



# API REST GF tests postman



16/05/2025

## Petit rappel :

Endpoint :

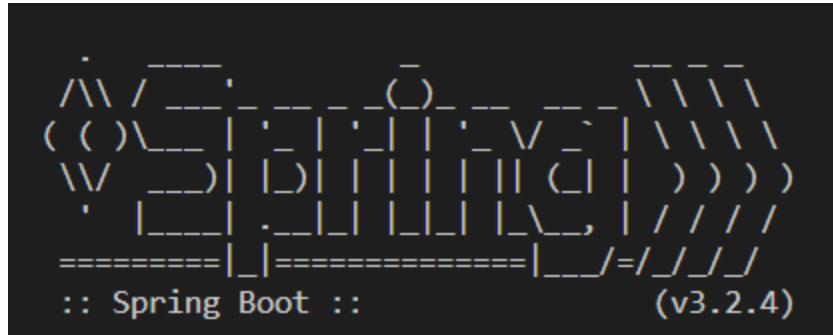
Un endpoint est une « adresse » (URL) d'une API à laquelle je peux envoyer un

C'est le point d'entrée de l'application pour communiquer avec elle depuis un c

**Problème rencontré : Pas de réponse car PhpMyAdmin déconnecté. Vérifier régulièrement que tout tourne bien, y compris Wamp et le serveur Spring Boot.**



**Erreur "Chaise clavier" classique ⇒ mauvaise méthode (verbe http) sélectionné dans postman.**



## Les principaux codes de retour HTTP :

Code	Signification	Explication simple
200	OK	Tout s'est bien passé, la requête a réussi.
201	Created	La ressource a été créée avec succès (ex : après un POST).
204	No Content	La requête a réussi, mais il n'y a rien à renvoyer.
400	Bad Request	La requête est mal formée ou il manque des informations.
401	Unauthorized	Tu n'es pas authentifié, il faut t'identifier (login manquant).
403	Forbidden	Tu n'as pas le droit d'accéder à cette ressource.
404	Not Found	La ressource demandée n'existe pas.
409	Conflict	Il y a un conflit (ex : doublon, ressource déjà existante).
500	Internal Server Error	Erreur côté serveur, ce n'est pas de ta faute.

### Explications courtes :

- **200 OK** : Tout est bon, tu reçois la réponse attendue.
- **201 Created** : Ta création (ex : nouvel utilisateur) a marché, l'objet a été ajouté.
- **204 No Content** : Ça a marché, mais il n'y a rien à afficher (souvent après un DELETE ou un PUT).

- **400 Bad Request** : Tu as fait une erreur dans ta requête (ex : JSON mal écrit, champ manquant).
- **401 Unauthorized** : Tu dois t'authentifier (manque un token, mot de passe, etc.).
- **403 Forbidden** : Tu es identifié, mais tu n'as pas le droit d'accéder à cette action.
- **404 Not Found** : L'URL ou l'objet demandé n'existe pas.
- **409 Conflict** : Il y a un problème de doublon ou d'état (ex : même email déjà utilisé).
- **500 Internal Server Error** : Le serveur a planté, ce n'est pas de ta faute.

### Couleurs des codes de retour dans Postman :

Code(s)	Couleur dans Postman	Signification rapide
2xx (200, 201)	 Vert	Succès : tout s'est bien passé
3xx (301, 302)	 Jaune	Redirection : la ressource a bougé
4xx (400, 404)	 Orange	Erreur côté client : problème dans la requête
5xx (500, 502)	 Rouge	Erreur côté serveur : bug ou crash

### Détail :

- **Vert (2xx)** : Succès, la requête a abouti (ex : 200 OK, 201 Created).
- **Jaune (3xx)** : Redirection, la ressource a changé d'adresse (rare en API REST).
- **Orange (4xx)** : Erreur de ta part (ex : mauvaise URL, mauvais JSON, accès refusé).
- **Rouge (5xx)** : Erreur du serveur, ce n'est pas ta faute (le backend a planté).






































### Voici tous les tests effectués :

Entité	Action	Méthode	URL	Body JSON (e
-----	-----	-----	-----	-----

Role	Lister tous	GET	/roles/	-
	Détail par ID	GET	/roles/1	-
	Créer	POST	/roles/create	{ "libelle": "Forr
	Modifier	PUT	/roles/update/1	{ "libelle": "Foi
	Désactiver	PUT	/roles/disable/1	-
Utilisateur	Lister tous	GET	/utilisateurs/	-
	Détail par matricule	GET	/utilisateurs/STG011	-
	Créer	POST	/utilisateurs/create	{ "matricule":
	Modifier	PUT	/utilisateurs/update/STG011	{ "nom":
	Désactiver	PUT	/utilisateurs/disable/STG011	-
Formation	Lister tous	GET	/formations/	-
	Détail par ID	GET	/formations/1	-
	Créer	POST	/formations/create	{ "libelle": "Ji
	Modifier	PUT	/formations/update/1	{ "libelle": "
	Désactiver	PUT	/formations/disable/1	-
Centre	Lister tous	GET	/centres/	-
	Détail par ID	GET	/centres/1	-
	Créer	POST	/centres/create	{ "nom": "Cen
	Modifier	PUT	/centres/update/1	{ "nom": "Ce
	Désactiver	PUT	/centres/disable/1	-
Module	Lister tous	GET	/modules/	-
	Détail par ID	GET	/modules/1	-
	Créer	POST	/modules/create	{ "nom": "Mo
	Modifier	PUT	/modules/update/1	{ "nom": "M
	Désactiver	PUT	/modules/disable/1	-
Sequence	Lister tous	GET	/sequences/	-
	Détail par ID	GET	/sequences/1	-
	Lister par module	GET	/sequences/module/1	-
	Créer	POST	/sequences/create	{ "libelle": "
	Modifier	PUT	/sequences/update/1	{ "libelle":
	Désactiver	PUT	/sequences/disable/1	-

### Détails test par test pour chaque entité :

# 1 → Entité Utilisateur

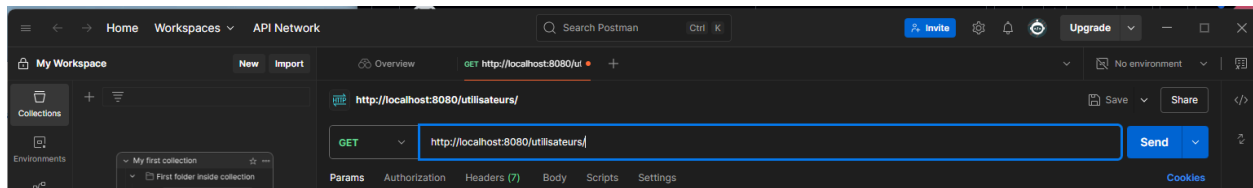
←		→		id	matricule	nom	prenom	adresse_mail	adresse_postal	code_postal	ville	mot_de_passe	statut	role_id
<input type="checkbox"/>				17	STG011	Lemoine	Alex	alex.lemoine@example.com	14 rue des Lilas	69003	Lyon2	stagiaire1	0	5
<input type="checkbox"/>				18	EMP001	Dupont	Marie	marie.dupont@example.com	12 rue de la République	75001	Paris	mdp123	0	5
<input type="checkbox"/>				19	STG012	Roux	Laura	laura.roux@example.com	7 chemin Vert	13002	Marseille	stagiaire2	1	7
<input type="checkbox"/>				20	EMP002	Martin	Luc	luc.martin@example.com	5 avenue des Champs-Élysées	75008	Paris	mdp456	1	7
<input type="checkbox"/>				21	STG013	Fournier	Thomas	thomas.fournier@example.com	22 impasse des Roses	59000	Lille	stagiaire3	0	5
<input type="checkbox"/>				22	EMP003	Bernard	Sophie	sophie.bernard@example.com	8 rue de la Paix	06000	Nice	mdp789	0	NULL
<input type="checkbox"/>				23	STG014	Girard	Manonin	manon.girard@example.com	3 allée des Tilleuls	33000	Bordeaux	stagiaire4	0	5
<input type="checkbox"/>				24	EMP004	Dubois	Pierre	pierre.dubois@example.com	15 cours Gambetta	69001	Lyon	mdp101	1	NULL
<input type="checkbox"/>				25	STG015	Boyer	Hugo	hugo.boyer@example.com	9 rue des Peupliers	75017	Paris	stagiaire5	0	NULL
<input type="checkbox"/>				26	EMP005	Leroy	Julie	julie.leroy@example.com	22 boulevard de la Liberté	59000	Lille	mdp112	1	NULL
<input type="checkbox"/>				27	EMP006	Moreau	Antoine	antoine.moreau@example.com	7 rue du Port	33000	Bordeaux	mdp131	1	NULL
<input type="checkbox"/>				28	EMP007	Laurent	Camille	camille.laurent@example.com	3 place Bellecour	69002	Lyon	mdp415	1	NULL
<input type="checkbox"/>				29	EMP008	Simon	Élodie	elodie.simon@example.com	18 rue de la Pompe	75116	Paris	mdp161	0	5

## Table utilisateur PhpMyAdmin AVANT TESTS.

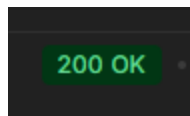
### TEST 01 :

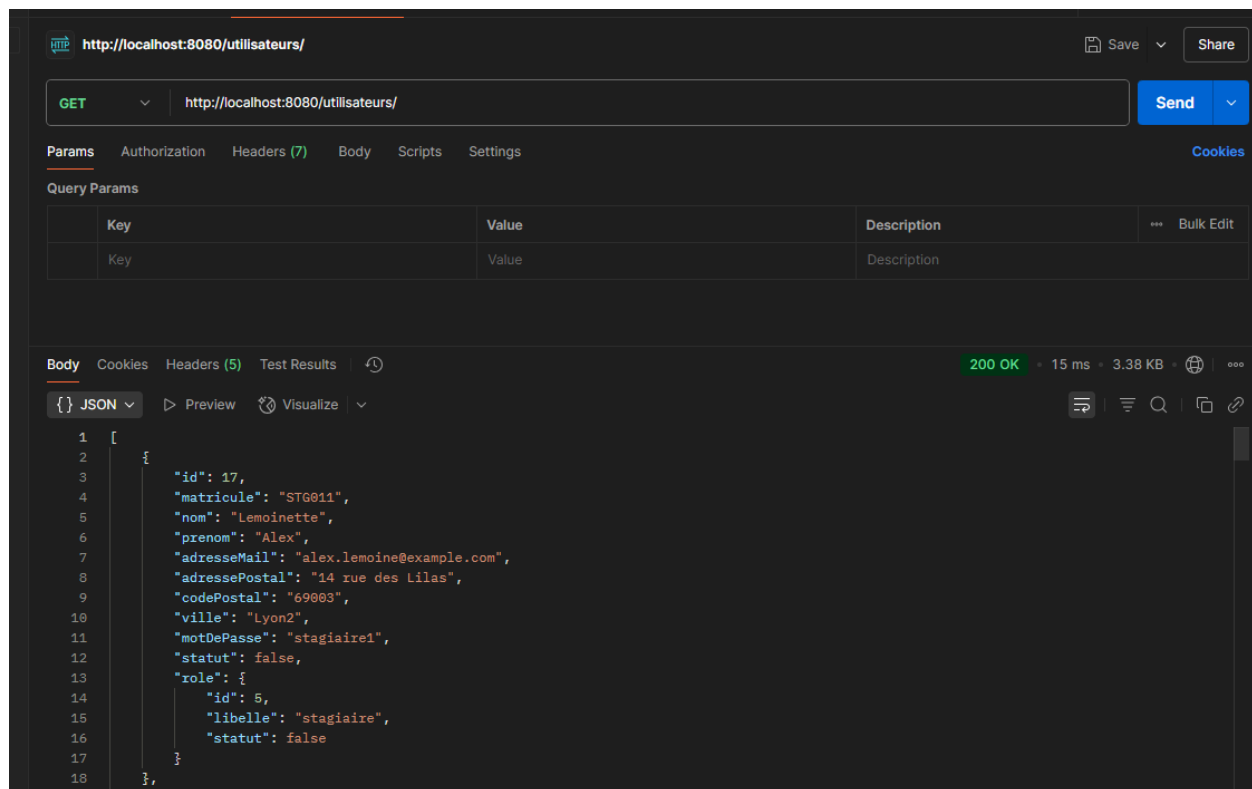
#### GET > RECUPERER / READ

GET <http://localhost:8080/utilisateurs/> → liste tous les utilisateurs (JSON)



### Résultat Postman :





### Liste des utilisateurs du gestionnaire de formation récupéré au format JSON :

```
[
  {
    "id": 17,
    "matricule": "STG011",
    "nom": "LemoINETTE",
    "prenom": "Alex",
    "adresseMail": "alex.lemoine@example.com",
    "adressePostal": "14 rue des Lilas",
    "codePostal": "69003",
    "ville": "Lyon2",
    "motDePasse": "stagiaire1",
    "statut": false,
    "role": {
      "id": 5,
      "libelle": "stagiaire",
      "statut": false
    }
  }
]
```

```

    }
  },
  {
    "id": 18,
    "matricule": "EMP001",
    "nom": "Dupont",
    "prenom": "Marie",
    "adresseMail": "marie.dupont@example.com",
    "adressePostal": "12 rue de la République",
    "codePostal": "75001",
    "ville": "Paris",
    "motDePasse": "mdp123",
    "statut": false,
    "role": {
      "id": 5,
      "libelle": "stagiaire",
      "statut": false
    }
  },
  {
    "id": 19,
    "matricule": "STG012",
    "nom": "Roux",
    "prenom": "Laura",
    "adresseMail": "laura.roux@example.com",
    "adressePostal": "7 chemin Vert",
    "codePostal": "13002",
    "ville": "Marseille",
    "motDePasse": "stagiaire2",
    "statut": true,
    "role": {
      "id": 7,
      "libelle": "intervenant",
      "statut": false
    }
  },
},

```

```

{
  "id": 20,
  "matricule": "EMP002",
  "nom": "Martin",
  "prenom": "Luc",
  "adresseMail": "luc.martin@example.com",
  "adressePostal": "5 avenue des Champs-Élysées",
  "codePostal": "75008",
  "ville": "Paris",
  "motDePasse": "mdp456",
  "statut": true,
  "role": {
    "id": 7,
    "libelle": "intervenant",
    "statut": false
  }
},
{
  "id": 21,
  "matricule": "STG013",
  "nom": "Fournier",
  "prenom": "Thomas",
  "adresseMail": "thomas.fournier@example.com",
  "adressePostal": "22 impasse des Roses",
  "codePostal": "59000",
  "ville": "Lille",
  "motDePasse": "stagiaire3",
  "statut": false,
  "role": {
    "id": 5,
    "libelle": "stagiaire",
    "statut": false
  }
},
{
  "id": 22,

```



```

    "matricule": "EMP003",
    "nom": "Bernard",
    "prenom": "Sophie",
    "adresseMail": "sophie.bernard@example.com",
    "adressePostal": "8 rue de la Paix",
    "codePostal": "06000",
    "ville": "Nice",
    "motDePasse": "mdp789",
    "statut": false,
    "role": null
  },
  {
    "id": 23,
    "matricule": "STG014",
    "nom": "Girard",
    "prenom": "Manonin",
    "adresseMail": "manon.girard@example.com",
    "adressePostal": "3 allée des Tilleuls",
    "codePostal": "33000",
    "ville": "Bordeaux",
    "motDePasse": "stagiaire4",
    "statut": false,
    "role": {
      "id": 5,
      "libelle": "stagiaire",
      "statut": false
    }
  },
  {
    "id": 24,
    "matricule": "EMP004",
    "nom": "Dubois",
    "prenom": "Pierre",
    "adresseMail": "pierre.dubois@example.com",
    "adressePostal": "15 cours Gambetta",
    "codePostal": "69001",

```

```
"ville": "Lyon",
"motDePasse": "mdp101",
"statut": true,
"role": null
},
{
  "id": 25,
  "matricule": "STG015",
  "nom": "Boyer",
  "prenom": "Hugo",
  "adresseMail": "hugo.boyer@example.com",
  "adressePostal": "9 rue des Peupliers",
  "codePostal": "75017",
  "ville": "Paris",
  "motDePasse": "stagiaire5",
  "statut": false,
  "role": null
},
{
  "id": 26,
  "matricule": "EMP005",
  "nom": "Leroy",
  "prenom": "Julie",
  "adresseMail": "julie.leroy@example.com",
  "adressePostal": "22 boulevard de la Liberté",
  "codePostal": "59000",
  "ville": "Lille",
  "motDePasse": "mdp112",
  "statut": true,
  "role": null
},
{
  "id": 27,
  "matricule": "EMP006",
  "nom": "Moreau",
  "prenom": "Antoine",
```

```

    "adresseMail": "antoine.moreau@example.com",
    "adressePostal": "7 rue du Port",
    "codePostal": "33000",
    "ville": "Bordeaux",
    "motDePasse": "mdp131",
    "statut": true,
    "role": null
  },
  {
    "id": 28,
    "matricule": "EMP007",
    "nom": "Laurent",
    "prenom": "Camille",
    "adresseMail": "camille.laurent@example.com",
    "adressePostal": "3 place Bellecour",
    "codePostal": "69002",
    "ville": "Lyon",
    "motDePasse": "mdp415",
    "statut": true,
    "role": null
  },
  {
    "id": 29,
    "matricule": "EMP008",
    "nom": "Simon",
    "prenom": "Élodie",
    "adresseMail": "elodie.simon@example.com",
    "adressePostal": "18 rue de la Pompe",
    "codePostal": "75116",
    "ville": "Paris",
    "motDePasse": "mdp161",
    "statut": false,
    "role": {
      "id": 5,
      "libelle": "stagiaire",
      "statut": false
    }
  }
]

```

```
}  
}  
]
```

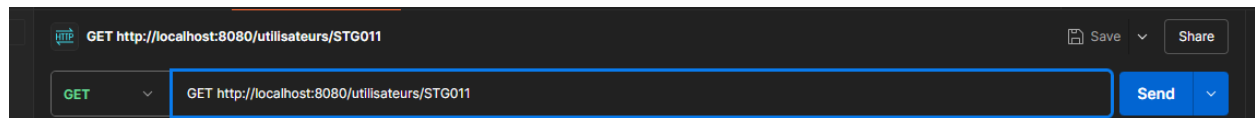
**TEST 01 :** → Résultat

**OK**

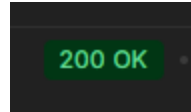
**GET > RECUPERER / READ**

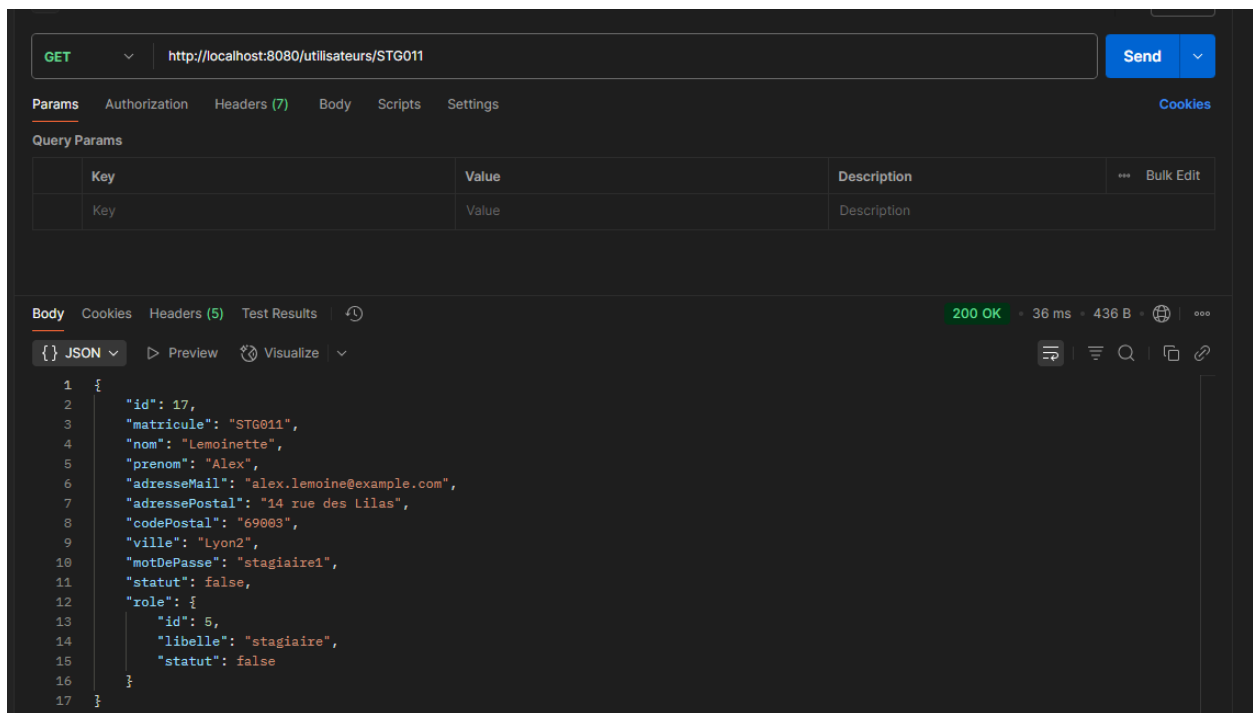
**TEST 02 :**

GET http://localhost:8080/utilisateurs/STG011 → détail d'un utilisateur (JSON)



**Résultat Postman :**





### L'utilisateur STG011 du gestionnaire de formation récupéré au format JSON :

```
{
  "id": 17,
  "matricule": "STG011",
  "nom": "LemoINETTE",
  "prenom": "Alex",
  "adresseMail": "alex.lemoine@example.com",
  "adressePostal": "14 rue des Lilas",
  "codePostal": "69003",
  "ville": "Lyon2",
  "motDePasse": "stagiaire1",
  "statut": false,
  "role": {
    "id": 5,
    "libelle": "stagiaire",
    "statut": false
  }
}
```

