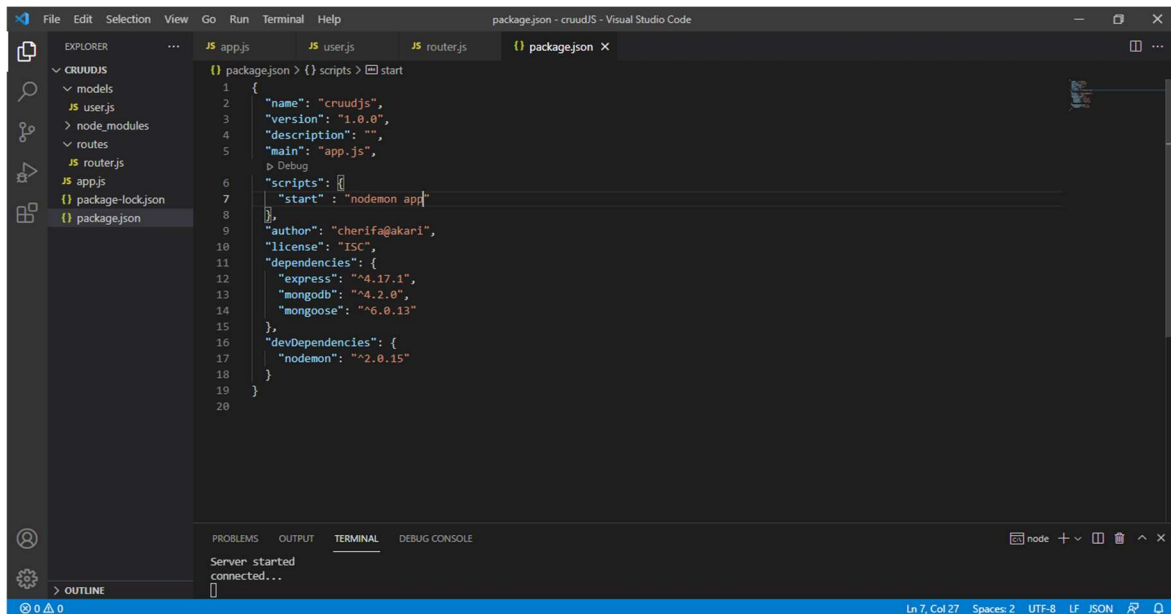


## Compte rendu de TP MongoDB

### CRUD avec Node-JS et MongoDB

**1<sup>ère</sup> étape** : j'ai fait la création d'un dossier CrudJs sur le bureau, j'ai l'ouvrir avec Visual code  
Après pour obtenir le fichier package.Json j'ai taper dans le terminal la commande « npm init »  
Ensuite j'ai installée mongoose , express et MongoDB a l'aide des commandes « npm install »  
Et l'installation de nodemon a travers la commande « npm install nodemon - -save-dev »



**2<sup>ème</sup> étape** : j'ai fait la création d'un fichier app.js dont laquelle j'ai fait le code pour avoir la liaison avec la base des Données 'ProjetDB'

```
const url = 'mongodb://localhost/ProjetDB'
```

lorsque le serveur est connecté sur le port 9000 il s'affiche dans la console Server Started Connected ...

```
con.on('open', () => {
  console.log('connected...')
})
```

```
app.listen(9000, () => {
  console.log('Server started')
})
```

```

1 const express = require('express')
2 const mongoose = require('mongoose')
3 const url = 'mongodb://localhost/ProjetDB'
4 const app = express()
5
6 mongoose.connect(url, {useNewUrlParser:true})
7 const con = mongoose.connection
8
9 con.on('open', () => {
10   console.log('connected...')
11 })
12
13 app.use(express.json())
14
15 const userRouter = require('./routes/router')
16 app.use('/router', userRouter)
17
18 app.listen(9000, () => {
19   console.log('Server started')
20 })

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Cherifa\Desktop\cruudJS\node\_modules\node-run-dev\node-run-dev

[nodemon] 2.0.15

[nodemon] to restart at any time, enter "rs"

[nodemon] watching path(s): \*.\*

[nodemon] watching extensions: js,mjs,json

[nodemon] starting 'node run dev app.js'

Server started

connected...

**3ème étape :** j'ai fait un fichier "user.js" dans le dossier models dont laquelle j'ai défini les attributs ( name , tech, sub)

```

1 const mongoose = require('mongoose')
2
3 const userSchema = new mongoose.Schema({
4   name: {
5     type: String,
6     required: true
7   },
8   tech: {
9     type: String,
10    required: true
11  },
12  sub: {
13    type: Boolean,
14    required: true,
15    default: false
16  }
17 })
18
19
20
21

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

connected...

[nodemon] restarting due to changes...

[nodemon] restarting due to changes...

[nodemon] starting 'node run dev app.js'

Server started

connected...

[nodemon] restarting due to changes...

[nodemon] starting 'node run dev app.js'

Server started

connected...

