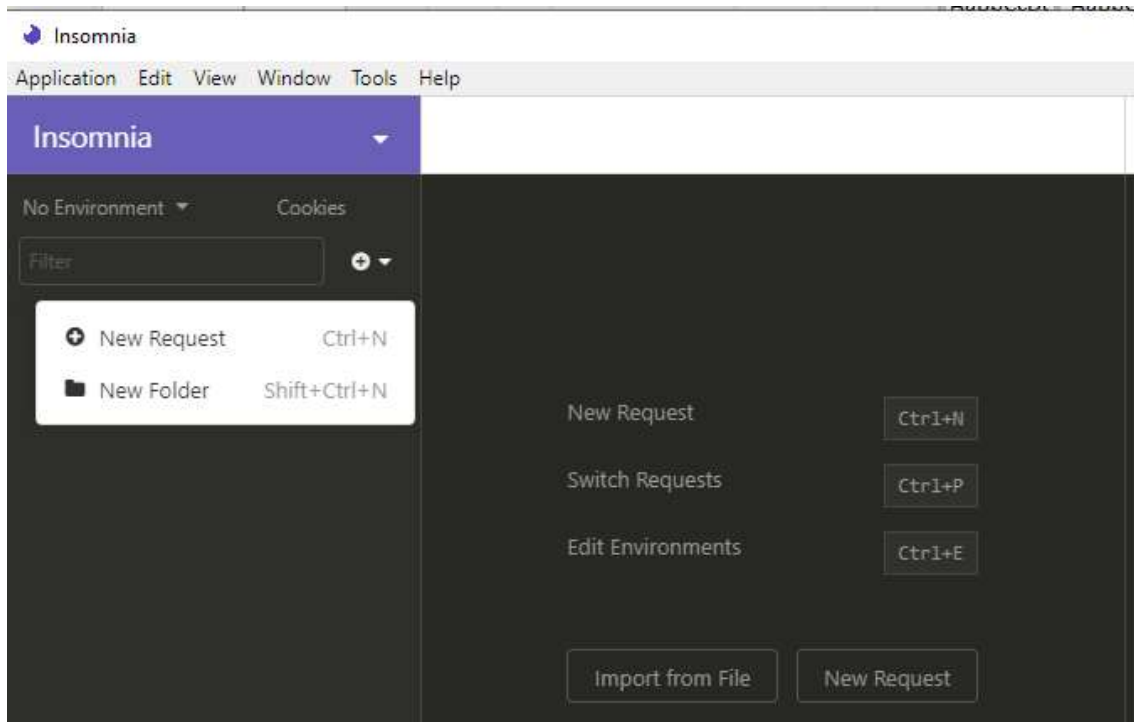


Utilizando o insomnia

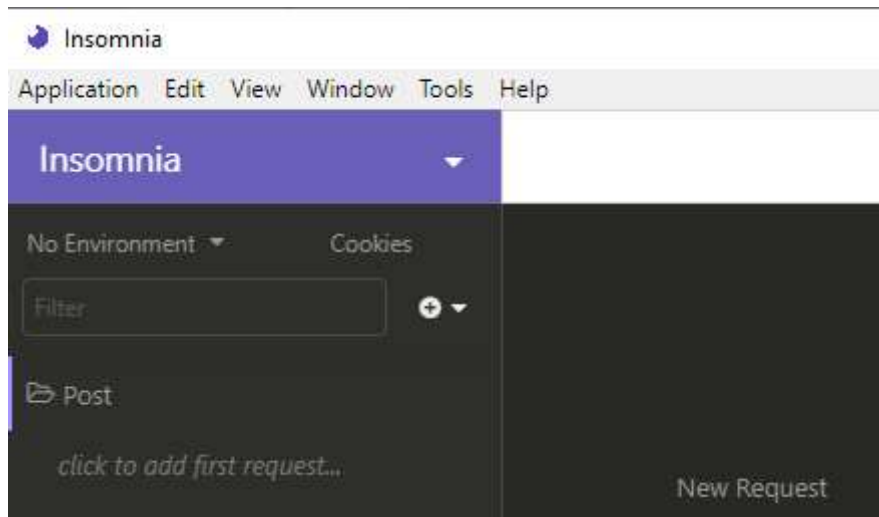
Ao abrir o insomnia, clique na seta ao lado do sinal de (+) e na opção New Folder.



