

# Procédure de Test

Pour effectuer les tests, j'ai choisi l'outil POSTMAN.

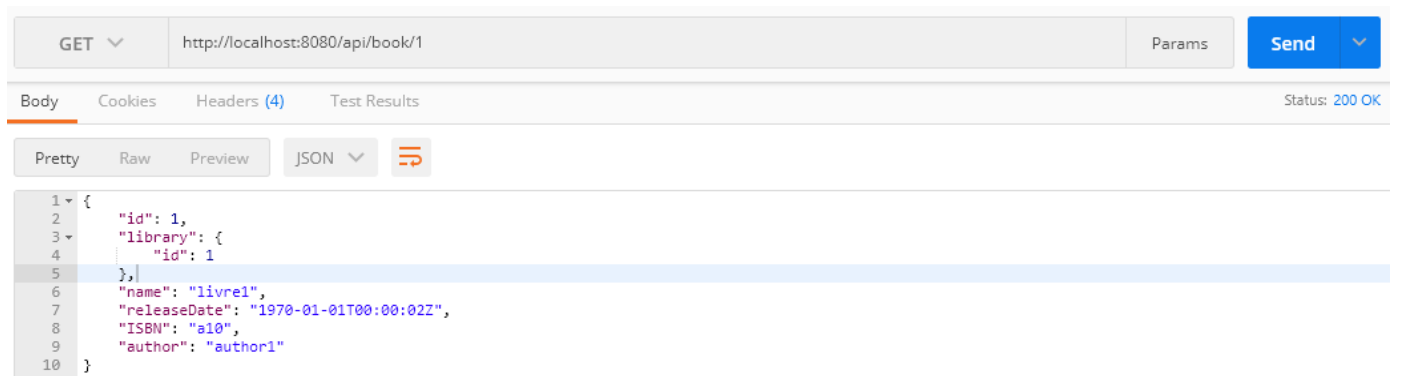
L'API réalisée comporte 3 méthodes :

## 1. La méthode book()

- **GET books :**

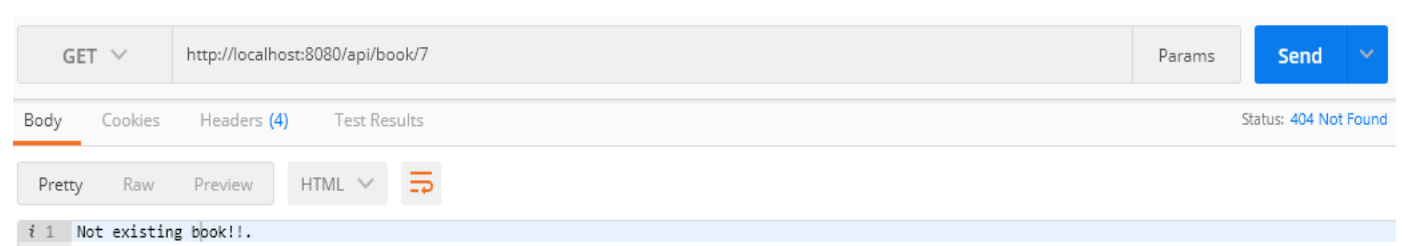


- **GET book :**



- **Cas d'erreurs :**

Si le livre de l'id choisi n'existe pas :



- **POST book :**

POST ▼ http://localhost:8080/api/books Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "library": {
3     "id": 1
4   },
5   "name": "livre4",
6   "releaseDate": "02/02/2000",
7   "ISBN": "a14",
8   "author": "author4"
9 }
```

Body Cookies Headers (4) Test Results Status: 201 Created

Pretty Raw Preview HTML ▼

1 1 Successfully saved.

- **Cas d'erreurs :**

Si l'id de la bibliothèque choisie n'existe pas (comme id = 8) :

POST ▼ http://localhost:8080/api/books Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "library": {
3     "id": 8
4   },
5   "name": "livre4",
6   "releaseDate": "02/02/2000",
7   "ISBN": "a14",
8   "author": "author4"
9 }
```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

1 1 Not Existing library.

