

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Cơ sở tại Tp. Hồ Chí Minh



**WEB SERVICE**

**ĐỀ TÀI: WEB SERVICE QUẢN LÝ SÁCH**

GIẢNG VIÊN HƯỚNG DẪN: **HUỲNH TRUNG TRỤ**

LỚP: **D16CQCP01-N**

NHÓM SINH VIÊN THỰC HIỆN:

**TRẦN CÔNG MINH – N16DCCN095**

**VŨ ĐĂNG KHOA – N16DCCN078**

**ĐẶNG CAO NGUYỄN – N16DCCN107**

**NGUYỄN THANH PHONG – N16DCCN116**

## NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

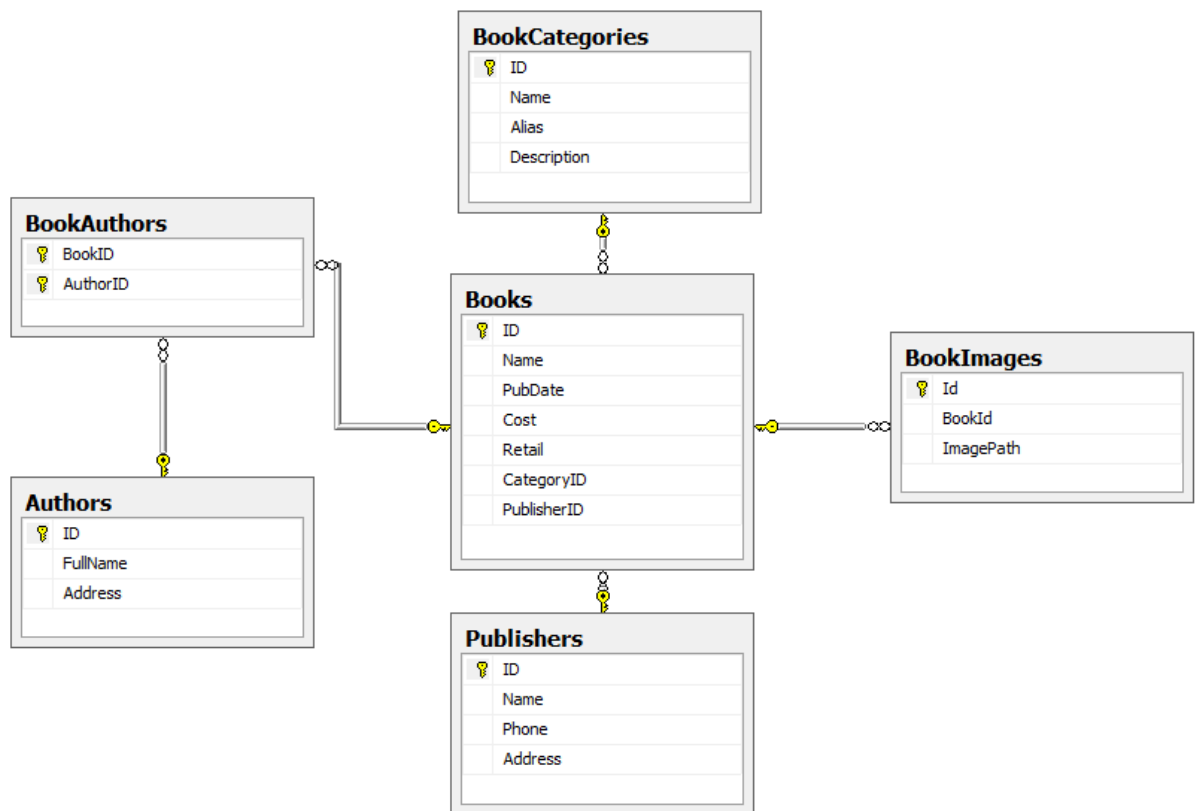
.....