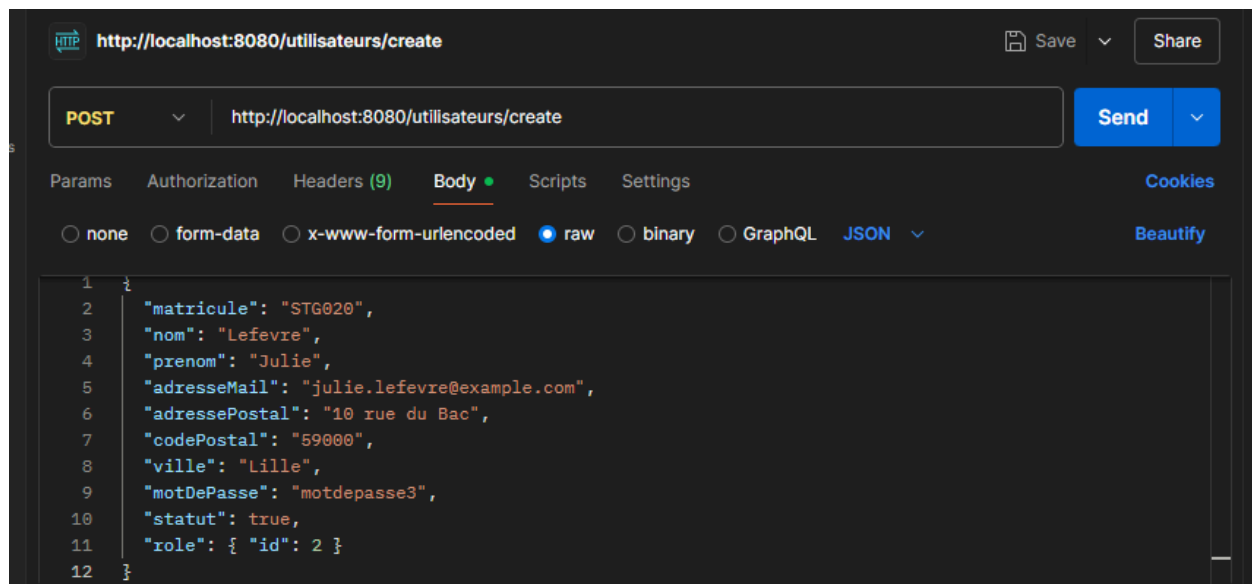
**TEST 02 :** → Résultat

OK

## POST > CREER / CREATE

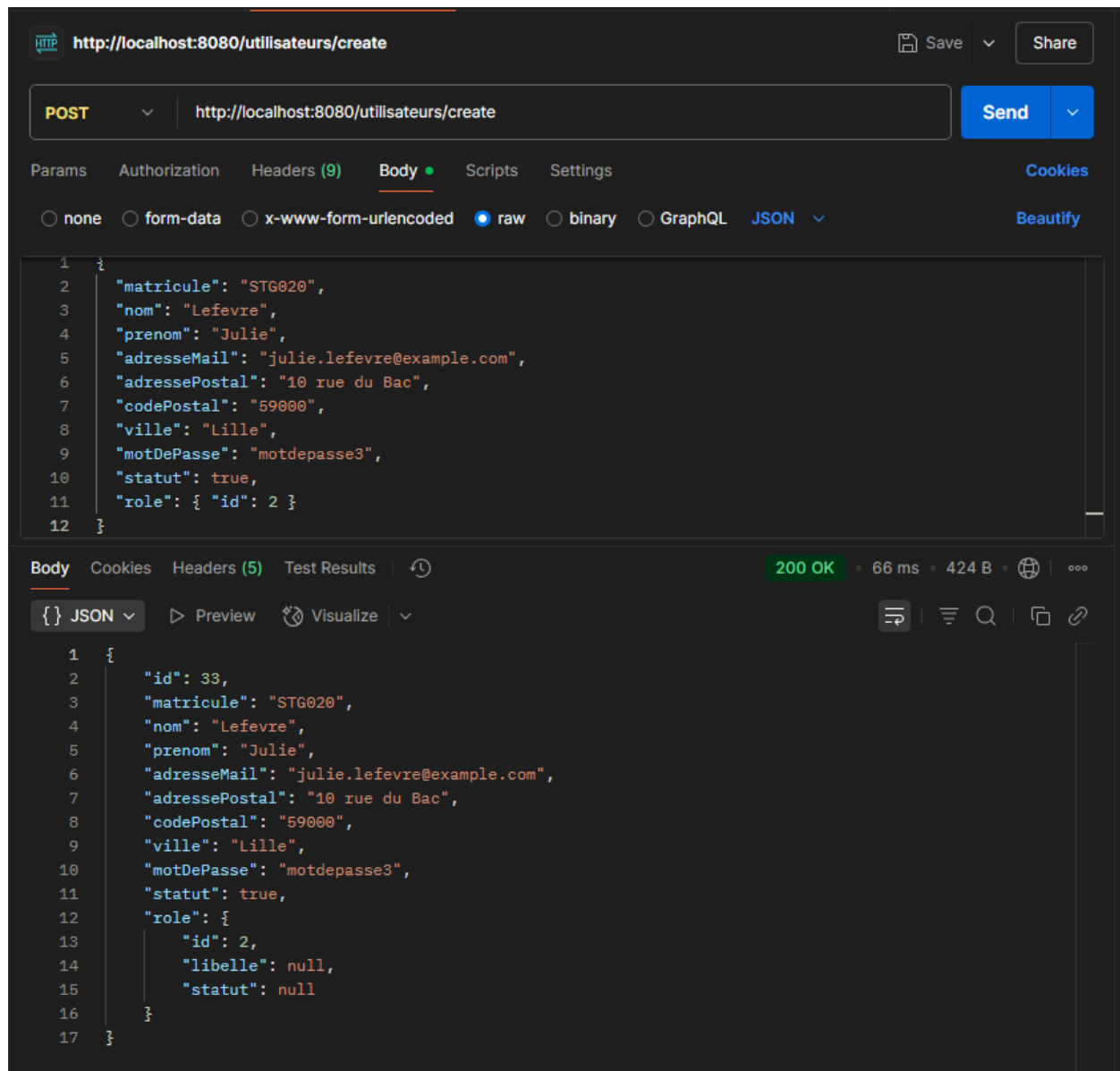
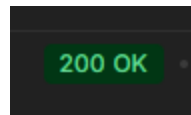
**TEST 03 :**

POST <http://localhost:8080/utilisateurs/create> → Créer un utilisateur (avec un bo



```
{
  "matricule": "STG020",
  "nom": "Lefevre",
  "prenom": "Julie",
  "adresseMail": "julie.lefevre@example.com",
  "adressePostal": "10 rue du Bac",
  "codePostal": "59000",
  "ville": "Lille",
  "motDePasse": "motdepasse3",
  "statut": true,
  "role": { "id": 2 }
}
```

## Résultat Postman :



## Résultat PhpMyAdmin : Le nouvel utilisateur STG020 a bien été créé.

	Éditer	Copier	Supprimer	33	STG020	Lefevre	Julie	julie.lefevre@example.com	10 rue du Bac	59000	Lille	motdepasse3	1
--	--------	--------	-----------	----	--------	---------	-------	---------------------------	---------------	-------	-------	-------------	---

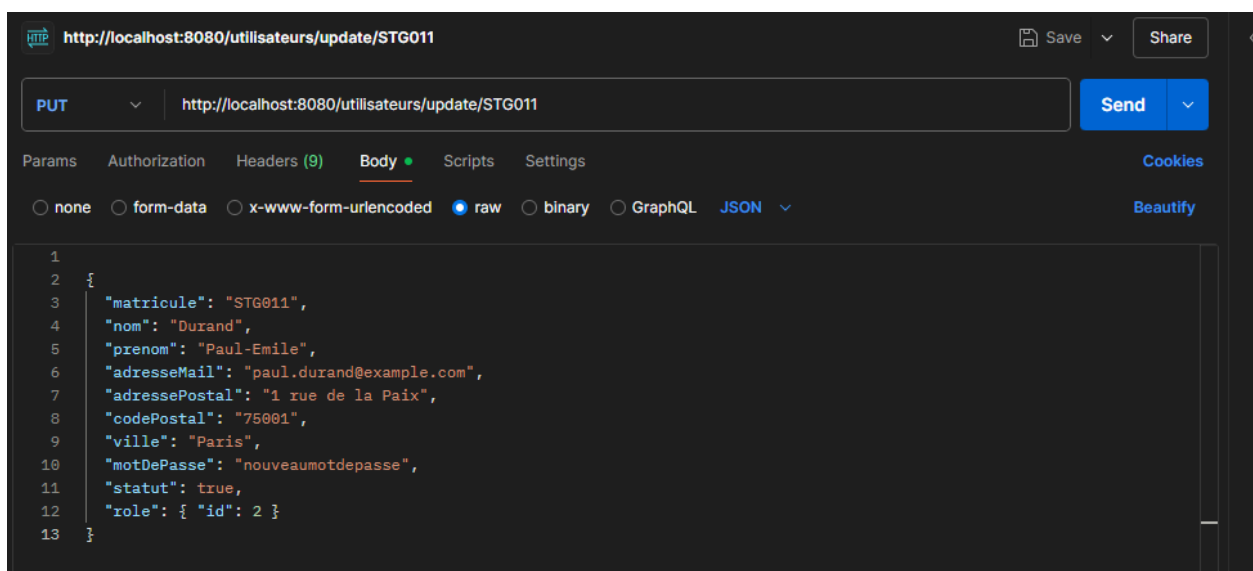
**TEST 03 :** → Résultat

OK

**TEST 04 :**

**PUT > MODIFIER / UPDATE**

PUT <http://localhost:8080/utilisateurs/update/STG011> → Modifier un utilisateur (a



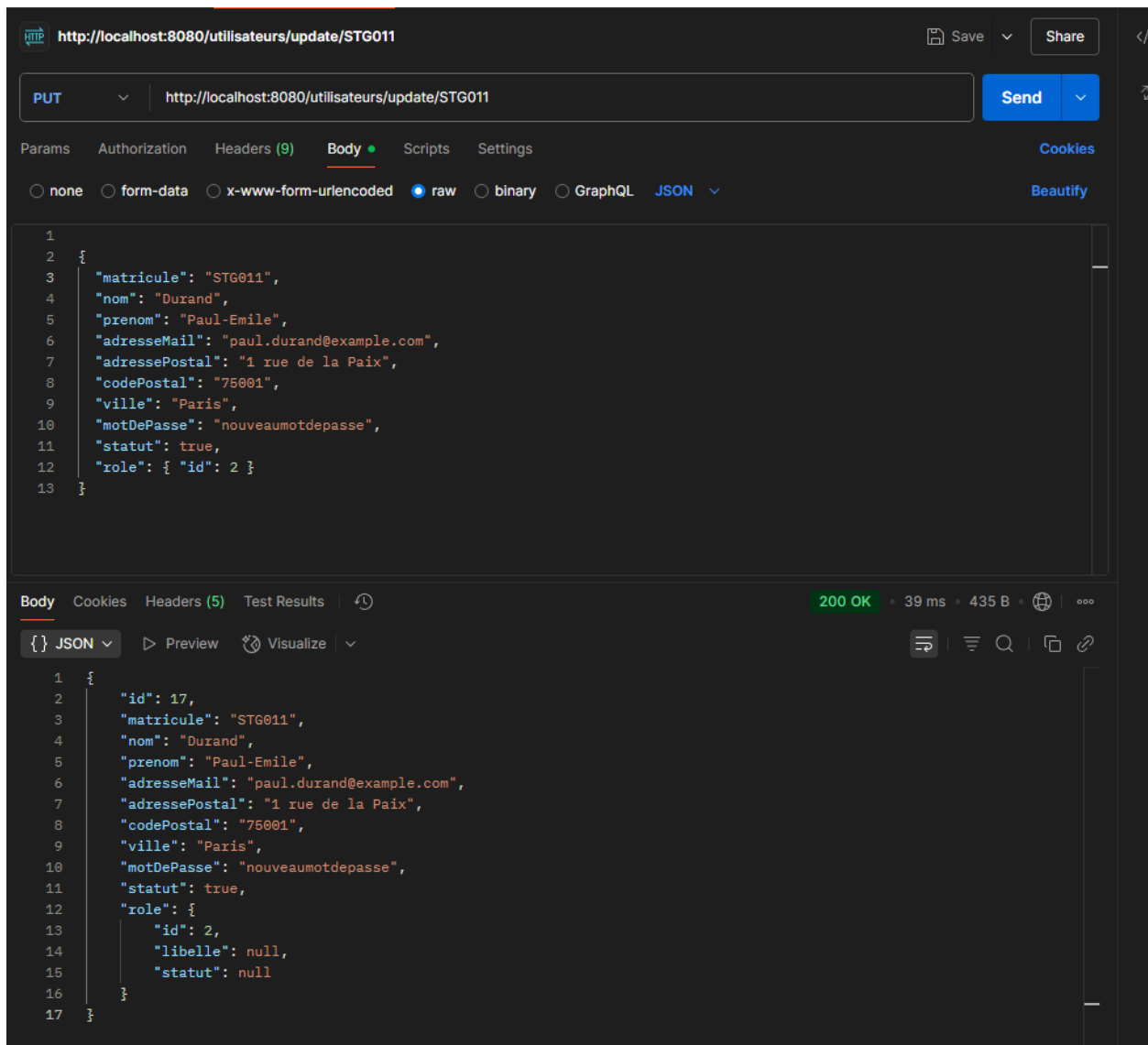
```
{
  "matricule": "STG011",
  "nom": "Durand",
  "prenom": "Paul-Emile",
  "adresseMail": "paul.durand@example.com",
  "adressePostal": "1 rue de la Paix",
  "codePostal": "75001",
  "ville": "Paris",
  "motDePasse": "nouveau mot de passe",
  "statut": true,
```



```
"role": { "id": 2 }  
}
```

## Résultat Postman :

200 OK



## Résultat PhpMyAdmin : L'utilisateur STG011 a bien été modifié.

Avant :

	id	matricule	nom	prenom	adresse_mail	adresse_postal	code_postal	ville	mot_de_passe	statut	role_id
<input type="checkbox"/>	17	STG011	Lemoine	Alex	alex.lemoine@example.com	14 rue des Lilas	69003	Lyon2	stagiaire1	0	5

Après :

	id	matricule	nom	prenom	adresse_mail	adresse_postal	code_postal	ville	mot_de_passe	statut	role_id
<input type="checkbox"/>	17	STG011	Durand	Paul-Emile	paul.durand@example.com	1 rue de la Paix	75001	Paris	nouveaumotdepasse	1	2

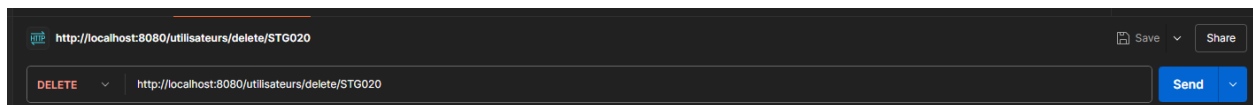
**TEST 04 :** → Résultat

**OK**

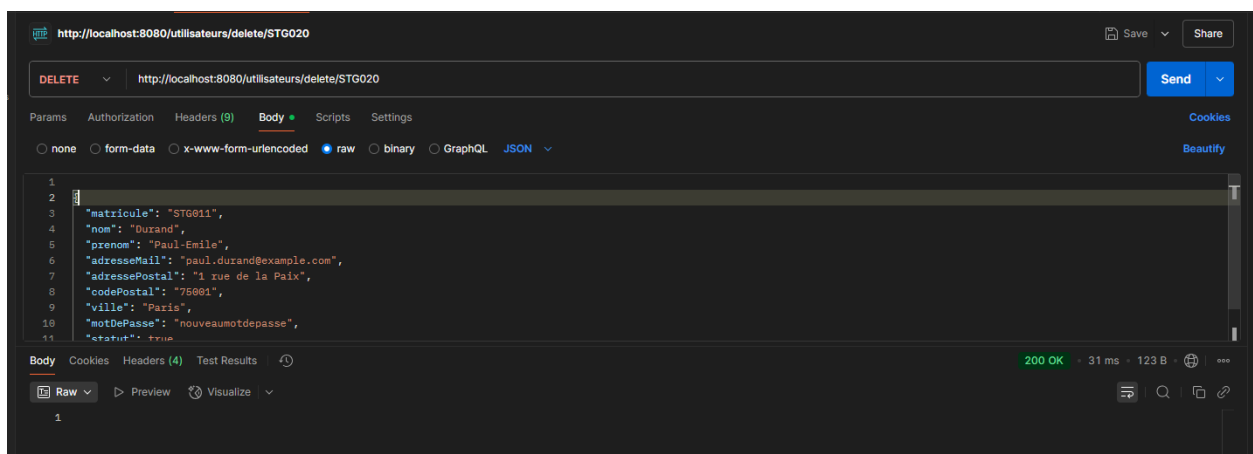
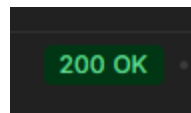
**TEST 05 :**

**DELETE [ NE JAMAIS UTILISER SAUF EXCEPTION ] > SUPPRIMER = NO RETURN**

DELETE http://localhost:8080/utilisateurs/delete/STG020 → Supprimer un utilisateur



**Résultat Postman :**



## Résultat PhpMyAdmin : L'utilisateur STG020 a bien été supprimé définitivement.

Avant :

<input type="checkbox"/>	Éditer	Copier	Supprimer	33	STG020	Lefevre	Julie	julie.lefevre@example.com	10 rue du Bac	59000	Lille	motdepasse3	1	2
--------------------------	--------	--------	-----------	----	--------	---------	-------	---------------------------	---------------	-------	-------	-------------	---	---

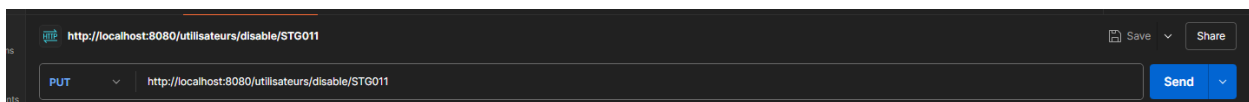
Après :

				id	matricule	nom	prenom	adresse_mail	adresse_postal	code_postal	ville	mot_de_passe	statut	role_id
<input type="checkbox"/>	Éditer	Copier	Supprimer	17	STG011	Durand	Paul-Emile	paul.durand@example.com	1 rue de la Paix	75001	Paris	nouveaumotdepasse	1	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	18	EMP001	Dupont	Marie	marie.dupont@example.com	12 rue de la République	75001	Paris	mdp123	0	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	19	STG012	Roux	Laura	laura.roux@example.com	7 chemin Vert	13002	Marseille	stagiaire2	1	7
<input type="checkbox"/>	Éditer	Copier	Supprimer	20	EMP002	Martin	Luc	luc.martin@example.com	5 avenue des Champs-Élysées	75008	Paris	mdp456	1	7
<input type="checkbox"/>	Éditer	Copier	Supprimer	21	STG013	Fournier	Thomas	thomas.fournier@example.com	22 impasse des Roses	59000	Lille	stagiaire3	0	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	22	EMP003	Bernard	Sophie	sophie.bernard@example.com	8 rue de la Paix	06000	Nice	mdp789	0	NULL
<input type="checkbox"/>	Éditer	Copier	Supprimer	23	STG014	Girard	Manonin	manon.girard@example.com	3 allée des Tilleuls	33000	Bordeaux	stagiaire4	0	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	24	EMP004	Dubois	Pierre	pierre.dubois@example.com	15 cours Gambetta	69001	Lyon	mdp101	1	NULL
<input type="checkbox"/>	Éditer	Copier	Supprimer	25	STG015	Boyer	Hugo	hugo.boyer@example.com	9 rue des Peupliers	75017	Paris	stagiaire5	0	NULL
<input type="checkbox"/>	Éditer	Copier	Supprimer	26	EMP005	Leroy	Julie	julie.leroy@example.com	22 boulevard de la Liberté	59000	Lille	mdp112	1	NULL
<input type="checkbox"/>	Éditer	Copier	Supprimer	27	EMP006	Moreau	Antoine	antoine.moreau@example.com	7 rue du Port	33000	Bordeaux	mdp131	1	NULL
<input type="checkbox"/>	Éditer	Copier	Supprimer	28	EMP007	Laurent	Camille	camille.laurent@example.com	3 place Bellecour	69002	Lyon	mdp415	1	NULL
<input type="checkbox"/>	Éditer	Copier	Supprimer	29	EMP008	Simon	Élodie	elodie.simon@example.com	18 rue de la Pompe	75116	Paris	mdp161	0	5

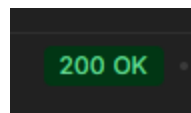
## TEST 06:

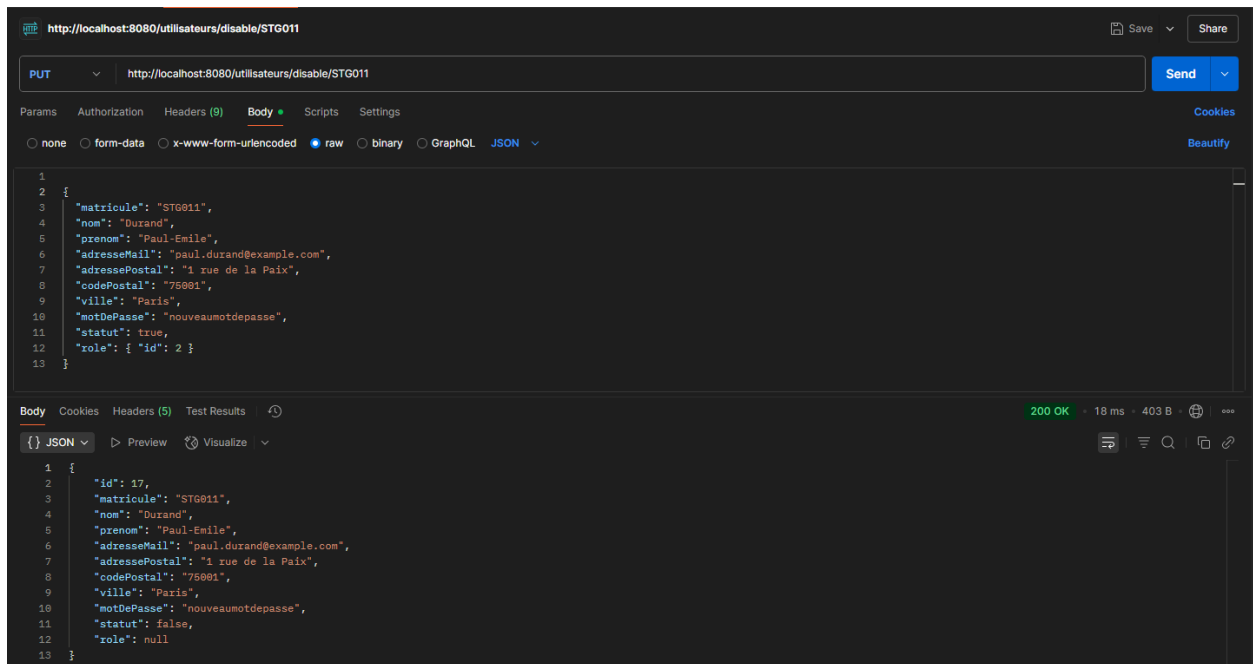
**PUT > MODIFIER / UPDATE** [ AU LIEU de DELETE → désactivation par le statut]

PUT <http://localhost:8080/utilisateurs/disable/STG011> → Désactiver un utilisateur



## Résultat Postman :





```
{
  "id": 17,
  "matricule": "STG011",
  "nom": "Durand",
  "prenom": "Paul-Emile",
  "adresseMail": "paul.durand@example.com",
  "adressePostal": "1 rue de la Paix",
  "codePostal": "75001",
  "ville": "Paris",
  "motDePasse": "nouveau mot de passe",
  "statut": false,
  "role": null
}
```

**Résultat PhpMyAdmin : L'utilisateur STG011 a bien été désactivé en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0) .**

	id	matricule	nom	prenom	adresse_mail	adresse_postal	code_postal	ville	mot_de_passe	statut	role_id
	17	STG011	Durand	Paul-Emile	paul.durand@example.com	1 rue de la Paix	75001	Paris	nouveaumotdepasse	0	NULL

## TEST 06 : → Résultat

OK

**Vs Code console terminal : Travail du package (maven) Hibernate ⇒ requêtes SQL envoyées avec Postman visibles.**

```
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0 where u1_0.matricule=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
2025-05-16T10:02:44.060+02:00 WARN 16944 --- [io-8080-exec-10] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.http.converter.HttpMessageNotReadableException: Required request body is missing: public com.emi.GestionnaireFormation.model.Utilisateur com.emi.GestionnaireFormation.controller.UtilisateurController.createUtilisateur(com.emi.GestionnaireFormation.model.Utilisateur)]
2025-05-16T10:03:04.084+02:00 WARN 16944 --- [nio-8080-exec-1] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.http.converter.HttpMessageNotReadableException: Required request body is missing: public com.emi.GestionnaireFormation.model.Utilisateur com.emi.GestionnaireFormation.controller.UtilisateurController.createUtilisateur(com.emi.GestionnaireFormation.model.Utilisateur)]
Hibernate: insert into utilisateur (adresse_mail,adresse_postal,code_postal,matricule,mot_de_passe,?, ?, ?, ?, ?, ?, ?)
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0 where u1_0.matricule=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: update utilisateur set adresse_mail=?,adresse_postal=?,code_postal=?,matricule=?,mot_de_passe=?,nom=?,prenom=?,role_id=?,statut=?,ville=? where id=?
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0 where u1_0.matricule=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: delete from utilisateur where id=?
Hibernate: select u1_0.id,u1_0.adresse_mail,u1_0.adresse_postal,u1_0.code_postal,u1_0.matricule,u1_0.mot_de_passe,u1_0.nom,u1_0.prenom,u1_0.role_id,u1_0.statut,u1_0.ville from utilisateur u1_0 where u1_0.matricule=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: update utilisateur set adresse_mail=?,adresse_postal=?,code_postal=?,matricule=?,mot_de_passe=?,nom=?,prenom=?,role_id=?,statut=?,ville=? where id=?
Hibernate: update utilisateur set adresse_mail=?,adresse_postal=?,code_postal=?,matricule=?,mot_de_passe=?,nom=?,prenom=?,role_id=?,statut=?,ville=? where id=?
```