**4ème étape :** j'ai fait un fichier "router.js" dans un dossier routes dont laquelle j'ai défini les méthodes à tester par Postman ( GET , POST , PATCH,DELETE ) . exemple de méthode POST :

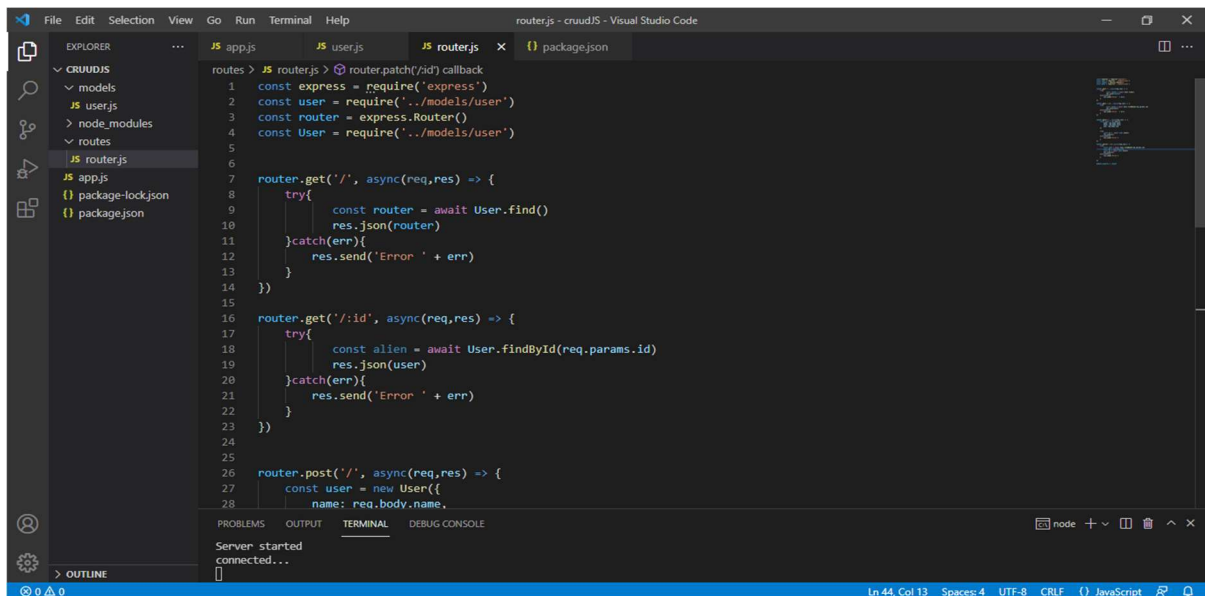
```

router.post('/', async(req,res) => {
  const user = new User({
    name: req.body.name,
    tech: req.body.tech,
    sub: req.body.sub
  }) try{
    const a1 = await user.save()
    res.json(a1)
  }catch(err){
    res.send('Error')
  }
})

```

Methode delete :

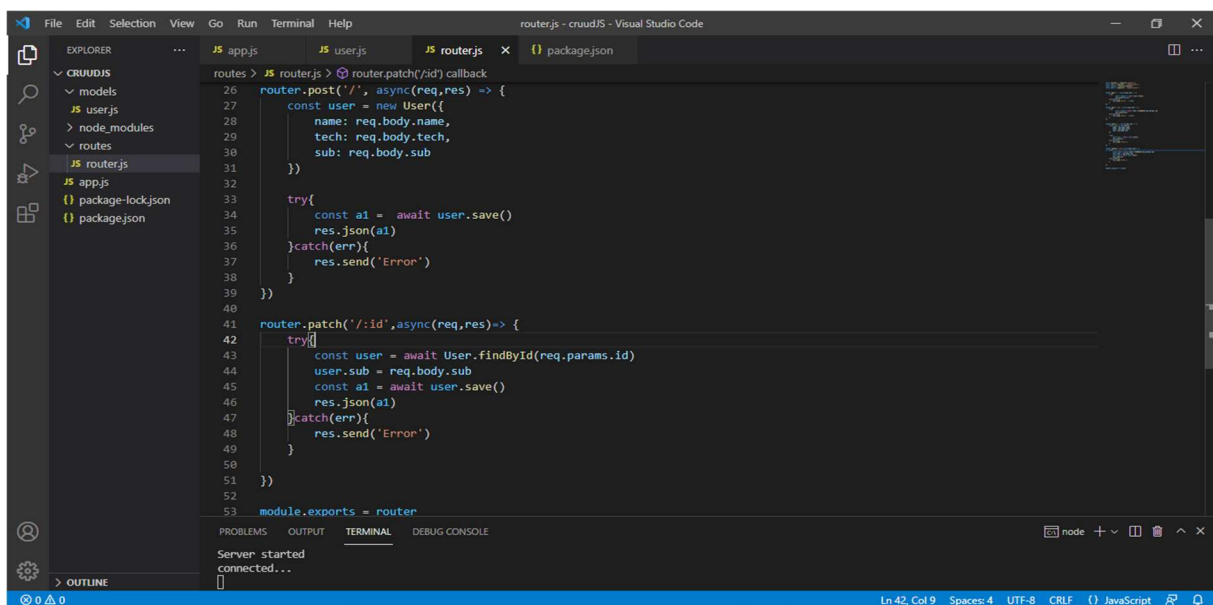
```
router.delete('/:id', async(req, res) => {
  try{
    const user = await User.findById(req.params.id)
    user.sub = req.body.sub
    const a1 = await user.remove()
    res.json('suppression avec succes')
  } catch(err){
    res.send('Error')
  }
})
```



