

Multer

HTML

```
<button type="button" (click)='filePicker.click()' class="btn btn-primary py-3 px-5">
    >Select profile image</button>
    <div class="form-group">
        <input style="visibility: hidden;" type="file"
class="form-control" #filePicker
        (change)='onImageSelected($event)' />
    </div>
    <img [src]="imagePreview" *ngIf="imagePreview"
width="100px" height="100px">
```

TS

```
onImageSelected(event: Event) {
    //Selection du fichier
    const file = (event.target as HTMLInputElement).files[0];
    // Ajout d'un attribut img dans l'objet Chef
    this.addTeamForm.patchValue({ img: file });
    // Mise à jour des valeurs du form
    this.addTeamForm.updateValueAndValidity();
    // Creation d'une variable reader pour lire le contenu de
fichiers
    const reader = new FileReader();
    //Déclenchement du event load lors d'une lecture de fichier
avec succès
    reader.onload = () => {
        //affecter le résultat de la lecture dans la variable
imagePreview
        this.imagePreview = reader.result as string
    };
    // lecture du contenu du fichier Blob ou File
    reader.readAsDataURL(file);
}
```

```

this.platService.addPlat(this.plat, this.platForm.value.img).subscribe(
  (data) => {
    console.log('message:', data.message);
  }
);

```

Service

```

addPlat(plat:any, img:File) {
  let formData = new FormData();
  formData.append('name', plat.name);
  formData.append('price', plat.price);
  formData.append('description', plat.description);
  formData.append('img', img);

  return
this.httpClient.post<{message:string}>(`${this.platURL}/api/addPlat`, formData);
}

```

APP.JS

1. Installation multer :

C'est un package qui nous permet de gérer les fichiers entrants dans les requêtes HTTP.

npm install multer

2. Importation du module path :

Le path module fournit de nombreuses fonctionnalités très utiles pour accéder et interagir avec le système de fichiers.

```
const path = require('path');
```

3. Accès aux fichiers images :

```
app.use('/images', express.static(path.join('backend/images')))
```

4. Configuration multer :

Définition des extensions selon le type de media

```

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

```

Traitement ajout image :

```

app.post('/api/addPlat', multer({ storage: storage
}).single('img'), (req, res) => {
  let plat= {};
  console.log('here in add plat');
  let url = req.protocol + '://' + req.get('host');
  console.log('here URL', url);
  plat = new Plat({
    name: req.body.name,
    price: (req.body.price),
    description : req.body.description ,
    img: url + '/images/' + req.file.filename
  });
});

```

```
console.log('added plat , plat);

plat.save();

res.status(200).json({
  message: "added"

});

});
```