Si on ne spécifie pas l'ID de la bibliothèque :

POST ▼ http://localhost:8080/api/books Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "livre4",
3   "releaseDate": "02/02/2000",
4   "ISBN": "a14",
5   "author": "author4"
6 }
```

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview HTML ▼

1 1 You must include the ID of the library in the JSON body

- **PUT book :**

PUT ▼ http://localhost:8080/api/book/4 Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "id": 4,
3   "library": {
4     "id": 2
5   },
6   "name": "livre",
7   "releaseDate": "04/05/1999",
8   "ISBN": "aaa14",
9   "author": "authorName"
10 }
```

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON ▼

```
1 {
2   "id": 4,
3   "library": {
4     "id": 2
5   },
6   "name": "livre",
7   "releaseDate": "1999-04-04T22:00:00Z",
8   "ISBN": "aaa14",
9   "author": "authorName"
10 }
```

- **Cas d'erreurs :**

Si le livre avec l'id choisi n'existe pas :

PUT ▼ http://localhost:8080/api/book/5 Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "id": 4,
3   "library": {
4     "id": 2
5   },
6   "name": "livre",
7   "releaseDate": "04/05/1999",
8   "ISBN": "aaa14",
9   "author": "authorName"
10 }
```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

```
1 1 Bpok not found
```

Si on ne spécifie pas l'id du livre :

PUT ▼ http://localhost:8080/api/book Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

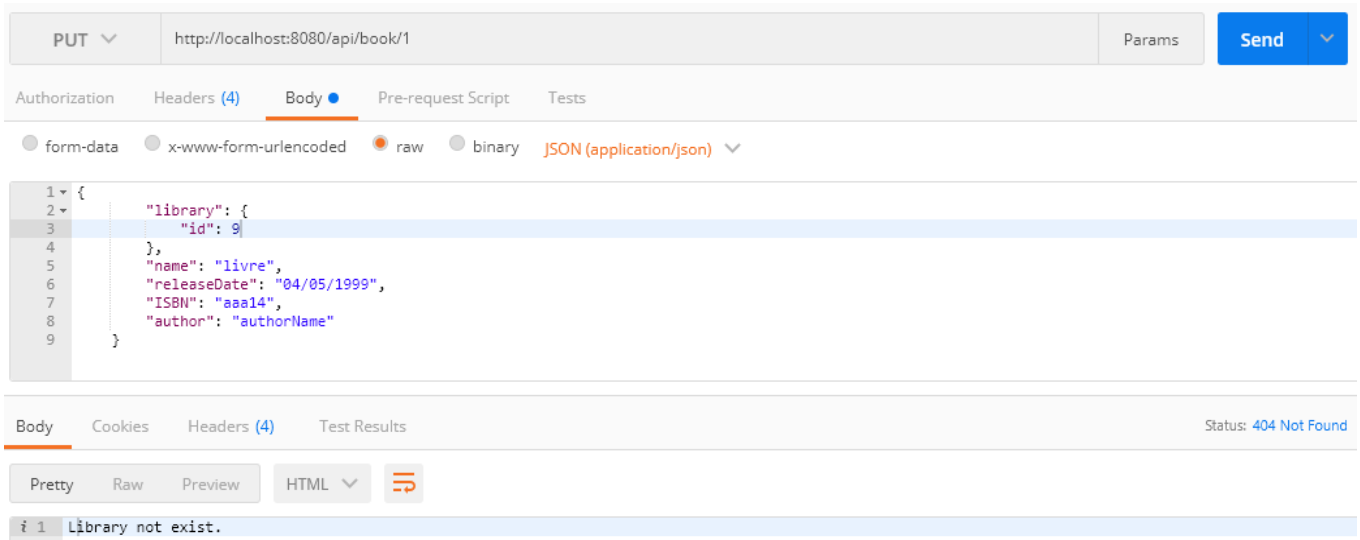
```
1 {
2   "library": {
3     "id": 1
4   },
5   "name": "livre",
6   "releaseDate": "04/05/1999",
7   "ISBN": "aaa14",
8   "author": "authorName"
9 }
```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼

```
1 1 You must include the ID of the book!
```

Si la bibliothèque choisie n'existe pas :




PUT ▼ http://localhost:8080/api/book/1 Params Send ▼

Authorization Headers (4) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

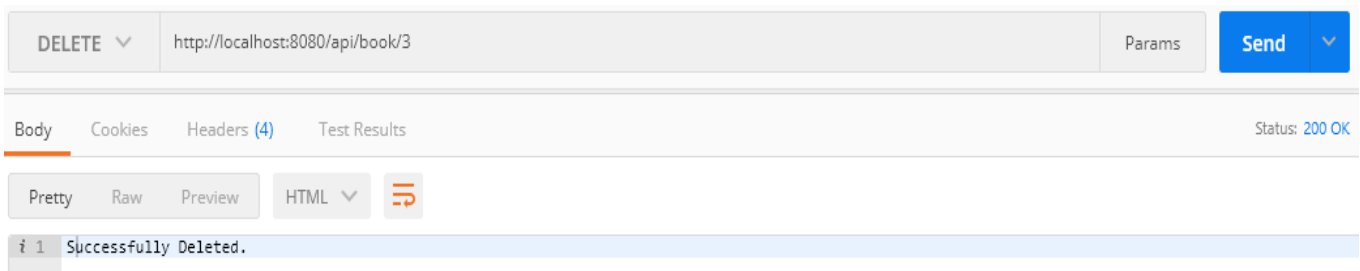
```
1 {  
2   "library": {  
3     "id": 9  
4   },  
5   "name": "livre",  
6   "releaseDate": "04/05/1999",  
7   "ISBN": "aaa14",  
8   "author": "authorName"  
9 }
```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼ 


i 1 Library not exist.

- **DELETE book :**



DELETE ▼ http://localhost:8080/api/book/3 Params Send ▼

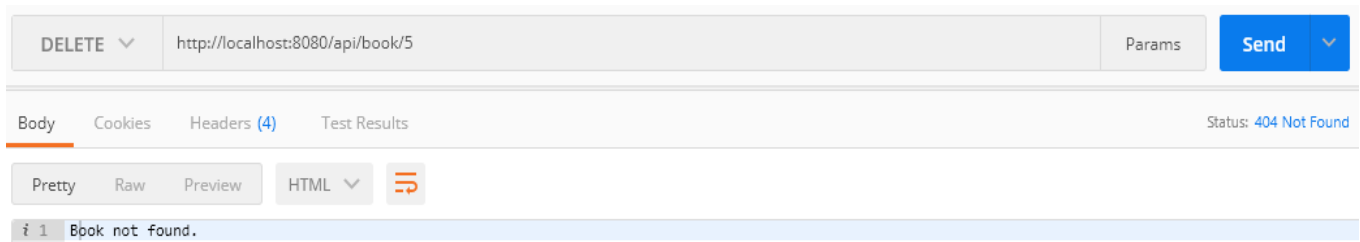
Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview HTML ▼ 

i 1 Successfully Deleted.


- **Cas d'erreurs :**

Si le livre n'existe pas :



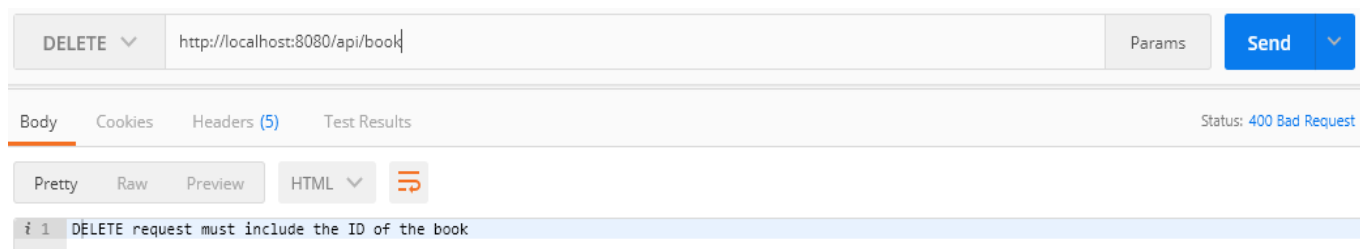
DELETE ▼ http://localhost:8080/api/book/5 Params Send ▼

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼ 


i 1 Book not found.

Si on ne spécifie pas l'id du livre à supprimer :



DELETE ▼ http://localhost:8080/api/book Params Send ▼

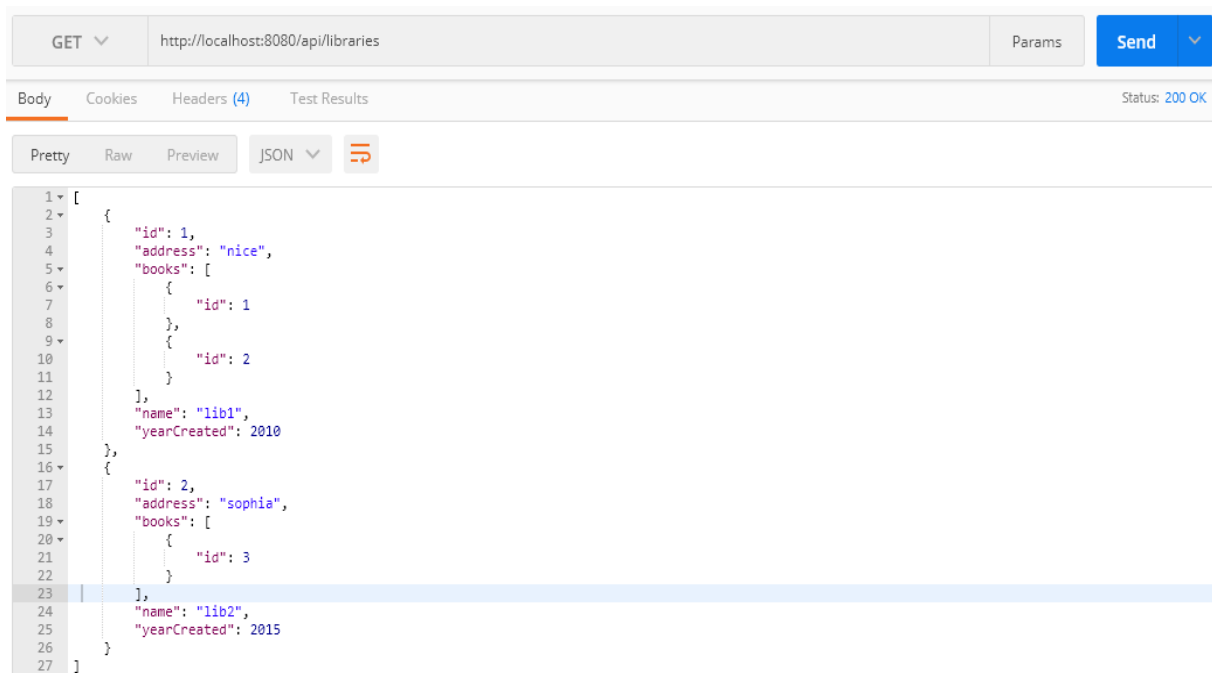
Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼ 

i 1 DELETE request must include the ID of the book


## 2. La méthode library() :

- **GET libraries :**



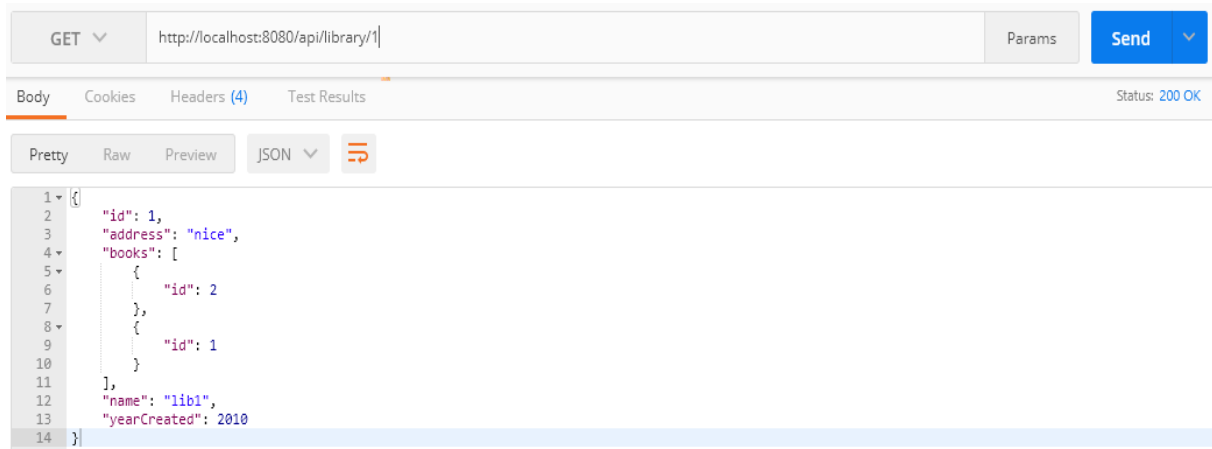
GET  Params

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON 