This screenshot shows the Visual Studio Code editor with the 'router.js' file open. The Explorer sidebar on the left shows the project structure: 'routes' > 'models' > 'user.js', 'node\_modules', 'routes', 'router.js', 'app.js', 'package-lock.json', and 'package.json'. The main editor area displays the following code:

```
1 const express = require('express')
2 const user = require('../models/user')
3 const router = express.Router()
4 const User = require('../models/user')
5
6
7 router.get('/', async(req, res) => {
8   try{
9     const router = await User.find()
10    res.json(router)
11  } catch(err){
12    res.send('Error ' + err)
13  }
14 })
15
16
17 router.get('/:id', async(req, res) => {
18   try{
19     const alien = await User.findById(req.params.id)
20     res.json(user)
21   } catch(err){
22     res.send('Error ' + err)
23   }
24 })
25
26 router.post('/', async(req, res) => {
27   const user = new User({
28     name: req.body.name,
29     tech: req.body.tech,
30     sub: req.body.sub
31   })
32
33   try{
34     const a1 = await user.save()
35     res.json(a1)
36   } catch(err){
37     res.send('Error')
38   }
39 })
40
41 router.patch('/:id', async(req, res) => {
42   try{
43     const user = await User.findById(req.params.id)
44     user.sub = req.body.sub
45     const a1 = await user.save()
46     res.json(a1)
47   } catch(err){
48     res.send('Error')
49   }
50 })
51
52
53 module.exports = router
```

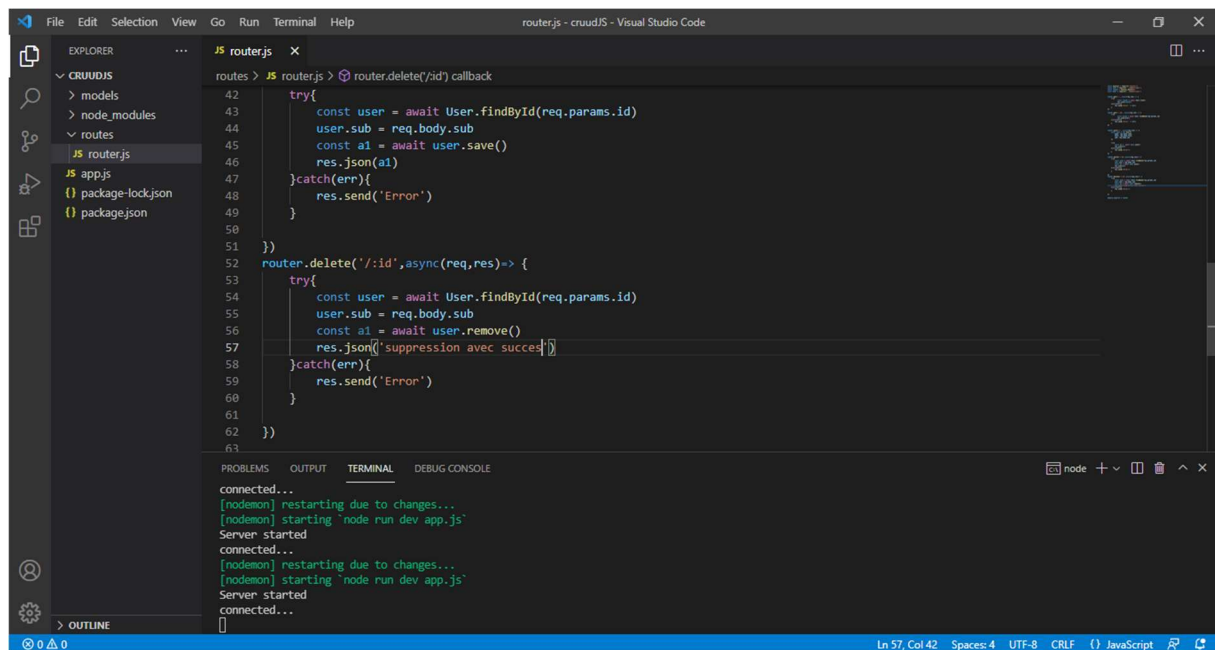
The bottom status bar indicates 'Ln 44, Col 13', 'Spaces: 4', 'UTF-8', 'CRLF', and 'JavaScript'.



This screenshot shows the Visual Studio Code editor with the 'router.js' file open. The Explorer sidebar on the left shows the project structure: 'routes' > 'models' > 'user.js', 'node\_modules', 'routes', 'router.js', 'app.js', 'package-lock.json', and 'package.json'. The main editor area displays the following code:

```
26 router.post('/', async(req, res) => {
27   const user = new User({
28     name: req.body.name,
29     tech: req.body.tech,
30     sub: req.body.sub
31   })
32
33   try{
34     const a1 = await user.save()
35     res.json(a1)
36   } catch(err){
37     res.send('Error')
38   }
39 })
40
41 router.patch('/:id', async(req, res) => {
42   try{
43     const user = await User.findById(req.params.id)
44     user.sub = req.body.sub
45     const a1 = await user.save()
46     res.json(a1)
47   } catch(err){
48     res.send('Error')
49   }
50 })
51
52
53 module.exports = router
```

The bottom status bar indicates 'Ln 42, Col 9', 'Spaces: 4', 'UTF-8', 'CRLF', and 'JavaScript'.