## Entité Role

SELECT \* FROM `role`

☐ Profilage

[\[ Éditer en ligne \]](#)

[\[ Éditer \]](#)

[\[ Expliquer SQL \]](#)

[\[ Créer le code source PHP \]](#)

[\[ Actualiser \]](#)

☐ Tout afficher

Nombre de lignes : 25

Filtrer les lignes: Chercher dans cette table

Trier par clé : Aucun(e)

Options supplémentaires

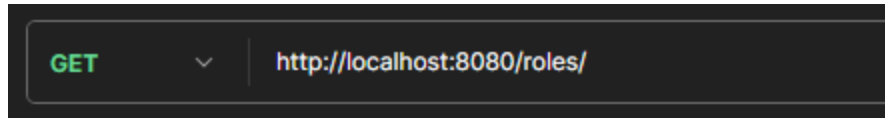
				id	libelle	statut
<input type="checkbox"/>		Éditer		Copier		Supprimer
				5	stagiaire	0
<input type="checkbox"/>		Éditer		Copier		Supprimer
				6	formateur	0
<input type="checkbox"/>		Éditer		Copier		Supprimer
				7	intervenant	0

**Table role PhpMyAdmin AVANT TESTS.**

## TEST 07 :

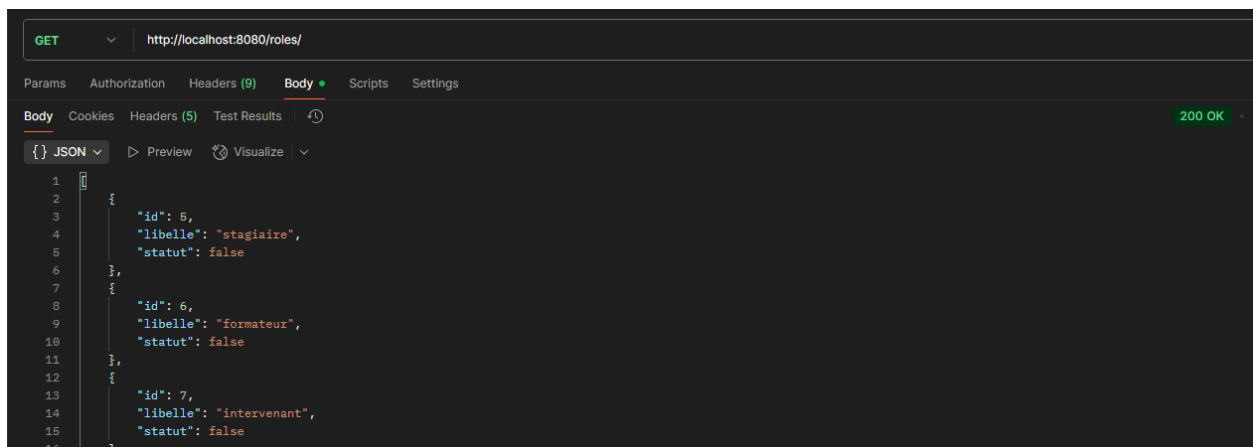
### GET > RECUPERER / READ

GET http://localhost:8080/roles/ → Récupère la liste de tous les rôles (JSON)



## Résultat Postman :

200 OK



## Liste des roles du gestionnaire de formation récupéré au format JSON :

```
[
  {
    "id": 5,
    "libelle": "stagiaire",
    "statut": false
  },
  {
    "id": 6,
    "libelle": "formateur",
    "statut": false
  },
  {
    "id": 7,
    "libelle": "intervenant",
    "statut": false
  }
]
```

```

    },
    {
      "id": 7,
      "libelle": "intervenant",
      "statut": false
    }
  ]

```

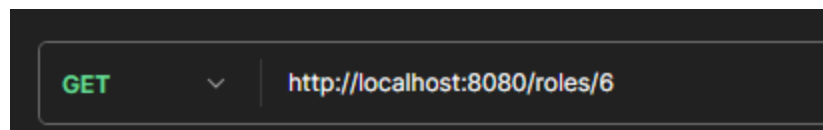
**TEST 07 :** → Résultat

OK

### GET > RECUPERER / READ

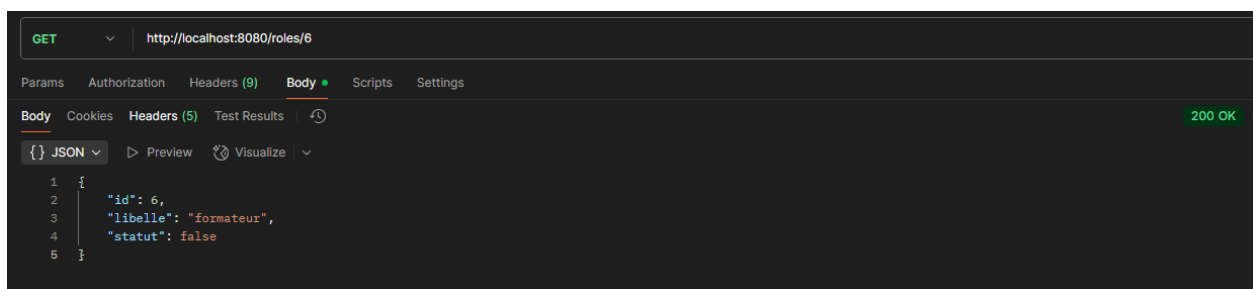
**TEST 08 :**

GET http://localhost:8080/roles/6 → Détail un rôle (JSON)



**Résultat Postman :**

200 OK



**Role id=6 du gestionnaire de formation récupéré au format JSON :**

```

{
  "id": 6,

```

```
"libelle": "formateur",  
"statut": false  
}
```

**TEST 08 :** → Résultat

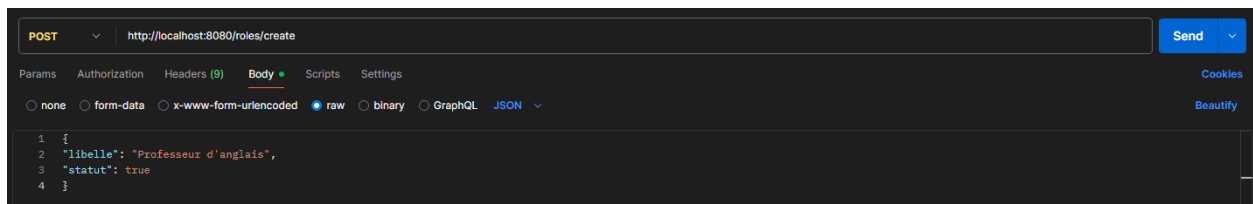
OK

**TEST 09 :**

**POST > CREER / CREATE**

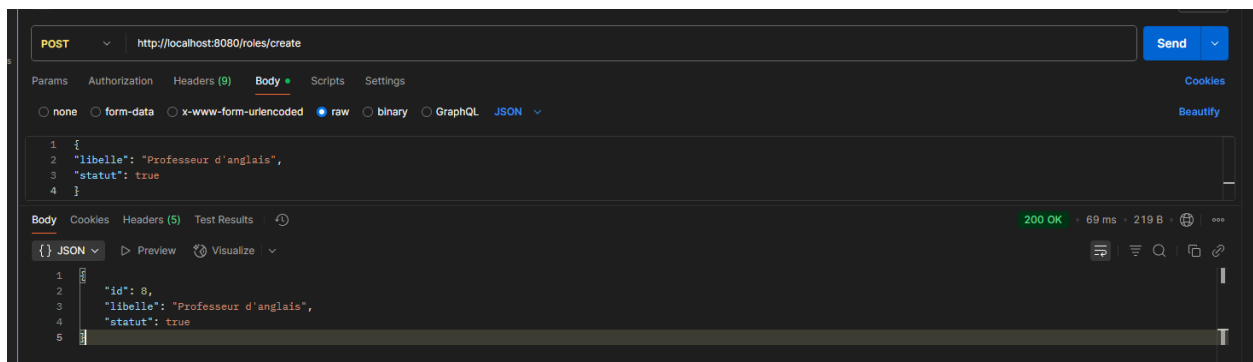
POST http://localhost:8080/roles/create → Créer un rôle (avec un body JSON)

```
{  
  "libelle": "Professeur d'anglais",  
  "statut": true  
}
```



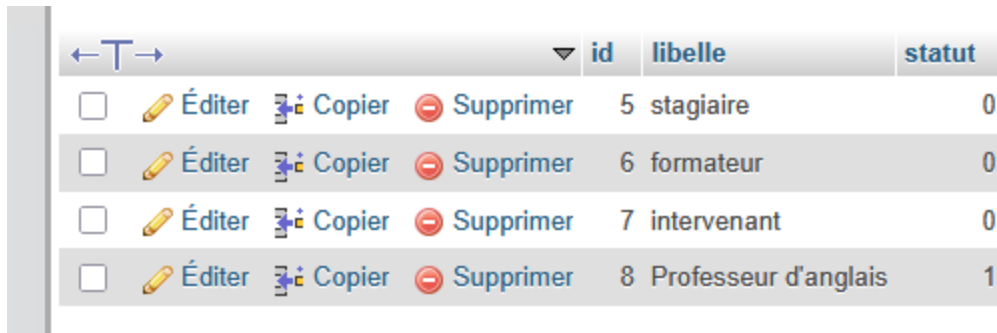
**Résultat Postman :**

200 OK





**Résultat PhpMyAdmin : Le nouveau role professeur d'anglais a bien été créé.**



				id	libelle	statut
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	stagiaire	0
<input type="checkbox"/>	Éditer	Copier	Supprimer	6	formateur	0
<input type="checkbox"/>	Éditer	Copier	Supprimer	7	intervenant	0
<input type="checkbox"/>	Éditer	Copier	Supprimer	8	Professeur d'anglais	1

**TEST 09 :** → Résultat

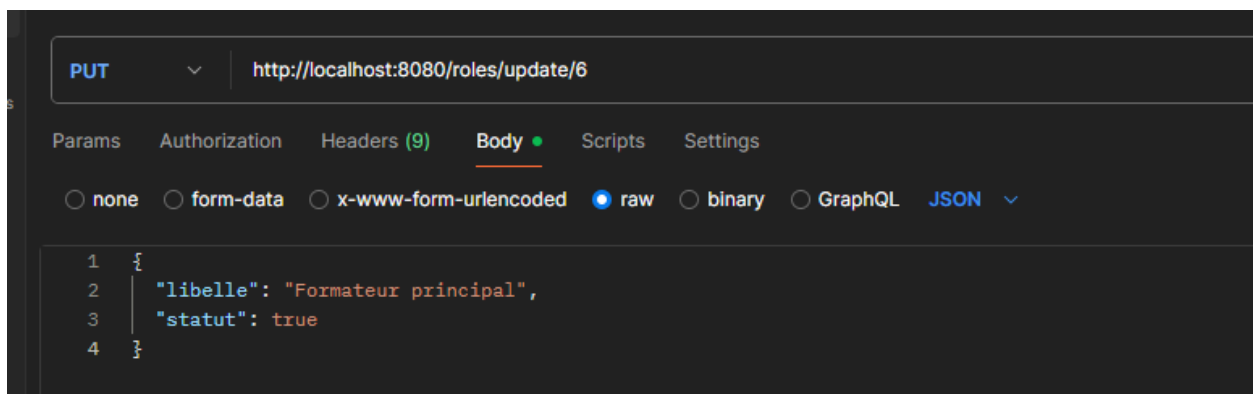
**OK**

**TEST 10 :**

**PUT > MODIFIER / UPDATE**

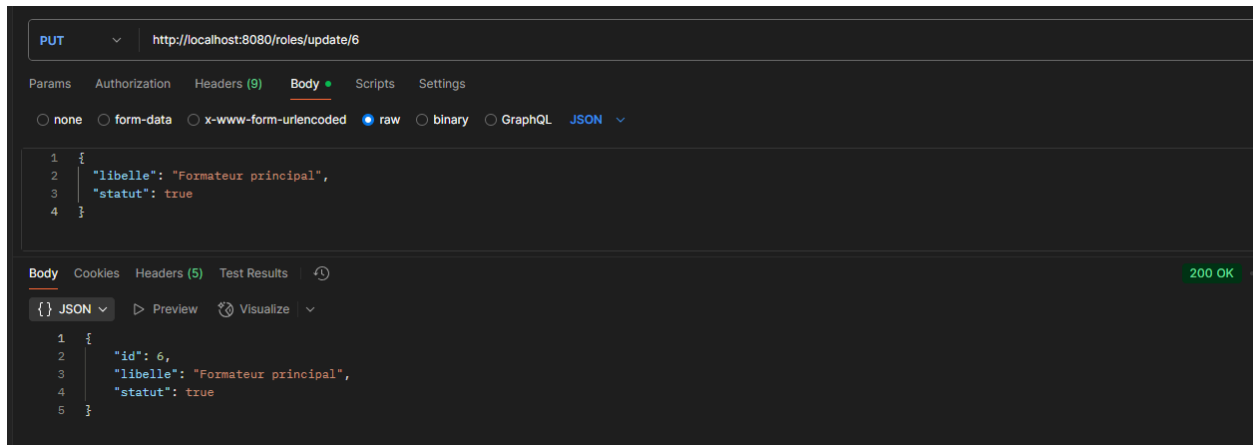
PUT <http://localhost:8080/roles/update/6> → Modifier un rôle (avec un body JSON)

```
{
  "libelle": "Formateur principal",
  "statut": true
}
```



**Résultat Postman :**

200 OK



**Résultat PhpMyAdmin : Le role id=6 a bien été modifié.**

	Éditer	Copier	Supprimer	Statut	
<input type="checkbox"/>	Éditer	Copier	Supprimer	6 Formateur principal	1

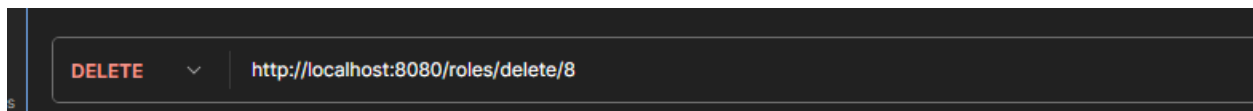
**TEST 10 : → Résultat**

**OK**

**TEST 11 :**

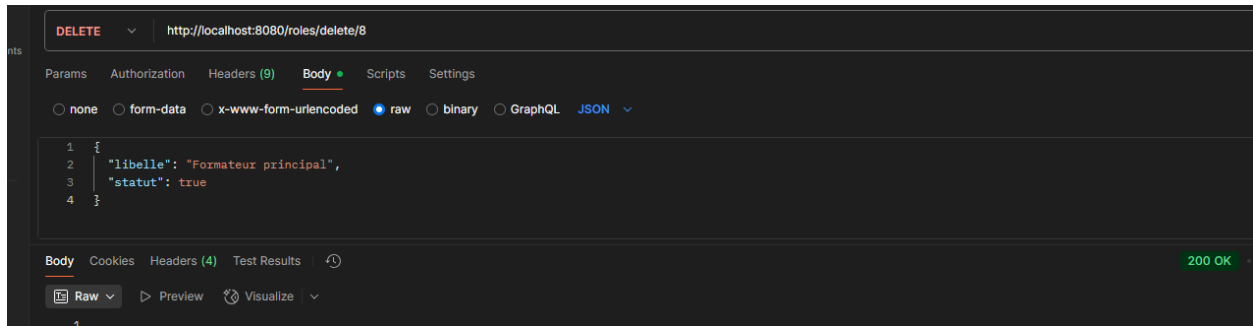
**DELETE [ NE JAMAIS UTILISER SAUF EXCEPTION ] > SUPPRIMER = NO RETURN**

DELETE `http://localhost:8080/roles/delete/8` → Supprimer un rôle



**Résultat Postman :**

200 OK



**Résultat PhpMyAdmin :** Le role id=8 a bien été supprimé définitivement.

					id	libelle	statut
<input type="checkbox"/>		Éditer		Copier		Supprimer	5 stagiaire 0
<input type="checkbox"/>		Éditer		Copier		Supprimer	6 Formateur principal 1
<input type="checkbox"/>		Éditer		Copier		Supprimer	7 intervenant 0

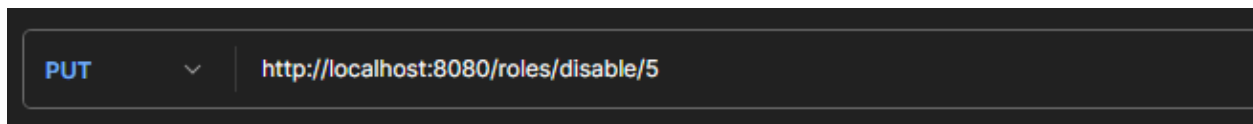
**TEST 11 :** → Résultat

OK

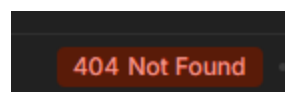
**TEST 12 :**

**PUT > MODIFIER / UPDATE** [ AU LIEU de DELETE → désactivation par le statut]

PUT `http://localhost:8080/roles/disable/5` → Désactiver un rôle



**Résultat Postman :**



```

1 {
2   "timestamp": "2025-05-16T09:59:35.343+00:00",
3   "status": 404,
4   "error": "Not Found",
5   "trace": "org.springframework.web.servlet.resource.NoResourceFoundException: No static resource roles/disable/5.\r\n\tat org.springframework.web.servlet.resource.ResourceHttpRequestHandler.handleRequest(ResourceHttpRequestHandler.java:585)\r\n\tat org.springframework.web.servlet.mvc.HttpRequestHandlerAdapter.handle(HttpRequestHandlerAdapter.java:52)\r\n\tat org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1089)\r\n\tat org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:979)\r\n\tat org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1814)\r\n\tat org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:885)\r\n\tat jakarta.servlet.http.HttpServlet.service(HttpServlet.java:659)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:205)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:149)\r\n\tat org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:51)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:174)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:149)\r\n\tat org.springframework.web.filter.RequestContextFilter.doFilterInternal(RequestContextFilter.java:108)\r\n\tat org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:116)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:174)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:149)\r\n\tat org.springframework.web.filter.FormContentFilter.doFilterInternal(FormContentFilter.java:93)\r\n\tat org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:116)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:174)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:149)\r\n\tat org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:201)\r\n\tat org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:116)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:174)\r\n\tat org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:149)\r\n\tat org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:167)\r\n\tat org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:98)\r\n\tat org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:482)\r\n\tat org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:115)\r\n\tat org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:93)\r\n\tat org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:74)\r\n\tat org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:344)\r\n\tat org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:391)\r\n\tat org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63)\r\n\tat org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:896)\r\n\tat org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1744)\r\n\tat org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52)\r\n\tat org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191)\r\n\tat org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659)\r\n\tat org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:63)\r\n\tat java.base/java.lang.Thread.run(Thread.java:1575)\r\n\t",
6   "message": "No static resource roles/disable/5.",
7   "path": "/roles/disable/5"
8 }

```

**Résultat PhpMyAdmin : Le role id=5 a bien été désactivé en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0).**

**TEST 12 : → Résultat NO TEST à refaire**

L'erreur rencontrée avec la requête

PUT http://localhost:8080/roles/disable/5

est un **404 Not Found** avec le message :

No static resource roles/disable/5.

**Explication :**

- Le serveur Spring Boot ne trouve **aucun contrôleur** ou endpoint correspondant à l'URL `/roles/disable/5 ???`

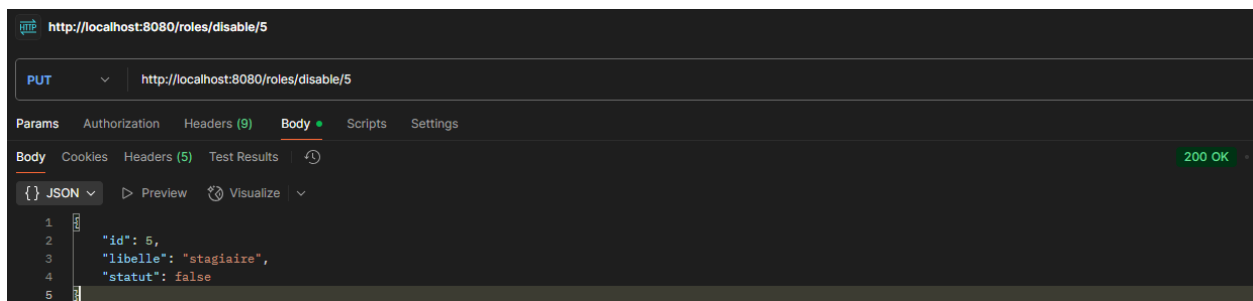
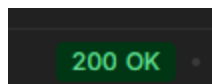
- Spring essaie alors de trouver une ressource statique à cette adresse (comme un fichier), mais il n'en existe pas, d'où l'erreur???
  - Cette URL n'est pas prévue dans l'API : la désactivation concerne les utilisateurs, pas les rôles???
- C'est bien cette erreur qui me concerne.

### Résultat Postman :

Après modification de mon code :

J'ai eu une **erreur 404 dans Postman** parce que j'essayais d'accéder à une URL `/roles/disable/5` qui n'existait pas dans mon contrôleur.

**J'ai compris que mon code n'avait pas de méthode pour gérer cette route.** J'ai corrigé le problème en ajoutant une méthode `disableRole` avec le mapping `/roles/disable/{id}` dans mon contrôleur, ce qui permet maintenant de désactiver un rôle sans erreur.