```
1 [
2   {
3     "id": 1,
4     "address": "nice",
5     "books": [
6       {
7         "id": 1
8       },
9       {
10        "id": 2
11      }
12    ],
13    "name": "lib1",
14    "yearCreated": 2010
15  },
16  {
17    "id": 2,
18    "address": "sophia",
19    "books": [
20      {
21        "id": 3
22      }
23    ],
24    "name": "lib2",
25    "yearCreated": 2015
26  }
27 ]
```

- **GET library :**



GET  Params

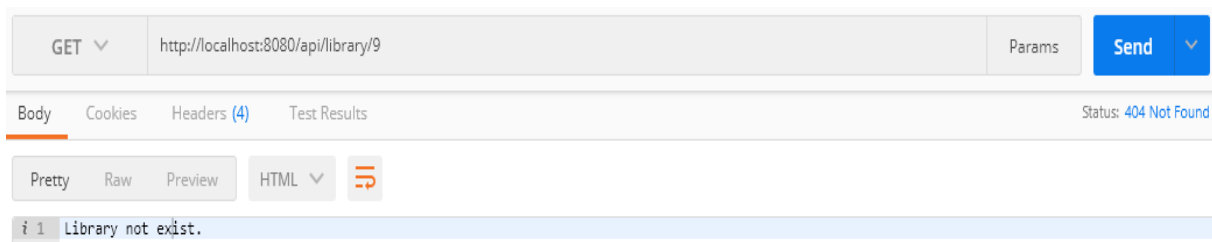
Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON 

