

# Projeto de Interfaces WEB

Express-Fileupload  
Aula 15

# Introdução

- O objetivo desse mini-projeto é:
  - Criar um servidor simples em Express;
  - Criar um serviço de upload
  - Criar uma página HTML simples para escolher e enviar o arquivo

# Criando o Projeto

- Para criar o projeto, vamos usar a `express-api`. Caso não tenha esse módulo do Node instalado, faça:
  - **`npm install -g express-generator-api`**
- Depois, pra criar o projeto, simplesmente faça:
  - **`express-api <nome_do_projeto>`**
    - cria a pasta de fato
  - **`cd <nome_do_projeto>`**
    - entra na pasta
  - **`npm install`**
    - instala as dependências na `node_modules`.

# Criando o Projeto

- Vamos usar express-fileupload
  - <https://www.npmjs.com/package/express-fileupload>
- Como instalar:
  - Na pasta do projeto recém criado do express faça:
    - `npm install --save express-fileupload`

# Criando a rota:

- **fileupload.routes.js**

```
const express = require('express');  
var router = express.Router();
```

```
router.post('/simplefile', function (req, res) {
```

```
  if (Object.keys(req.files).length == 0) {  
    return res.status(400).send('Nenhum arquivo enviado.');
```

Caso o cliente não tenha selecionado nada.



```
  }  
  // 'sampleFile' é o nome do campo do input no HTML que chamou esse serviço
```

```
  let file = req.files.sampleFile;
```

```
  console.log("Nome do arquivo: " + file.name);
```

```
  console.log("Tamanho do arquivo: " + file.size);
```

```
  //salvando
```

```
  file.mv('/home/jefferson/piw/PIW-2019_1/13-Express-Mongoose-Relacionamentos/uploaded-files/' + file.name,
```

```
  function(err){
```

```
    if(err) return res.status(500).send(err);
```


```
    res.send('Arquivo enviado!');
```

```
  });
```

```
});
```

```
module.exports = router;
```

Nome do arquivo que vem no request, mas você pode escolher outro.



**ATENÇÃO** para o caminho que o arquivo será salvo! Não é recomendado salvar os arquivos do seu projeto na MESMA pasta que o projeto. O interessante é deixar uma pasta separada apenas para isso.

# O app.js

- Configure o arquivo principal do express para aceitar upload de arquivos.

```
var express = require('express');  
var path = require('path');  
var cookieParser = require('cookie-parser');  
var bodyParser = require('body-parser');
```

Importanto o arquivo de rotas.



```
var upload = require('./routes/fileupload.routes');
```

```
var app = express();
```

```
//file upload default
```

```
const fileUpload = require('express-fileupload');  
app.use(fileUpload());
```

Iniciando o express-fileupload



```
app.use(bodyParser.json());  
app.use(bodyParser.urlencoded({ extended: false }));  
app.use(cookieParser());
```

```
app.use('/upload', upload);
```

Criando o endpoint raiz



```
module.exports = app;
```

# Cliente HTML


- Basta agora criar uma página simples HTML para subir nosso arquivo:

```
<html>
<body>
  <form ref='uploadForm'
    id='uploadForm'
    action='http://localhost:3000/upload/simplefile'
    method='post'
    encType="multipart/form-data">
    <input type="file" name="sampleFile" />
    <input type='submit' value='Upload!' />
  </form>
</body>
</html>
```

Nosso serviço Express



**ATENÇÃO** para esse nome `sampleFile`.  
Deve ser usado lá no Express.



Abra esse arquivo com qualquer navegador.