Contents

[**1.** **Class Student** 2](#_Toc184204763)

[**2.** **Class Printer Configuration** 3](#_Toc184204764)

[**3.** **Class Printer** 5](#_Toc184204765)

[**4.** **Class Login** 6](#_Toc184204766)

[**5.** **Class SPSO** 6](#_Toc184204767)

[**6.** **Class Printing System (thấy không cần thiết lắm)** 7](#_Toc184204768)

[**7.** **Class Print Modification** 8](#_Toc184204769)

[**8.** **Class Buy Log** 8](#_Toc184204770)

[**Body:** 8](#_Toc184204771)

[{ 8](#_Toc184204772)

[“id”: buy log id (long) 8](#_Toc184204773)

[“paperSize”: cỡ giấy (String) 8](#_Toc184204774)

[“boughtPageNum”: số trang (int), 9](#_Toc184204775)

[“price”: tổng giá, 9](#_Toc184204776)

[“paymentTime”: thời gian thanh toán, 9](#_Toc184204777)

[“student”: { 9](#_Toc184204778)

[} 9](#_Toc184204779)

[} 9](#_Toc184204780)

**List API**

1. **Class Student**

**Body:**

{

“id”: MSSV (long),

“name”: Tên sinh viên (String),

“balance”: Số trang giấy còn lại (int),

“phoneNum”: số điện thoại (String),

“doB”: ngày sinh (Date),

“email”: email hcmut.edu.vn (String),

“login”: (object của class login)

{

“id”: tạo tự động (long),

“username”: tên đăng nhập (String),

“password”:mật khẩu (String)

}

}

* 1. ***GET: lấy danh sách tất cả sinh viên có trong database và thông tin của họ.***

API: <http://localhost:8080/api/v1/students>

* 1. ***GET: lấy thông tin của sinh viên ứng với mã số sinh viên***

API: [http://localhost:8080/api/v1/student/{id}](http://localhost:8080/api/v1/student/%7bid%7d)

* 1. ***GET: lấy thông tin đăng nhập của sinh viên***

*API:* <http://localhost:8080/api/v1/student>/{id}/login

* 1. ***POST: tạo một đối tượng sinh viên mới***

API: <http://localhost:8080/api/v1/student>

yêu cầu nhập vào thông tin cho một đối tượng sinh viên mới

Nếu nhập vào sinh viên đã có MSSV trong database rồi thì nó sẽ ném lỗi. Nên muốn chỉnh sửa thông tin sinh viên thì dùng Put Resquest.

* 1. ***PUT: cập nhật thông tin sinh viên bằng id sinh viên***

***(Thông tin login của sinh viên có thể được cập nhật bằng API này, hoặc bằng username của tài khoản sinh viên)***

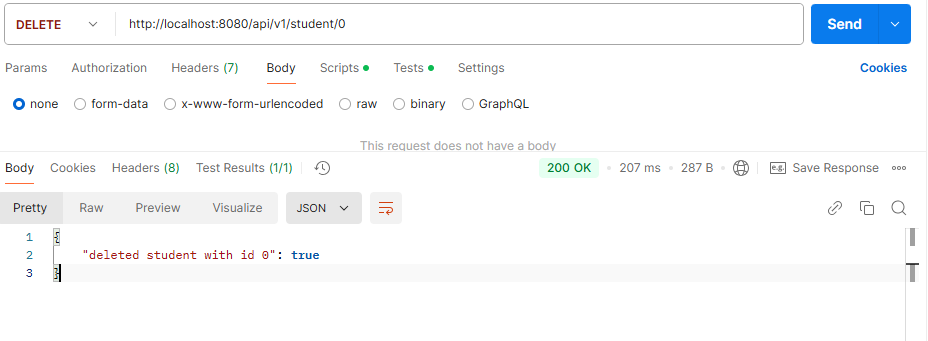
API: [http://localhost:8080/api/v1/student/{id}](http://localhost:8080/api/v1/student/%7bid%7d)

* 1. DELETE: xóa tất cả sinh viên

API: <http://localhost:8080/api/v1/students>

* 1. DELETE: xóa sinh viên có MSSV id

API: [http://localhost:8080/api/v1/student/{id}](http://localhost:8080/api/v1/student/%7bid%7d)



1. **Class Printer Configuration**

**Body:**

{

“id”: tạo tự động (long),

“model”: mẫu máy in (String).

