

Configurando o multer para salvar a imagem.

- Dentro da pasta backend, crie a pasta uploads e dentro dela resized.
- Dentro da pasta src, crie a pasta config e arquivo upload.js conforme código abaixo.

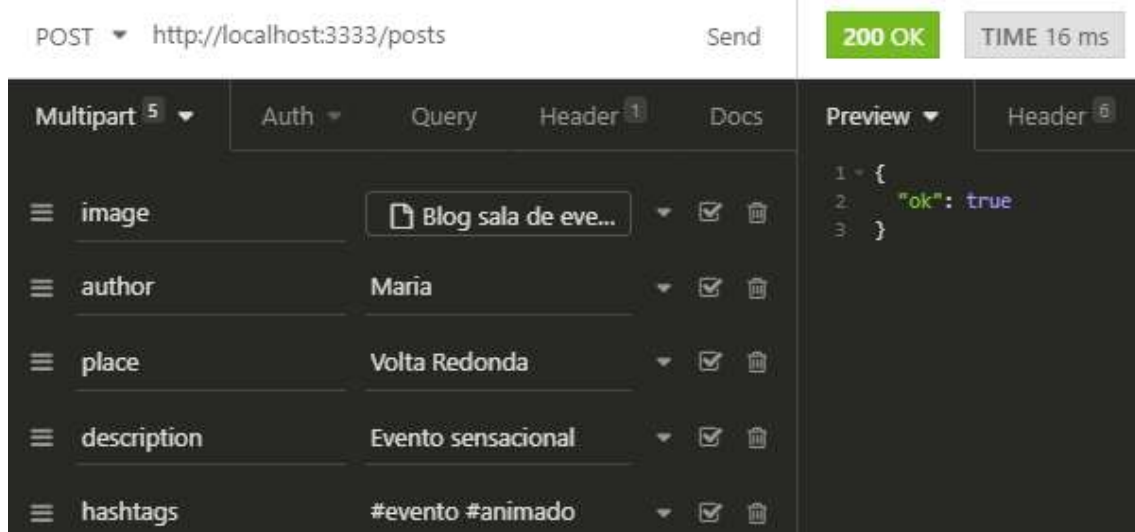
```
1  const multer = require('multer')
2  const path = require('path')
3
4  module.exports = {
5    storage: new multer.diskStorage({
6      destination: path.resolve(__dirname, '..', '..', 'uploads')
7    })
8  }
```

Edite o arquivo src/routes.js conforme print abaixo.

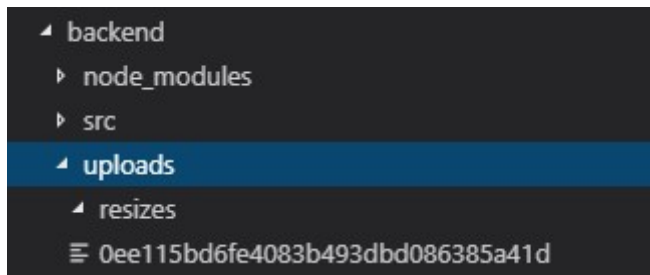
- Insira a linha 3 para carregar a configuração do src/config/upload.js
- Altere a linha 7 para o multer carregar as configurações do arquivo upload.js

```
1  const { Router } = require('express')
2  const multer = require('multer')
3  const uploadConfig = require('./config/upload')
4  const PostController = require('./controllers/PostController')
5
6  const routes = new Router();
7  const upload = multer(uploadConfig);
8
9  routes.post('/posts', upload.single('image'), PostController.store)
10
11 module.exports = routes;
```

Abra o insomnia e envia novamente a requisição conforme print abaixo.



Observe que a imagem será gravada no caminho backend/uploads conforme print abaixo num formato totalmente ilegível.