**Résultat PhpMyAdmin : Le role id=5 a bien été désactivé en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0) .**

	ID	libelle	statut
<input type="checkbox"/>	5	stagiaire	0

**Vs Code console terminal : Travail du package (maven) Hibernate ⇒ requêtes SQL envoyé avec Postman visibles.**

```
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
2025-05-16T11:38:40.204+02:00 WARN 17108 --- [nio-8080-exec-4] .w.s.m.s.DefaultHandlerExceptionResolve
r : Resolved [org.springframework.web.method.annotation.MethodArgumentTypeMismatchException: Failed to
convert value of type 'java.lang.String' to required type 'java.lang.Long'; For input string: "create"]
Hibernate: insert into role (libelle,statut) values (?,?)
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: update role set libelle=?,statut=? where id=?
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: delete from role where id=?
```

```
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
```

## Entité Formation

SELECT \* FROM `formation`

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Expliquer SQL](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:  Trier par clé :

Options supplémentaires

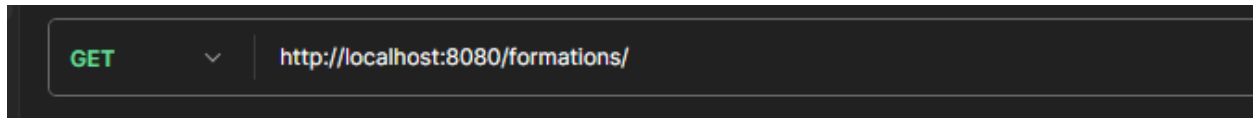
	ID	libelle	numero_offre	date_debut	date_fin	date_debut_pe	date_fin_pe	statut	id_centre
<input type="checkbox"/>	1	CDA	02KL56I	2025-05-04	2025-12-12	2025-05-21	2025-09-16	1	NULL
<input type="checkbox"/>	2	ADRN	HFJ03H	2025-05-13	2025-05-17	2025-05-23	2025-09-18	0	NULL

**Table formation PhpMyAdmin AVANT TESTS.**

**TEST 13 :**

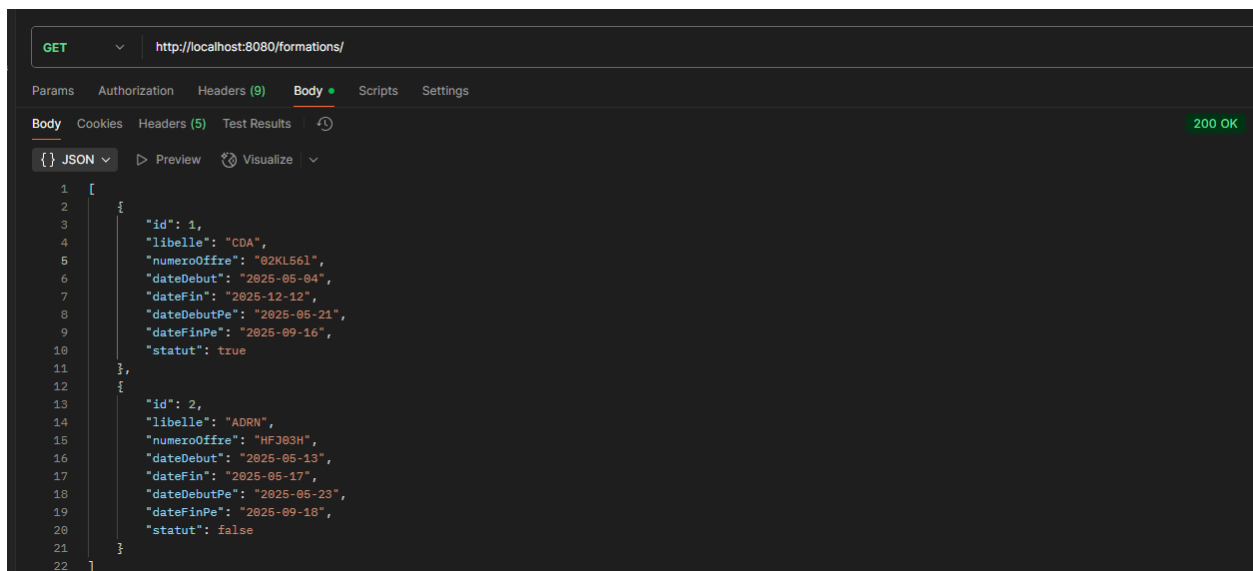
**GET > RECUPERER / READ**

GET http://localhost:8080/formations/ → liste toutes les formations



## Résultat Postman :

200 OK



## Liste des formations du gestionnaire de formation récupéré au format JSON :

```
[
  {
    "id": 1,
    "libelle": "CDA",
    "numeroOffre": "02KL561",
    "dateDebut": "2025-05-04",
    "dateFin": "2025-12-12",
    "dateDebutPe": "2025-05-21",
    "dateFinPe": "2025-09-16",
    "statut": true
  }
]
```

```
},  
{  
  "id": 2,  
  "libelle": "ADRN",  
  "numeroOffre": "HFJ03H",  
  "dateDebut": "2025-05-13",  
  "dateFin": "2025-05-17",  
  "dateDebutPe": "2025-05-23",  
  "dateFinPe": "2025-09-18",  
  "statut": false  
}  
]
```

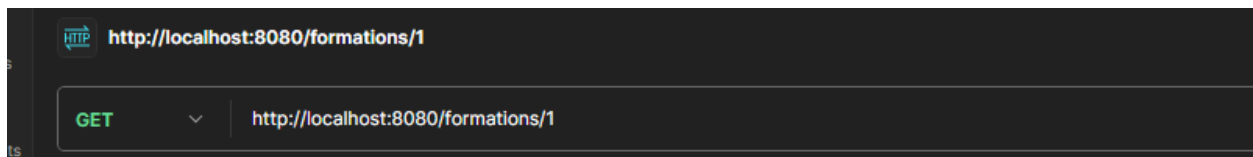
**TEST 13 :** → Résultat

**OK**

#### GET > RECUPERER / READ

**TEST 14 :**

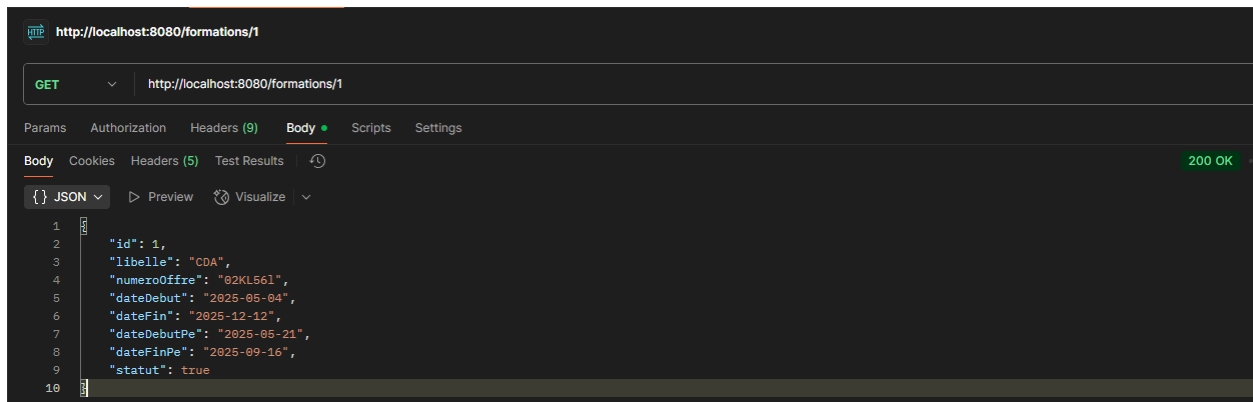
GET http://localhost:8080/formations/1 → détail d'une formation



**Résultat Postman :**

**200 OK**





### Formation id = 1 du gestionnaire de formation récupéré au format JSON :

```
{
  "id": 1,
  "libelle": "CDA",
  "numeroOffre": "02KL561",
  "dateDebut": "2025-05-04",
  "dateFin": "2025-12-12",
  "dateDebutPe": "2025-05-21",
  "dateFinPe": "2025-09-16",
  "statut": true
}
```

**TEST 14 :** → Résultat

**OK**

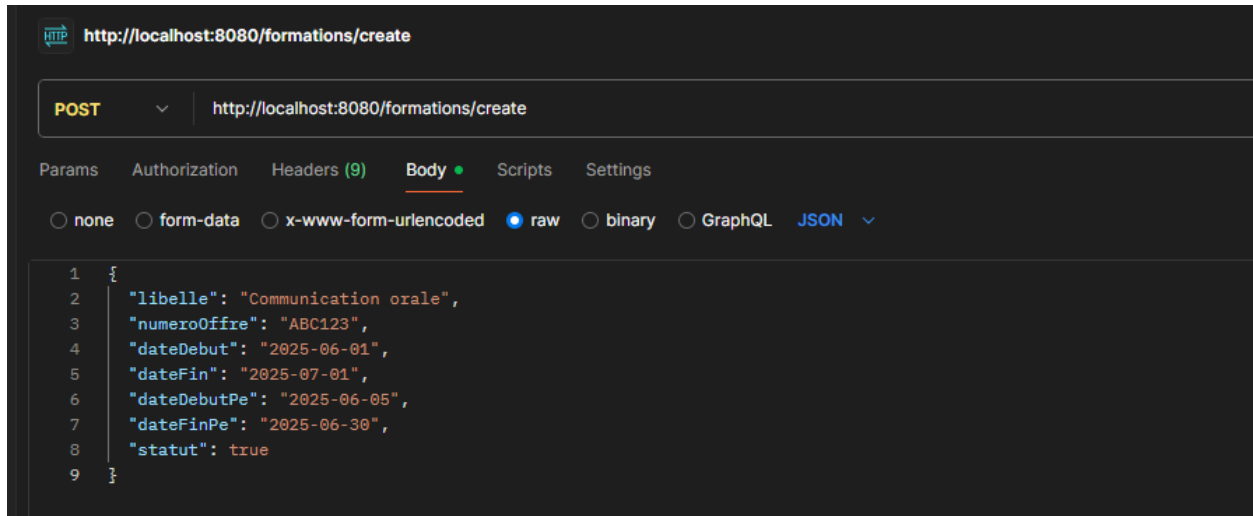
### **TEST 15 :**

#### **POST > CREER / CREATE**

POST http://localhost:8080/formations/create → Créer une formation (avec un b

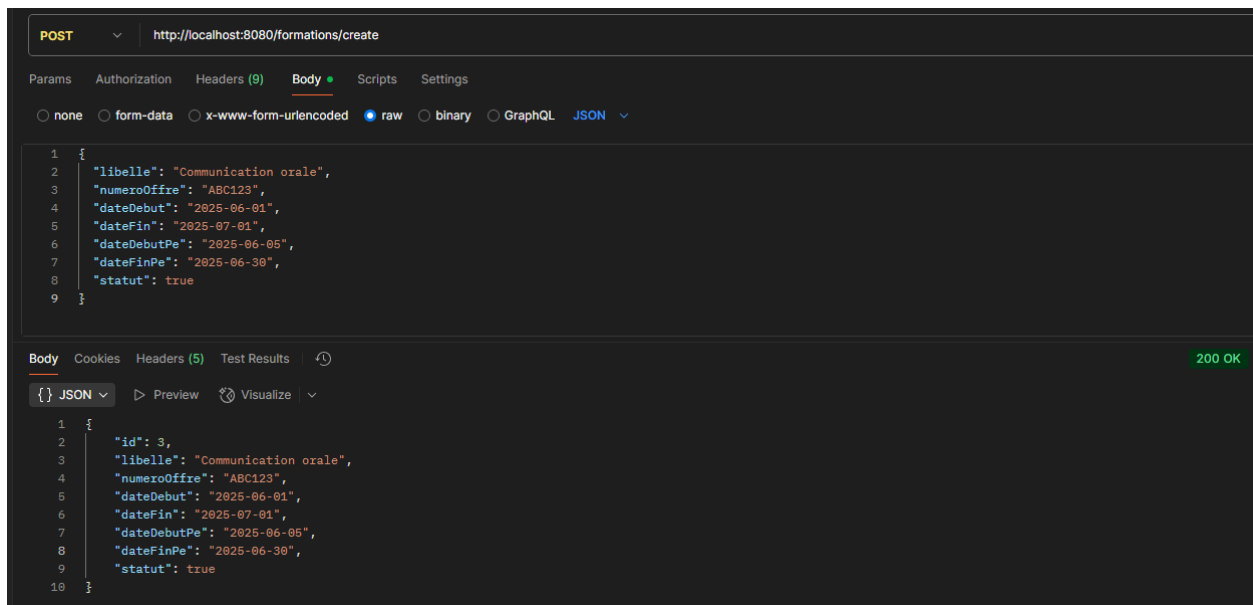
```
{
  "libelle": "Communication orale",
  "numeroOffre": "ABC123",
```

```
"dateDebut": "2025-06-01",  
"dateFin": "2025-07-01",  
"dateDebutPe": "2025-06-05",  
"dateFinPe": "2025-06-30",  
"statut": true  
}
```



### Résultat Postman :

200 OK



**Résultat PhpMyAdmin : Le nouvel formation id = 3 Communication orale a bien été créée.**

SELECT \* FROM `formation`

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:  | Trier par clé :

Options supplémentaires

	id	libelle	numero_offre	date_debut	date_fin	date_debut_pe	date_fin_pe	statut	id_centre
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	1	CDA	02KL56I	2025-05-04	2025-12-12	2025-05-21	2025-09-16	1	NULL
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	2	ADRN	HFJ03H	2025-05-13	2025-05-17	2025-05-23	2025-09-18	0	NULL
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	3	Communication orale	ABC123	2025-06-01	2025-07-01	2025-06-05	2025-06-30	1	NULL

**TEST 15 : → Résultat**

**OK**

**TEST 16 :**

**PUT > MODIFIER / UPDATE**

PUT http://localhost:8080/formations/update/3 → Modifier une formation (avec u

```

{
  "id": 3,

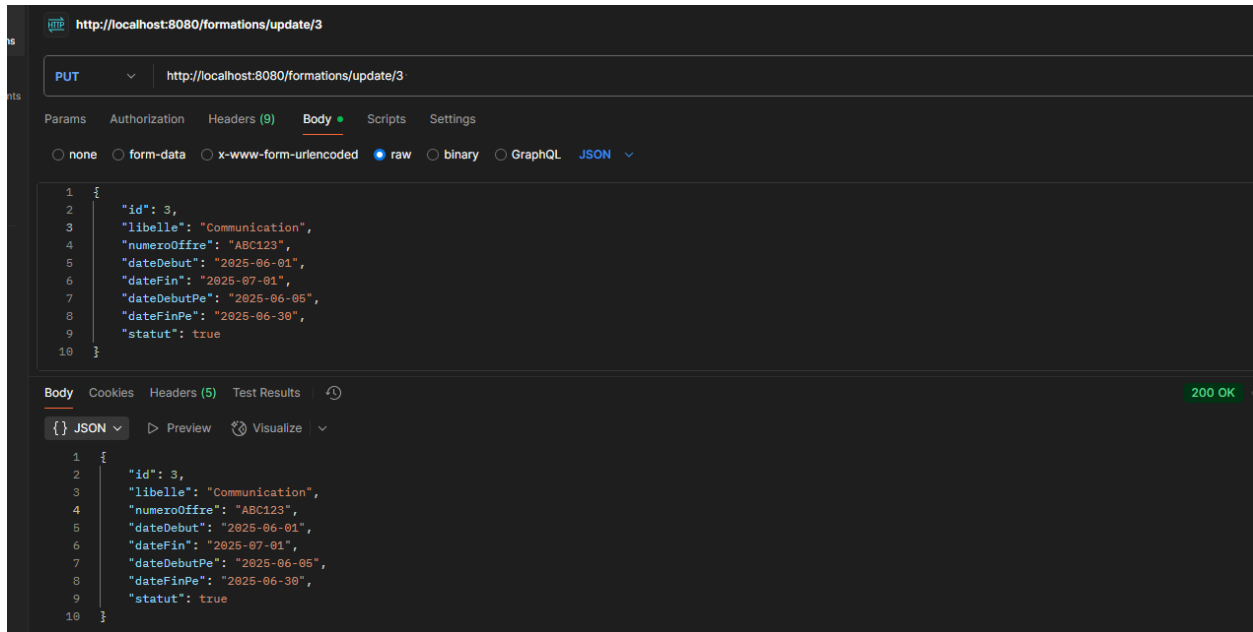
```

```
"libelle": "Communication",
"numeroOffre": "ABC123",
"dateDebut": "2025-06-01",
"dateFin": "2025-07-01",
"dateDebutPe": "2025-06-05",
"dateFinPe": "2025-06-30",
"statut": true
}
```



### Résultat Postman :

200 OK



**Résultat PhpMyAdmin : Résultat PhpMyAdmin : Le nouvel formation id = 3 Communication orale a bien été modifiée.**

<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Communication	ABC123	2025-06-01	2025-07-01	2025-06-05	2025-06-30	1	NULL
--------------------------	--------	--------	-----------	---	---------------	--------	------------	------------	------------	------------	---	------

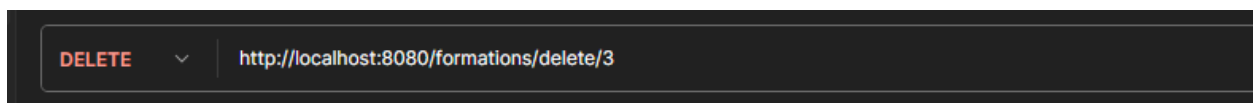
**TEST 16 : → Résultat**

**OK**

**TEST 17 :**

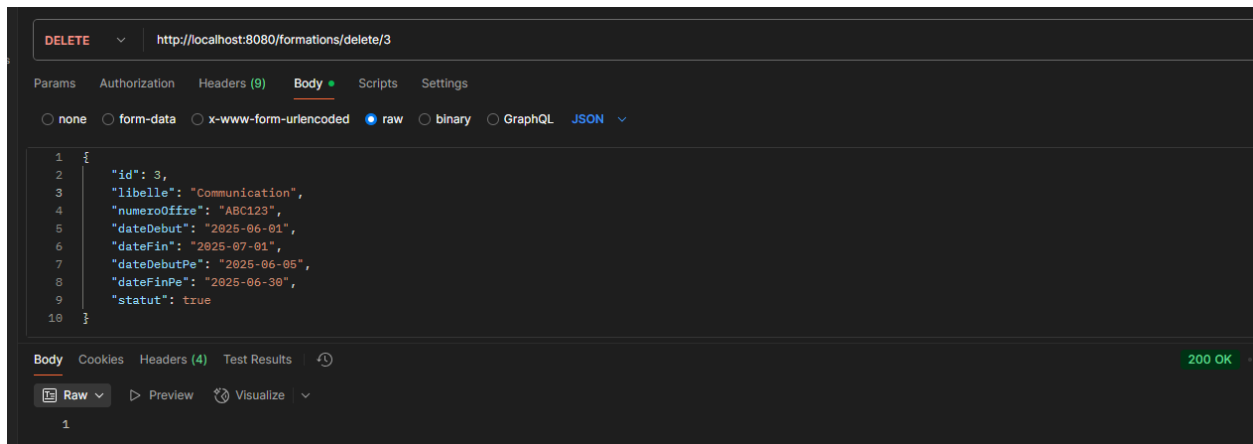
**DELETE [ NE JAMAIS UTILISER SAUF EXCEPTION ] > SUPPRIMER = NO RETURN**

DELETE http://localhost:8080/formations/delete/3 → Supprimer une formation



**Résultat Postman :**

**200 OK**



**Résultat PhpMyAdmin : Le nouvel formation id = 3 Communication a bien été supprimée définitivement.**

	id	libelle	numero_offre	date_debut	date_fin	date_debut_pe	date_fin_pe	statut	id_centre
<input type="checkbox"/>	1	CDA	02KL56I	2025-05-04	2025-12-12	2025-05-21	2025-09-16	1	NULL
<input type="checkbox"/>	2	ADRN	HFJ03H	2025-05-13	2025-05-17	2025-05-23	2025-09-18	0	NULL

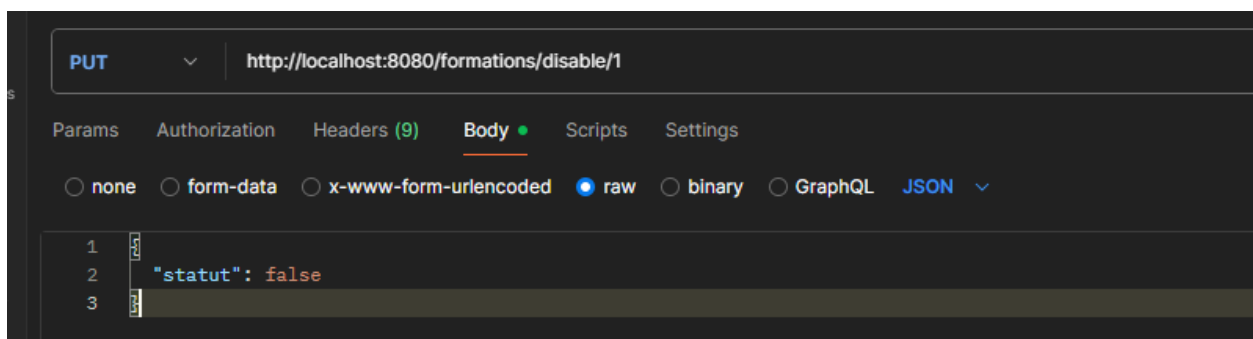
**TEST 17 : → Résultat**

**OK**

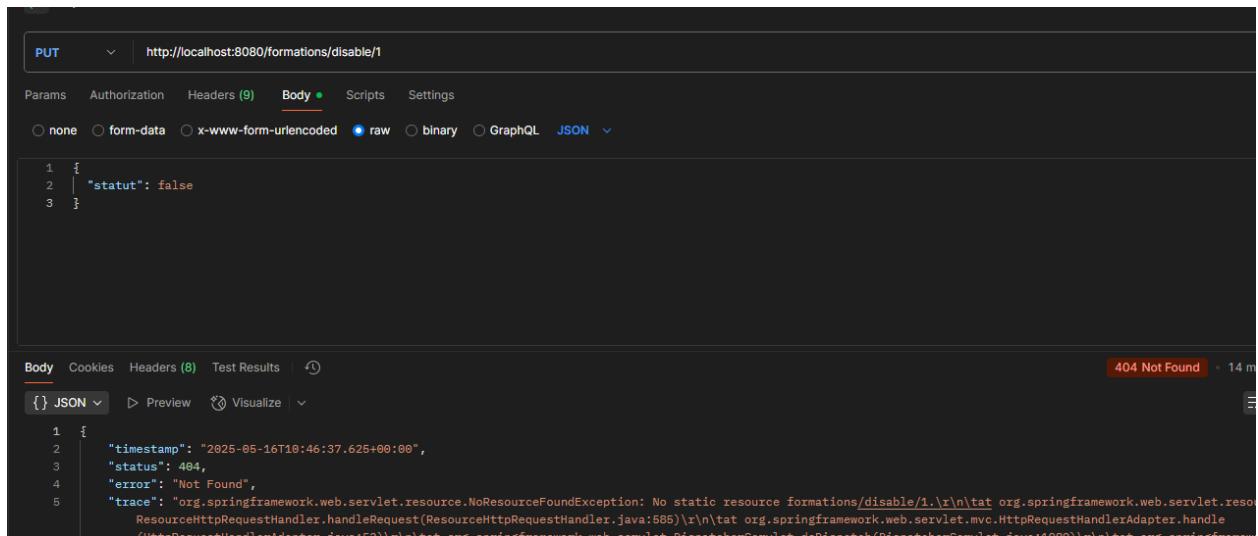
**TEST 18 :**

**PUT > MODIFIER / UPDATE [ AU LIEU de DELETE → désactivation par le statut]**

PUT http://localhost:8080/formations/disable/1 → Désactiver une formation



**Résultat Postman :**



- Cette URL n'est pas prévue dans l'API : la désactivation concerne les utilisateurs, pas les formations???

C'est bien cette erreur qui me concerne.

