

Multer config

```
// add-X.html
<button type="button" (click)='filePicker.click()' class="btn btn-
success">Sélectionner image</button>
  <div class="form-group">
    <input style="visibility: hidden;" type="file" class="form-control"
#filePicker (change)='onImageSelected($event)' />
  </div>
  <img [src]="imagePreview">
```

```
// add-X.ts
onImageSelected(event: Event) {
  const file = (event.target as HTMLInputElement).files[0];
  this.matchForm.patchValue({ img: file });
  this.matchForm.updateValueAndValidity();
  const reader = new FileReader();
  reader.onload = () => {
    this.imagePreview = reader.result as string
  };
  reader.readAsDataURL(file);
}
```

```
// app.js
const path = require('path');
const multer = require('multer');
const app = express();
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use('/images', express.static(path.join('backend/images')))
```

```
const MIME_TYPE = {
  'image/png': 'png',
  'image/jpeg': 'jpg',
  'image/jpg': 'jpg'
}
```

```
const storage = multer.diskStorage({
  // destination
  destination: (req, file, cb) => {
    const isValid = MIME_TYPE[file.mimetype];
    let error = new Error("Mime type is invalid");
    if (isValid) {
      error = null;
    }
  }
});
```

```
    }  
    cb(null, 'backend/images')  
  },  
  filename: (req, file, cb) => {  
    const name = file.originalname.toLowerCase().split(' ').join('-');  
    const extension = MIME_TYPE[file.mimetype];  
    const imgName = name + '-' + Date.now() + '-crocodocder-' + '.' +  
extension;  
    cb(null, imgName);  
  }  
});
```

```
// Traitement logique  
app.post("/api/addX", multer({ storage: storage }).single('img'), (req, res) => {  
  console.log('file', req.file);  
  url = req.protocol + '://' + req.get('host');  
  img: url + '/images/' + req.file.filename
```