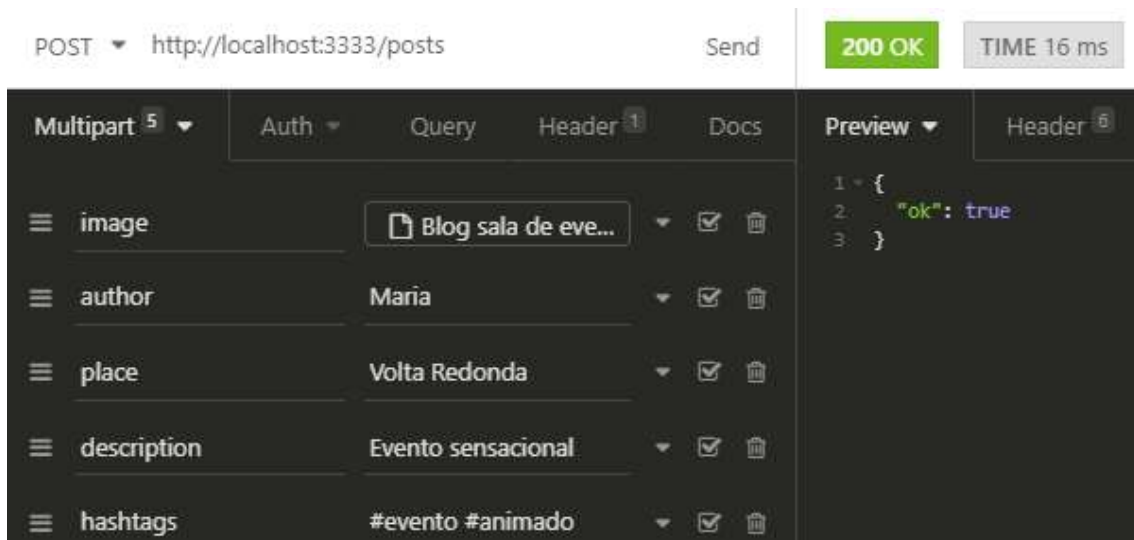
Alterando o código para gravar a imagem num formato legível

Edite o arquivo src/config/uploads.js e insira as linhas 7, 8 e 9.

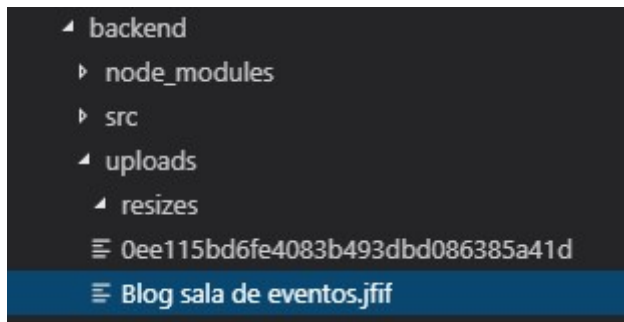
```

1  const multer = require('multer')
2  const path = require('path')
3
4  module.exports = {
5    storage: new multer.diskStorage({
6      destination: path.resolve(__dirname, '..', '..', 'uploads'),
7      filename: function(req, file, cb) {
8        cb(null, file.originalname)
9      }
10   })
11 }
```

Abra o insomnia e envie novamente a requisição conforme print abaixo



Observe que a imagem será gravada no caminho backend/uploads conforme print abaixo e com o mesmo nome que tem no seu computador, ou seja, o nome original do arquivo.



Gravando os dados do formulário e a imagem no banco de dados.

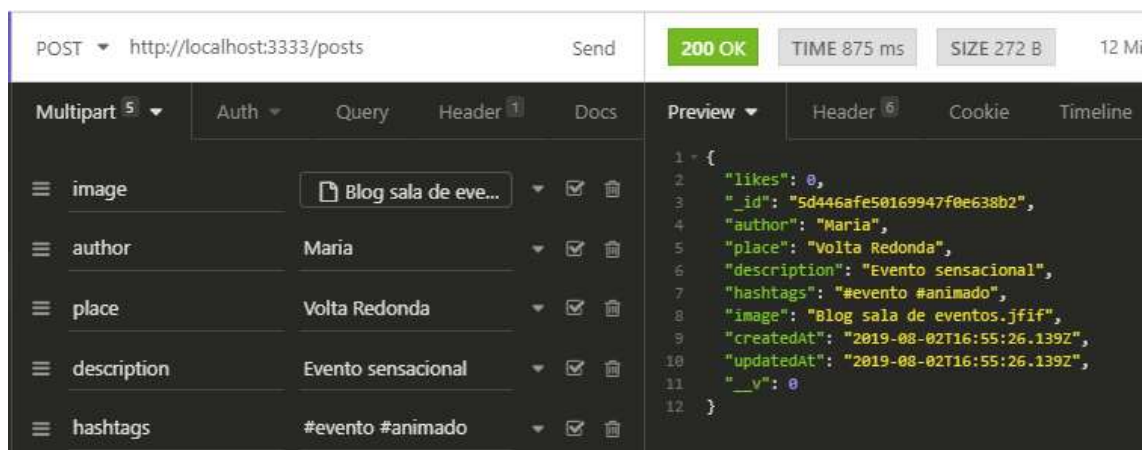
Edite o arquivo `src/controllers/PostController.js` conforme print abaixo. Insira da linha 11 até 21.

```

1  const Post = require('../models/Posts')
2
3  module.exports = {
4    // o método index lista os posts
5    async index(req, res) {
6      return res.json({ ok: true })
7    },
8
9    // o método store cria os posts
10   async store(req, res) {
11     const { author, place, description, hashtags } = req.body;
12     const { filename: image } = req.file;
13
14     const post = await Post.create({
15       author,
16       place,
17       description,
18       hashtags,
19       image
20     })
21     return res.json(post)
22   }
23 }