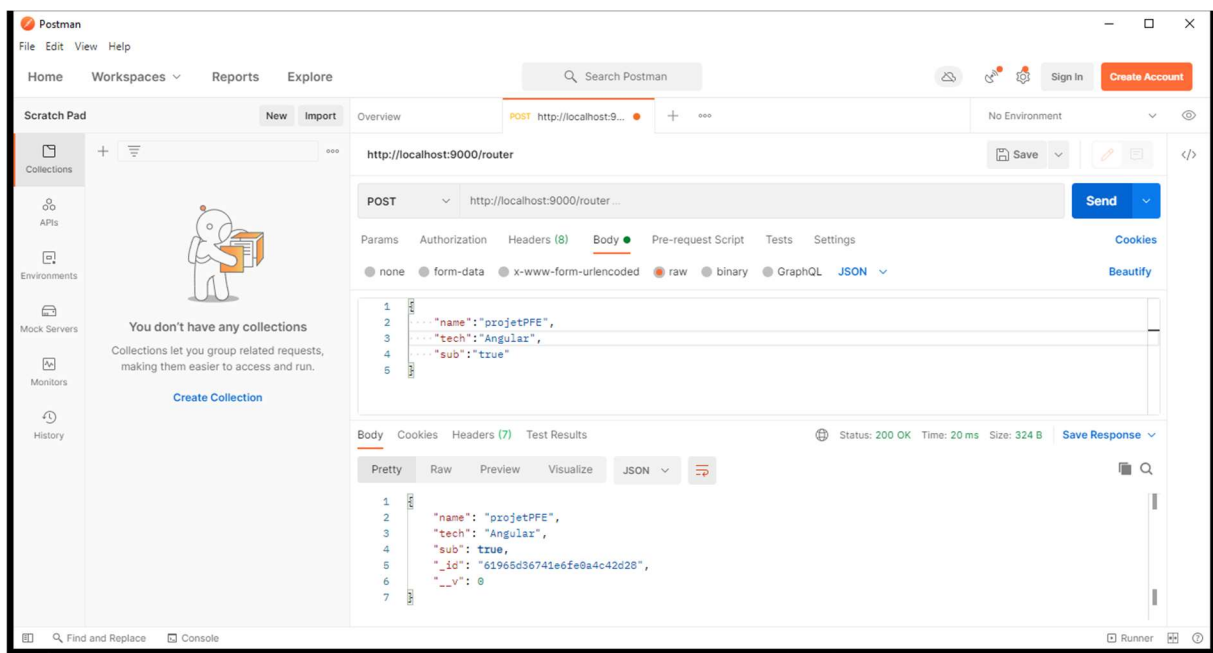
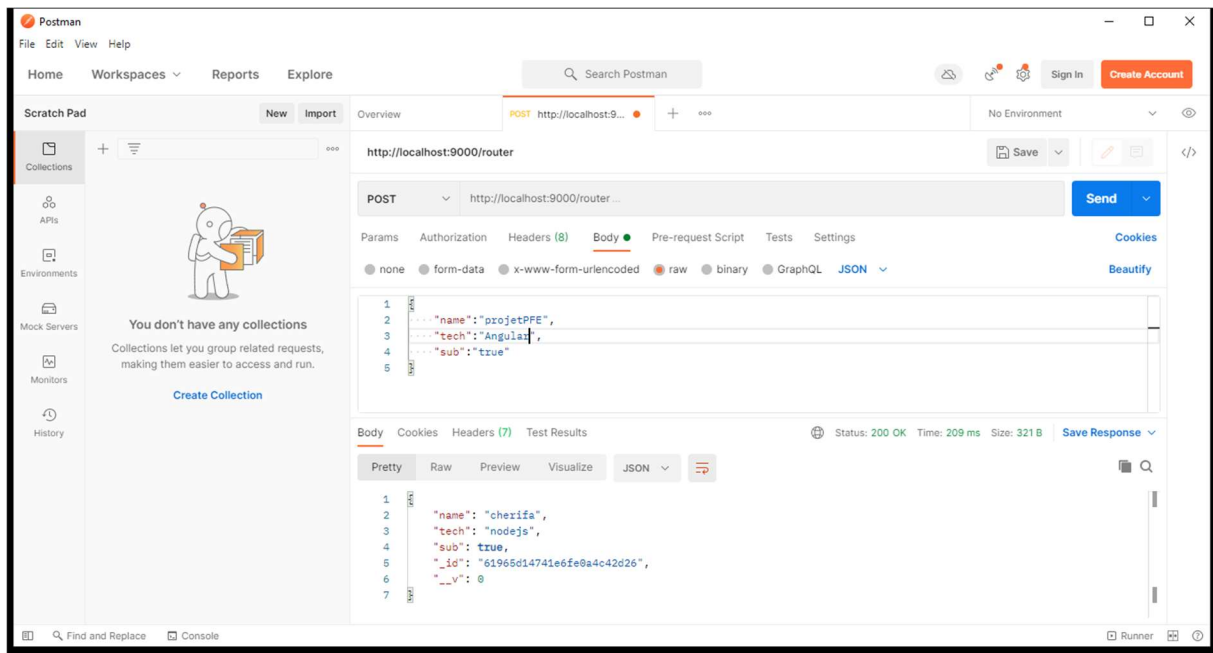
```
42     try{
43         const user = await User.findById(req.params.id)
44         user.sub = req.body.sub
45         const a1 = await user.save()
46         res.json(a1)
47     }catch(err){
48         res.send('Error')
49     }
50 }
51 })
52 router.delete('/:id', async(req, res)=> {
53     try{
54         const user = await User.findById(req.params.id)
55         user.sub = req.body.sub
56         const a1 = await user.remove()
57         res.json({suppression avec succes:1})
58     }catch(err){
59         res.send('Error')
60     }
61 }
62 })
63 }
```

