

NODE

EXPRESS ROUTES

Install Packages

`npm i mongoose express ejs`

Connecting mongodb

mongoose

```
.connect("mongodb://localhost/sp19bsea")  
.then(() => {  
  console.log("connected to db");  
})  
.catch((err) => {  
  console.log(err);  
});
```

Creating a Model

```
const mongoose = require("mongoose");
const gamesSchema = mongoose.Schema({
  name: String,
  year: Number,
});
const Game = mongoose.model("Game", gamesSchema);
module.exports = Game;
```

Fire Up the server

```
const express = require("express");
const mongoose = require("mongoose");
const Game = require("./models/Game");
const app = express();
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
```

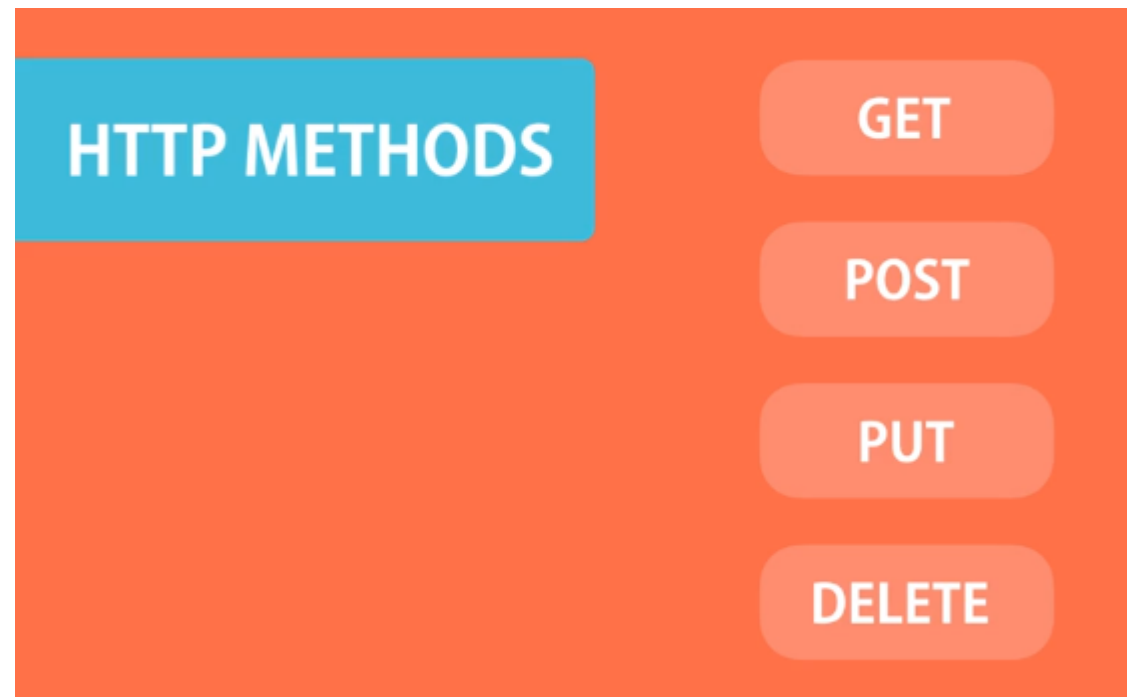
middlewares

express.json() is a built express middleware that convert request body to JSON.

express.urlencoded() just like **express.json()** but also convert form-data to JSON etc.

- **Extended:true** This option allows to choose between parsing the URL-encoded data with the `querystring` library (when false) or the `qs` library (when true). The “extended” syntax allows for rich objects and arrays to be encoded into the URL-encoded format, allowing for a JSON-like experience with URL-encoded. For more information, please see the `qs` library.

localhost:3300/api/games



All RESTFUL calls

```
GET /api/customers  
GET /api/customers/1  
PUT /api/customers/1  
DELETE /api/customers/1  
POST /api/customers
```