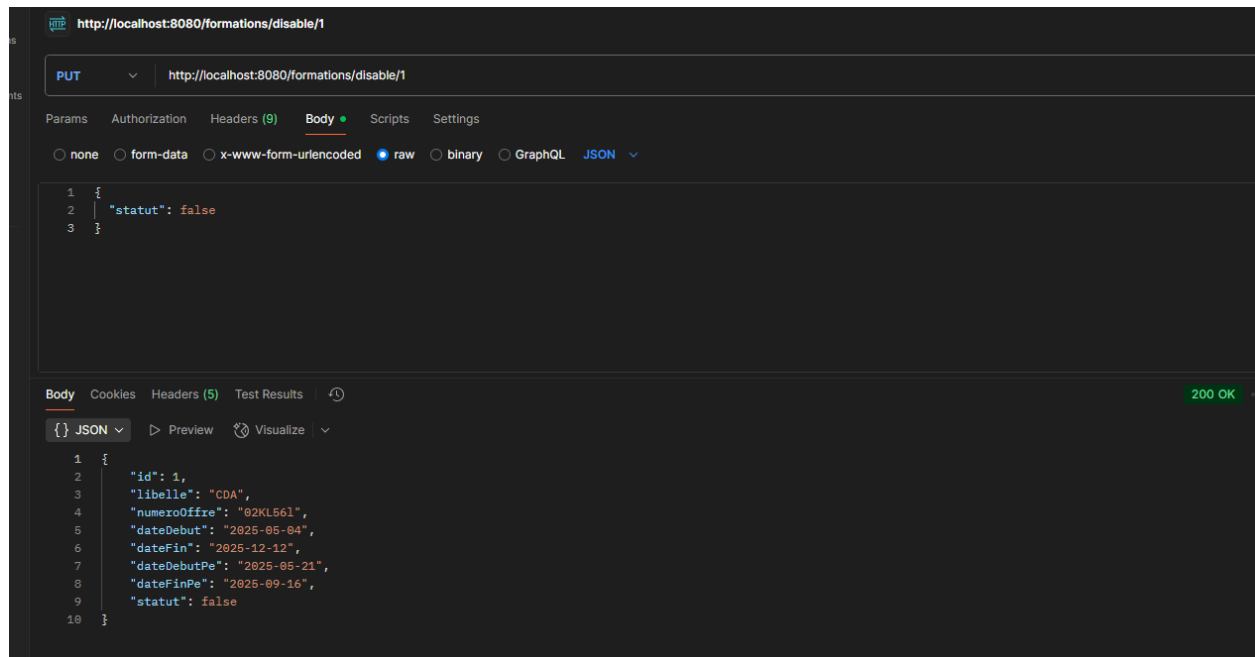
J'avais une erreur 404 dans Postman parce que j'essayais d'appeler une route `/formations/disable/{id}` qui n'existait pas dans mon contrôleur. J'ai compris que je devais

ajouter une méthode pour gérer cette URL.

J'ai corrigé le problème en ajoutant la méthode

`disableFormation` avec le mapping `/formations/disable/{id}` pour pouvoir désactiver une formation sans erreur.

200 OK



**Résultat PhpMyAdmin : Résultat PhpMyAdmin : La formation id = 1 a bien été désactivée en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0) .**

**TEST 18 : → Résultat**

**OK**

**Vs Code console terminal : Travail du package (maven) Hibernate ⇒ requêtes SQL envoyé avec Postman visibles.**



```

Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select f1_0.id,f1_0.date_debut,f1_0.date_debut_pe,f1_0.date_fin,f1_0.date_fin_pe,f1_0.libelle,f1_0.numero_offre,f1_0.statut from formation f1_0
Hibernate: select f1_0.id,f1_0.date_debut,f1_0.date_debut_pe,f1_0.date_fin,f1_0.date_fin_pe,f1_0.libelle,f1_0.numero_offre,f1_0.statut from formation f1_0 where f1_0.id=?
2025-05-16T12:34:43.021+02:00 WARN 12620 --- [io-8080-exec-10] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.method.annotation.MethodArgumentTypeMismatchException: Failed to convert value of type 'java.lang.String' to required type 'java.lang.Long'; For input string: "create"]
Hibernate: insert into formation (date_debut,date_debut_pe,date_fin,date_fin_pe,libelle,numero_offre,statut) values (?, ?, ?, ?, ?, ?, ?)
Hibernate: select f1_0.id,f1_0.date_debut,f1_0.date_debut_pe,f1_0.date_fin,f1_0.date_fin_pe,f1_0.libelle,f1_0.numero_offre,f1_0.statut from formation f1_0 where f1_0.id=?
Hibernate: update formation set date_debut=?,date_debut_pe=?,date_fin=?,date_fin_pe=?,libelle=?,numero_offre=?,statut=? where id=?
2025-05-16T12:40:12.803+02:00 WARN 12620 --- [nio-8080-exec-7] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]
2025-05-16T12:40:22.360+02:00 WARN 12620 --- [nio-8080-exec-4] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not supported]
Hibernate: select f1_0.id,f1_0.date_debut,f1_0.date_debut_pe,f1_0.date_fin,f1_0.date_fin_pe,f1_0.libelle,f1_0.numero_offre,f1_0.statut from formation f1_0 where f1_0.id=?
Hibernate: delete from formation where id=?

```

```

Hibernate: select f1_0.id,f1_0.date_debut,f1_0.date_debut_pe,f1_0.date_fin,f1_0.date_fin_pe,f1_0.libelle,f1_0.numero_offre,f1_0.statut from formation f1_0 where f1_0.id=?
Hibernate: update formation set date_debut=?,date_debut_pe=?,date_fin=?,date_fin_pe=?,libelle=?,numero_offre=?,statut=? where id=?

```

## Entité Module

`SELECT * FROM `module``

☐ Profilage [\[ Éditer en ligne \]](#) [\[ Éditer \]](#) [\[ Expliquer SQL \]](#) [\[ Créer le code source PHP \]](#) [\[ Actualiser \]](#)

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:  Trier par clé :

Options supplémentaires

			id	libelle	description	nom	statut
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Découverte de la programmation orientée objet avec...	Java Fondamentaux	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	HTML, CSS, JavaScript pour le développement web	Web Frontend	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Développement d'API REST avec Spring Boot	Spring Boot	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	Introduction à SQL et à la modélisation	Base de données	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	Mise en production et CI/CD	Déploiement	0

### Table module PhpMyAdmin AVANT TESTS.

## TEST 19 :

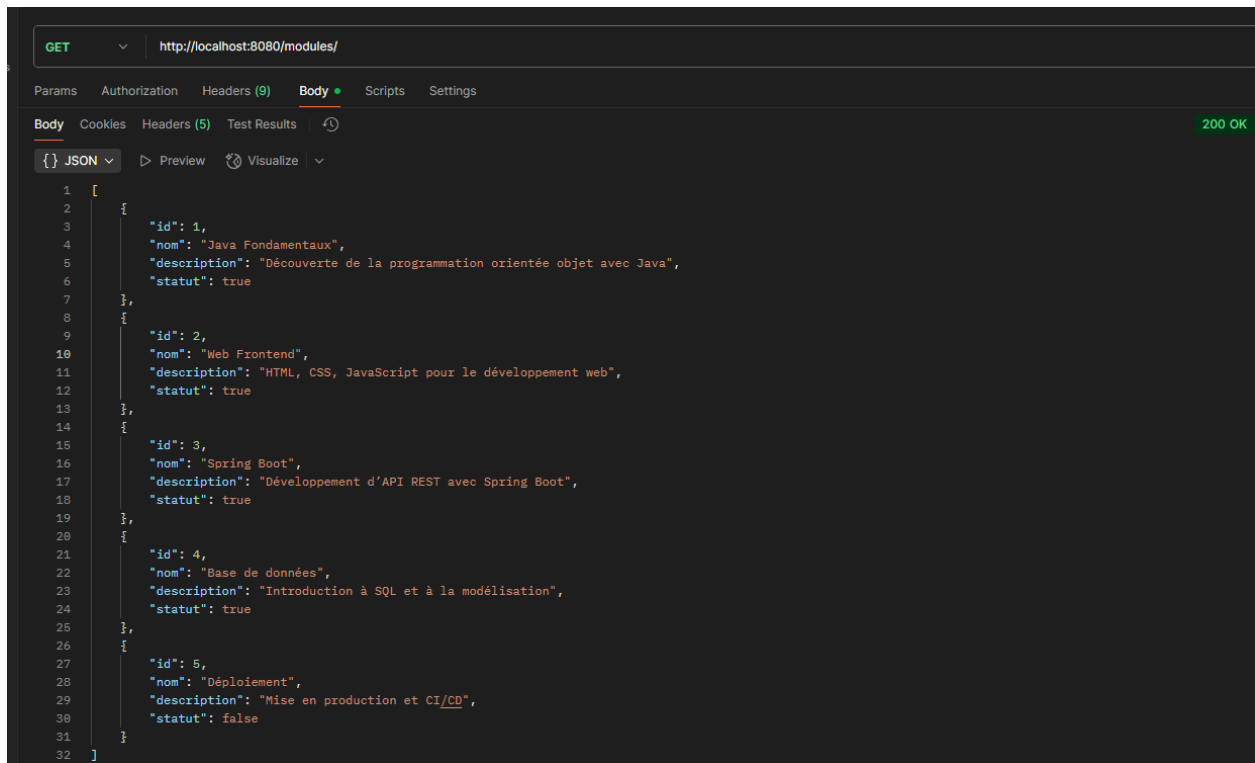
### GET > RECUPERER / READ

GET http://localhost:8080/modules/ → liste tous les modules



## Résultat Postman :

200 OK



## Liste des modules du gestionnaire de formation récupéré au format JSON :

```
[
  {
```

```
"id": 1,
"nom": "Java Fondamentaux",
"description": "Découverte de la programmation orientée objet avec Java",
"statut": true
},
{
  "id": 2,
  "nom": "Web Frontend",
  "description": "HTML, CSS, JavaScript pour le développement web",
  "statut": true
},
{
  "id": 3,
  "nom": "Spring Boot",
  "description": "Développement d'API REST avec Spring Boot",
  "statut": true
},
{
  "id": 4,
  "nom": "Base de données",
  "description": "Introduction à SQL et à la modélisation",
  "statut": true
},
{
  "id": 5,
  "nom": "Déploiement",
  "description": "Mise en production et CI/CD",
  "statut": false
}
]
```

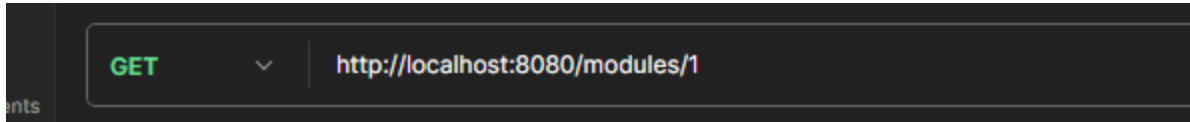
**TEST 19 :** → Résultat

**OK**

**GET > RECUPERER / READ**

## TEST 20 :

http://localhost:8080/modules/1 → détail d'un module



## Résultat Postman :

200 OK



## Détail module id = 1 du gestionnaire de formation récupéré au format JSON :

```
{
  "id": 1,
  "nom": "Java Fondamentaux",
  "description": "Découverte de la programmation orientée objet avec Java",
  "statut": true
}
```

TEST 20 : → Résultat

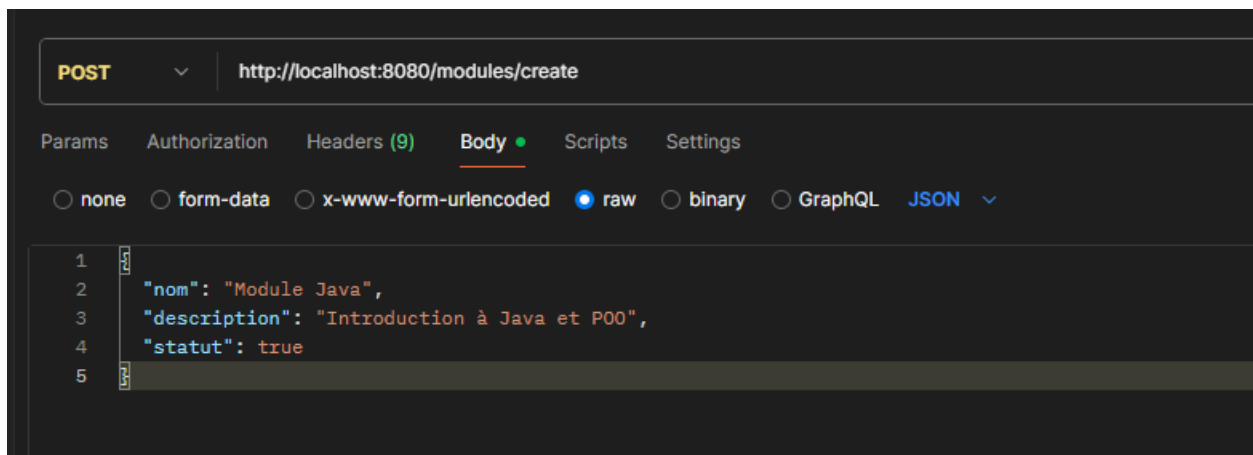
OK

## TEST 21 :

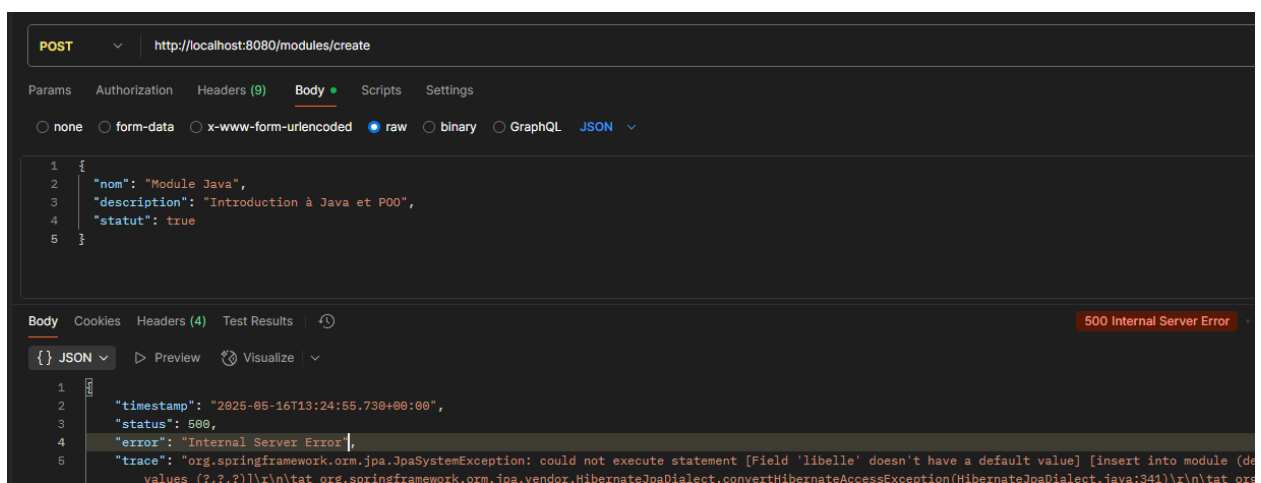
POST > CREER / CREATE

POST http://localhost:8080/modules/create → Création d'un module

```
{
  "nom": "Module Java",
  "description": "Introduction à Java et POO",
  "statut": true
}
```



## Résultat Postman :



Erreur 500 lors de la création d'un module indique :

## "Field 'libelle' doesn't have a default value"

Cela signifie que, dans ma base de données, la colonne `libelle` de la table `module` est obligatoire (NOT NULL, sans valeur par défaut), mais je ne lui fournis pas de valeur lors de l'insertion.

The screenshot shows a database management interface. At the top, there's a SQL editor with the following queries:

```
SELECT * FROM `module` ORDER BY `module`.`libelle` DESC
```

```
UPDATE `module` SET `libelle` = 'partie 5' WHERE `module`.`id` = 5;
```

Below the queries, there are controls for the table: "Tout afficher", "Nombre de lignes: 25", "Filtrer les lignes: Chercher dans cette table", and "Trier par clé: Aucun(e)".

Under "Options supplémentaires", there's a table with 6 columns: `id`, `libelle`, `description`, `nom`, and `statut`. The table contains 5 rows of data:

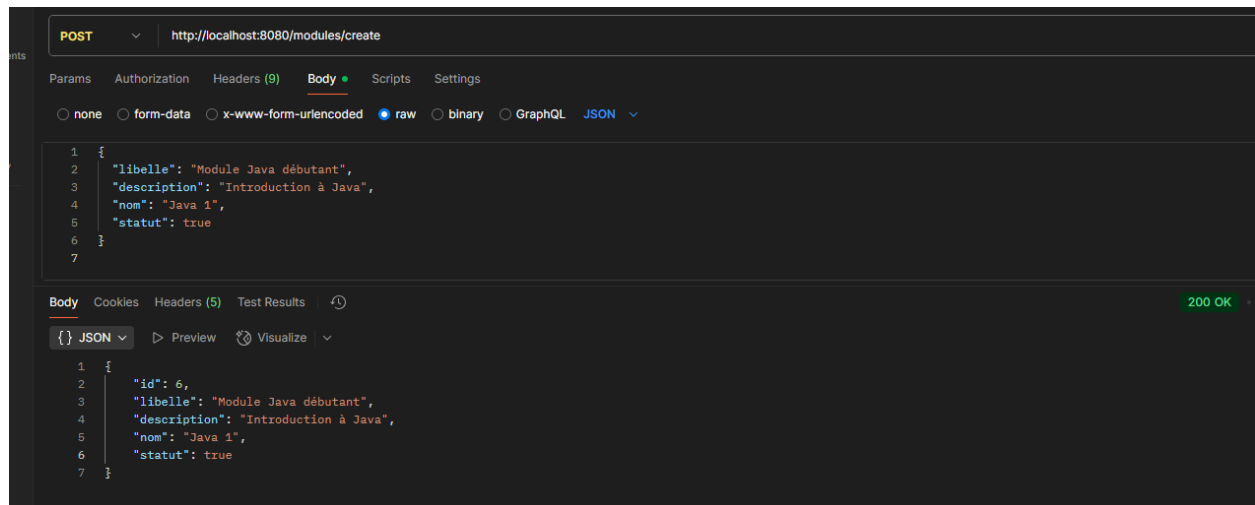
	id	libelle	description	nom	statut
<input type="checkbox"/>	1	partie 1	Découverte de la programmation orientée objet avec...	Java Fondamentaux	1
<input type="checkbox"/>	2	partie 2	HTML, CSS, JavaScript pour le développement web	Web Frontend	1
<input type="checkbox"/>	3	partie 3	Développement d'API REST avec Spring Boot	Spring Boot	1
<input type="checkbox"/>	4	partie 4	Introduction à SQL et à la modélisation	Base de données	1
<input type="checkbox"/>	5	partie 5	Mise en production et CI/CD	Déploiement	0

## Correction de l'erreur : Modification des champs et de getter/setter du model → Module.java

The screenshot shows a REST client interface. The method is **POST** and the URL is `http://localhost:8080/modules/create. The "Body" tab is selected, and the request body is raw JSON:`

```
{
  "libelle": "Module Java débutant",
  "description": "Introduction à Java",
  "nom": "Java 1",
  "statut": true
}
```

At the bottom, a green box displays the response: **200 OK**.



## Format JSON :

```
{
  "id": 6,
  "libelle": "Module Java débutant",
  "description": "Introduction à Java",
  "nom": "Java 1",
  "statut": true
}
```

## Résultat PhpMyAdmin : Le nouveau module id = 6 a bien été créé.

			id	libelle	description	nom	statut	
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	5	partie 5	Mise en production et CI/CD	Déploiement	0
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	4	partie 4	Introduction à SQL et à la modélisation	Base de données	1
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	3	partie 3	Développement d'API REST avec Spring Boot	Spring Boot	1
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2	partie 2	HTML, CSS, JavaScript pour le développement web	Web Frontend	1
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1	partie 1	Découverte de la programmation orientée objet avec...	Java Fondamentaux	1
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	6	Module Java débutant	Introduction à Java	Java 1	1

## TEST 21 : → Résultat

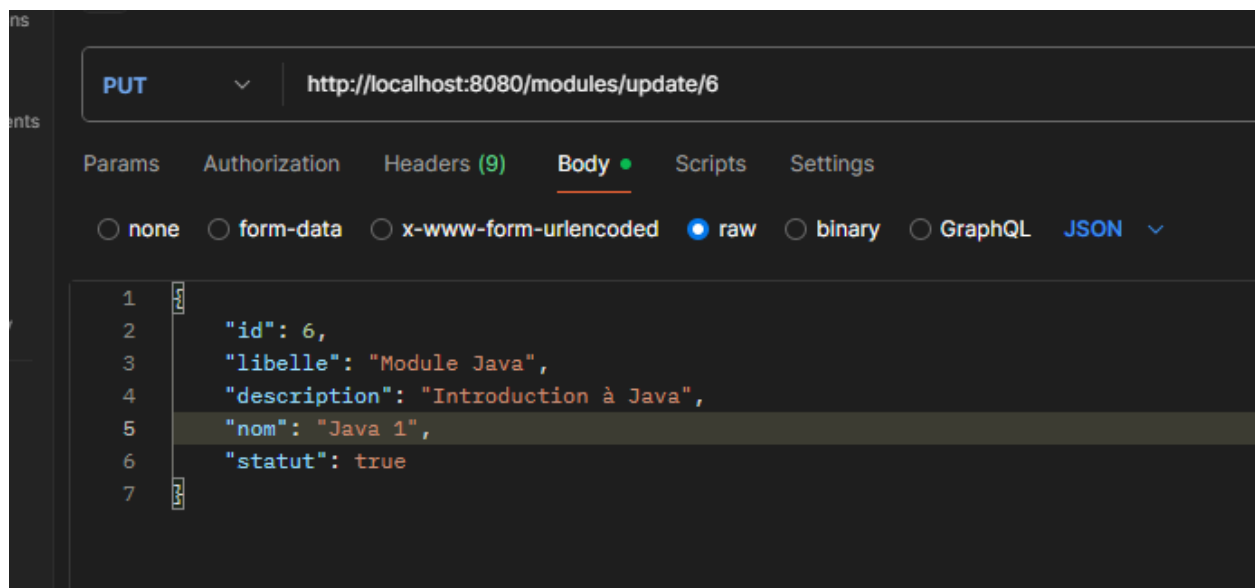
OK

## TEST 22 :

**PUT > MODIFIER / UPDATE**

PUT http://localhost:8080/modules/update/6 → Modifier un module

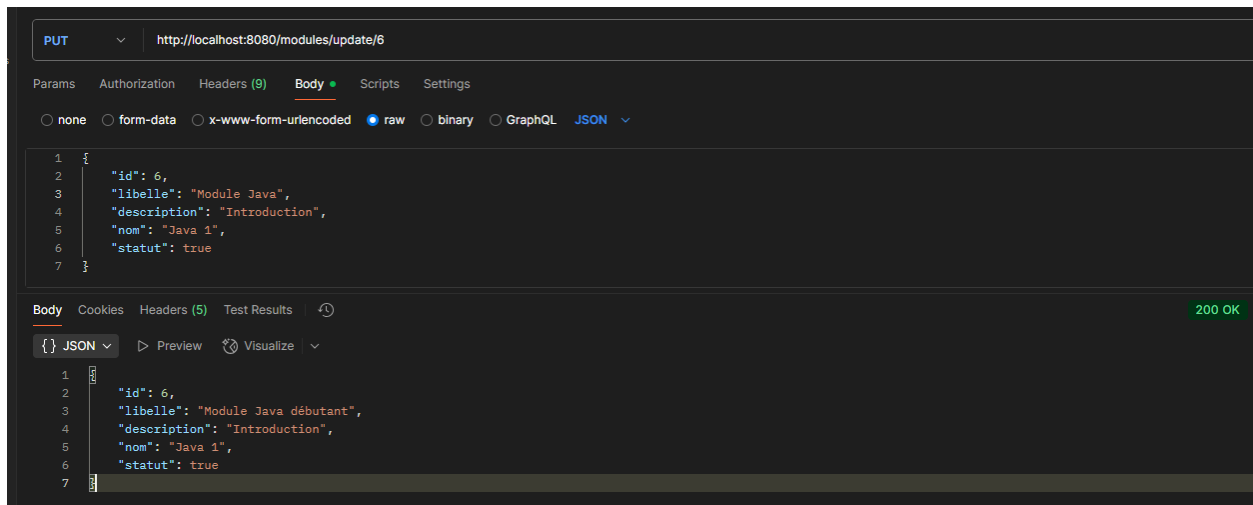
```
{
  "id": 6,
  "libelle": "Module Java débutant",
  "description": "Introduction",
  "nom": "Java 1",
  "statut": true
}
```