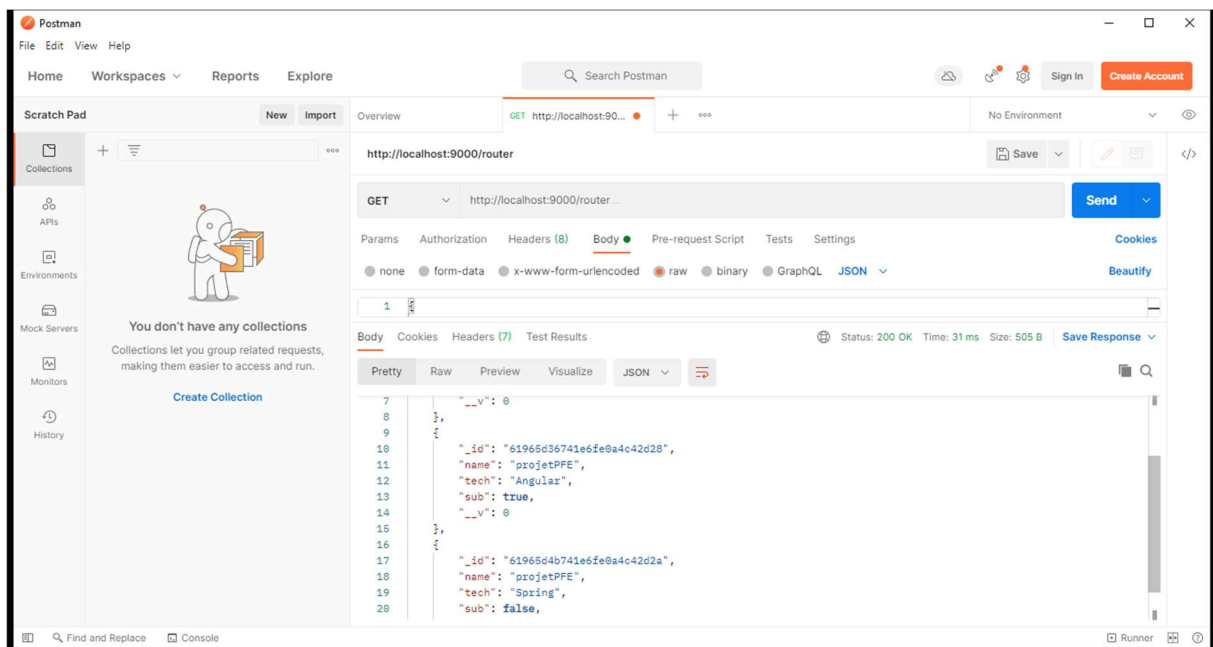
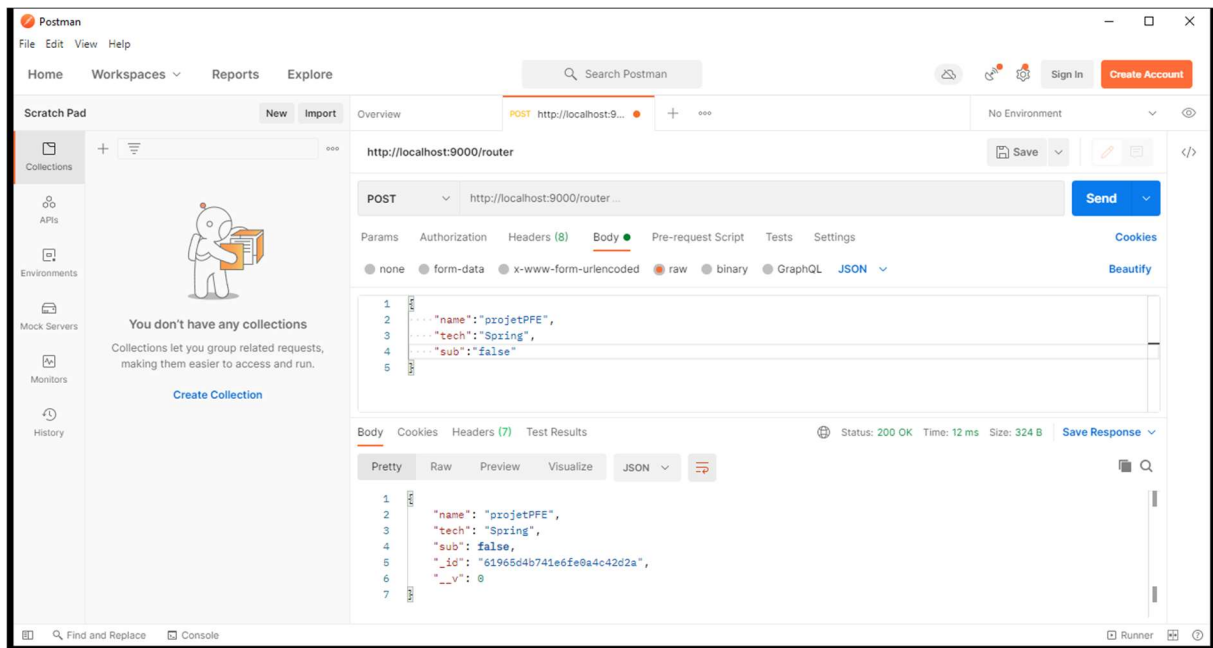
connected...  
[nodemon] restarting due to changes...  
[nodemon] starting 'node run dev app.js'  
Server started  
connected...  
[nodemon] restarting due to changes...  
[nodemon] starting 'node run dev app.js'  
Server started  
connected...