GET /api/games

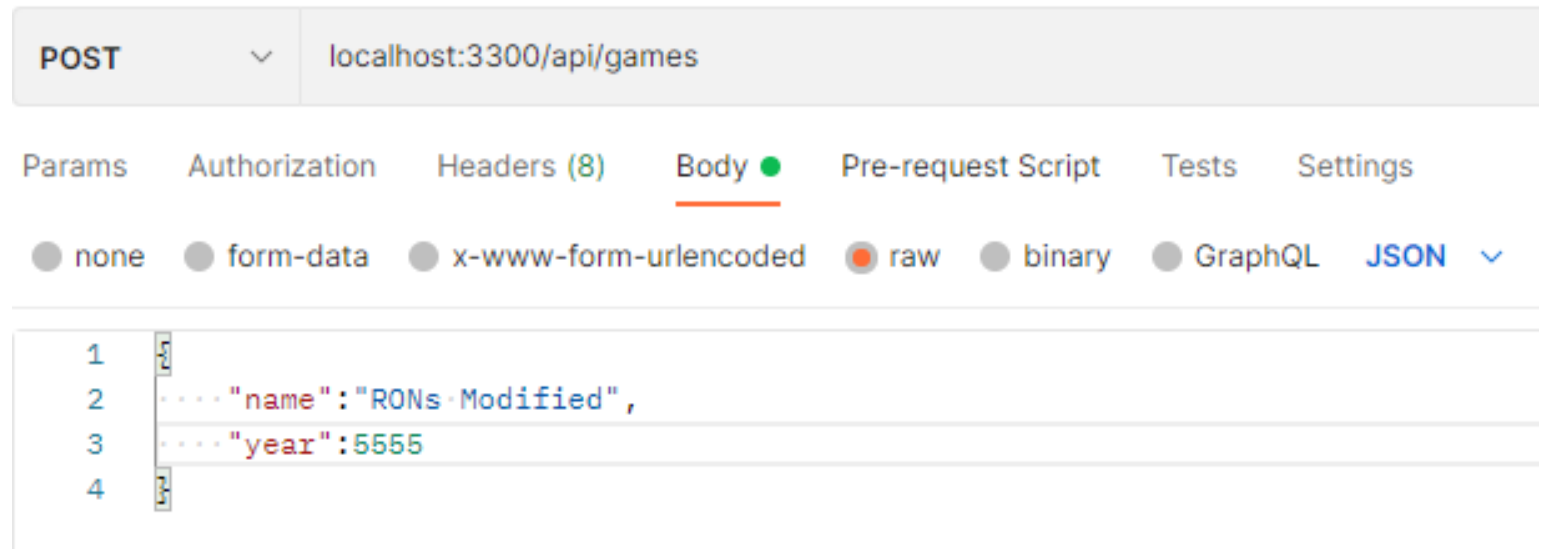
```
app.get("/api/games", async (req, res) => {  
    let games = new Game.find();  
    res.send(games);  
});
```

GET /api/games/123

```
app.get("/api/games/:id", async (req, res) => {  
  const game = await Game.findById(req.params.id);  
  res.send(game);  
});
```

POST /api/games

```
app.post("/api/games", async (req, res) => {  
  let game = new Game(req.body);  
  await game.save();  
  res.send(game);  
});
```