```
1 {
2   "id": 1,
3   "address": "nice",
4   "books": [
5     {
6       "id": 2
7     },
8     {
9       "id": 1
10    }
11  ],
12  "name": "lib1",
13  "yearCreated": 2010
14 }
```


- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :



GET  Params

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML 

```
1 Library not exist.
```

- **POST library :**

POST ▼ http://localhost:8080/api/libraries Params Send ▼

Authorization Headers (4) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "libParis",
3   "address": "Paris",
4   "yearCreated": 2015
5 }
```

Body Cookies Headers (4) Test Results Status: 201 Created

Pretty Raw Preview HTML ▼ ≡

**1** Library saved

- **Cas d'erreurs :**

Si on ne fournit pas l'un des attributs de la bibliothèque :

POST ▼ http://localhost:8080/api/libraries Params Send ▼

Authorization Headers (4) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "address": "Paris",
3   "yearCreated": 2015
4 }
```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼ ≡

**1** Library not saved (Bad Request)

- **PUT library :**

PUT ▼ http://localhost:8080/api/library/3 Params Send ▼

Authorization Headers (4) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "libraryParis",
3   "address": "Paris",
4   "yearCreated": 2015
5 }
```

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON ▼ ≡

```
1 {
2   "id": 3,
3   "address": "Paris",
4   "books": [],
5   "name": "libraryParis",
6   "yearCreated": 2015
7 }
```

- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :

PUT ▼ http://localhost:8080/api/library/7 Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "libraryParis",
3   "address": "Paris",
4   "yearCreated": 2015
5 }
```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

```
1 This library is not existing
```

Si on ne spécifie pas l'id de la bibliothèque à l'URL :

PUT ▼ http://localhost:8080/api/library Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "libraryParis",
3   "address": "Paris",
4   "yearCreated": 2015
5 }
```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼

```
1 PUT request must include the ID of the library
```

- **DELETE library :**

DELETE ▼ http://localhost:8080/api/library/3 Params Send ▼

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview HTML ▼

```
1 Successfully Deleted.
```

- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :

DELETE ▼ http://localhost:8080/api/library/3 Params Send ▼

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

```
1 Library not found.
```

Si on ne spécifie pas l'id de la bibliothèque à l'URL :

DELETE  Params

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML

```
1 DELETE request must include the ID of the library
```

### 3. La méthode libraryBooks :

- GET library 's books :

GET  Params

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 [
2   {
3     "id": 2,
4     "library": {
5       "id": 1
6     },
7     "name": "livre2",
8     "releaseDate": "1970-01-01T00:00:02Z",
9     "ISBN": "a11",
10    "author": "author2"
11  },
12  {
13    "id": 1,
14    "library": {
15      "id": 1
16    },
17    "name": "livre1",
18    "releaseDate": "1970-01-01T00:00:02Z",
19    "ISBN": "a10",
20    "author": "author1"
21  }
22 ]
```

- Cas d'erreurs :

Si la bibliothèque n'existe pas :

GET  Params

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML