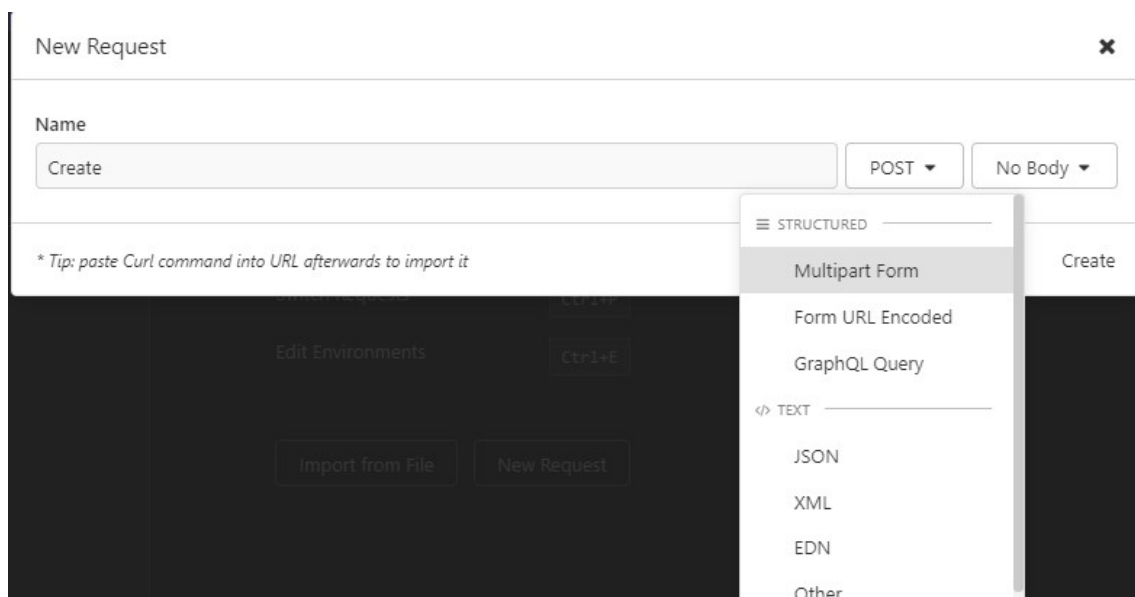
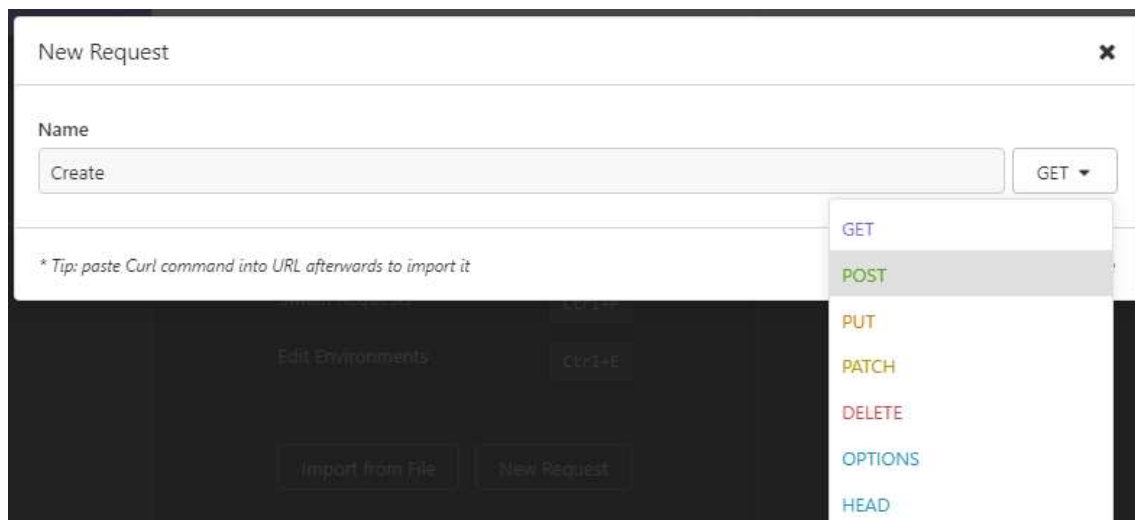
Crie a pasta Post.