## Mục Lục

<b>SERVER.....</b>	<b>3</b>
<b>1. Database.....</b>	<b>3</b>
<b>2. APIs.....</b>	<b>5</b>
2.1. Tổng quát.....	5
2.2. Author APIs.....	6
2.3. Book APIs .....	9
2.4. BookAuthor APIs .....	15
2.5. BookCategory APIs.....	15
2.6. Publisher APIs.....	19
<b>CLIENT – Winform.....</b>	<b>22</b>
<b>Công cụ:.....</b>	<b>22</b>
<b>Các công nghệ hỗ trợ chính:.....</b>	<b>22</b>
<b>Chức năng chính:</b> Lấy dữ liệu về, thêm, xóa, sửa dữ liệu lên Server API dưới dạng JSON .....	22
<b>Các bước thực hiện chính: .....</b>	<b>22</b>
<b>CLIENT – Android.....</b>	<b>27</b>
I. Các công cụ và thư viện cần dùng:.....	27
II. Các bước thực hiện:.....	27
1. Implementation các thư viện cần thiết vào file <b>build.gradle</b> . .....	27
2. Tạo một interface để thực hiện giao tiếp.....	27
3. Tạo một object class thực thi các service phía trên và trả về kết quả dưới dạng đọc được.....	29
III. Giao diện Demo .....	30
<b>CLIENT - REACTJS.....</b>	<b>31</b>
<b>Công cụ:.....</b>	<b>31</b>
<b>Các công nghệ hỗ trợ chính:.....</b>	<b>31</b>
<b>Chức năng chính:</b> Lấy dữ liệu về, thêm, xóa, sửa dữ liệu lên Server API dưới dạng JSON .....	31
<b>Các bước thực hiện chính: .....</b>	<b>31</b>

# SERVER

## 1. Database



**Books:** ID, Name, PubDate, Cost, Retail, CategoryID, PublisherID

**Publishers:** ID, Name, Phone, Address

**BookImages:** ID, BookID, ImagePath

**BookCategories:** ID, Name, Alias, Description

**BookAuthors:** BookID, AuthorID

**Author:** ID, FullName, Address

Bảng Books

STT	Field	Value type
1	ID	Int(auto generate)
2	Name	nvarchar(256)
3	Pubdate	Date

4	Cost	Money
5	Retail	Money
6	CategoryID	Int
7	PublisherID	Int

Bảng Publishers:

STT	Field	Value type
1	ID	Int(auto generate)
2	Name	nvarchar(256)
3	Phone	char(256)
4	Address	nvarchar(256)

Bảng BookImages:

STT	Field	Value type
1	ID	Int(auto generate)
2	BookID	Int
3	ImagePath	nvarchar(256)

Bảng BookCategories:

STT	Field	Value type
1	ID	Int(auto generate)
2	Name	nvarchar(256)
3	Alias	nvarchar(256)

4	Description	nvarchar(256)
---	-------------	---------------

Bảng BookAuthors:

STT	Field	Value type
1	AuthorID	Int
2	BookID	Int

Bảng Authors:

STT	Field	Value type
1	ID	Int(auto generate)
2	FullName	nvarchar(256)
3	Address	nvarchar(256)

## 2. APIs

### 2.1. Tổng quát

Web sử dụng công ASP.NET MVC WebAPI nên nền .NET 4.7.2

Tất cả các request đều có url theo mẫu:

<https://webapiptit.azurewebsites.net/{controller}/{action}/{id}>

### INT1448.WebApi

Author	Show/Hide	List Operations	Expand Operations
Book	Show/Hide	List Operations	Expand Operations
BookAuthor	Show/Hide	List Operations	Expand Operations
BookCategory	Show/Hide	List Operations	Expand Operations
Publisher	Show/Hide	List Operations	Expand Operations

## Hình 2.1

Hệ thống cung cấp các API để thao tác với hệ thống qua từng object:  
Author, Book, BookAuthor, BookCategory, Publisher như hình 2.1