```

Abra o insomnia e envie novamente a requisição conforme print abaixo.



Veja no print acima que as informações foram gravadas no banco de dados. Além dos campos `author`, `place`, `description`, `hashtags` e `image` que estavam no formulário, foram retornados outros como `likes`, `_id`, `createdAt` e `updatedAt`.

Insira mais três registros no banco de dados com informações diferentes.

Listando os dados do banco de dados

Definindo a rota para listar os dados do banco de dados.

Edite o arquivo `src/routes.js`, conforme print a seguir. Insira a linha 9 para que ao acessar a rota `/posts` com o método `http get`, o método `index` do `PostController` seja executado.

```

1  const { Router } = require('express')
2  const multer = require('multer')
3  const uploadConfig = require('./config/upload')
4  const PostController = require('./controllers/PostController')
5
6  const routes = new Router();
7  const upload = multer(uploadConfig);
8
9  routes.get('/posts', PostController.index);
10 routes.post('/posts', upload.single('image'), PostController.store)
11
12 module.exports = routes;

```

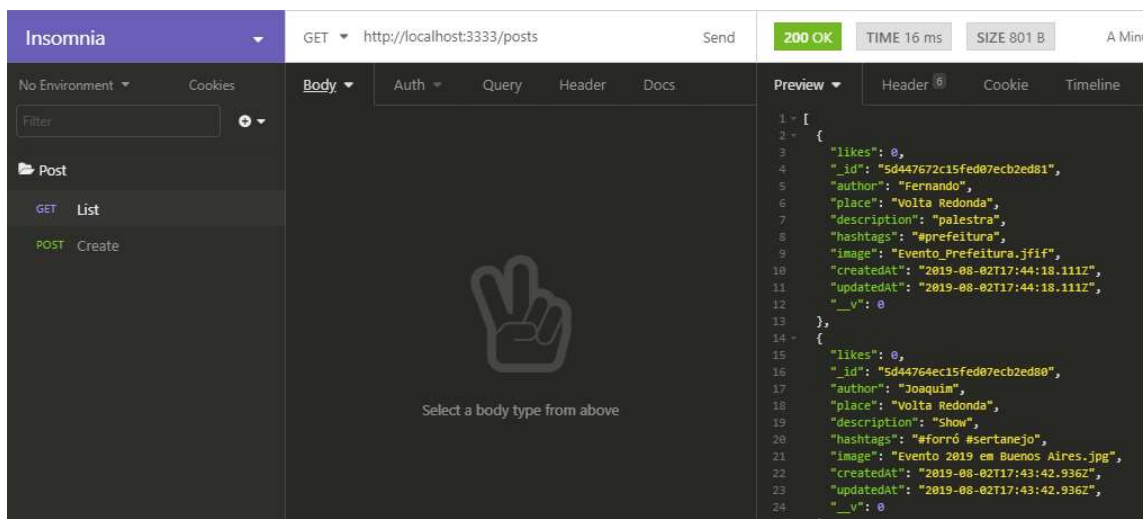
Edite o arquivo `src/controllers/PostController.js` para implementar o método `index` para listar todos os posts gravados no banco de dados, conforme print abaixo. Insira as linhas 6 e 7.

```

1  const Post = require('../models/Posts')
2
3  module.exports = {
4    // o método index lista os posts
5    async index(req, res) {
6      const posts = await Post.find().sort('-createdAt')
7      return res.json(posts)
8    },
9
10   // o método store cria os posts
11   async store(req, res) {
12     const { author, place, description, hashtags } = req.body;
13     const { filename: image } = req.file;
14
15     const post = await Post.create({
16       author,
17       place,
18       description,
19       hashtags,
20       image
21     })
22     return res.json(post)
23   }
24 }

