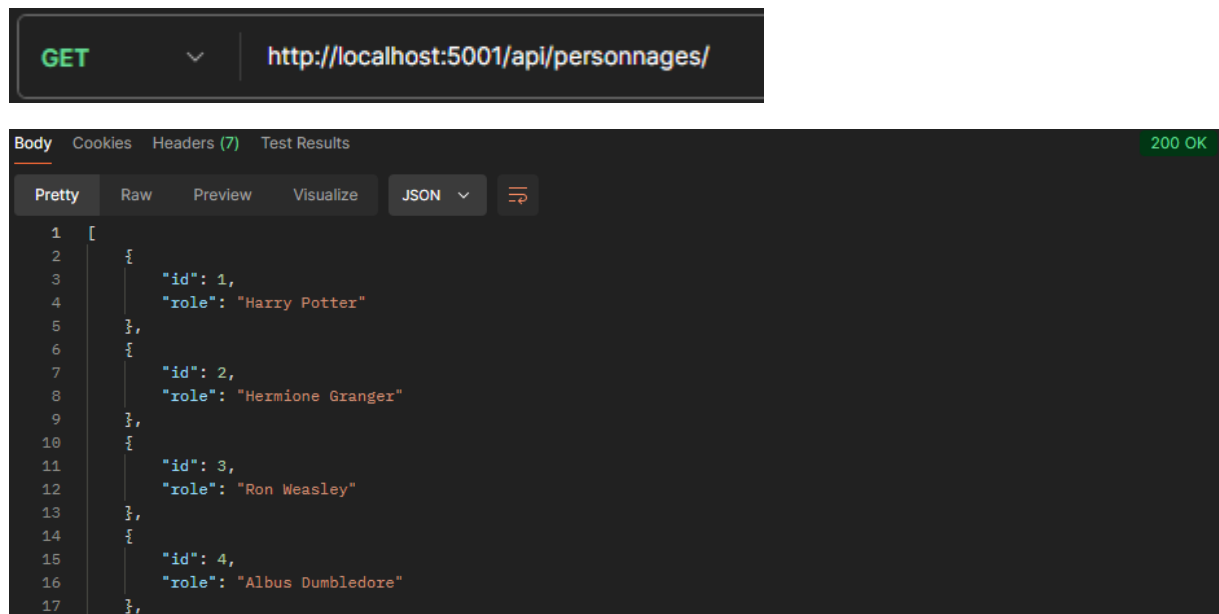


API Node JS

Méthode HTTP : GET

Chemin : /api/personnages

Fonction : Renvoyer la liste complète des personnages sous forme de JSON.

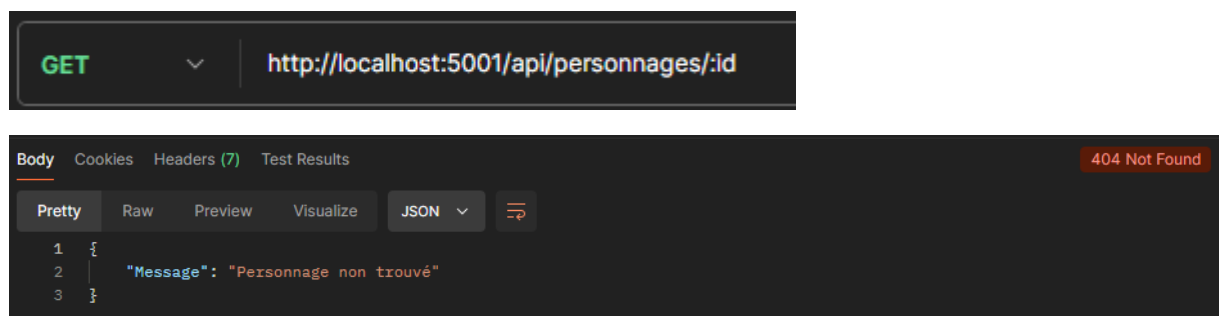


Méthode HTTP : GET

Chemin : /api/personnages/:id

Fonction : Rechercher un personnage par ID et le renvoyer en JSON. Si le personnage n'est pas trouvé, renvoyer un statut 404 avec un message d'erreur.

ID inexistant



ID existant

GET `http://localhost:5001/api/personnages/:id`

Params • Authorization Headers (9) Body • Scripts Settings

Query Params

	Key	Value
	Key	Value

Path Variables

	Key	Value
	id	1

Body Cookies Headers (7) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {  
2   "name": "Harry Potter"  
3 }
```

Méthode HTTP : POST

Chemin : /api/personnages

Fonction : Ajouter un nouveau personnage avec les données fournies dans le corps de la requête et le renvoyer en JSON avec un statut 201.

POST `http://localhost:5001/api/personnages/`

Params Authorization Headers (9) Body • Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {  
2   "role": "Michael Kyle",  
3   "age": 33  
4 }
```

```
Body Cookies Headers (7) Test Results 201 Created
Pretty Raw Preview Visualize JSON
1 {
2   "id": 11,
3   "role": "Michael Kyle"
4 }
```

Méthode HTTP : PUT

Chemin : /api/personnages/:id

Fonction : Trouver un personnage par ID et mettre à jour ses informations avec les données fournies dans le corps de la requête. Renvoyer le personnage mis à jour en JSON.

ID inexistant

```
PUT http://localhost:5001/api/personnages/:id
Params Authorization Headers (9) Body Scripts Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   "role": "Michael Jackson"
3 }
```

Path Variables		
	Key	Value
	id	11

```
Body Cookies Headers (7) Test Results 404 Not Found
Pretty Raw Preview Visualize JSON
1 "Cet ID n'existe pas, il est donc inchangeable."
```

ID existant

PUT ▼ http://localhost:5001/api/personnages/:id

Params ● Authorization Headers (9) **Body** ● Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   |   "role": "Michael Jackson"
3 }
```

Path Variables

Key	Value
id	10

Body Cookies Headers (7) Test Results 200 OK

Pretty Raw Preview Visualize **JSON** ▼

```
1 "Utilisateur avec l'id : 10 a été mis à jour "
```

Méthode HTTP : DELETE

Chemin : /api/personnages/:id

Fonction : Trouver un personnage par ID et le supprimer de la liste.

Renvoyer le personnage supprimé en JSON.

ID inexistant

DELETE ▼ http://localhost:5001/api/personnages/:id

Params ● Authorization Headers (9) **Body** ● Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

Path Variables

Key	Value	Description
id	11	Description

Body Cookies Headers (7) Test Results 404 Not Found

Pretty Raw Preview Visualize **HTML** ▼

```
1 Utilisateur a supprimé non trouvé
```

ID existant

DELETE

▼

http://localhost:5001/api/personnages/id

Params

Authorization

Headers (9)

Body

Scripts

Settings

Query Params

	Key	Value	Description
	Key	Value	Description

Path Variables

	Key	Value	Description
	id	10	Description

Body

Cookies

Headers (7)

Test Results

200 OK

Pretty

Raw

Preview

Visualize

JSON

▼

↺

1 "ID : 10 a été supprimé"