

Início da Aula

Explicar que é necessário refatorar a estrutura do projeto para deixar no padrão MVC

MODEL, VIEW e CONTROLLER

no INDEX.JS, exportar o server para ser usado

```
28 })
29
30 server.delete('/movies/deleteall', async(req,res) => {
31
32     try{
33         await Movie.deleteMany()
34         res.status(200).json({message: "Todos os filmes deletados"})
35     } catch (erro){
36         res.status(500).json({message: erro})
37     }
38 })
39
40 module.exports = server
41
42
```

Criar o arquivo SERVER.JS e colocar toda a configuração da porta do server nesse arquivo

```
const server = require('./index')
const PORT = process.env.PORT || 3000
server.listen(PORT, () => {
  console.log("Servidor funcionando...")
})
```

no INDEX.JS apagar o server.listen na conexão com o Banco

```
//Configurar a porta de conexão
mongoose.connect('mongodb+srv://vinicius:super123@apiclusterv.mongodb.net/bancoapi?retryWrites=true&w=majority')
  .then(() => {
    console.log("servidor conectado ao MONGO DB ATLAS...")
  })
  .catch((erro) => {console.log(erro)})
```

Alterar o PACKAGE.JSON para separar o ambiente de desenvolvimento do ambiente de produção

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "dev": "nodemon ./server.js",
  "start": "node ./server.js"
},
```

Para rodar o projeto localmente em modo desenvolvedor devemos utilizar o código

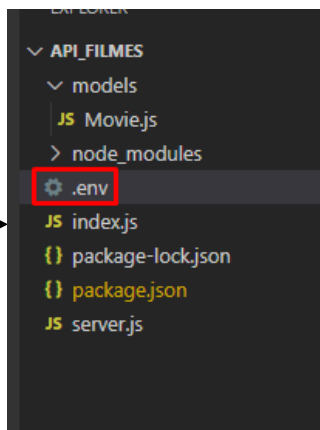
`npm run dev`

Explicar que devemos separar a conexão do banco do arquivo do index.js por conta da segurança (a senha do banco está no caminho)

Adicionar o pacote no projeto de variáveis de ambiente

```
npm install dotenv
```

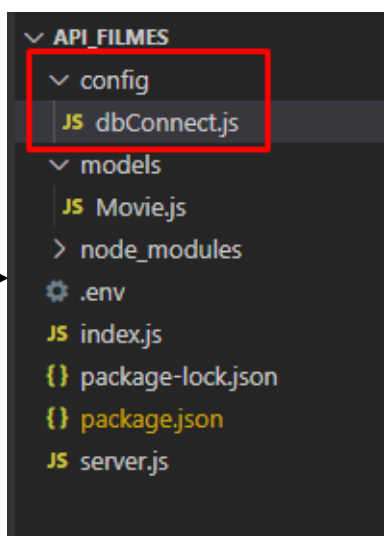
Criar um arquivo .env na raiz do projeto



Adicionar a url do banco

```
.env
1 DATABASE_URL = mongodb+srv://vinicius:super123@apicluster.eaftv.mongodb.net/bancoapi?retryWrites=true&w=majority
```

na raiz do projeto criar uma pasta chamada CONFIG e dentro desta pasta criar um arquivo chamado dbConnect.js



no DBCONNECT.JS, realizar a conexão com o banco

```
> JS dbConnect.js > [?] <unknown>
const mongoose = require("mongoose")

require('dotenv').config()

mongoose.connect(process.env.DATABASE_URL, {
  useNewUrlParser: true
})

const db = mongoose.connection

module.exports = db
```

no INDEX.JS, substituir a importação do mongo pelo arquivo da configuração e substituir a conexão pelo tratamento e abertura

```
//Importar o pacote do express
const express = require("express")

//Importar a conexão com o banco
const db = require("../config/dbConnect")

//Importar a entidade Movie
const Movie = require('../models/Movie')

//Verificar se deu erro na conexão ou tentar abrir para utilizar
db.on("error", console.log.bind(console, 'Erro de conexão'))
db.once("open", () => {
  console.log('Conexão com o banco feita com sucesso...')
})

//Inicializar o pacote
const server = express()