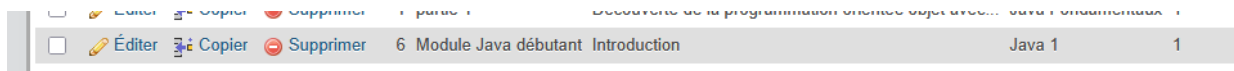
**Résultat Postman :**

200 OK





**Résultat PhpMyAdmin : Le module id = 6 a bien été modifié.**



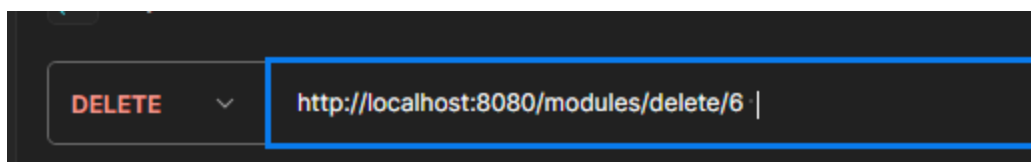
**TEST 22 : → Résultat**

**OK**

**TEST 23 :**

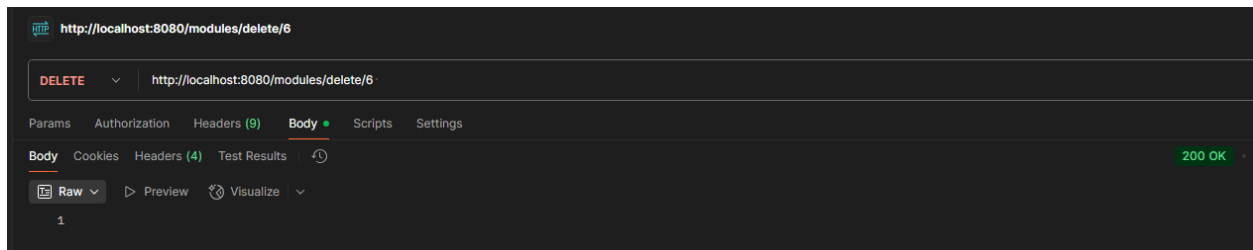
**DELETE [ NE JAMAIS UTILISER SAUF EXCEPTION ] > SUPPRIMER = NO RETURN**

DELETE http://localhost:8080/modules/delete/6 → Supprimer un module



**Résultat Postman :**

**200 OK**



**Résultat PhpMyAdmin : Le module id = 6 a bien été supprimé définitivement.**

				id	libelle	description	nom	statut
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	partie 5	Mise en production et CI/CD	Déploiement	0
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	partie 4	Introduction à SQL et à la modélisation	Base de données	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	partie 3	Développement d'API REST avec Spring Boot	Spring Boot	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	partie 2	HTML, CSS, JavaScript pour le développement web	Web Frontend	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	partie 1	Découverte de la programmation orientée objet avec...	Java Fondamentaux	1

↑ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

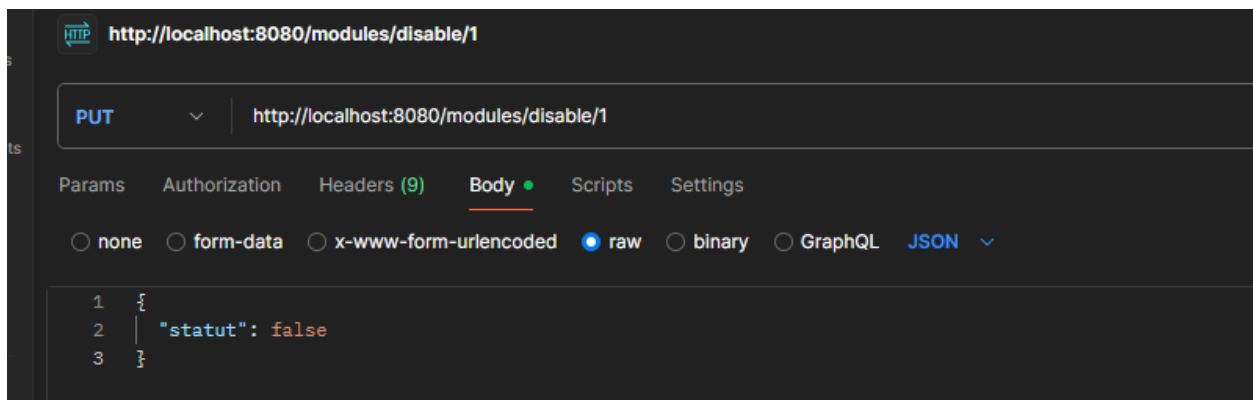
**TEST 23 : → Résultat**

**OK**

**TEST 24 :**

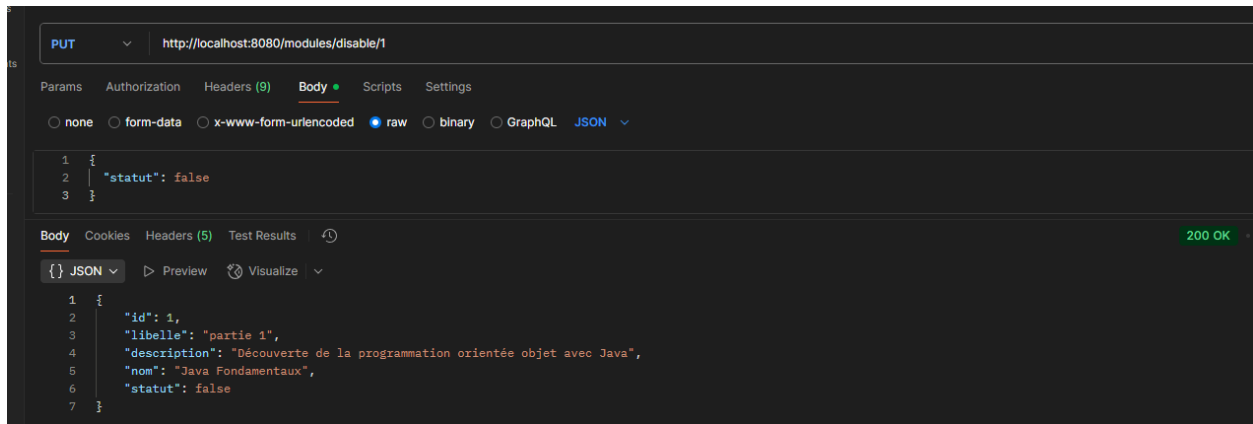
**PUT > MODIFIER / UPDATE [ AU LIEU de DELETE → désactivation par le statut]**

PUT http://localhost:8080/modules/disable/1 → Désactiver un module



**Résultat Postman :**

200 OK



**Résultat PhpMyAdmin : Le module id = 1 a bien été désactivé en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0) .**

	id	libelle	description	nom	statut
<input type="checkbox"/>	5	partie 5	Mise en production et CI/CD	Déploiement	0

**TEST 24 : → Résultat**

**OK**

**Vs Code console terminal : Travail du package (maven) Hibernate ⇒ requêtes SQL envoyé avec Postman visibles.**

```
Hibernate: select m1_0.id,m1_0.description,m1_0.libelle,m1_0.nom,m1_0.statut from module m1_0 where m1_0.id=?
Hibernate: select m1_0.id,m1_0.description,m1_0.libelle,m1_0.nom,m1_0.statut from module m1_0 where m1_0.id=?
Hibernate: update module set description=?,libelle=?,nom=?,statut=? where id=?
Hibernate: select m1_0.id,m1_0.description,m1_0.libelle,m1_0.nom,m1_0.statut from module m1_0 where m1_0.id=?
Hibernate: select m1_0.id,m1_0.description,m1_0.libelle,m1_0.nom,m1_0.statut from module m1_0 where m1_0.id=?
Hibernate: delete from module where id=?
Hibernate: select m1_0.id,m1_0.description,m1_0.libelle,m1_0.nom,m1_0.statut from module m1_0 where m1_0.id=?
Hibernate: update module set description=?,libelle=?,nom=?,statut=? where id=?
```

## Entité Séquence

SELECT \* FROM `sequence`

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

				id	libelle	description	ordre	statut	module_id
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Variables et types	Premiers pas avec les variables Java	1	1	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Boucles et conditions	Comprendre les structures de contrôle	2	1	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Classes et objets	POO en Java	3	1	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	HTML de base	Structure d'une page web	1	1	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	CSS fondamental	Mettre en forme ses pages	2	1	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	6	JavaScript initiation	Dynamiser le web	3	1	2
<input type="checkbox"/>	Éditer	Copier	Supprimer	7	Création d'un projet Spring Boot	Initialisation et configuration	1	1	3
<input type="checkbox"/>	Éditer	Copier	Supprimer	8	Contrôleurs REST	Créer des endpoints REST	2	1	3
<input type="checkbox"/>	Éditer	Copier	Supprimer	9	JPA et persistance	Connexion à la base de données	3	1	3
<input type="checkbox"/>	Éditer	Copier	Supprimer	10	Modélisation SQL	Concevoir un schéma relationnel	1	1	4
<input type="checkbox"/>	Éditer	Copier	Supprimer	11	Requêtes de base	SELECT, INSERT, UPDATE, DELETE	2	1	4
<input type="checkbox"/>	Éditer	Copier	Supprimer	12	Déploiement sur serveur	Publier une application	1	0	5

## Table `sequence` PhpMyAdmin AVANT TESTS.

### TEST 25:

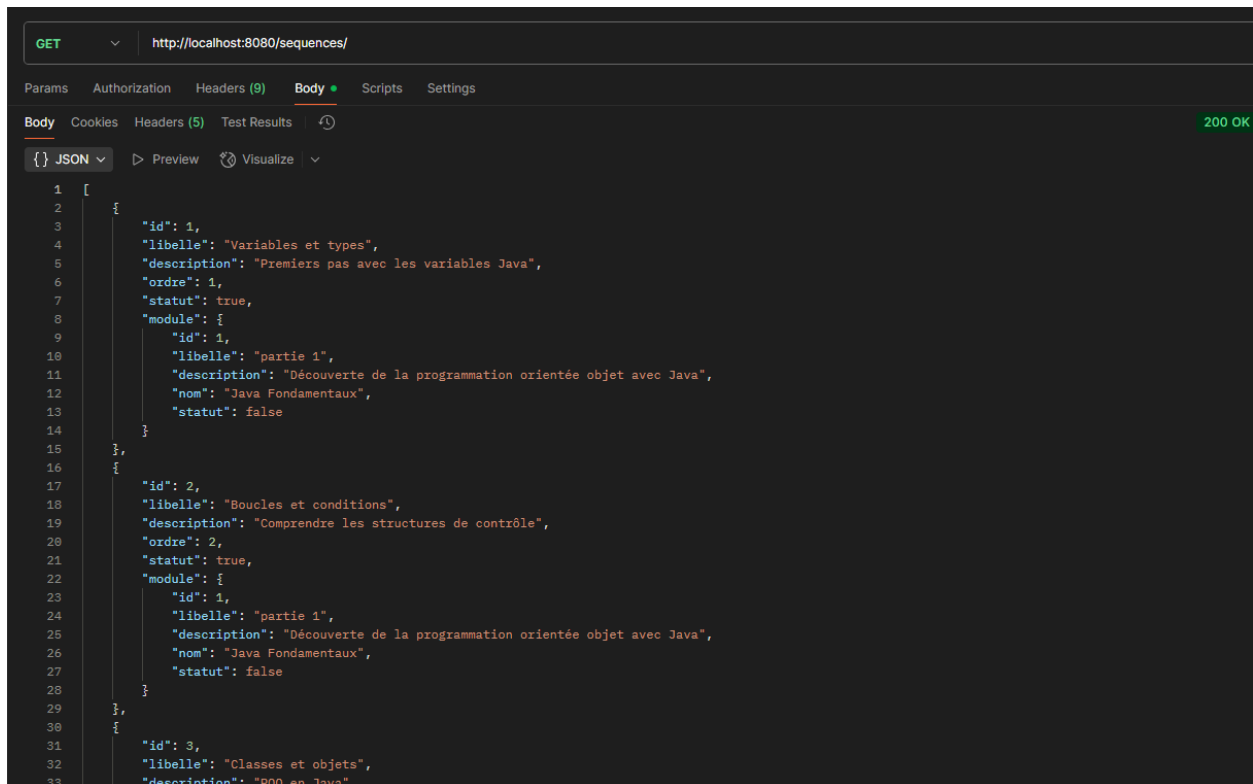
#### GET > RECUPERER / READ

GET `http://localhost:8080/sequences/` → liste toutes les séquences

GET `http://localhost:8080/sequences/`

### Résultat Postman :

200 OK



The screenshot shows a Postman interface with a GET request to `http://localhost:8080/sequences/`. The 'Body' tab is selected, displaying a JSON array of three objects. The status bar indicates '200 OK'.

```
1  [
2    {
3      "id": 1,
4      "libelle": "Variables et types",
5      "description": "Premiers pas avec les variables Java",
6      "ordre": 1,
7      "statut": true,
8      "module": {
9        "id": 1,
10       "libelle": "partie 1",
11       "description": "Découverte de la programmation orientée objet avec Java",
12       "nom": "Java Fondamentaux",
13       "statut": false
14     }
15   },
16   {
17     "id": 2,
18     "libelle": "Boucles et conditions",
19     "description": "Comprendre les structures de contrôle",
20     "ordre": 2,
21     "statut": true,
22     "module": {
23       "id": 1,
24       "libelle": "partie 1",
25       "description": "Découverte de la programmation orientée objet avec Java",
26       "nom": "Java Fondamentaux",
27       "statut": false
28     }
29   },
30   {
31     "id": 3,
32     "libelle": "Classes et objets",
33     "description": "POO en Java",
```

### Liste des sequences du gestionnaire de formation récupéré au format JSON :

```
[
  {
    "id": 1,
    "libelle": "Variables et types",
    "description": "Premiers pas avec les variables Java",
    "ordre": 1,
    "statut": true,
    "module": {
      "id": 1,
      "libelle": "partie 1",
      "description": "Découverte de la programmation orientée objet avec Java",
      "nom": "Java Fondamentaux",
      "statut": false
    }
  },
  {
```

```

    "id": 2,
    "libelle": "Boucles et conditions",
    "description": "Comprendre les structures de contrôle",
    "ordre": 2,
    "statut": true,
    "module": {
      "id": 1,
      "libelle": "partie 1",
      "description": "Découverte de la programmation orientée objet avec Java",
      "nom": "Java Fondamentaux",
      "statut": false
    }
  },
  {
    "id": 3,
    "libelle": "Classes et objets",
    "description": "POO en Java",
    "ordre": 3,
    "statut": true,
    "module": {
      "id": 1,
      "libelle": "partie 1",
      "description": "Découverte de la programmation orientée objet avec Java",
      "nom": "Java Fondamentaux",
      "statut": false
    }
  },
  {
    "id": 4,
    "libelle": "HTML de base",
    "description": "Structure d'une page web",
    "ordre": 1,
    "statut": true,
    "module": {
      "id": 2,
      "libelle": "partie 2",

```

```

    "description": "HTML, CSS, JavaScript pour le développement web",
    "nom": "Web Frontend",
    "statut": true
  }
},
{
  "id": 5,
  "libelle": "CSS fondamental",
  "description": "Mettre en forme ses pages",
  "ordre": 2,
  "statut": true,
  "module": {
    "id": 2,
    "libelle": "partie 2",
    "description": "HTML, CSS, JavaScript pour le développement web",
    "nom": "Web Frontend",
    "statut": true
  }
},
{
  "id": 6,
  "libelle": "JavaScript initiation",
  "description": "Dynamiser le web",
  "ordre": 3,
  "statut": true,
  "module": {
    "id": 2,
    "libelle": "partie 2",
    "description": "HTML, CSS, JavaScript pour le développement web",
    "nom": "Web Frontend",
    "statut": true
  }
},
{
  "id": 7,
  "libelle": "Création d'un projet Spring Boot",

```

```

    "description": "Initialisation et configuration",
    "ordre": 1,
    "statut": true,
    "module": {
      "id": 3,
      "libelle": "partie 3",
      "description": "Développement d'API REST avec Spring Boot",
      "nom": "Spring Boot",
      "statut": true
    }
  },
  {
    "id": 8,
    "libelle": "Contrôleurs REST",
    "description": "Créer des endpoints REST",
    "ordre": 2,
    "statut": true,
    "module": {
      "id": 3,
      "libelle": "partie 3",
      "description": "Développement d'API REST avec Spring Boot",
      "nom": "Spring Boot",
      "statut": true
    }
  },
  {
    "id": 9,
    "libelle": "JPA et persistance",
    "description": "Connexion à la base de données",
    "ordre": 3,
    "statut": true,
    "module": {
      "id": 3,
      "libelle": "partie 3",
      "description": "Développement d'API REST avec Spring Boot",
      "nom": "Spring Boot",

```



```

    "statut": true
  }
},
{
  "id": 10,
  "libelle": "Modélisation SQL",
  "description": "Concevoir un schéma relationnel",
  "ordre": 1,
  "statut": true,
  "module": {
    "id": 4,
    "libelle": "partie 4",
    "description": "Introduction à SQL et à la modélisation",
    "nom": "Base de données",
    "statut": true
  }
},
{
  "id": 11,
  "libelle": "Requêtes de base",
  "description": "SELECT, INSERT, UPDATE, DELETE",
  "ordre": 2,
  "statut": true,
  "module": {
    "id": 4,
    "libelle": "partie 4",
    "description": "Introduction à SQL et à la modélisation",
    "nom": "Base de données",
    "statut": true
  }
},
{
  "id": 12,
  "libelle": "Déploiement sur serveur",
  "description": "Publier une application",
  "ordre": 1,

```

```
"statut": false,  
"module": {  
  "id": 5,  
  "libelle": "partie 5",  
  "description": "Mise en production et CI/CD",  
  "nom": "Déploiement",  
  "statut": false  
}  
}  
]
```

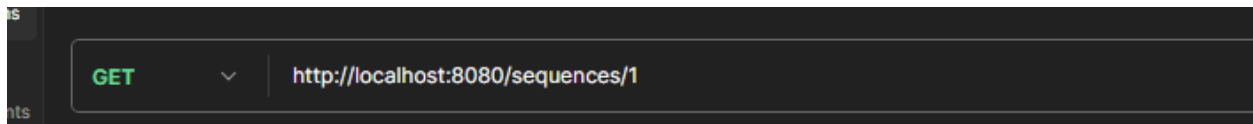
**TEST 25 :** → Résultat

OK

### GET > RECUPERER / READ

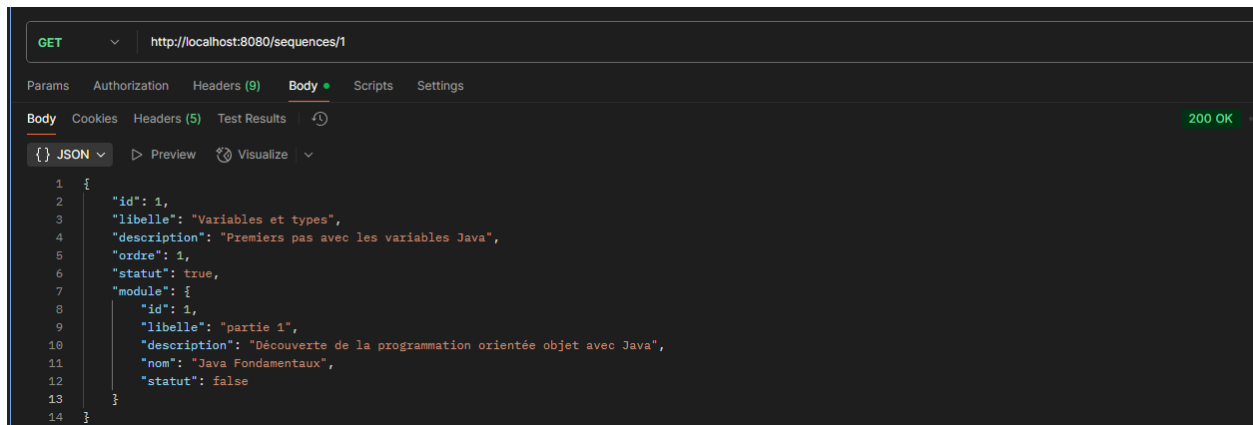
**TEST 26 :**

GET <http://localhost:8080/sequences/1> → détail d'une séquence



**Résultat Postman :**

200 OK



### Sequence id = 1 du gestionnaire de formation récupéré au format JSON :

```
{
  "id": 1,
  "libelle": "Variables et types",
  "description": "Premiers pas avec les variables Java",
  "ordre": 1,
  "statut": true,
  "module": {
    "id": 1,
    "libelle": "partie 1",
    "description": "Découverte de la programmation orientée objet avec Java",
    "nom": "Java Fondamentaux",
    "statut": false
  }
}
```

**TEST 26 :** → Résultat

**OK**

**TEST 27 :**

**POST > CREER / CREATE**

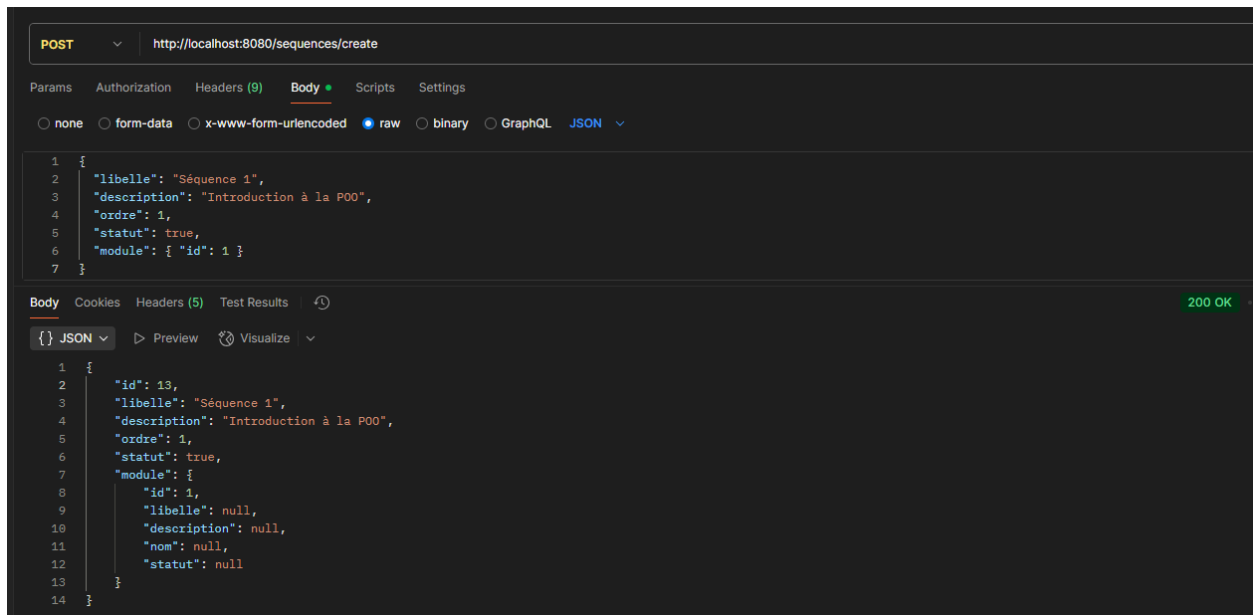
POST http://localhost:8080/sequences/create → Création d'une séquence

```
{
  "libelle": "Séquence 1",
  "description": "Introduction à la POO",
  "ordre": 1,
  "statut": true,
  "module": { "id": 1 }
}
```



### Résultat Postman :

200 OK



**Résultat PhpMyAdmin : La nouvelle sequence a bien été créée.**

<input type="checkbox"/>	Éditer	Copier	Supprimer	12 Déploiement sur serveur	Publier une application	1	0	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	13 Séquence 1	Introduction à la POO	1	1	1