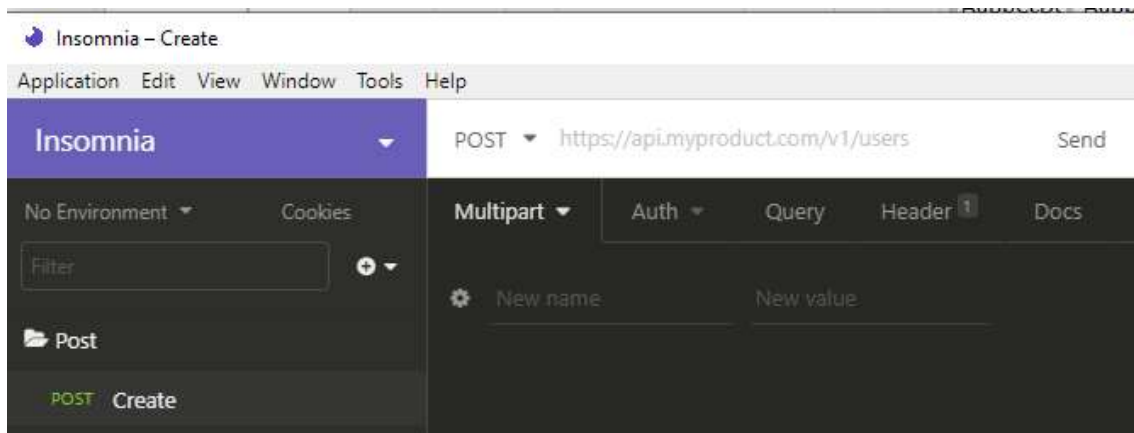
Abaixo da pasta Post, clique na opção **click to add first request...** conforme print abaixo, para criar uma nova requisição.



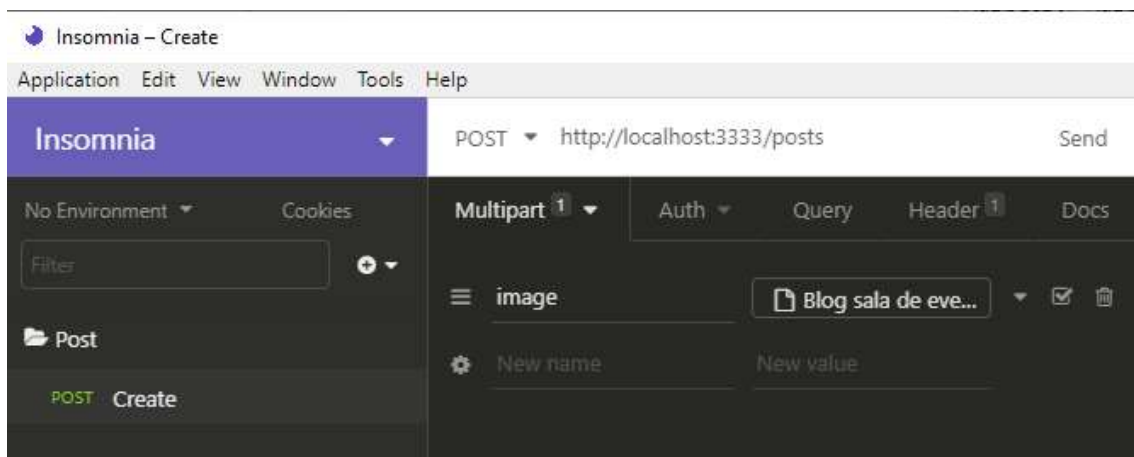
Crie uma request chamada Create do tipo Post e no formato Multipart Form conforme prints abaixo:



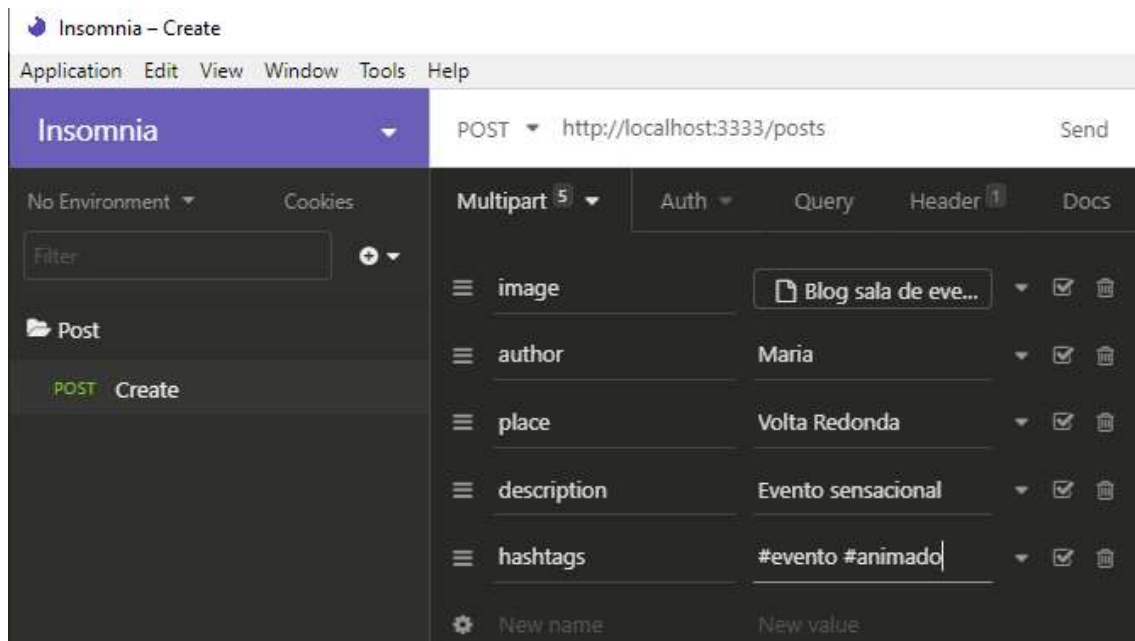
Sua tela deverá ficar conforme abaixo.