```
1 Not existing library
```

- GET book from library :

GET  Params

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 {
2   "id": 1,
3   "library": {
4     "id": 1
5   },
6   "name": "livre1",
7   "releaseDate": "1970-01-01T00:00:02Z",
8   "ISBN": "a10",
9   "author": "author1"
10 }
```



- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :

GET ▼ http://localhost:8080/api/library/11/book/1 Params Send ▼

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

```
1 Not existing library
```

Si le livre n'existe pas :

GET ▼ http://localhost:8080/api/library/1/book/10 Params Send ▼

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

```
1 This book is not existing!!
```

Si le livre n'appartient pas à la bibliothèque choisie :

GET ▼ http://localhost:8080/api/library/2/book/1 Params Send ▼

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

```
1 This book is not existing in the library!!
```

- **POST book on library :**

POST ▼ http://localhost:8080/api/library/1/books Params Send ▼

Authorization Headers (4) Body ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {
2   "name": "livreName",
3   "releaseDate": "02/01/1999",
4   "ISBN": "aaa12",
5   "author": "author"
6 }
```

Body Cookies Headers (4) Test Results Status: 201 Created

Pretty Raw Preview HTML ▼

```
1 Successfully saved.
```

- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :

POST ▼ http://localhost:8080/api/library/12/books Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name": "livreName",  
3   "releaseDate": "02/01/1999",  
4   "ISBN": "aaa12",  
5   "author": "author"  
6 }
```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼ ≡

```
1 can't attach a book to a not existent library
```

Si on ne fournit pas un des attributs obligatoires :

POST ▼ http://localhost:8080/api/library/2/books Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "releaseDate": "02/01/1999",  
3   "ISBN": "aaa12",  
4   "author": "author"  
5 }  
6 }
```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼ ≡

```
1 Book not saved.
```

- **PUT library's book :**

PUT ▼ http://localhost:8080/api/library/1/book/1 Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name": "livre01",  
3   "releaseDate": "02/01/1999",  
4   "ISBN": "cc12",  
5   "author": "authorN"  
6 }
```

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview HTML ▼ ≡

```
1 Successfully updated.
```

- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :

PUT ▼ http://localhost:8080/api/library/11/book/1 Params Send ▼

Authorization Headers (4) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name": "livre01",  
3   "releaseDate": "02/01/1999",  
4   "ISBN": "cc12",  
5   "author": "authorN"  
6 }
```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

**1** Not existing library

Si le livre n'existe pas :

PUT ▼ http://localhost:8080/api/library/1/book/10 Params Send ▼

Authorization Headers (4) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name": "livre01",  
3   "releaseDate": "02/01/1999",  
4   "ISBN": "cc12",  
5   "author": "authorN"  
6 }
```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

**1** Not existing Book

Si on ne spécifie pas l'id du livre :

PUT ▼ http://localhost:8080/api/library/1/books Params Send ▼

Authorization Headers (4) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "name": "livre01",  
3   "releaseDate": "02/01/1999",  
4   "ISBN": "cc12",  
5   "author": "authorN"  
6 }
```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼

**1** Update request must include the ID of the book

Si on ne spécifie pas l'id de la bibliothèque :

PUT ▼ http://localhost:8080/api/library Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```

1 {
2   "name": "livre01",
3   "releaseDate": "02/01/1999",
4   "ISBN": "cc12",
5   "author": "authorN"
6 }

```

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML ▼

**1** PUT request must include the ID of the library

Si on utilise un ID d'un livre qui n'appartient pas à la bibliothèque choisie :

PUT ▼ http://localhost:8080/api/library/2/book/1 Params Send ▼

Authorization Headers (4) **Body** • Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```

