

La connexion à l'aide du jeton généré par JWT

The screenshot displays the Postman interface with a REST client request configured for a POST method to the endpoint `http://localhost:8083/authenticate`. The request body is a JSON object containing the username `houssine` and password `1234`. The response status is `200 OK`, and the response body contains a JWT token.

Request Details:

- Method: POST
- URL: `http://localhost:8083/authenticate`
- Body (JSON):

```
{  "username": "houssine",  "password": "1234"}
```

Response Details:

- Status: 200 OK
- Time: 218 ms
- Size: 539 B
- Body (JSON):

```
{  "jwtToken": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJob3Vzc2luZSIsImh0dCI6MTY4MzQ4NzUzMywiZXhwIjojNjgzNTA1NTMzZjQ4IjE7RmV6pg2qU2JGxjSNYbCQ423mwBZ9mYq8pg2aZhbxrZgIAryJ8zC9YzqHVNxVHdG5Bf0DhYPzjg_tz4w"}
```

The interface also shows a sidebar with collections and environments, and a bottom status bar with system icons and the date `07/05/2023`.

Home page

The screenshot displays the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar labeled 'Search Postman' is also present. Below this, the 'Team Workspace' section shows a list of collections, including 'api rest spring boot'. The main panel shows a selected request with the URL 'http://localhost:8083/'. The request is a GET method. The 'Headers' tab is active, showing a table with headers. The 'Body' tab is also visible, showing a JSON response.

Navigation: Home Workspaces API Network Explore Search Postman

Team Workspace New Import Overview GET http://localhost:8083/ GET http://localhost:8083/ GET http://localhost:8083/ GET http://localhost:8083/ POST http://localhost:8083/ GET http://localhost:8083/ No Environment

Collections + api rest spring boot

- GET http://localhost:8082/produit/r...
- POST http://localhost:8082/produit/c...
- DEL http://localhost:8082/produit/d...
- PUT http://localhost:8082/produit/u...
- GET http://localhost:8083/students...
- GET http://localhost:8083/students...
- GET http://localhost:8083/students...
- GET http://localhost:8083/students...
- POST http://localhost:8083/authenti...

http://localhost:8083/ Save

GET http://localhost:8083/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Headers 7 hidden

Key	Value	Description
Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJob3...	
Key	Value	Description

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 39 ms Size: 394 B Save as Example

Pretty Raw Preview Visualize Text

```
1 Welcome to application School management (Anywr)
```

Online Find and Replace Console Cookies Capture requests Runner Trash

Taper ici pour rechercher 21°C 20:30 07/05/2023

Filtres : nom de la classe (avec la pagination)

The screenshot shows the Postman interface with a REST client request configured. The request is a GET call to `http://localhost:8083/students/getStudentsByClassRoomName/class_2`. The response is a JSON array of student objects. The first object has `id: 2`, `firstName: "st2"`, `lastName: "name_2"`, and a `classRoom` object with `id: 2`, `name: "class_2"`, and a `teacher` object with `id: 2`, `firstName: "Ali"`, and `lastName: "feya9"`. The second object has `id: 5` and `firstName: "absent"`.

Request Details:

- Method: GET
- URL: `http://localhost:8083/students/getStudentsByClassRoomName/class_2`
- Headers (8): Authorization (Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWiiOiJob3...

Response Details:

- Status: 200 OK
- Time: 52 ms
- Size: 990 B
- Save as Example

Response Body (JSON):

```
4  {
5    "id": 2,
6    "firstName": "st2",
7    "lastName": "name_2",
8    "classRoom": {
9      "id": 2,
10     "name": "class_2",
11     "teacher": {
12       "id": 2,
13       "firstName": "Ali",
14       "lastName": "feya9",
15       "fullName": "Ali feya9"
16     }
17   },
18   {
19     "id": 5,
20     "firstName": "absent",
```

Filtres : nom complet de l'enseignant (avec la pagination)

The screenshot displays the Postman interface with a REST client request configured. The request is a GET call to the endpoint `http://localhost:8083/students/getByTeacherByFullName/Ali feya9`. The Authorization header is set to `Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJob3...`. The response is a JSON array of student objects, filtered by the teacher's full name 'Ali feya9'.

