Atelier 2:

REST APIs

1. Creation de fichier ‘packages.json:

```
houdaifa@houdaifa:~/everything/node/ateliers/atelier2$ npm init -y
Wrote to /home/houdaifa/everything/node/ateliers/atelier2/package.json:

{
  "name": "atelier2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}
```

2. Installer les modules (express) :

```
houdaifa@houdaifa:~/everything/node/ateliers/atelier2$ npm install express

added 65 packages, and audited 66 packages in 5s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

3. Fichier functions (success et error) :

atelier2 > JS functions.js > ...

```
1   const success = (result) => {
2       return {
3           status: 'success',
4           result: result
5       }
6   }
7
8   const error = (message) => {
9       return {
10          status : 'error',
11          result: message
12      }
13  }
14
15  exports.success = success
16  exports.error = error
```

4. Fichier index.js :
    a. get (read), post (update) et delete par id:

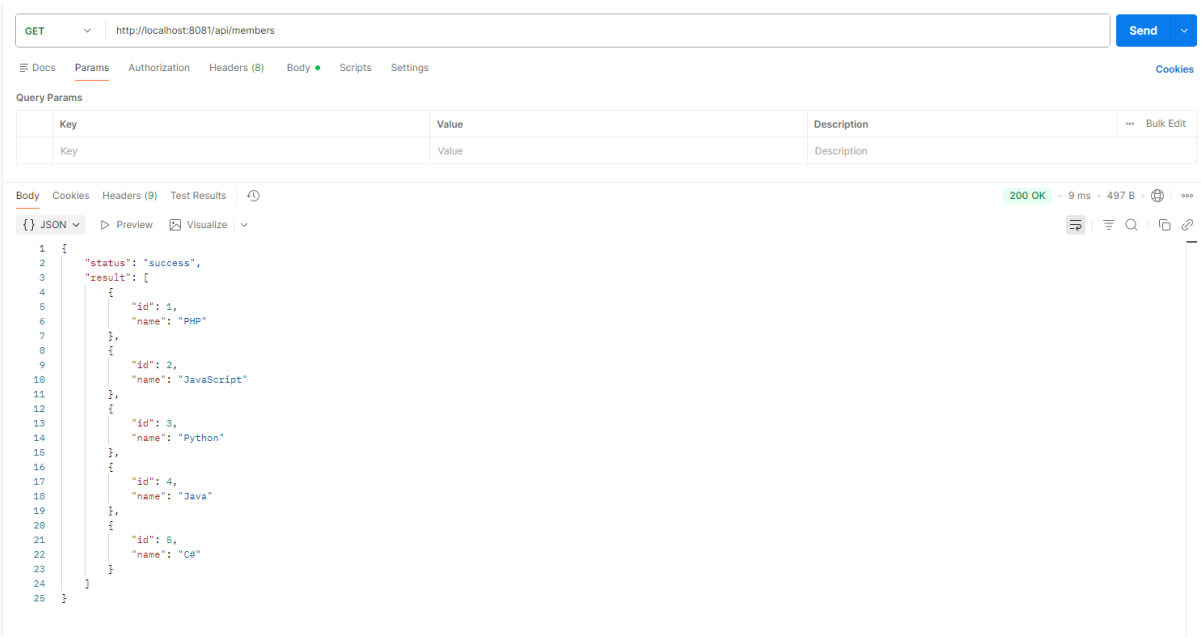
```
membersRouter.route('/:id')
    .get((req,res) => {
        let id = parseInt(req.params.id)
        let member = members.find(m => m.id === id)

        if (!member) {
            return res.json(error("Member not found"))
        }
        res.json(success(member))
    })

    .post((req,res) => {
        let id = parseInt(req.params.id)
        let member = members.find(m => m.id === id)

        if (!member) {
            return res.json(error("Member not found"))
        }

        for (let i = 0; i < members.length; i++) {
            if (req.body.name === members[i].name && members[i].id !== id) {
                return res.json(error("Member name already exists"))
            }
        }

        member.name = req.body.name
        res.json(success(true))
    })
    .delete((req,res) => {
        let id = parseInt(req.params.id)
        let memberIndex = members.findIndex(m => m.id === id)

        if (memberIndex === -1) {
            return res.json(error("Member not found"))
        }