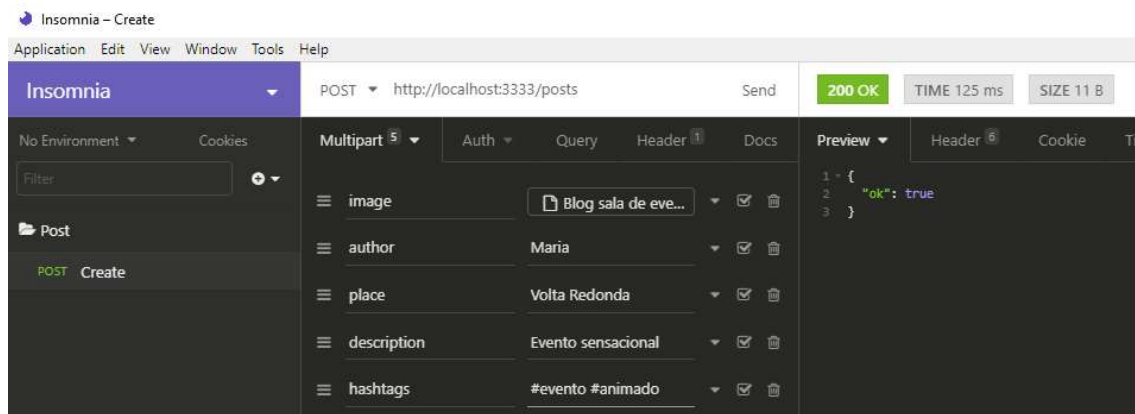
Na janela acima, clique no campo New Value, escolha File, depois clique em Choose File para escolher uma imagem, defina o nome do campo como image, e o endereço da rota que será acessada <http://localhost:3333/posts> conforme print abaixo:



Preencha os campos author, place, description e hashtags, conforme tela abaixo:



Clique no botão Send e verifique se será retornado um json com **ok: true**, conforme print a seguir.



Para o express entender uma requisição no formato multipart form data será necessário instalar uma dependência chamada multer, digitando **yarn add multer**. Ele é uma biblioteca que lida com estas informações e automatiza o processo de uploads dentro do backend.

Configurando o multer para capturar as informações do formulário com imagem

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

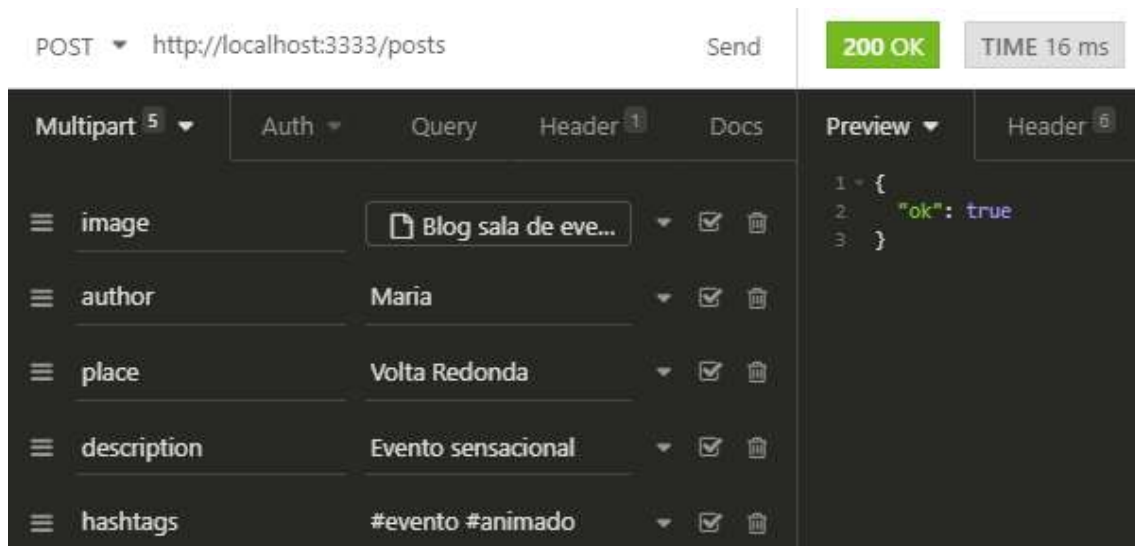
- Insira a linha 2 para carregar o multer.
- Insira a linha 6 para executar o multer e armazenar na variável upload
- Altere a linha 8, inserindo o comando upload.single para permitir o envio de um arquivo.