“brand”: thương hiệu máy in (String),

“technology”: công nghệ in (laser, phun) (String)

“duplexPrinting”: in hai mặt thủ công hay tự động (automatic or manual) (String),

“printer”:{

“name”: tên máy in (key)(String),

“location”: vị trí (String),

“state”: trạng thái máy in (on/off) (String)

}

}

**Class Printer Configuration có lưu một đối tượng Printer, nhưng đối tượng Printer thì không.**

* 1. **GET: lấy danh sách tất cả cấu hình máy in**

**API:** http://localhost:8080/api/v1/configurations

* 1. **GET: lấy cấu hình máy in tương ứng với ID của cấu hình**

**API:** [http://localhost:8080/api/v1/configuration/{id}](http://localhost:8080/api/v1/configuration/%7bid%7d)

**Note: id của cấu hình máy in được tạo tự động**

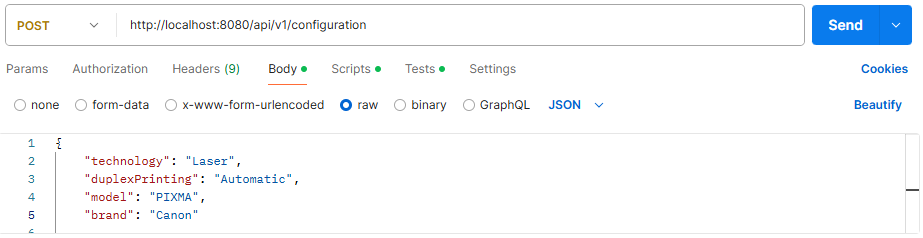
* 1. **GET: lấy cấu hình máy in tương ứng với Printer Name**

**API:** [http://localhost:8080/api/v1/configuration/printer/{printer\_name}](http://localhost:8080/api/v1/configuration/printer/%7bprinter_name%7d)

* 1. **POST: tạo một configuration**

**API:** <http://localhost:8080/api/v1/configuration>

**Example:**

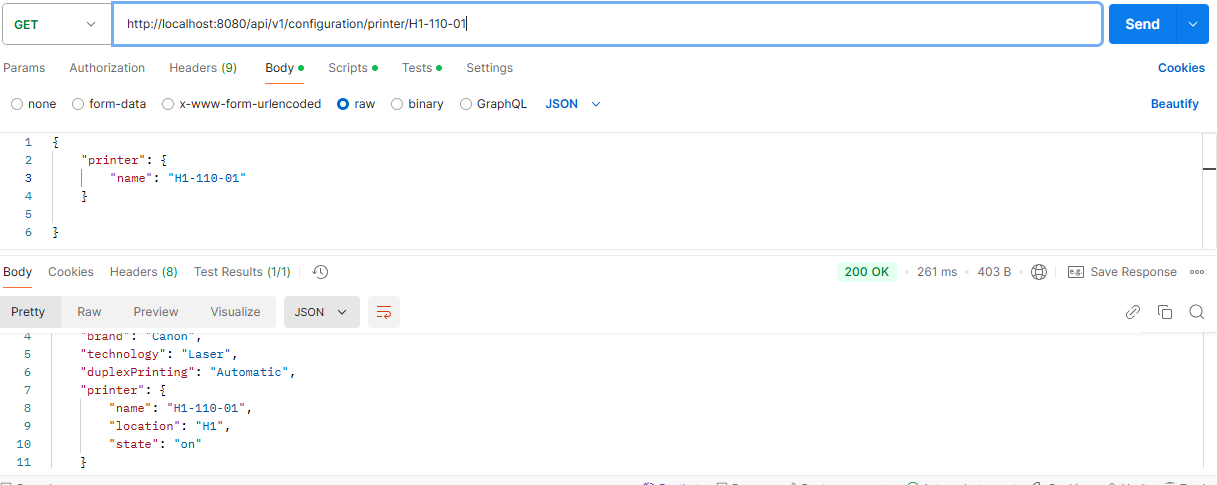


**2.5 PUT: sửa một cấu hình máy in tương ứng với id của nó**

API: [http://localhost:8080/api/v1/configuration/{id}](http://localhost:8080/api/v1/configuration/%7bid%7d)

* 1. PUT: sửa một cấu hình máy in tương ứng với máy in

API: [http://localhost:8080/api/v1/configuration/printer/{printer\_name}](http://localhost:8080/api/v1/configuration/printer/%7bprinter_name%7d)



* 1. DELETE: xóa tất cả cấu hình của máy in

API: <http://localhost:8080/api/v1/configurations>

* 1. DELETE: xóa cấu hình máy in tương ứng với id của nó

API: [http://localhost:8080/api/v1/configuration{id}](http://localhost:8080/api/v1/configuration%7bid%7d)

* 1. DELETE: xóa cấu hình máy in tương ứng với tên máy in

**API:** [http://localhost:8080/api/v1/configuration/printer/{printer\_name}](http://localhost:8080/api/v1/configuration/printer/%7bprinter_name%7d)

1. **Class Printer**

{

“name”: tên máy in (key)(String),

“location”: vị trí (String),

“state”: trạng thái máy in (on/off) (String)

}

* 1. GET: lấy danh sách tất cả máy in

API: <http://localhost:8080/api/v1/printers>

* 1. GET: lấy máy in tương ứng với tên máy in (tên máy in là key của máy in)

API: [http://localhost:8080/api/v1/printer/{printer\_name}](http://localhost:8080/api/v1/printer/%7bprinter_name%7d)

* 1. POST: thêm một máy in vào hệ thống

API: <http://localhost:8080/api/v1/printer>

* 1. PUT: Sửa thông số (bao gồm trạng thái) của một máy in dựa vào tên của máy in

API: [http://localhost:8080/api/v1/printer/{printer\_name}](http://localhost:8080/api/v1/printer/%7bprinter_name%7d)