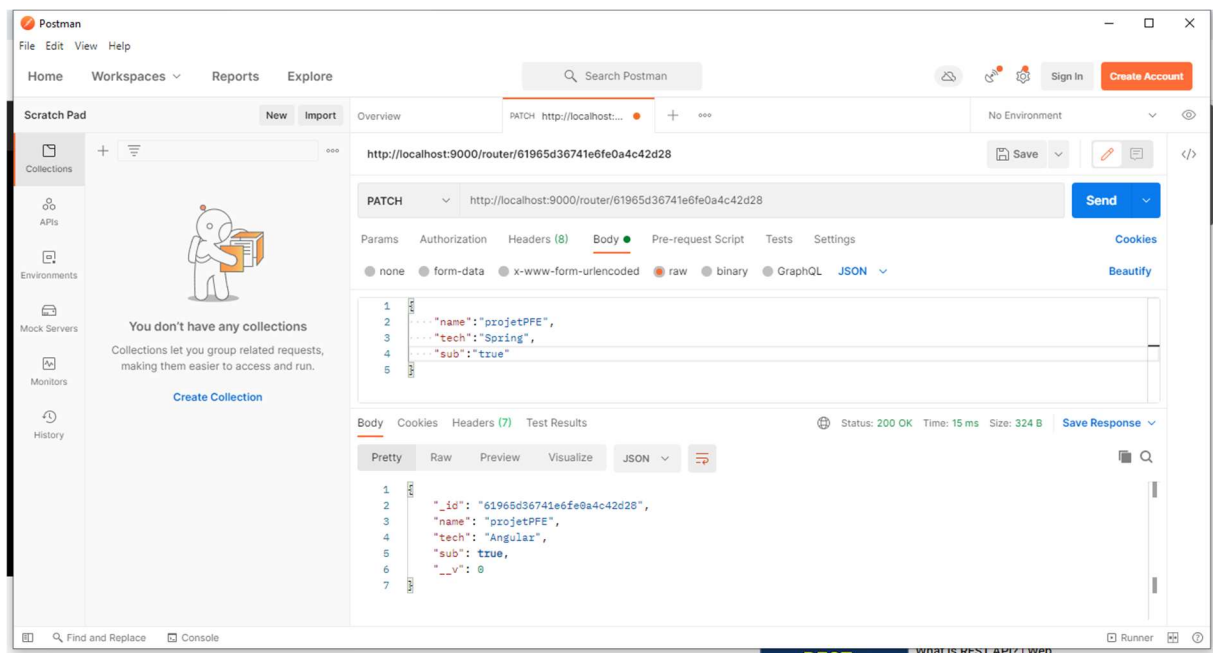
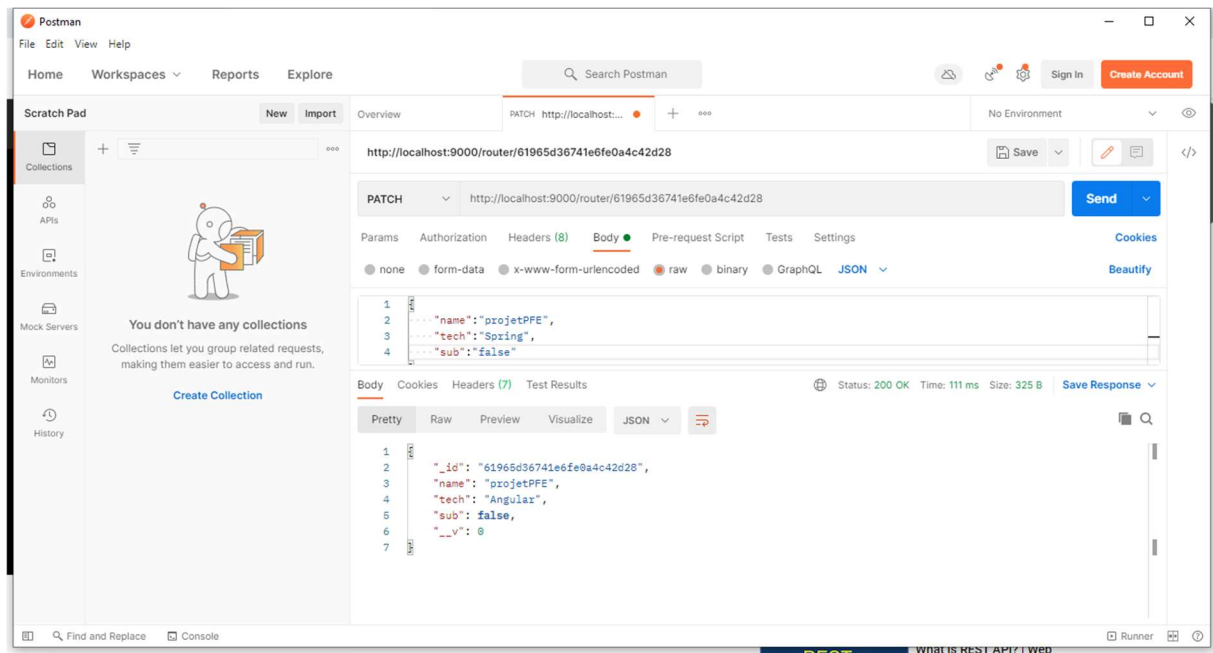
Et pour lancer le serveur a l'aide de commande « nodemon run dev »