Request Details:

- Method: GET
- URL: `http://localhost:8083/students/getByTeacherByFullName/Ali feya9`
- Headers (8):
 - Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJob3...

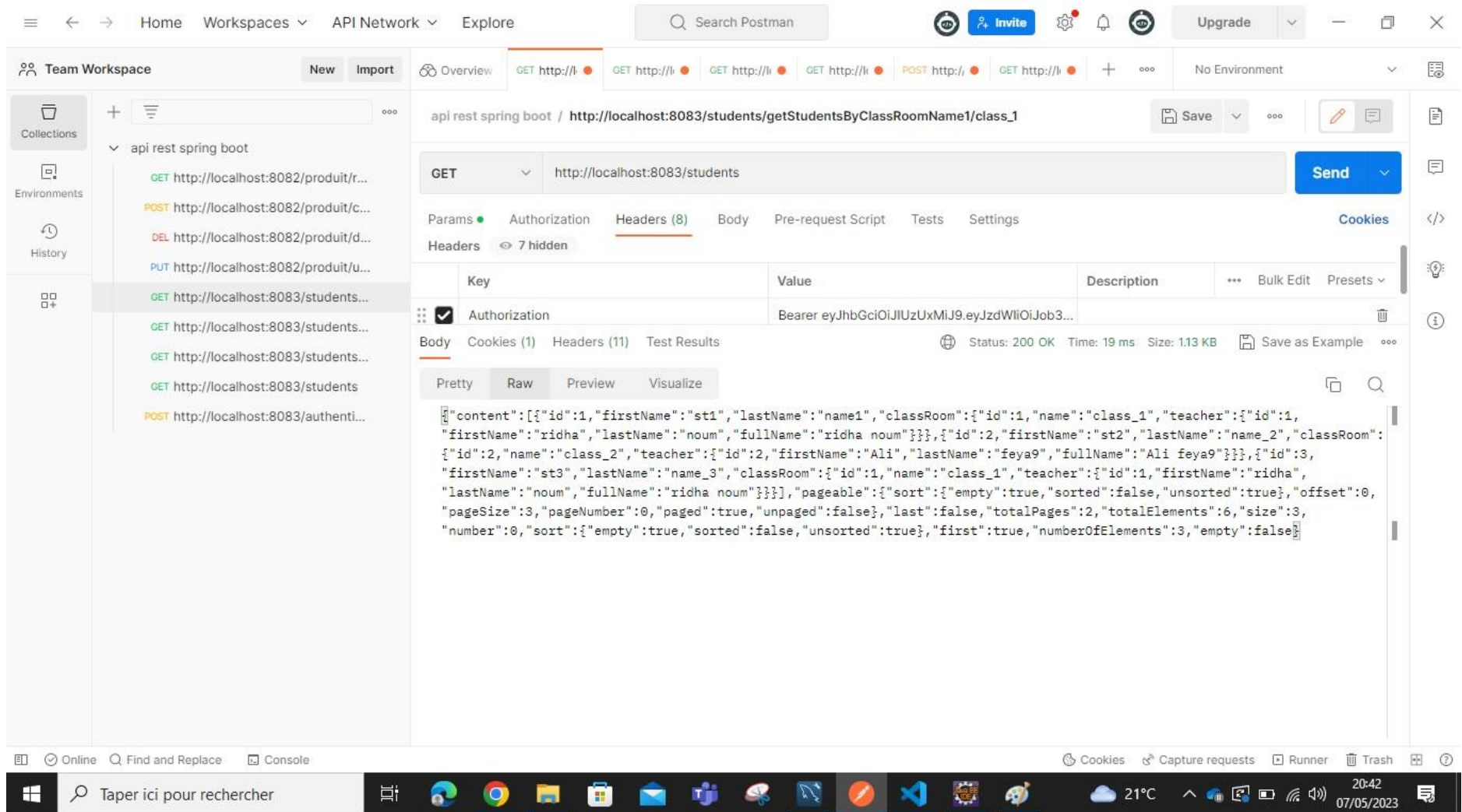
Response Details:

- Status: 200 OK
- Time: 27 ms
- Size: 990 B
- Save as Example

Response Body (JSON):

```
[{"id": 2, "firstName": "st2", "lastName": "name_2", "classRoom": {"id": 2, "name": "class_2", "teacher": {"id": 2, "firstName": "Ali", "lastName": "feya9", "fullName": "Ali feya9"}}, {"id": 5, "firstName": "absent",
```

La liste de tous les étudiants sera renvoyée en cas d'absence de valeur de filtre (avec la pagination)



The screenshot displays the Postman interface with a REST client request configured. The request is a GET request to the endpoint `http://localhost:8083/students`. The response is a JSON object containing a list of students, with a status of 200 OK and a response time of 19 ms.

Request Details:

- Method: GET
- URL: `http://localhost:8083/students`
- Headers (8):
 - Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJob3...

Response Details:

- Status: 200 OK
- Time: 19 ms
- Size: 1.13 KB

Response Body (JSON):

```
{
  "content": [
    {
      "id": 1,
      "firstName": "st1",
      "lastName": "name1",
      "classRoom": {
        "id": 1,
        "name": "class_1",
        "teacher": {
          "id": 1,
          "firstName": "ridha",
          "lastName": "noum",
          "fullName": "ridha noum"
        }
      }
    },
    {
      "id": 2,
      "firstName": "st2",
      "lastName": "name_2",
      "classRoom": {
        "id": 2,
        "name": "class_2",
        "teacher": {
          "id": 2,
          "firstName": "Ali",
          "lastName": "feya9",
          "fullName": "Ali feya9"
        }
      }
    },
    {
      "id": 3,
      "firstName": "st3",
      "lastName": "name_3",
      "classRoom": {
        "id": 1,
        "name": "class_1",
        "teacher": {
          "id": 1,
          "firstName": "ridha",
          "lastName": "noum",
          "fullName": "ridha noum"
        }
      }
    }
  ],
  "pageable": {
    "sort": {
      "empty": true,
      "sorted": false,
      "unsorted": true
    },
    "offset": 0,
    "pageSize": 3,
    "pageNumber": 0,
    "paged": true,
    "unpaged": false,
    "last": false,
    "totalPages": 2,
    "totalElements": 6,
    "size": 3,
    "number": 0,
    "sort": {
      "empty": true,
      "sorted": false,
      "unsorted": true
    },
    "first": true,
    "numberOfElements": 3,
    "empty": false
  }
}
```


La liste de tous les étudiants sera renvoyée en cas d'absence de valeur de filtre (pagination avec PathVariable)

The screenshot displays the Postman application interface. On the left, the 'Team Workspace' sidebar shows a collection named 'api rest spring boot' with several API requests. The main panel shows a selected GET request to 'http://localhost:8083/students'. The 'Headers' tab is active, showing an 'Authorization' header with a Bearer token. The 'Body' tab shows the response in JSON format, which is a list of two student objects. The status bar at the bottom indicates a successful 200 OK response with a time of 179 ms and a size of 989 B.

Request Details:

- Method: GET
- URL: http://localhost:8083/students/0/2
- Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWl0OiJob3...

Response Body (JSON):

```
4  {
5    "id": 1,
6    "firstName": "st1",
7    "lastName": "name1",
8    "classRoom": {
9      "id": 1,
10     "name": "class_1",
11     "teacher": {
12       "id": 1,
13       "firstName": "ridha",
14       "lastName": "noum",
15       "fullName": "ridha noum"
16     }
17   },
18   },
19   {
20     "id": 2,
21     "firstName": "st2",
22     "lastName": "name_2",
```

Status Bar: Status: 200 OK, Time: 179 ms, Size: 989 B, Save as Example