### 2.2. Author APIs

#### Author

Show/Hide | List Operations | Expand Operations

GET	/api/authors/getall
GET	/api/authors/getbyid/{id}
GET	/api/authors/getallpublisherbook/{id}
PUT	/api/authors/update
POST	/api/authors/create
DELETE	/api/authors/delete/{id}

Author				
Method	Url pattern	Parameter		Description
GET	authors/getall			Lấy tất cả tác giả.
	authors/getbyid/{id:int}	id	int	Lấy tác giả theo id. Id(int) id của tác giả muốn lấy
PUT	authors/update	author	Object	Cập nhật tác giả. author(object): tác giả muốn cập nhật
POST	authors/create	author	Object	Tạo mới tác giả. author(object): Tác giả muốn tạo.
DELETE	authors/delete	id	int	Xóa tác giả theo id. Id(int) id của tác giả muốn xóa

#### 2.2.1. authors/getall

##### Request URL

```
https://webapiptit.azurewebsites.net/api/authors/getall
```

Kết quả trả về dưới dạng JSON:

```

1  [
2    {
3      "ID": 1,
4      "FullName": "Hồ Chí Minh",
5      "Address": "Làng Sen, xã Kim Liên, huyện Nam Đàn, tỉnh Nghệ An"
6    },
7    {
8      "ID": 2,
9      "FullName": "Nguyễn Hữu Phước",
10     "Address": "Xã Khánh Hội, huyện U Minh, Cà Mau"
11   },
12   {
13     "ID": 3,
14     "FullName": "Ngô Minh Vân",
15     "Address": "778/12 Nguyen Kiem, Ward 4, Phu Nhuan Dist, Ho Chi Minh City"
16   },
17   {
18     "ID": 4,
19     "FullName": "Ngọc Linh",
20     "Address": "17 St. 8, Ward 7, Go Vap Dist, Ho Chi Minh City"
21   },
22   {
23     "ID": 5,
24     "FullName": "Lê Văn Thông",
25     "Address": "8 Hoang Van Thu Street, Phu Nhuan District, Ho Chi Minh City"
26   },
27   {
28     "ID": 6,
29     "FullName": "PGS.TS.Võ Văn Nhị",

```

### 2.2.2. authors/getbyid/{id:int}

#### Request URL

```
https://webapiptit.azurewebsites.net/api/authors/getbyid/1
```

Kết quả trả về dưới dạng JSON:

```

{
  "ID": 1,
  "FullName": "Hồ Chí Minh",
  "Address": "Làng Sen, xã Kim Liên, huyện Nam Đàn, tỉnh Nghệ An"
}

```

### 2.2.3. authors/update

#### Request URL

```
https://webapiptit.azurewebsites.net/api/authors/update
```

Kèm theo object muốn update theo format:



```
{
  "ID": 0,
  "FullName": "string",
  "Address": "string"
}
```

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Update Author succeeded."
}
```

#### 2.2.4. authors/create

##### Request URL

```
https://webapiptit.azurewebsites.net/api/authors/create
```

Kèm theo object muốn tạo theo format:

```
{
  "ID": 0,
  "FullName": "string",
  "Address": "string"
}
```

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Create Author succeeded."
}
```