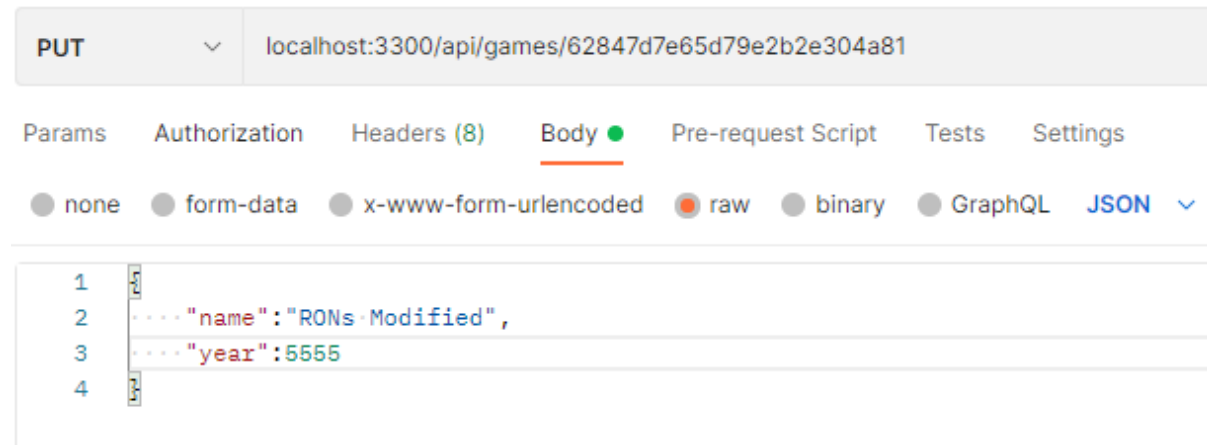


DELETE /api/games/123

```
app.delete("/api/games/:id", async (req, res) => {  
  const game = await Game.findById(req.params.id);  
  await game.delete();  
  res.send(game);  
});
```

PUT /api/games/123

```
app.put("/api/games/:id", async (req, res) => {  
  const game = await Game.findById(req.params.id);  
  game.name = req.body.name;  
  game.year = req.body.year;  
  await game.save();  
  res.send(game);  
});
```



Our API is Done

Now Some code clean up

express.Router

- class to create modular, mountable route handlers

express.Router

```
const express = require('express')
const router = express.Router()
router.delete("/:id", async (req, res) => {
  const game = await Game.findById(req.params.id);
  await game.delete();
  res.send(game);
});
module.exports = router
```

Connect Router in main file as

```
app.use("/api/games",require("./api/games"))
```


Set Up View Engine (npm i ejs)

```
app.set('views', path.join(__dirname, 'views'));  
app.set('view engine', 'ejs');
```

//path is built in module

//npm i ejs

List Records in EJS

```
router.get("/", async (req, res) => {  
  const games = await Game.find();  
  res.render("games/index", { games });  
});  
// res.render will look for index.ejs file in views/games folder
```

Index.ejs

```
<html>
  <head></head>
  <body>
    <a href="/games/create">Create New Game</a>
    <ul>
      <%for(let i=0;i<games.length;i++){%>
        <li>
          <%-games[i].name%>
          <a href="/games/edit/<%-games[i]._id%>">Edit</a>
          <a href="/games/delete/<%-games[i]._id%>">Delete</a>
        </li>
      <%}%>
    </ul>
  </body>
</html>
```

Create a Record

```
router.get("/create", async (req, res) => {  
    res.render("games/create");  
}); // render form
```

```
router.post("/create", async (req, res) => {  
    let record = new Game(req.body);  
    await record.save();  
    res.redirect("/games");  
}); // handle form submission
```

Create Form

```
<html>
  <head></head>
  <body>
    <form action="/games/create" method="post">
      Name: <input type="text" name="name" /> <br />
      Year: <input type="text" name="year" /> <br />
      <input type="submit" value="Create Game" />
    </form>
  </body>
</html>
```

Data will have the shape

```
{
  name:"",
  Year:""
}
```

Edit a record

```
router.get("/edit/:id", async (req, res) => {
  let game = await Game.findById(req.params.id);
  res.render("games/edit", { game });
});

router.post("/edit/:id", async (req, res) => {
  let game = await Game.findById(req.params.id);
  game.name = req.body.name;
  game.year = req.body.year;
  await game.save();
  res.redirect("/games");
});
```

Edit Form

```
<form action="/games/edit/<%-game._id%>" method="post">  
  Name: <input type="text" name="name" value="<%-game.name%>" /> <br />  
  Year: <input type="text" name="year" value="<%-game.year%>" /> <br />  
  <input type="submit" value="Edit Game" />  
</form>
```

Delete Require no view to render

```
router.get("/delete/:id", async (req, res) => {  
  let game = await Game.findById(req.params.id);  
  await game.delete();  
  res.redirect("/games");  
});
```


Check Final Code

<https://github.com/mua22/super-simple-api-and-server-side-crud>

Run following commands

```
> npm install
```

```
> nodemon server.js
```

If nodemon is not installed

```
> npm install nodemon -g
```

Application will run on localhost:3300/games for server side and
localhost:3300/api/games for api