**Voici les Tests a travers POSTMAN**

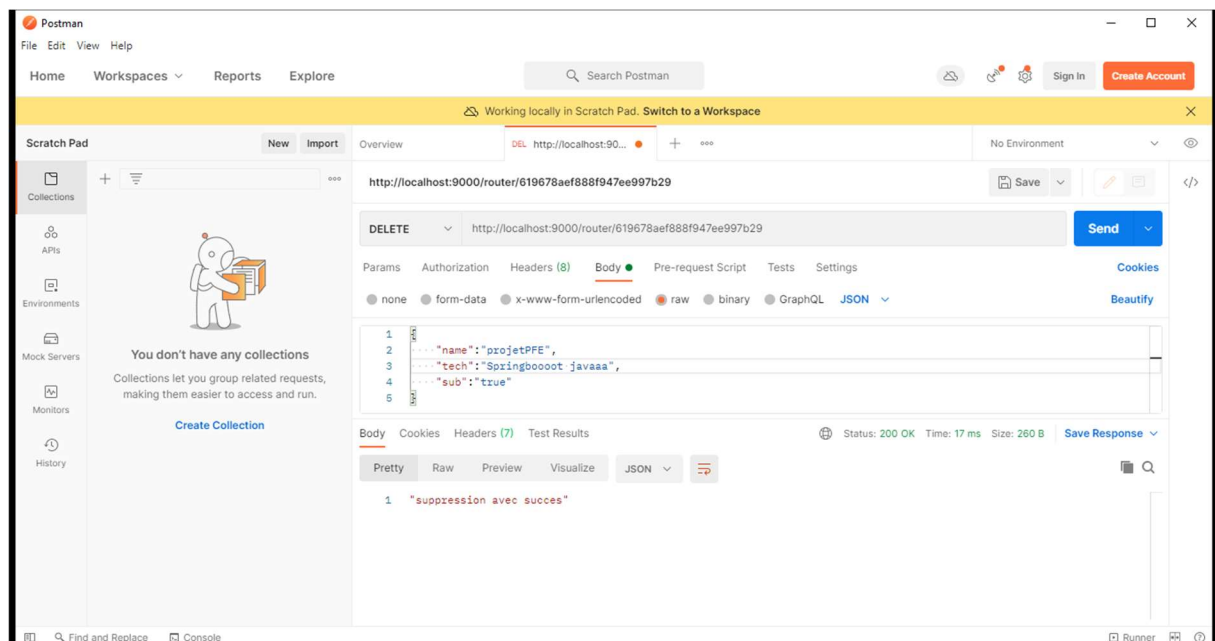
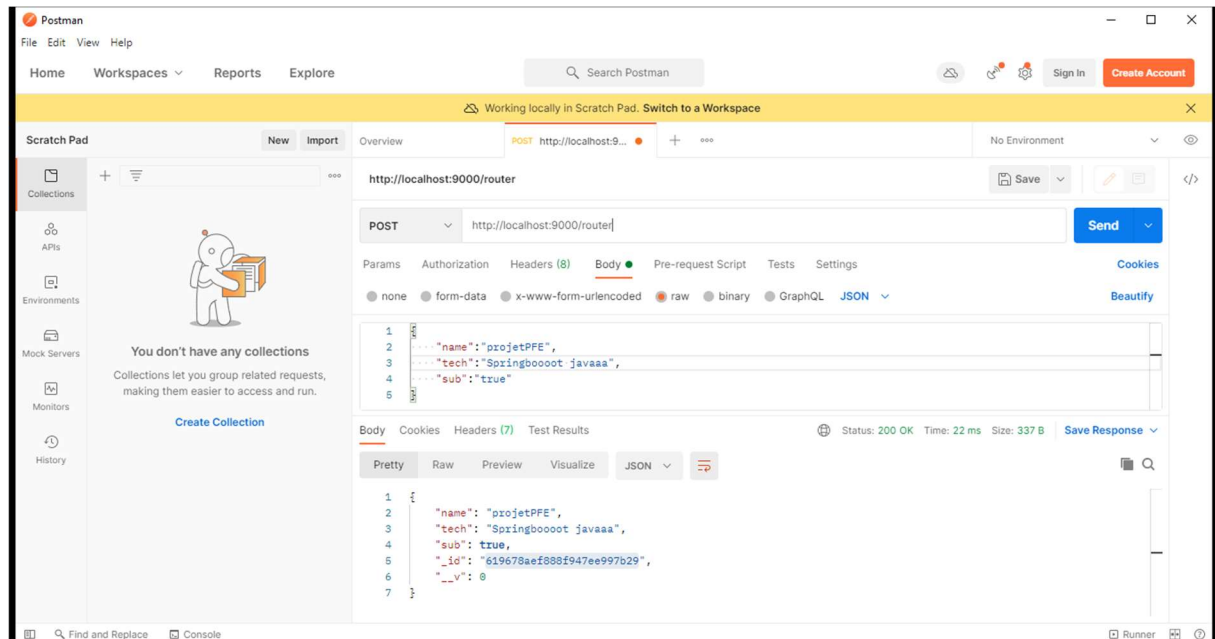
<http://localhost:9000/router>