        members.splice(memberIndex, 1)
        res.json(success(true))
    })
```

b. get (read tous les members), post (create):

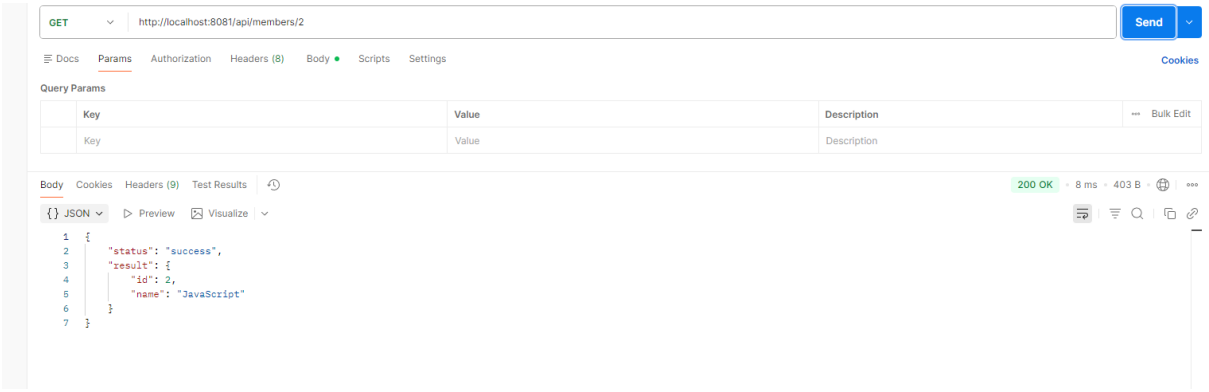
```
membersRouter.route('/')
    .get((req, res) => {
        res.json(success(members))
    })
    .post((req, res) => {
        if (!req.body.name){
            return res.json(error("Missing required parameter: name"))
        }
        for (let i = 0; i < members.length; i++) {
            if (req.body.name === members[i].name) {
                return res.json(error("Member name already exists"))