//Habilitar a leitura e recebimento de json -> Middleware (use)
server.use(
  express.urlencoded({
    extended: true,
  })
)
```

na raiz do projeto criar uma pasta chamada CONTROLLERS e dentro dela criar o arquivo movieController.js

colocar o código da primeira rota de boas vindas

```
ollers > JS movieController.js > MovieController > giveWelcome
const Movie = require("../models/Movie")

class MovieController {

  static giveWelcome = (req, res) => {
    res.json({
      message: "Bem Vindo a nossa API com MongoDB."
    })
  }

}

module.exports = MovieController
```

na raiz do projeto criar uma pasta chamada ROUTES e dentro criar o arquivo MOVIEROUTE.JS

mapear somente a rota de boas vindas

```
s > JS movieRoute.js > <unknown>
const express = require('express')

const MovieController = require("../controllers/movieController")

const router = express.Router()

router.get("/", MovieController.giveWelcome)

module.exports = router
```

no INDEX.JS, importar as rotas para utilizar e apagar as rotas escritas

```
//Importar a entidade Movie
const Movie = require('./models/Movie')

//Importar as rotas
const movieRoutes = require('./routes/movieRoute')

//Verificar se deu erro na conexão ou tentar abrir para utilizar
db.on("error", console.log.bind(console, 'Erro de conexão'))
db.once("open", () => {
  console.log('Conexão com o banco feita com sucesso...')
})

//Inicializar o pacote
const server = express()

//Habilitar a leitura e recebimento de json -> Middleware (use)
server.use(
  express.urlencoded({
    extended: true,
  })
)

server.use(
  express.json(),
  movieRoutes
)
```

Repetir esse processo para todas as rotas

movieController.js

```
const Movie = require("../models/Movie")

class MovieController {

  static giveWelcome = (req, res) => {
    res.json({
      message: "Bem Vindo a nossa API com MongoDB."
    })
  }

  static registerMovie = async (req, res) => {
    //Pegar as informações do corpo da nossa requisição
    const {name, year, streaming} = req.body

    //Colocar as informações em um objeto
    const movie = {
      name,
      year,
      streaming
    }

    try{
      // Esperar para garantir que os dados foram cadastrados
      await Movie.create(movie)

      //enviar uma resposta com sucesso
      res.status(201).json({message: 'Filme cadastrado no banco com sucesso !'})
    } catch (erro) {
      //Enviar uma resposta com o erro
      res.status(500).json({message: erro})
    }
  }

  static getAllMovies = async (req, res) => {
    // buscar os filmes -> Operação que falhar
    try{
      //metodo .find() lista todos os filmes do banco e colocar em uma variavel
      const movies = await Movie.find()
    }
  }
}
```



```
es > JS movieRoute.js > ...
1  const express = require('express')
2
3
4  const MovieController = require("../controllers/movieController")
5
6  const router = express.Router()
7
8  router.get("/", MovieController.giveWelcome)
9
10 router.post("/movie", MovieController.registerMovie)
11
12 router.get("/movies", MovieController.getAllMovies)
13
14 router.put("/movie/:id", MovieController.updateMovie)
15
16 router.delete("/movie/:id", MovieController.deleteMovie)
17
18 router.get("/movie/:id", MovieController.getMovieById)
19
20 router.delete("/movies/deleteall", MovieController.deleteAllMovies)
21
22
23 module.exports = router
```



```
JS index.js > ...
1  //Importar o pacote do express
2  const express = require("express")
3
4  //Importar a conexão com o banco
5  const db = require("../config/dbConnect")
6
7  //Importar as rotas
8  const movieRoutes = require('../routes/movieRoute')
9
10 //Verificar se deu erro na conexão ou tentar abrir para utilizar
11 db.on("error", console.log.bind(console, 'Erro de conexão'))
12 db.once("open", () => {
13   console.log('Conexão com o banco feita com sucesso...')
14 })
15
16 //Inicializar o pacote
17 const server = express()
18
19 //Habilitar a leitura e recebimento de json -> Middleware (use)
20 server.use(
21   express.urlencoded({
22     extended: true,
23   })
24 )
25
26 server.use(
27   express.json(),
28   movieRoutes
29 )
30
31 module.exports = server
32
33
```