1 {
2   "name": "livre01",
3   "releaseDate": "02/01/1999",
4   "ISBN": "cc12",
5   "author": "authorN"
6 }

```

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

**1** This book is not existing in the library!!

- **DELETE book from library**

DELETE ▼ http://localhost:8080/api/library/1/book/4 Params Send ▼

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview HTML ▼

**1** Successfully Deleted.

- **Cas d'erreurs :**

Si la bibliothèque n'existe pas :

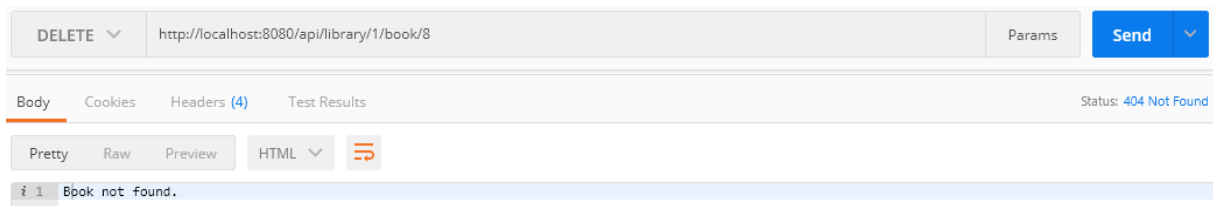
DELETE ▼ http://localhost:8080/api/library/10/book/4 Params Send ▼

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML ▼

**1** Library not found.

Si le livre n'existe pas :



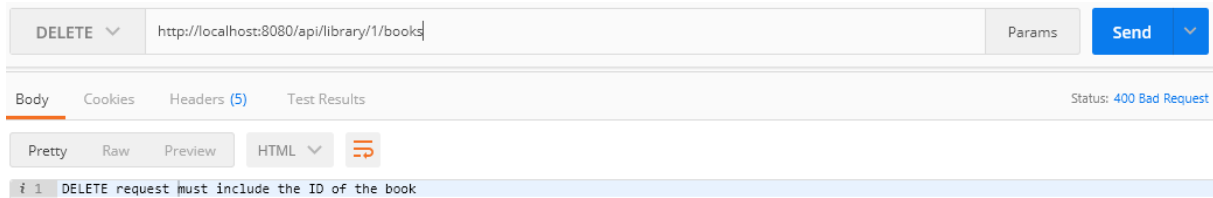
DELETE  Params

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML

**1** Bpok not found.

Si on ne spécifie pas l'id du livre :



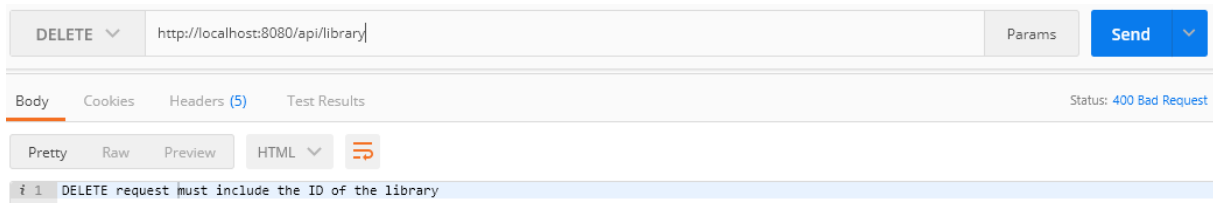
DELETE  Params

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML

**1** DELETE request must include the ID of the book

Si on ne spécifie pas l'id de la bibliothèque :



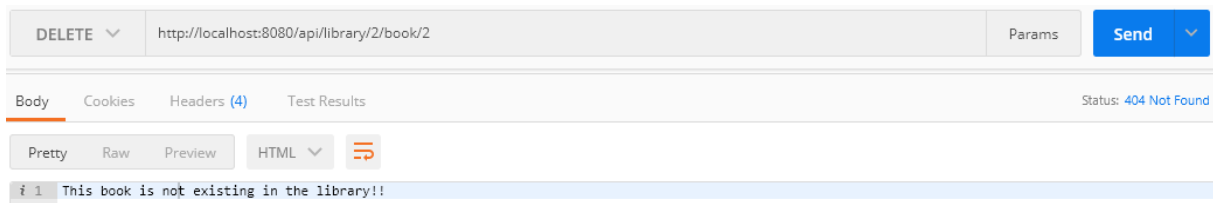
DELETE  Params

Body Cookies Headers (5) Test Results Status: 400 Bad Request

Pretty Raw Preview HTML

**1** DELETE request must include the ID of the library

Si on utilise un ID d'un livre qui n'appartient pas à la bibliothèque choisie :



DELETE  Params

Body Cookies Headers (4) Test Results Status: 404 Not Found

Pretty Raw Preview HTML

**1** This book is not existing in the library!!