```

Abra o insomnia, crie uma requisição chamada `List` com o método `http get` para acessar a rota <http://localhost:3333/posts> e ao clicar no botão `Send` será listado todos os posts do banco de dados, conforme print abaixo.



Implementando a funcionalidade de like na api

Crie o arquivo `src/controllers/LikeController.js` conforme print abaixo.

```

1  const Post = require('../models/Posts')
2
3  module.exports = {
4    // o método store cria os likes
5    async store(req, res) {
6      return res.json({ ok: true })
7    }
8  }

```

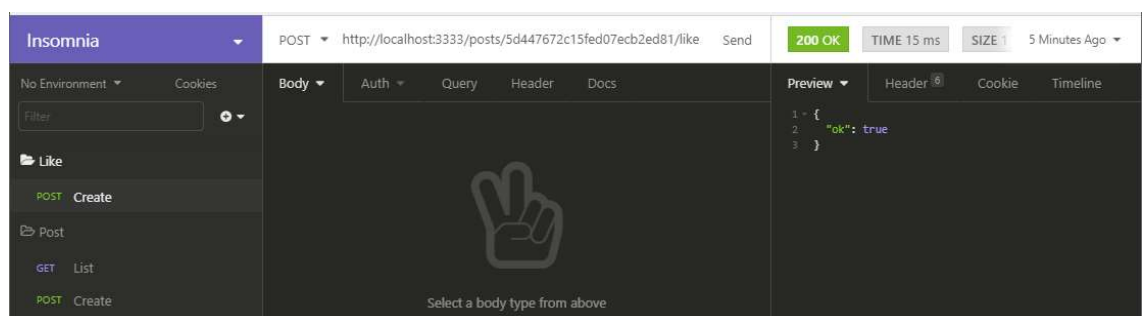
Edite o arquivo `src/routes.js` para importar o `LikeController` e criar a rota com o método `post` para gerenciar o like dos usuários. Insira as linhas 5 e 13.

```

1  const { Router } = require('express')
2  const multer = require('multer')
3  const uploadConfig = require('./config/upload')
4  const PostController = require('./controllers/PostController')
5  const LikeController = require('./controllers/LikeController')
6
7  const routes = new Router();
8  const upload = multer(uploadConfig);
9
10 routes.get('/posts', PostController.index);
11 routes.post('/posts', upload.single('image'), PostController.store)
12
13 routes.post('/posts/:id/like', LikeController.store)
14
15 module.exports = routes;

```

Abra o insomnia, crie uma pasta chamada Like e uma requisição Create do tipo `post` para acessar a rota http://localhost:3333/posts/id_de_post_gravado/like, por exemplo (<http://localhost:3333/posts/5d447672c15fed07ecb2ed81/like>). Clique no botão enviar e verifique se retornará um json com `ok: true`, conforme print abaixo.



Edite o arquivo `src/controllers/LikeController.js` conforme print abaixo para implementar a quantidade de likes num post. Insira as linhas 6, 7, 8 e 9.

```

1  const Post = require('../models/Posts')
2
3  module.exports = {
4    // o método store cria os likes
5    async store(req, res) {
6      const post = await Post.findById(req.params.id)
7      post.likes += 1;
8      await post.save();
9      return res.json(post)
10   }
11 }

```

Abra o insomnia, acesse a requisição create da pasta Like, defina um id do banco de dados, por exemplo, (<http://localhost:3333/posts/5d447672c15fed07ecb2ed81/like>). Clique no botão enviar e verifique se o campo likes será exibido com o conteúdo 1. Se você clicar novamente no botão Send este número será incrementado automaticamente.