* 1. DELETE: Xóa tất cả máy in

API: <http://localhost:8080/api/v1/printers>

* 1. DELETE: xóa máy in tương ứng với tên máy in

API: [http://localhost:8080/api/v1/printer/{printer\_name}](http://localhost:8080/api/v1/printer/%7bprinter_name%7d)

1. **Class Login**

Body:

{

“id”: id của login được tạo tự động, id ảo, không có giá trị sử dụng (long),

“username”: tên đăng nhập (String),

“password”: mật khẩu (String)

}

* 1. GET: lấy danh sách tất cả thông tin log in (bao gồm cả sinh viên và SPSO)

API: <http://localhost:8080/api/v1/logins>

* 1. GET: lấy thông tin login dựa vào id của login

API: [http://localhost:8080/api/v1/login/{id}](http://localhost:8080/api/v1/login/%7bid%7d)

* 1. POST: tạo một thông tin login (không phân biệt sinh viên hay SPSO)

API: <http://localhost:8080/api/v1/login>

Note: không thể tạo trùng username

* 1. PUT: cập nhật thông tin login dựa vào id của login (không phân biệt sinh viên hay SPSO)

API: [http://localhost:8080/api/v1/login/{id}](http://localhost:8080/api/v1/login/%7bid%7d)

Note: chỉ được cập nhật mật khẩu, không được cập nhật username

* 1. DELETE: xóa thông tin login dựa vào id của login

API: [http://localhost:8080/api/v1/login/{id}](http://localhost:8080/api/v1/login/%7bid%7d)

* 1. DELETE: xóa hết thông tin login

API: <http://localhost:8080/api/v1/logins>

1. **Class SPSO**

Body:

{

“id”: do người dùng nhập (long),

“name”: tên nhân viên (String),

“phoneNum”: số điện thoại (String),

“doB”: ngày sinh nhân viên (Date),

“email” : email nhân viên (String)

“login”: {

“id”: long

“username”: String,

“password”: String

}

}

* 1. GET: Lấy danh sách tất cả SPSO

API: <http://localhost:8080/api/v1/spsos>

5.2 GET: Lấy SPSO dựa vào id

API: : [http://localhost:8080/api/v1/spso/{id}](http://localhost:8080/api/v1/spso/%7bid%7d)

* 1. POST: Thêm 1 SPSO

API: : <http://localhost:8080/api/v1/spso>

Lưu ý: nhập trùng id với spso đã có sẽ báo lỗi, muốn update thì dùng put

5.4. PUT: Chỉnh sửa SPSO dựa vào id

API: : [http://localhost:8080/api/v1/spso/{id}](http://localhost:8080/api/v1/spso/%7bid%7d)

* 1. DELETE: Xóa SPSO dựa vào id

API: : [http://localhost:8080/api/v1/spso/{id}](http://localhost:8080/api/v1/spso/%7bid%7d)

* 1. DELETE: xóa tất cả SPSO:

API: : <http://localhost:8080/api/v1/spsos>

5.7: GET: lấy thông tin login của SPSO tương ứng với id

API: [http://localhost:8080/api/v1/spso/{id}/login](http://localhost:8080/api/v1/spso/%7bid%7d/login)

1. **Class Printing System (thấy không cần thiết lắm)**

Body:

{

“id”:

“printers”: list các máy in,

“printlogs”: list các print log

}

1. **Class Print Modification**

Body:

{

“id”: tự động tạo(long),

“paperSize”: cỡ giấy (String),

“copies”: số bản sao (in),

“doubleSize”: có in hai mặt hay không(boolean)

}

* 1. GET: lấy danh sách tất cả print modification

API: <http://localhost:8080/api/v1/printModifications>

* 1. GET: lấy print modification ứng với id

API: [http://localhost:8080/api/v1/printModification/{id}](http://localhost:8080/api/v1/printModification/%7bid%7d)

* 1. POST: tạo một print modification mới

API: <http://localhost:8080/api/v1/printModification>

* 1. PUT: cập nhật print modification

API: [http://localhost:8080/api/v1/printModification/{id}](http://localhost:8080/api/v1/printModification/%7bid%7d)

* 1. DELETE: xóa tất cả

API: <http://localhost:8080/api/v1/printModifications>

* 1. DELETE: xóa 1 print modification dựa vào id

API: [http://localhost:8080/api/v1/printModification/{id}](http://localhost:8080/api/v1/printModification/%7bid%7d)

1. **Class Buy Log**

**Body:**

{

“id”: buy log id (long)

“paperSize”: cỡ giấy (String)

“boughtPageNum”: số trang (int),

“price”: tổng giá,

“paymentTime”: thời gian thanh toán,

“student”: {

“id”:MSSV,

và các thông tin khác

}

}

* 1. GET: lấy tất cả buy log
  2. GET: lấy buy log ứng với id của nó
  3. GET: lấy tất cả buy log của một sinh viên

1. **Class Print Log**

Body:

{

“id”: tự động tạo (long),

“status”: trạng thái (String),

“startTime”: Timestamp,

“finishedTime”: Timestamp,

“document”:

{

}

}