            }
        }
        let newMember = {
            id: currentId++,
            name: req.body.name
        }
        members.push(newMember)
        res.json(success(newMember))
    })
```

5. Tests:

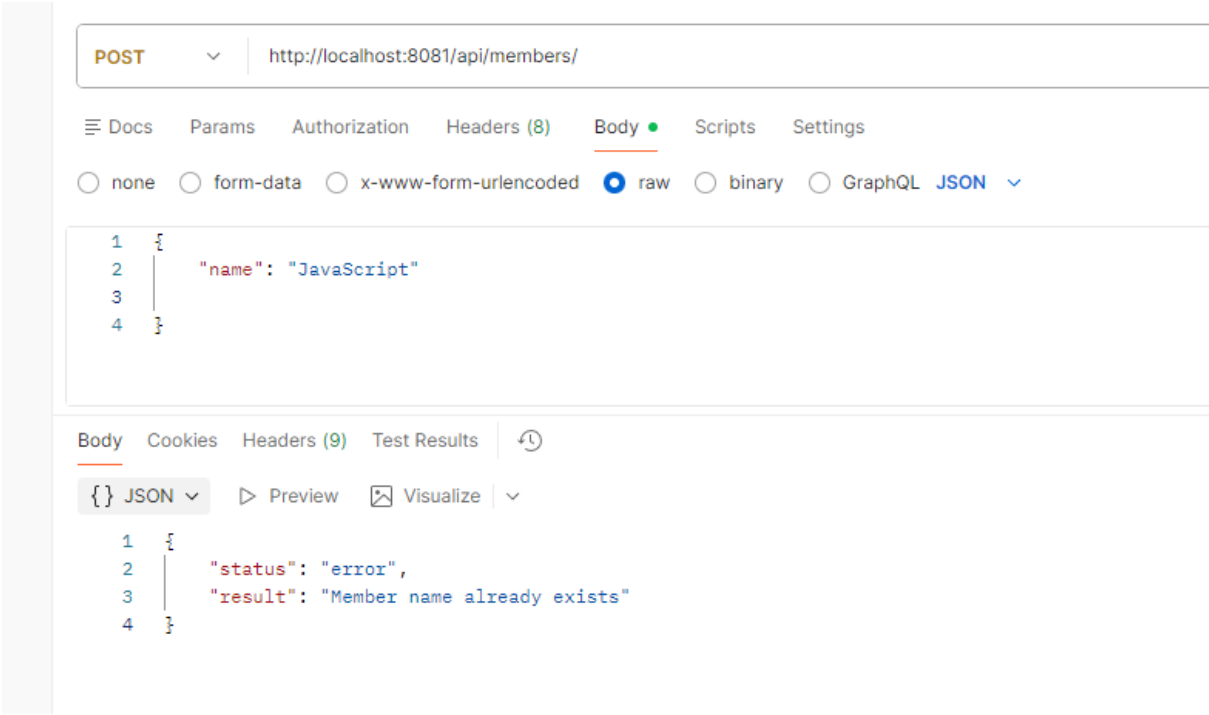
a. Recuperer tous les members:



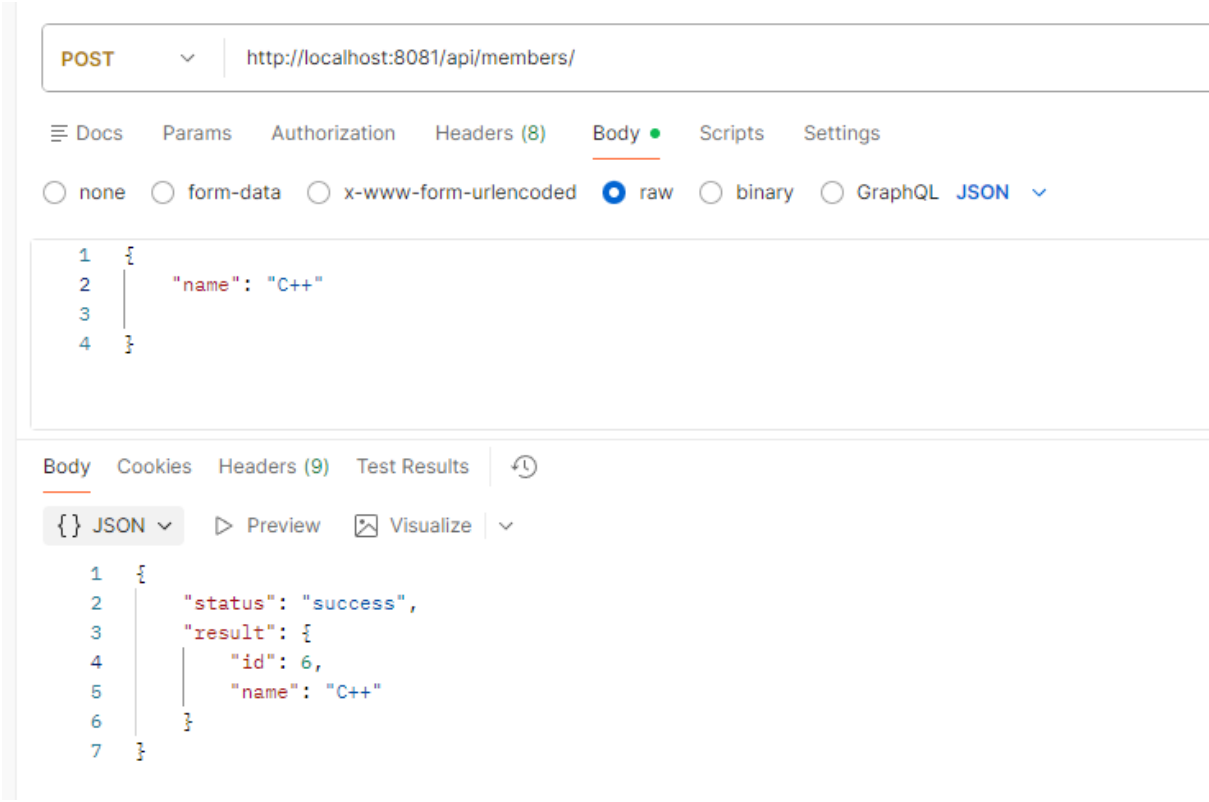
b. Recuperer par ID:



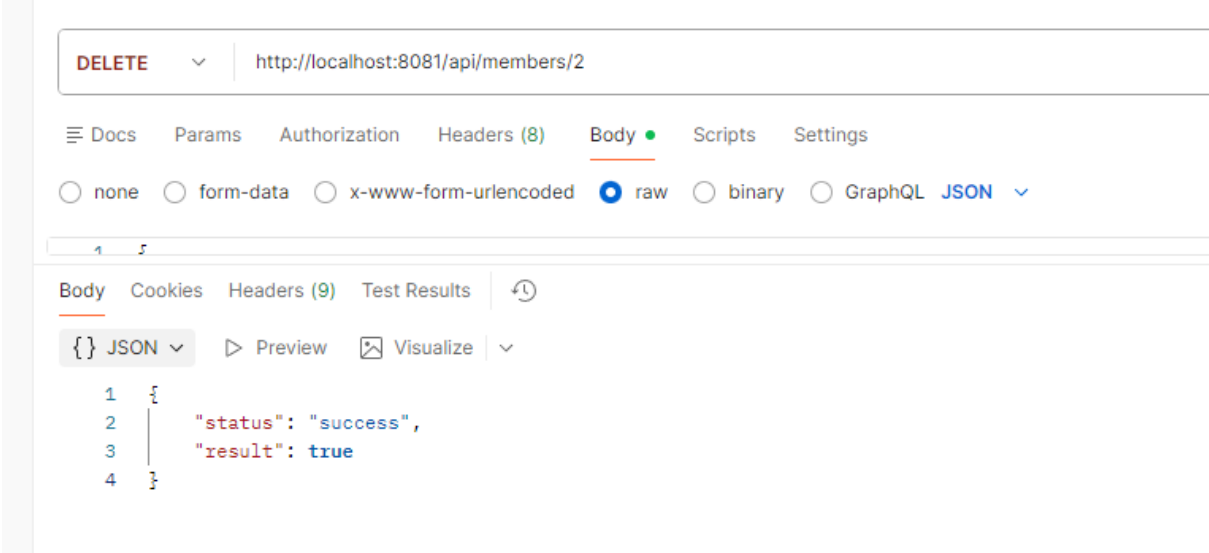
c. Ajouter un member qui existe déjà (error) :



d. Ajouter un neaveau membre (success) :



e. Supprimer un membre (success) :



f. Metre a jour un membre (success) :