

Configurando o backend

- Crie as pastas 4oanosi e dentro dela backend
- Entre na pasta backend e digite o comando `yarn init -y` para criar o projeto
 - Instalar o express, digitando `yarn add express`
 - Instale o nodemon digitando `yarn add nodemon -D`
 - Observe no package.json que criou a dependência do express e a pasta node_modules e o arquivo yarn.lock que é um arquivo de cache
 - Crie a pasta src e dentro dela o arquivo index.js. Digite o código abaixo neste arquivo

```
const express = require('express')

const app = express()

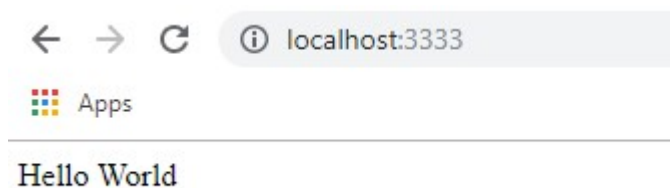
app.get('/', (req, res) => {
  res.send('Hello World')
})

app.listen(3333)
```

Para facilitar a inicialização da aplicação, edite o arquivo backend/package.json e inclua as linhas 6, 7 e 8 conforme print abaixo.

```
1  {
2    "name": "backend",
3    "version": "1.0.0",
4    "main": "index.js",
5    "license": "MIT",
6    "scripts": {
7      "dev": "nodemon src/index.js"
8    },
9    "dependencies": {
10     "express": "^4.17.1"
11   },
12   "devDependencies": {
13     "nodemon": "^1.19.1"
14   }
15 }
```

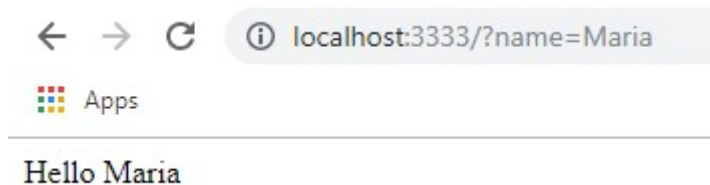
- Inicie a aplicação digitando `yarn dev`
- Abra o navegador digitando `localhost:3333` e será exibida a janela abaixo



Edite o arquivo src/index.js conforme abaixo. Insira a linha 6 e altere a linha 7

```
1  const express = require('express')
2
3  const app = express()
4
5  app.get('/', (req, res) => {
6    const { name } = req.query
7    res.send(`Hello ${name}`)
8  })
9
10 app.listen(3333)
```

Abra o navegador e digite <http://localhost?name=Maria> e será exibida a tela abaixo



Mongoose

É uma biblioteca instalada no node para manipular dados criados no banco de dados de MongoDB. Para instá-lo digite **yarn add mongoose**

Edite o arquivo src/index.js conforme print abaixo. Insira as linhas 2, 6, 7 e 8. A linha 6 conecta no servidor onde está instalado o mongodb na porta 27017 e no banco de dados bdsi. O comando useNewUrlParser: true deve ser informado para você utilizar url de conexão no banco de dados.

```
1  const express = require('express')
2  const mongoose = require('mongoose')
3
4  const app = express()
5
6  mongoose.connect(`mongodb://localhost:27017/bdsi`, {
7    useNewUrlParser: true,
8  })
9
10 app.get('/', (req, res) => {
11   const { name } = req.query
12   res.send(`Hello ${name}`)
13 })
14
15 app.listen(3333)
```

Criando as rotas da aplicação

Crie o arquivo `src/routes.js`

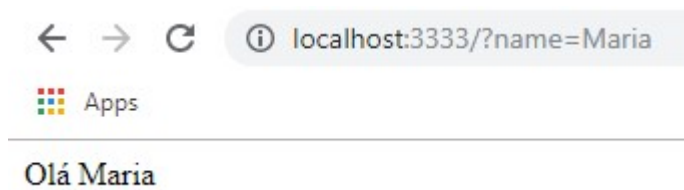
```
1  const { Router } = require('express')
2
3  const routes = new Router();
4
5  routes.get('/', (req, res) => {
6    const { name } = req.query
7    return res.send(`Olá ${name}`)
8  })
9
10 module.exports = routes;
```

Importando as rotas no arquivo `src/index.js`

Insira a linha 10 e deixe o seu arquivo conforme o print abaixo.

```
1  const express = require('express')
2  const mongoose = require('mongoose')
3
4  const app = express()
5
6  mongoose.connect(`mongodb://localhost:27017/bdsi`, {
7    useNewUrlParser: true,
8  })
9
10 app.use(require('./routes'))
11
12 app.listen(3333)
```

Abra o navegador e digite <http://localhost?name=Maria> e será exibida a página abaixo.