**TEST 27 : → Résultat**

**OK**

**TEST 28 :**

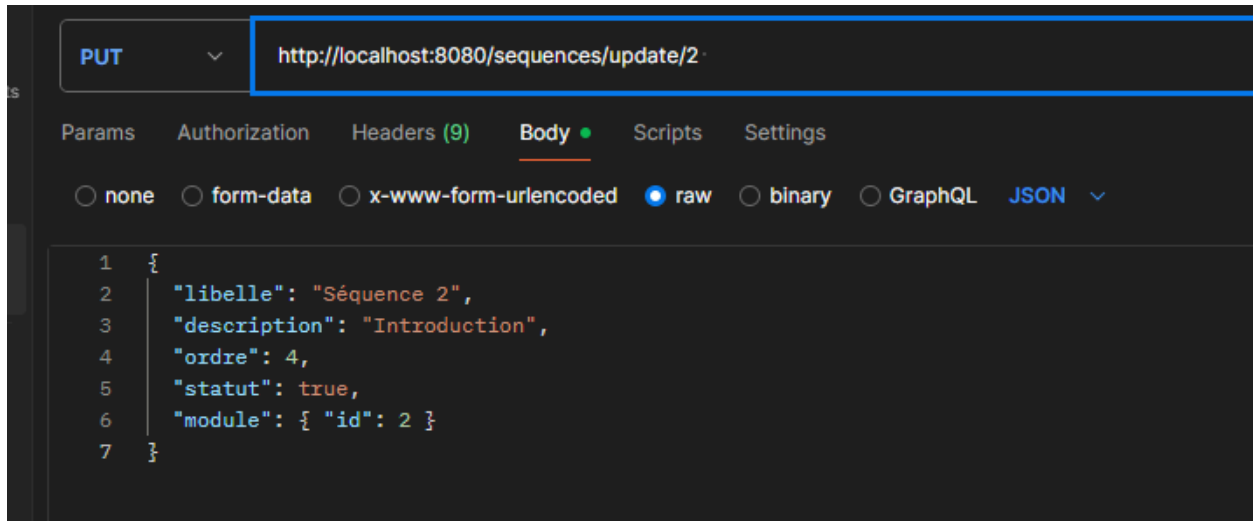
**PUT > MODIFIER / UPDATE**

PUT http://localhost:8080/sequences/update/2 → Modifier une séquence

```

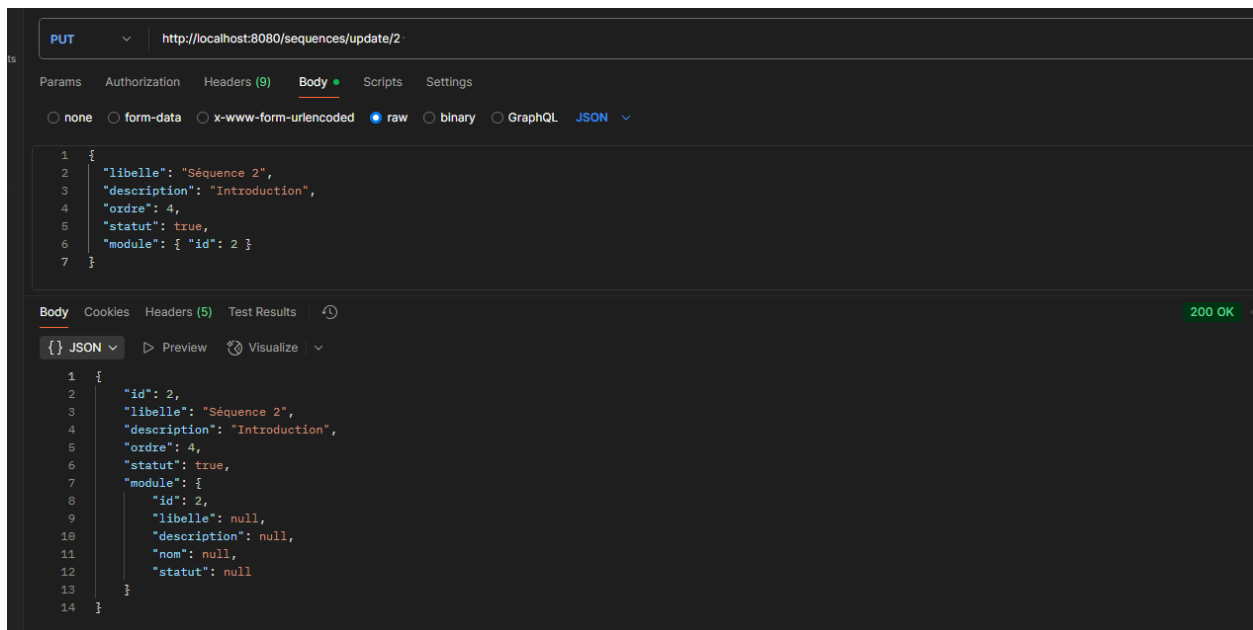
{
  "libelle": "Séquence 2",
  "description": "Introduction",
  "ordre": 4,
  "statut": true,
  "module": { "id": 2 }
}

```



## Résultat Postman :

200 OK



## Résultat PhpMyAdmin : La sequence id = 2 a bien été modifiée.

<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Séquence 2	Introduction	4	1	2
--------------------------	--------	--------	-----------	---	------------	--------------	---	---	---

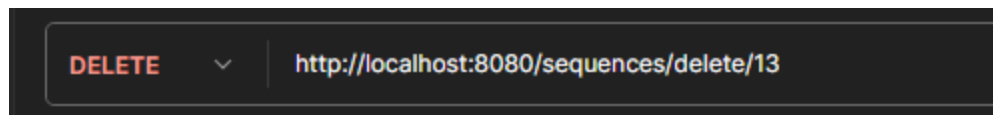
## TEST 28 : → Résultat

OK

## TEST 29 :

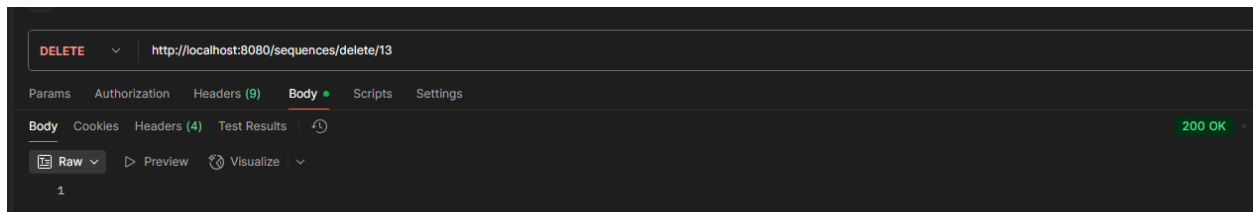
**DELETE [ NE JAMAIS UTILISER SAUF EXCEPTION ] > SUPPRIMER =  
NO RETURN**

DELETE http://localhost:8080/sequences/delete/13 → Supprimer une séquence



## Résultat Postman :

200 OK



Résultat PhpMyAdmin : La sequence id = 13 a bien été supprimée définitivement.

 		id	libelle	description	ordre	statut	module_id
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1 Variables et types	Premiers pas avec les variables Java	1 1	1
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2 Séquence 2	Introduction	4 1	2
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	3 Classes et objets	POO en Java	3 1	1
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	4 HTML de base	Structure d'une page web	1 1	2
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	5 CSS fondamental	Mettre en forme ses pages	2 1	2
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	6 JavaScript initiation	Dynamiser le web	3 1	2
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	7 Création d'un projet Spring Boot	Initialisation et configuration	1 1	3
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	8 Contrôleurs REST	Créer des endpoints REST	2 1	3
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	9 JPA et persistance	Connexion à la base de données	3 1	3
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	10 Modélisation SQL	Concevoir un schéma relationnel	1 1	4
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	11 Requêtes de base	SELECT, INSERT, UPDATE, DELETE	2 1	4
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	12 Déploiement sur serveur	Publier une application	1 0	5

☐ Tout cocher Avec la sélection :  Éditer  Copier  Supprimer  Exporter

**TEST 29 :** → Résultat

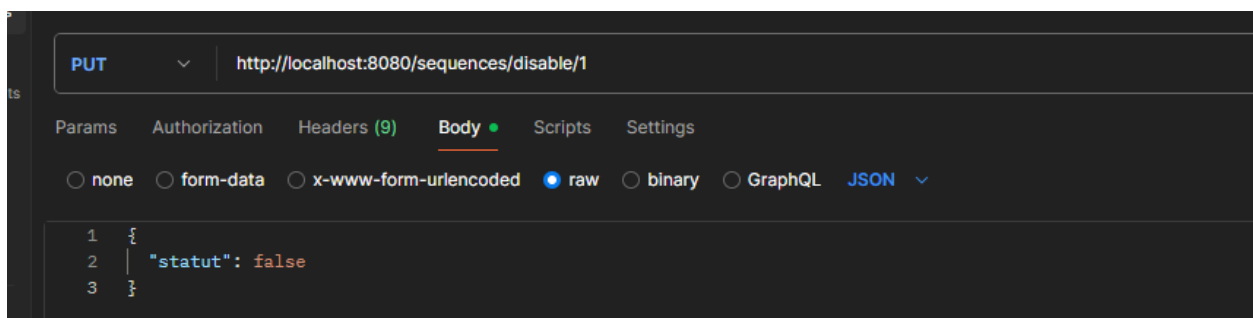
OK

**TEST 30 :**

**PUT > MODIFIER / UPDATE**      **[ AU LIEU de DELETE → désactivation par le statut]**

PUT <http://localhost:8080/roles/disable/1> → Désactiver un rôle

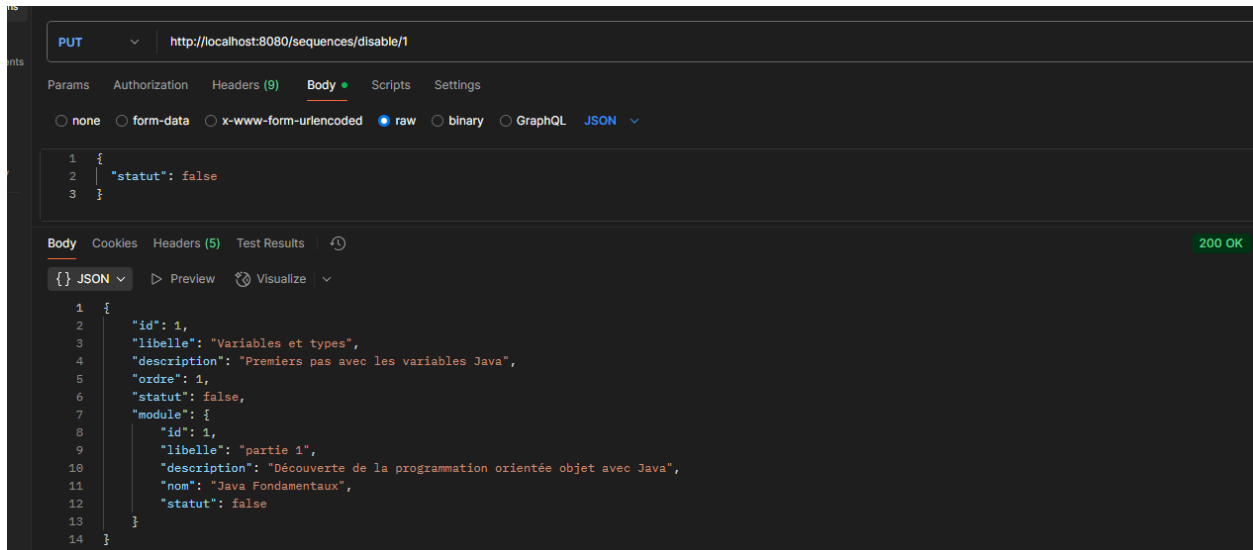
```
{
  "statut": false
}
```





## Résultat Postman :

200 OK



**Résultat PhpMyAdmin :** La sequence id = 1 a bien été désactivée en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0) .

	id	libelle	description	ordre	statut	module_id
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	Variables et types	Premiers pas avec les variables Java	1	0	1

**TEST 30 :** → Résultat

OK

**Vs Code console terminal :** Travail du package (maven) Hibernate ⇒ requêtes SQL envoyé avec Postman visibles.

```

Hibernate: insert into sequence (description,libelle,module_id,ordre,statut) values (?,?,,?,?)
Hibernate: select r1_0.id,r1_0.libelle,r1_0.statut from role r1_0 where r1_0.id=?
Hibernate: select s1_0.id,s1_0.description,s1_0.libelle,s1_0.module_id,m1_0.id,m1_0.description,m1_0.li
belle,m1_0.nom,m1_0.statut,s1_0.ordre,s1_0.statut from sequence s1_0 join module m1_0 on m1_0.id=s1_0.m
odule_id where s1_0.id=?
Hibernate: update sequence set description=?,libelle=?,module_id=?,ordre=?,statut=? where id=?
Hibernate: select s1_0.id,s1_0.description,s1_0.libelle,s1_0.module_id,m1_0.id,m1_0.description,m1_0.li
belle,m1_0.nom,m1_0.statut,s1_0.ordre,s1_0.statut from sequence s1_0 join module m1_0 on m1_0.id=s1_0.m
odule_id where s1_0.id=?
Hibernate: delete from sequence where id=?
Hibernate: select s1_0.id,s1_0.description,s1_0.libelle,s1_0.module_id,m1_0.id,m1_0.description,m1_0.li
belle,m1_0.nom,m1_0.statut,s1_0.ordre,s1_0.statut from sequence s1_0 join module m1_0 on m1_0.id=s1_0.m
odule_id where s1_0.id=?
Hibernate: update sequence set description=?,libelle=?,module_id=?,ordre=?,statut=? where id=?

```

## Entité Centre

`SELECT * FROM `centre``

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

`UPDATE `centre` SET `nom` = 'Bouquet' WHERE `centre`.`id` = 4;`

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

		id	nom	adresse_postal	ville	code_postal	telephone	libelle	adresse	statut
<input type="checkbox"/>	Éditer	1	Candau	9, rue Ferdinand de Candau	Mont-de-Marsan	40000	0005532154	1		0
<input type="checkbox"/>	Éditer	2	Bosquet	10, rue lala	Mont-de-Marsan	40000	0002335566	2		0
<input type="checkbox"/>	Éditer	3	Kandau	3 lala	Mont	40000	0000000	2		0
<input type="checkbox"/>	Éditer	4	Bouquet	4 fibxd	Mont	40000	2222222	1		0

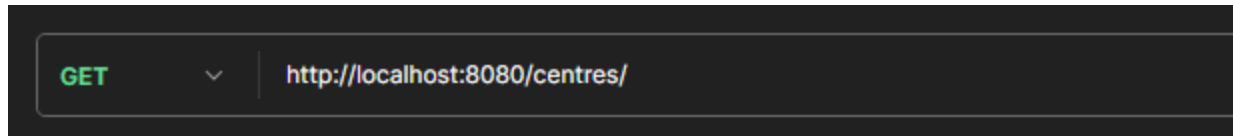
↑ ☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

## Table centre PhpMyAdmin AVANT TESTS.

### TEST 31 :

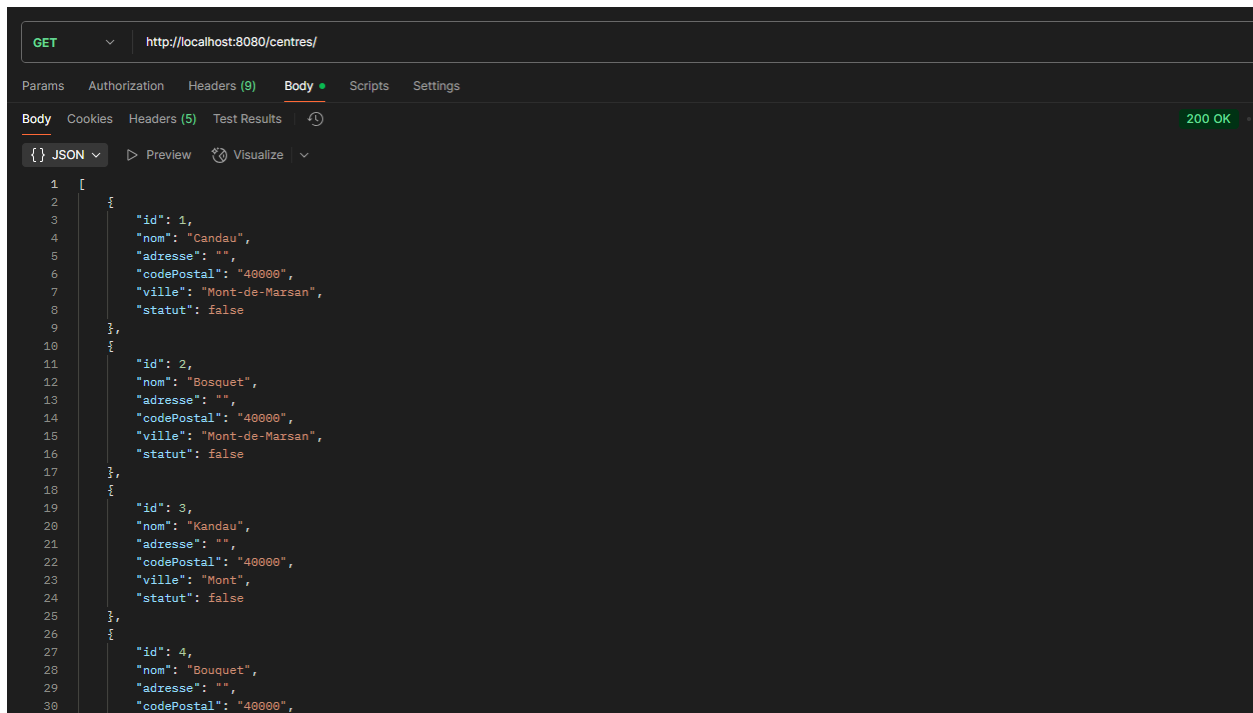
#### GET > RECUPERER / READ

GET <http://localhost:8080/centres/> → liste tous les centres



## Résultat Postman :

200 OK



## Liste des centres du gestionnaire de formation récupéré au format JSON :

```
[
  {
    "id": 1,
    "nom": "Candau",
    "adresse": "",
    "codePostal": "40000",
    "ville": "Mont-de-Marsan",
    "statut": false
  },
  {
```

```
[
  {
    "id": 2,
    "nom": "Bosquet",
    "adresse": "",
    "codePostal": "40000",
    "ville": "Mont-de-Marsan",
    "statut": false
  },
  {
    "id": 3,
    "nom": "Kandau",
    "adresse": "",
    "codePostal": "40000",
    "ville": "Mont",
    "statut": false
  },
  {
    "id": 4,
    "nom": "Bouquet",
    "adresse": "",
    "codePostal": "40000",
    "ville": "Mont",
    "statut": false
  }
]
```

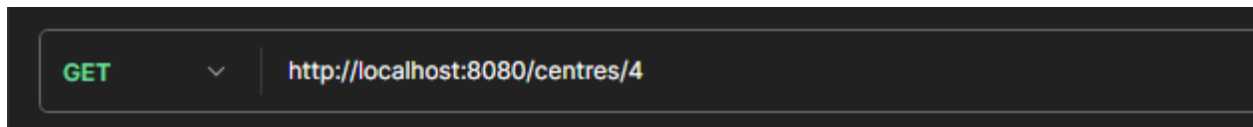
**TEST 31 :** → Résultat

**OK**

**GET > RECUPERER / READ**

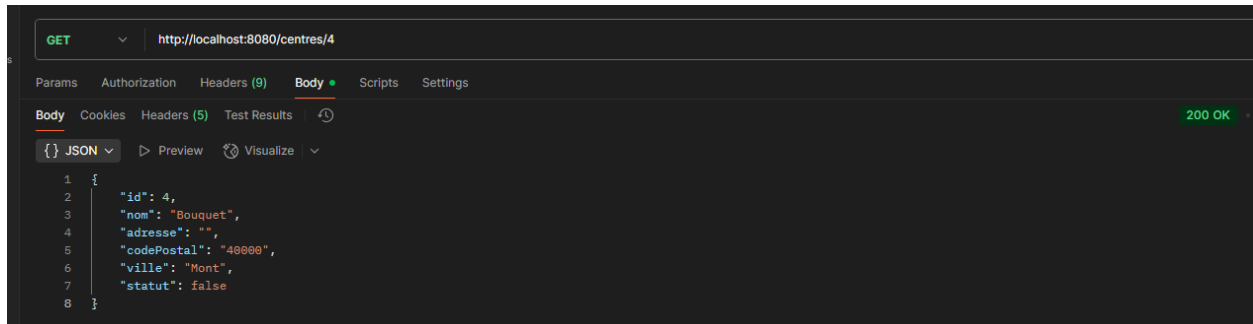
**TEST 32 :**

GGET http://localhost:8080/centres/4 → détail d'un centre



## Résultat Postman :

200 OK



## Centre id = 4 du gestionnaire de formation récupéré au format JSON :

```
{
  "id": 4,
  "nom": "Bouquet",
  "adresse": "",
  "codePostal": "40000",
  "ville": "Mont",
  "statut": false
}
```

## TEST 32 : → Résultat

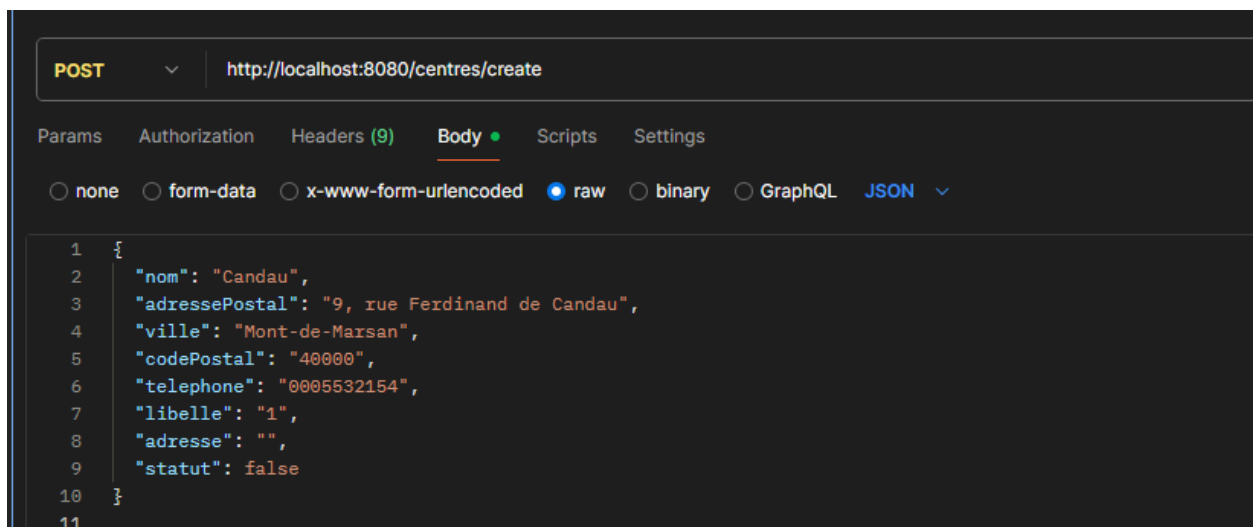
OK

## TEST 33 :

### POST > CREER / CREATE

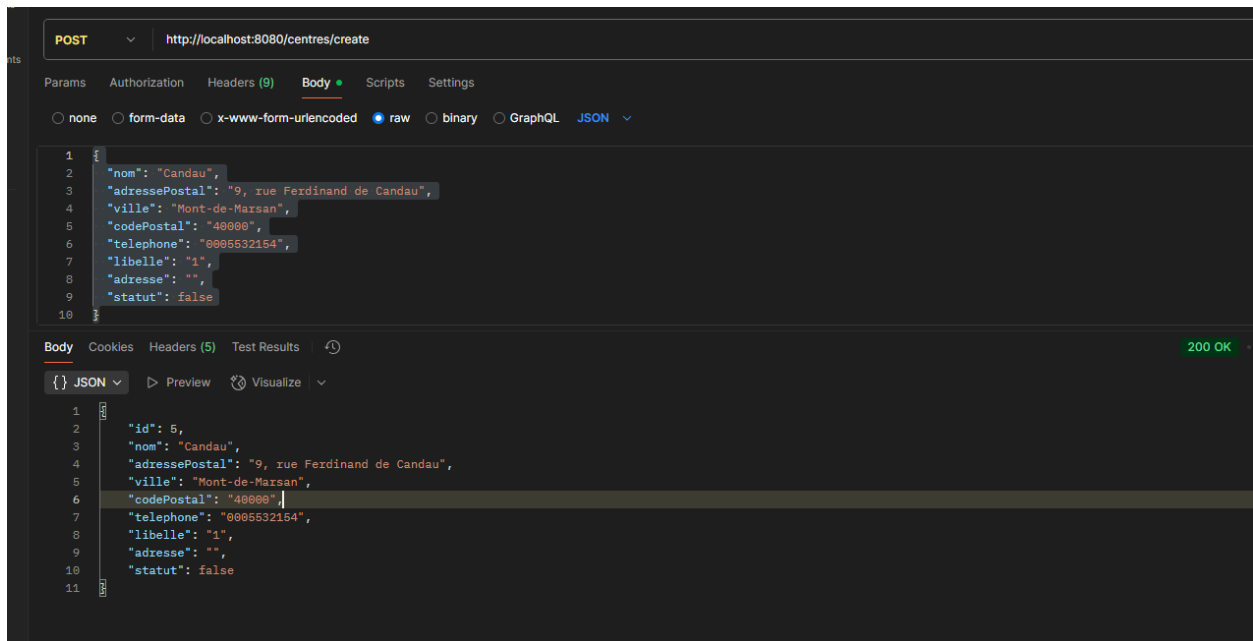
POST http://localhost:8080/centres/create → Création d'un centre

```
{
  "nom": "Candau",
  "adressePostal": "9, rue Ferdinand de Candau",
  "ville": "Mont-de-Marsan",
  "codePostal": "40000",
  "telephone": "0005532154",
  "libelle": "1",
  "adresse": "",
  "statut": false
}
```