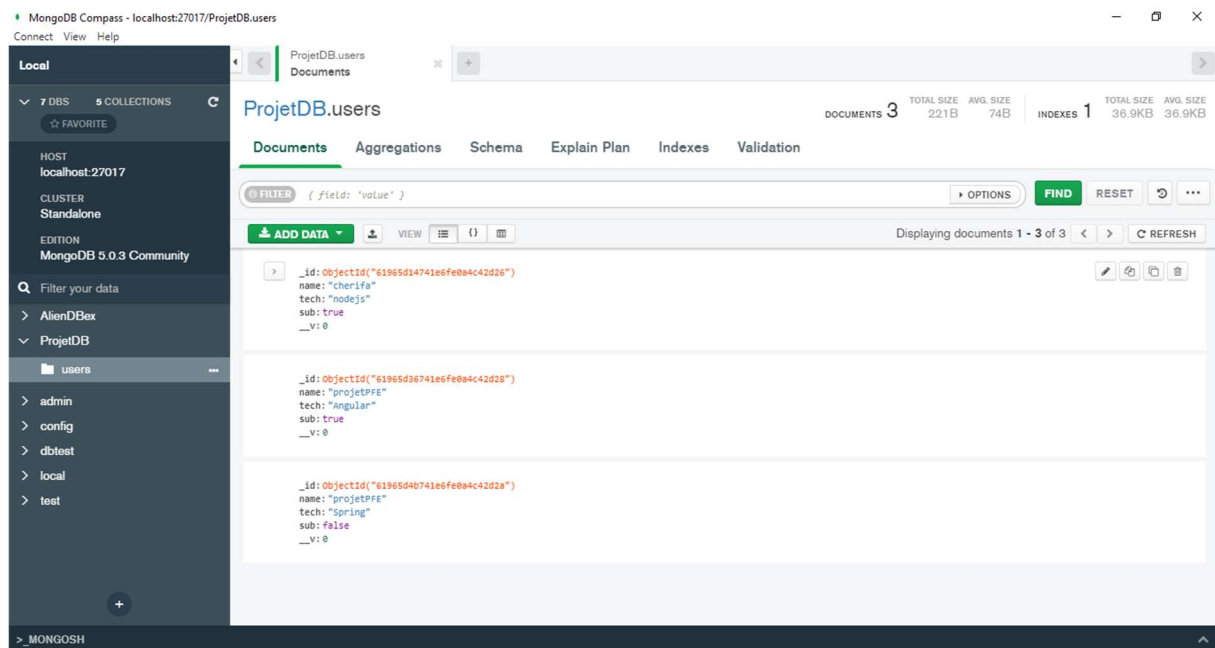
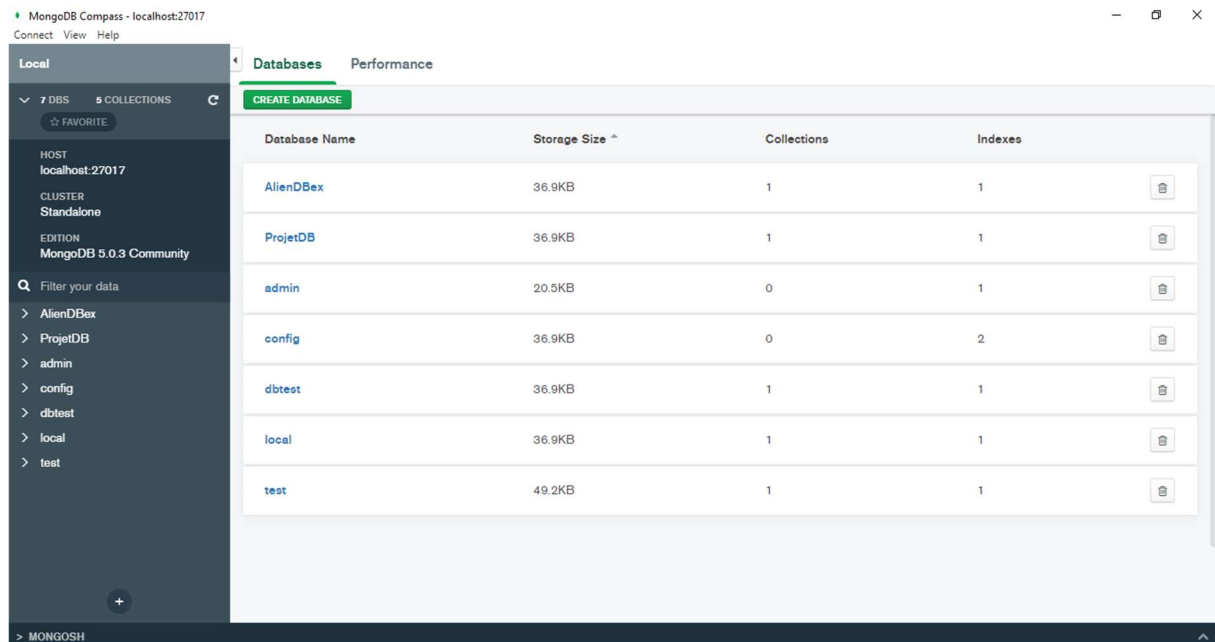




Voici la méthode DELETE :







**Chérifa Akkari**

**3LM3/TP1**