Criando outras pastas do projeto

Dentro da pasta `src`, crie as pastas `models` (que terá como objetivo definir as tabelas se for um banco de dados relacionais ou schema para banco de dados não relacionais), `controllers` (onde serão definidas todas as regras de negócio da aplicação) e `config`.

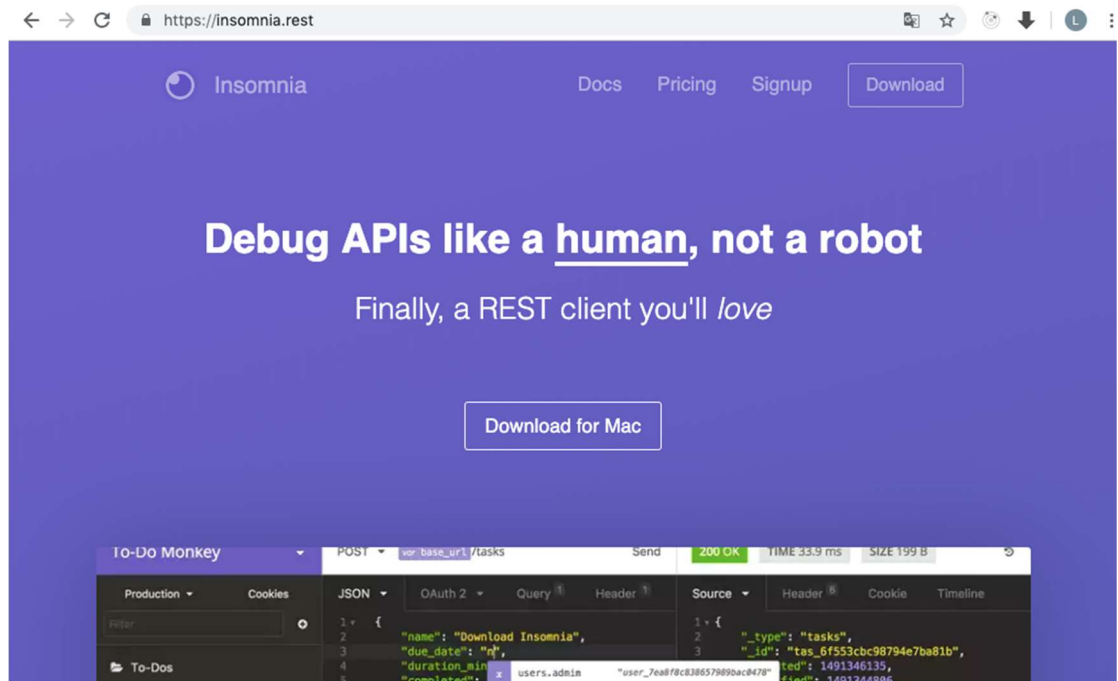
Criando o Model

Crie o arquivo `src/models/Posts.js` conforme print abaixo.

```
1  const mongoose = require('mongoose');
2
3  const PostSchema = new mongoose.Schema({
4    author: String,
5    place: String,
6    description: String,
7    hashtags: String,
8    image: String,
9    likes: {
10     type: Number,
11     default: 0,
12   }
13 }, {
14   timestamps: true
15 })
16
17 module.exports = mongoose.model('Post', PostSchema)
```

Utilizando o insomnia

O navegador testa rotas somente do tipo GET, para as rotas do tipo POST, PUT ou DELETE será necessário utilizar um software chamado **insomnia** através do endereço (<https://insomnia.rest/>) veja abaixo o print.



Baixe e instale-o. Depois que abrir o insomnia será exibida a janela a seguir.