### Résultat Postman :

200 OK



**Résultat PhpMyAdmin : Le nouveau centre id = 5 bien été créé.**

	id	nom	adresse_postal	ville	code_postal	telephone	libelle	adresse	statut
<input type="checkbox"/>	1	Candau	9, rue Ferdinand de Candau	Mont-de-Marsan	40000	0005532154	1		0
<input type="checkbox"/>	4	Bouquet	4 fibxd	Mont	40000	2222222	1		0
<input type="checkbox"/>	5	Candau	9, rue Ferdinand de Candau	Mont-de-Marsan	40000	0005532154	1		0
<input type="checkbox"/>	2	Bosquet	10, rue lala	Mont-de-Marsan	40000	0002335566	2		0
<input type="checkbox"/>	3	Kandau	3 lala	Mont	40000	0000000	2		0

**TEST 33 : → Résultat**

**OK**

**TEST 34 :**

**PUT > MODIFIER / UPDATE**

PUT <http://localhost:8080/centres/update/1> → Modifier un centre

```

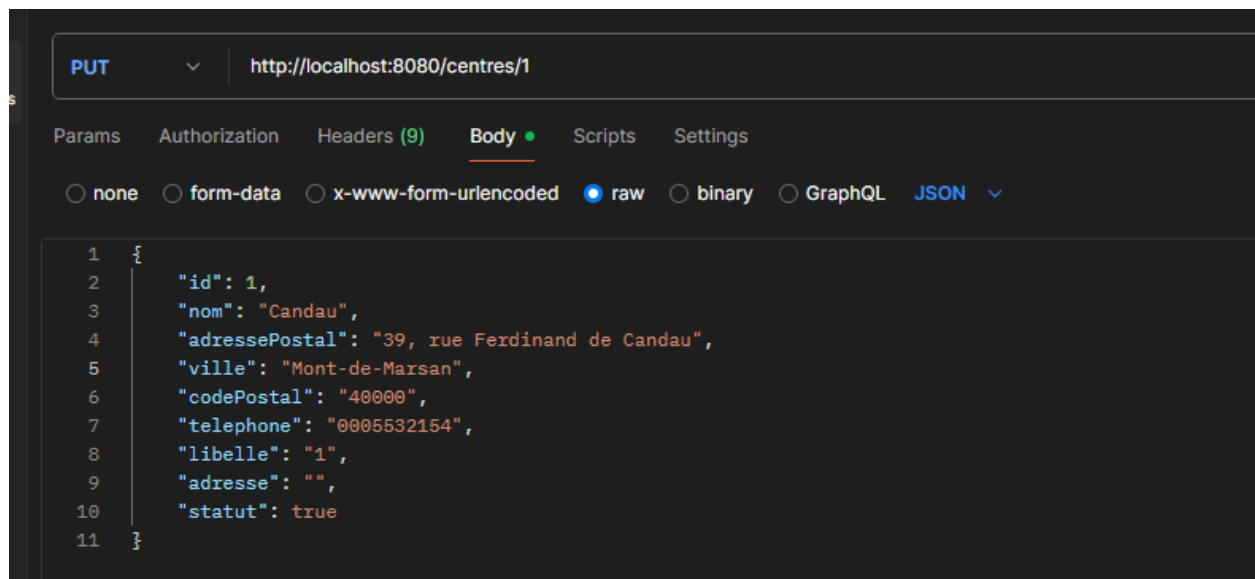
{
  "nom": "Candau",
  "adressePostal": "39, rue Ferdinand de Candau",
  "ville": "Mont-de-Marsan",
  "codePostal": "40000",

```

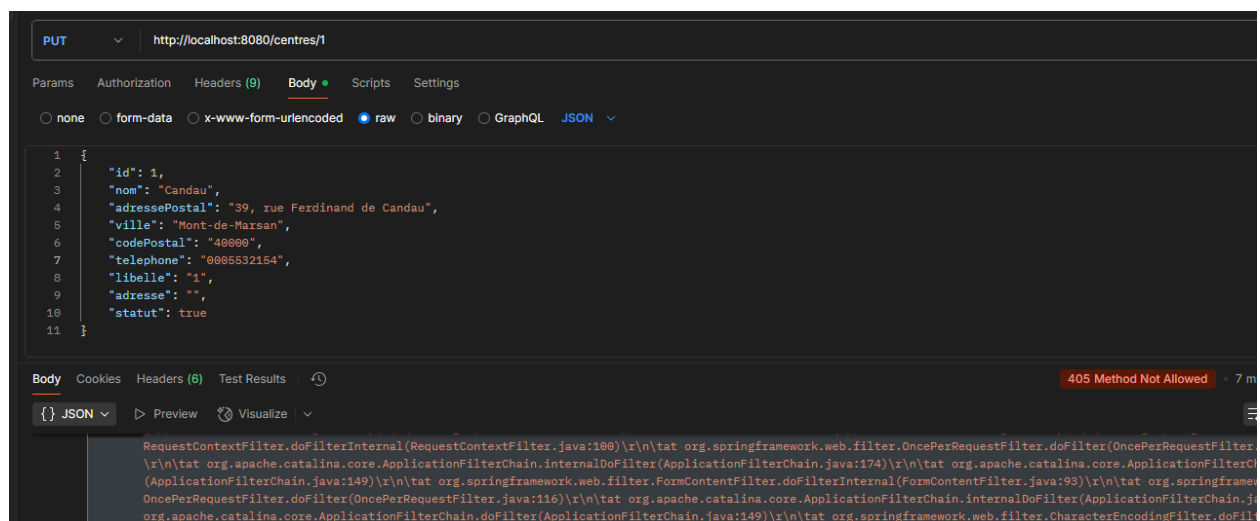
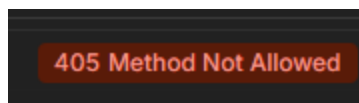
```

"telephone": "0005532154",
"libelle": "1",
"adresse": "",
"statut": true
}

```



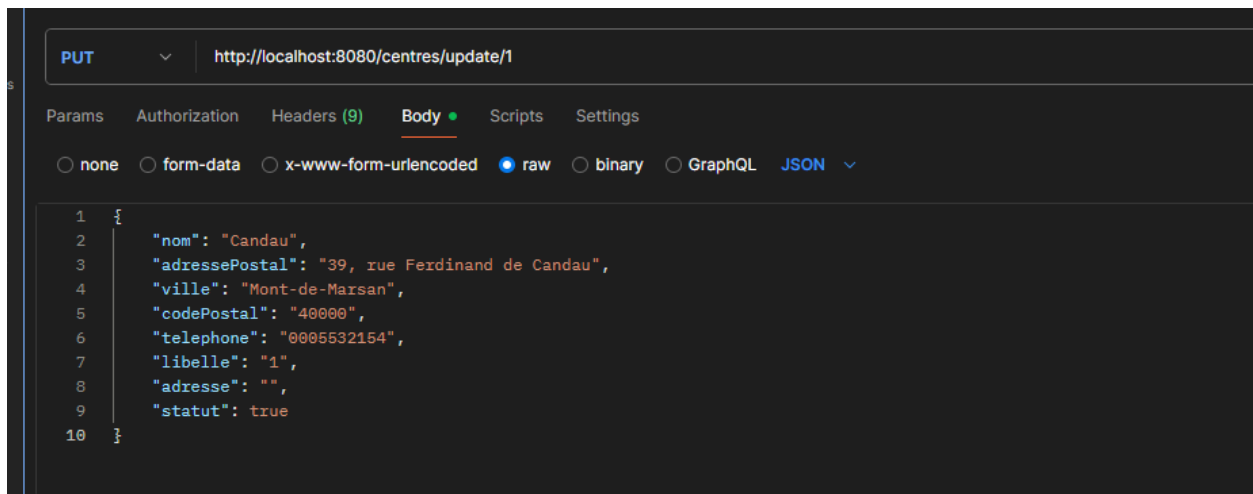
## Résultat Postman :



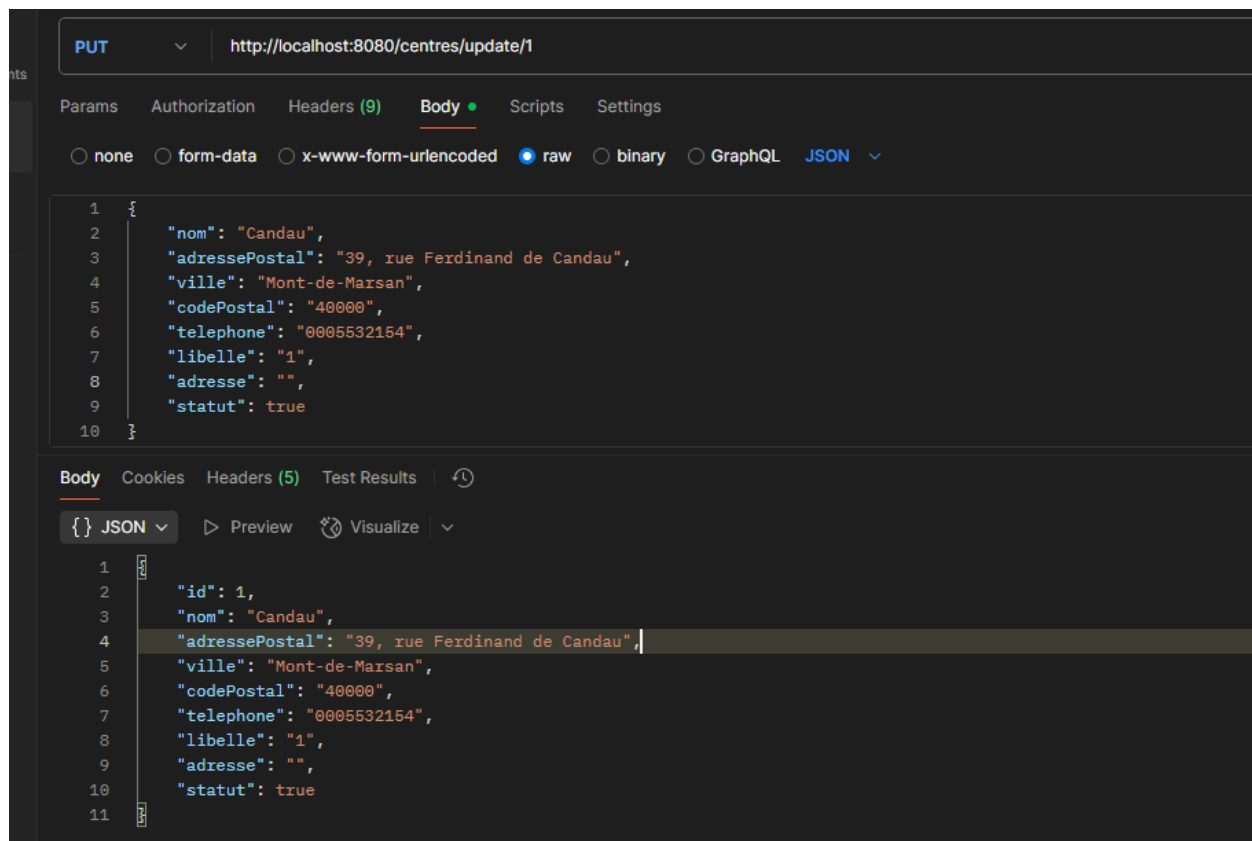


## Erreur 405 "Method Not Allowed" avec `/centres/1` (PUT)

- j'ai eu une erreur 405 car tu utilisais une URL qui n'accepte pas la méthode PUT dans ton contrôleur Spring.
- **Pour corriger, il faut soit utiliser l'URL `/centres/update/1` dans Postman, soit** modifier le contrôleur pour accepter aussi `/centres/1` en PUT.
- Une fois la bonne URL ou le bon mapping utilisé, ta requête passera en 200 et la mise à jour fonctionnera



200 OK



**Résultat PhpMyAdmin : Le centre id = 1 bien été modifié.**

	id	nom	adresse_postal	ville	code_postal	telephone	libelle	adresse	statut
<input type="checkbox"/>	1	Candau	39, rue Ferdinand de Candau	Mont-de-Marsan	40000	0005532154	1		1

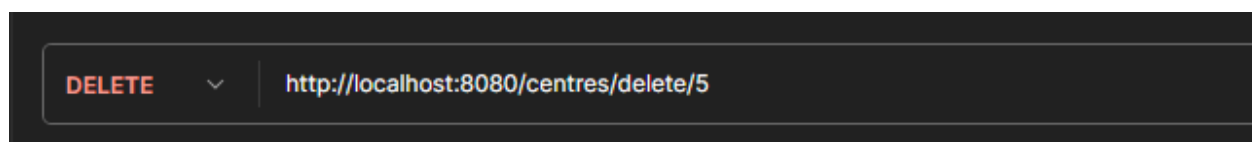
**TEST 34 : → Résultat**

**OK**

**TEST 35 :**

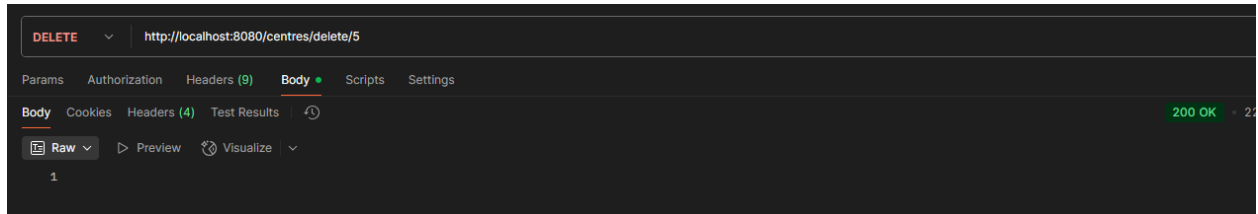
**DELETE [ NE JAMAIS UTILISER SAUF EXCEPTION ] > SUPPRIMER = NO RETURN**

DELETE http://localhost:8080/centres/delete/5 → Supprimer un centre



## Résultat Postman :

200 OK



## Résultat PhpMyAdmin : Le centre id = 5 bien été supprimé définitivement.

				id	nom	adresse_postal	ville	code_postal	telephone	libelle	adresse	statut
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Candau	39, rue Ferdinand de Candau	Mont-de-Marsan	40000	0005532154	1		1
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	Bouquet	4 fibxd	Mont	40000	2222222	1		0
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Bosquet	10, rue lala	Mont-de-Marsan	40000	0002335566	2		0
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Kandau	3 lala	Mont	40000	0000000	2		0

Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

## TEST 35 : → Résultat

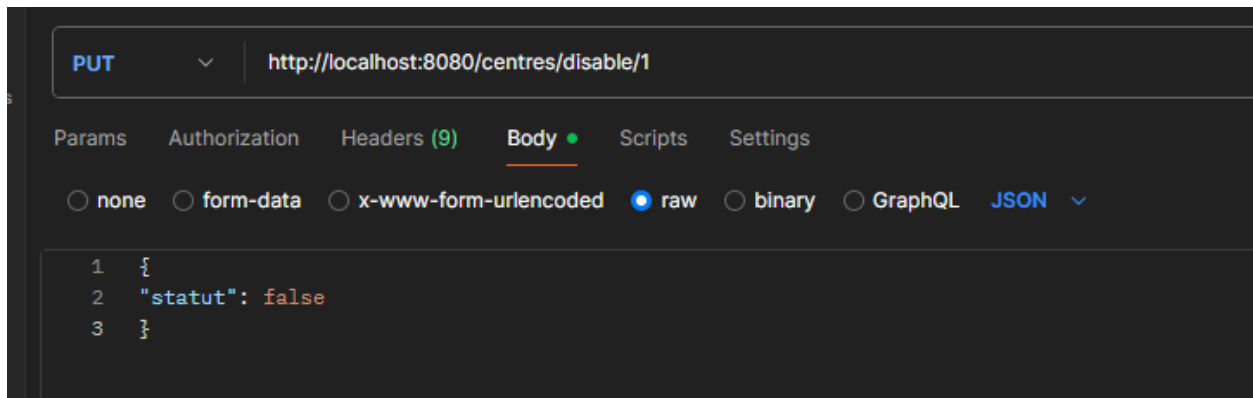
OK

## TEST 36 :

PUT > MODIFIER / UPDATE [ AU LIEU de DELETE → désactivation par le statut]

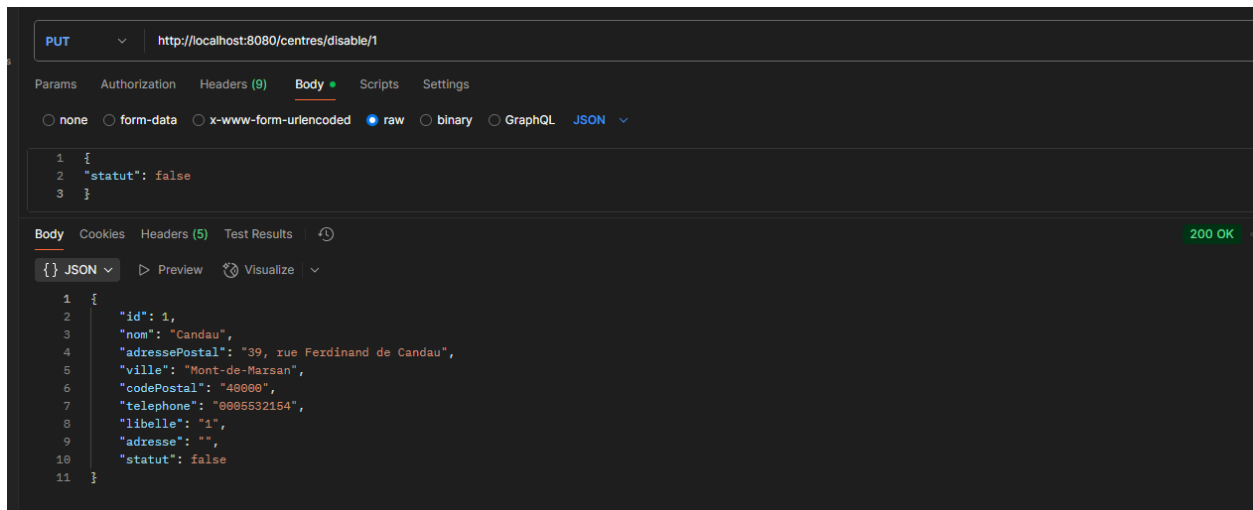
PUT `http://localhost:8080/centres/disable/1` → Désactiver un centre

```
{
  "statut": false
}
```



## Résultat Postman :

200 OK



**Résultat PhpMyAdmin : Le centre id = 1 a bien été activé en utilisant son statut qui est passé de "actif" (true = 1) à "inactif" (false = 0) .**

	id	nom	adresse_postal	ville	code_postal	telephone	libelle	1	adresse	statut
<input type="checkbox"/>	1	Candau	39, rue Ferdinand de Candau	Mont-de-Marsan	40000	0005532154	1			0

**TEST 36 : → Résultat**

**OK**

**Vs Code console terminal : Travail du package (maven) Hibernate ⇒ requêtes SQL envoyé avec Postman visibles.**

```

Hibernate: select c1_0.id,c1_0.adresse,c1_0.code_postal,c1_0.nom,c1_0.statut,c1_0.ville from centre c1_0
Hibernate: select c1_0.id,c1_0.adresse,c1_0.code_postal,c1_0.nom,c1_0.statut,c1_0.ville from centre c1_0 where c1_0.id=?
Hibernate: insert into centre (adresse,code_postal,nom,statut,ville) values (?,?,,?,?)
2025-05-16T16:47:22.088+02:00 WARN 20356 --- [nio-8080-exec-2] o.h.engine.jdbc.spi.SqlExceptionHelper
: SQL Error: 1364, SQLState: HY000
2025-05-16T16:47:22.089+02:00 ERROR 20356 --- [nio-8080-exec-2] o.h.engine.jdbc.spi.SqlExceptionHelper
: Field 'adresse_postal' doesn't have a default value
2025-05-16T16:47:22.098+02:00 ERROR 20356 --- [nio-8080-exec-2] o.a.c.c.C.[.][/].[dispatcherServlet]
: Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request
processing failed: org.springframework.orm.jpa.JpaSystemException: could not execute statement [Field '
adresse_postal' doesn't have a default value] [insert into centre (adresse,code_postal,nom,statut,ville
) values (?,?,,?,?)]] with root cause

```

Completed initialization in 1 ms

```

Hibernate: select c1_0.id,c1_0.adresse,c1_0.adresse_postal,c1_0.code_postal,c1_0.libelle,c1_0.nom,c1_0.
statut,c1_0.telephone,c1_0.ville from centre c1_0 where c1_0.id=?
Hibernate: update centre set adresse=?,adresse_postal=?,code_postal=?,libelle=?,nom=?,statut=?,telephon
e=?,ville=? where id=?
Hibernate: select c1_0.id,c1_0.adresse,c1_0.adresse_postal,c1_0.code_postal,c1_0.libelle,c1_0.nom,c1_0.
statut,c1_0.telephone,c1_0.ville from centre c1_0 where c1_0.id=?
Hibernate: delete from centre where id=?
Hibernate: select c1_0.id,c1_0.adresse,c1_0.adresse_postal,c1_0.code_postal,c1_0.libelle,c1_0.nom,c1_0.
statut,c1_0.telephone,c1_0.ville from centre c1_0 where c1_0.id=?
Hibernate: update centre set adresse=?,adresse_postal=?,code_postal=?,libelle=?,nom=?,statut=?,telephon
e=?,ville=? where id=?

```

**Les 36 tests sont ok.**