Campus rooms API

José Pablo Alvarado

USE CASES

- Log in
  - Student logs in
  - Admin logs in

  * Password must be encrypted
  * Students must have limited privileges to execute request

- Search rooms
  - search single room
  - search all rooms
  - search by available rooms (rooms with available beds)
  - search students by room (Admin only)

  * searching all rooms must allow to be paginated

- Add room (Admin only)
- Remove room (Admin only)
- Add student to a room

  * Students should only be associated to 1 room at a time
- Remove student from room (Admin only)

  * Student must contact admin to change/leave room

- Search students (Admin only)
  - search by student's name
  - search by student's code
  - search all students

  * searching all students must allow to be paginated
  * search should allow order students by name

- Add new student (Admin only)
- Remove student (Admin only)

ENTITIES

ROOMS

- id SMALLINT(2)
- available_beds SMALLINT(1)
- total_beds SMALLINT(1)

STUDENTS

- id SMALLINT(3)
- code VARCHAR(10)
- name VARCHAR(50)
- password VARCHAR(40)

ADMINS

- id SMALLINT(1)
- code VARCHAR(10)
- name VARCHAR(50)
- password VARCHAR(40)

ROOMS_STUDENTS

- student_id SMALLINT(3)
- room_id SMALLINT(2)

API USE CASES INFORMATION

END_POINT = https://Escuela.edu:443/studentRoomsService

Logging in

REQUEST HEADERS

Http method : POST
End point : **END_POINT**
Protocol : Https
URI : **END_POINT**/login
Content-type : application/json

REQUEST BODY example

```
{
  "id" : 2132458,
  "password" : "ahkajshdSS54$JFH°#eiurkjsah;.&&7is9-aois*akAJAKA"   (encrypted)
}
```

RESPONSE HEADERS

Http status code : 200 = Login successful
400 = Id or password invalid
503 = Service unavailable
Content type : text/plain; charset=UTF-8
Cookie : ACCESS_TOKEN=<Authorization token>;
EXPIRES=<date and time>
Allow : (Students) GET, PUT
(Admins) GET, POST, PUT, UPDATE, DELETE, PATCH

RESPONSE BODY

(Login successful / Id or password invalid)

Search rooms

REQUEST HEADERS

Http method : GET
End point : **END_POINT**
Authorization : < Authorization token>
Protocol : Https
URI : **END_POINT**/rooms

URI : **END_POINT**/rooms/{id}
URI : **END_POINT**/rooms/{id}?show_students=<show_students>
URI : **END_POINT**/rooms?limit=<limit>&startingElement=<starting_element_number>
URI : **END_POINT**/rooms?has_available_beds=<has_beds>
URI : **END_POINT**/rooms?show_students=<show_students>

Query parameters : <limit> = 5, 10, 15, 20 (pagination)
default : 10
optional
<starting_element_number> = index of room (pagination)
default : 1
optional
<has_beds> = true, false, null
default : null
optional
<show_students> = true, false
default : false
optional

REQUEST BODY

(---)

RESPONSE HEADERS

Content type : application/json
Cache-control : no-cache
Http status code : 200 = success
400 = Invalid arguments
401 = User not logged in
403 = User can’t execute request
404 = Resource not found
429 = User must wait before make request again
503 = Service unavailable

RESPONSE BODY example, search all rooms

```
{
 "rooms" :
 [
  {
   "id" : 7,
   "total_beds" : 3,
   "available_beds" : 1
  },
  {
   "id" : 8,
   "total_beds" : 1,
   "available_beds" : 0
  },
  {
   "id" : 9,
   "total_beds" : 2,
   "available_beds" : 2
  }
 ]
}
```

RESPONSE BODY example, search single room

```
{
 "room" :
   {
     "id" : 9,
     "total_beds" : 2,
     "available_beds" : 2
   }
}
```

RESPONSE BODY example, search single room showing students

```
{
 "room" :
   {
     "id" : 8,
     "total_beds" : 1,
     "available_beds" : 0
     "students" :
        [
          {
            "id" : 7,
            "code" : 221477,
            "name" : "Jessica Jauregui"
          }
        ]
   }
}
```

Add room / Remove room / Remove student from room (Admin)

REQUEST HEADERS

HTTP method : POST, PUT, DELETE
End point : **END_POINT**
Authorization : < Authorization token>
Protocol : Https

URI : (POST) **END_POINT**/rooms/{room_id}?total_beds=<beds_number>
URI : (DELETE) **END_POINT**/rooms/{room_id}
URI : (PUT) **END_POINT**/rooms/{room_id}/{student_id} (remove student from room)

Query parameters : <beds_number> = number of beds a room will have
required

REQUEST BODY

(---)

RESPONSE HEADERS

Http status code : 200 = Success
201 = Room added
400 = Invalid arguments
404 = resource not found
503 = Service unavailable

RESPONSE BODY

(---)

Search students (Admin)

REQUEST HEADERS

HTTP method : GET
End point : END_POINT
Authorization : < Authorization token>
Protocol : Https

URI : END_POINT/students
URI : END_POINT/students?limit=<limit>&startingElement=<student_id>
URI : END_POINT/students?order=<order>
URI : END_POINT/students?code=<student_code>
URI : END_POINT/students?name=<student_name>
(search by name ignores case, matches any that contains argument in name)

Query parameters : <limit> = 5, 10, 15, 20
default: 10
optional
<student_id> = student id, integer
optional
<student_code> = student code, integer
optional
<student_name> = string to match
optional
<order> = ASC, DESC (sorts alphabetically)
default: ASC
optional

REQUEST BODY

(---)

RESPONSE HEADERS

Content type : application/json
Cache-control : max-age=3600
Http status code : 200 = Success
400 = Invalid arguments
401 = User not logged in
403 = User not allowed to execute request
404 = Resource not found
429 = User must wait before make request again
503 = Service unavailable

RESPONSE BODY example

```
{
 "students" :
 [
  {
   "id" : 7,
   "code" : 158874,
   "name" : "Ramiro Ramirez"
  },
  {
   "id" : 35,
      "code" : 6487,
      "name" : "Raul Ramirez"
    }
    ~ ~ ~ ~ ~ ~ ~ ~
    ~ ~ ~ ~ ~ ~ ~ ~
  ]
}
```

RESPONSE BODY example, single student

```
{
  "student" :
    {
      "id" : 22,
      "code" : 154237,
      "name" : "Jimena Jimenez"
    }
}
```

Add student to a room

REQUEST HEADERS

HTTP method : PUT
End point : **END_POINT**
Authorization : <Authorization token>
Protocol : Https
URI : **END_POINT**/rooms/{room_id}/add_student/{student_id}
(associates a student to room)

REQUEST BODY

(---)

RESPONSE HEADERS

Http status code : 200 = success
400 = Invalid arguments
401 = User not logged in
403 = Student already has a room
429 = User must wait before make request
404 = Resource not found
503 = Service unavailable

RESPONSE BODY example

(---)

Add new student / Remove student (Admin)

REQUEST HEADERS

Http method : POST, DELETE
End point : **END_POINT**
Authorization : < Authorization token>
Protocol : Https

URI : (POST) **END_POINT**/students?code=<code>
URI : (DELETE) **END_POINT**/students?code=<code>

Query parameters : <code> = student code, integer
required

REQUEST BODY

(---)

RESPONSE HEADERS

Http status code : 200 = Success
400 = Invalid arguments
404 = Student code not found
503 = Service unavailable

RESPONSE BODY

(---)