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

Edite o arquivo src/controllers/PostController.js inserindo a linha 11 para imprimir o corpo da requisição, utilizando o req.body

```
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     console.log(req.body)
12     res.json({ ok: true })
13   }
14 }
```

Abra o insomnia e envia novamente a requisição conforme print abaixo



Veja o resultado do console.log(req.body)

```
[nodemon] starting `node src/index.js`
[Object: null prototype] {
  author: 'Maria',
  place: 'Volta Redonda',
  description: 'Evento sensacional',
  hashtags: '#evento #animado' }

```

Visualizando as informações da imagem.

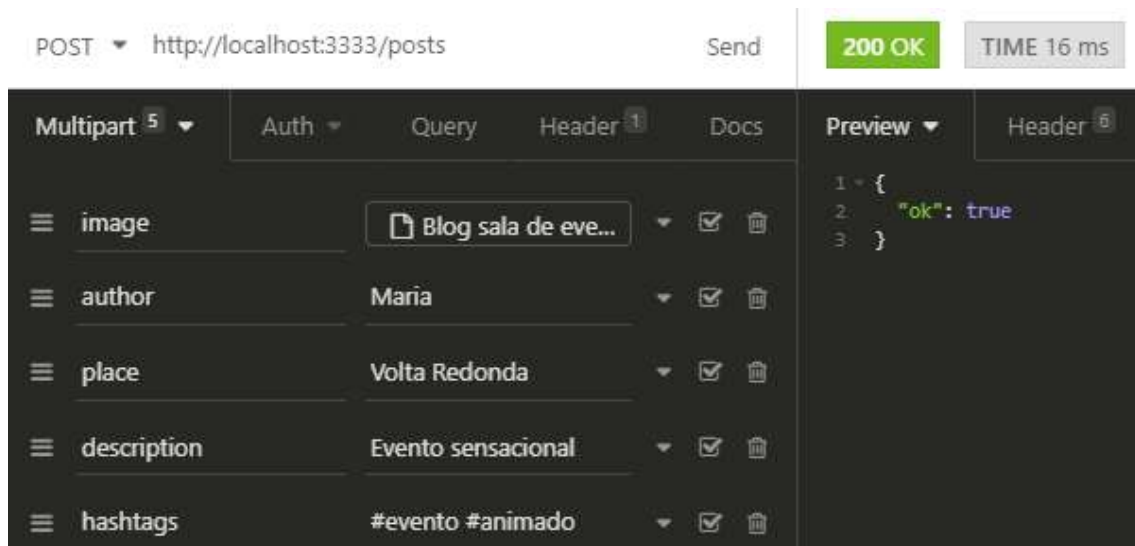
Edite o arquivo src/controllers/PostController.js alterando a linha 11 para imprimir o req.file, conforme abaixo.

```

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     console.log(req.file)
12     res.json({ ok: true })
13   }
14 }

```

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



Veja o resultado do console.log(req.file)

```
[nodemon] restarting due to changes...
[nodemon] starting `node src/index.js`
{ fieldname: 'image',
  originalname: 'Blog sala de eventos.jfif',
  encoding: '7bit',
  mimetype: 'application/octet-stream',
  buffer:
    <Buffer ff d8 ff e0 00 10 4a 46 49 46 00 01 01 00 00 01 00 01 00 00 ff db 00 84 00 09 06 07 13 13
12 15 13 13 13 16 16 15 17 18 1d 1a 17 18 18 17 17 1f 18 19 ... >,
  size: 13644 }
```