port ${port}`);

// Sending data to table
    let Customer1 = await  Customer.create({custName:"Anish Kumar", custAddr:"Delhi"});
    // console.log(Customer1);

// Reading data from table.
    let data = await Customer.findAll();
    // console.log(data);
})
```