#### 2.2.5. authors/delete

##### Request URL

```
https://webapiptit.azurewebsites.net/api/authors/delete/15
```

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Delete Author succeeded."
}
```

## 2.3. Book APIs

### Book

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/books/getall
GET	/api/books/getalltoview
GET	/api/books/getbyid/{id}
GET	/api/books/getbyidtoview/{id}
PUT	/api/books/update
POST	/api/books/create
DELETE	/api/books/delete/{id}

Mô tả:

Book					
Method	Url pattern		Parameter	Return	Description
GET	books/getall			List result	Lấy danh sách tất cả sách.
	books/getalltoview			List result	Lấy danh sách tất cả sách để xem.
	book/getbyid/{id:int}		id	List result	Lấy sách theo id
	book/getbyidtoview/{id:int}		id	List result	Lấy sách theo id để xem
PUT	books/update		<a href="#">book</a>	Message	Cập nhật thông tin cho sách.
POST	books/create		<a href="#">book</a>	Message	Tạo sách mới
DELETE	books/delete/{int:id}		id	Message	Xóa sách theo id

### 2.3.1. books/getall

Request URL:

Request URL

`https://webapiptit.azurewebsites.net/api/books/getall`

Object parameter:

Kết quả trả về dưới dạng JSON:

```
[
  {
    "ID": 1,
    "Name": "Bộ Sách Hồ Chí Minh - Tác Phẩm (Hộp 5 Cuốn)",
    "PubDate": "2020-04-13T00:00:00",
    "Cost": 148000,
    "Retail": 118000,
    "CategoryID": 1,
    "PublisherID": 2,
    "BookImages": [
      {
        "Id": 2,
        "BookID": 1,
        "ImagePath": "https://www.dtv-ebook.com/images/sach-moi-phat-hanh/2020/bo-sach-ho-chi-minh-tac-pham-hop-5-cuon-tac-gia-ho-chi-minh.jpg"
      }
    ],
    "BookAuthors": [
      {
        "BookID": 1,
        "AuthorID": 1
      }
    ]
  },
  {
    "ID": 2,
    "Name": "Di Chúc Của Chủ Tịch Hồ Chí Minh",
    "PubDate": "2020-04-13T00:00:00",
    "Cost": 30000,

```

### 2.3.2. books/getalltoview

Request URL:

Request URL

<https://webapiptit.azurewebsites.net/api/books/getalltoview>

Object parameter:

Kết quả trả về dưới dạng JSON:

```

{
  "ID": 1,
  "Name": "Bộ Sách Hồ Chí Minh - Tác Phẩm (Hộp 5 Cuốn)",
  "PubDate": "2020-04-13T00:00:00",
  "Cost": 148000,
  "Retail": 118000,
  "BookCategoryName": "Chính trị - pháp luật",
  "PublisherName": "Nhà xuất bản Trẻ",
  "Images": [
    "https://www.dtv-ebook.com/images/sach-moi-phat-hanh/2020/bo-sach-ho-chi-minh-tac-pham-hop-5-cuon-1"
  ],
  "AuthorNames": [
    "Hồ Chí Minh"
  ]
},
{
  "ID": 2,
  "Name": "Di Chúc Của Chủ Tịch Hồ Chí Minh",
  "PubDate": "2020-04-13T00:00:00",
  "Cost": 30000,
  "Retail": 24000,
  "BookCategoryName": "Văn hóa xã hội - Lịch sử",
  "PublisherName": "Nhà xuất bản Trẻ",
  "Images": [
    "https://newshop.vn/public/uploads/products/19709/di-chuc-cua-chu-tich-ho-chi-minh-bia-truoc.jpg"
  ],
  "AuthorNames": [
    "Hồ Chí Minh"
  ]
}

```

### 2.3.3. books/getbyid/{id}

Request URL:

Request URL

<https://webapiptit.azurewebsites.net/api/books/getbyid/2>

Object parameter:

Kết quả trả về dưới dạng JSON:

```

{
  "ID": 2,
  "Name": "Đi Chúc Của Chủ tịch Hồ Chí Minh",
  "PubDate": "2020-04-13T00:00:00",
  "Cost": 30000,
  "Retail": 24000,
  "CategoryID": 3,
  "PublisherID": 2,
  "BookImages": [
    {
      "Id": 3,
      "BookId": 2,
      "ImagePath": "https://newshop.vn/public/uploads/products/19709/di-chuc-cua-chu-tich-ho-chi-minh-bia-truoc.jpg"
    }
  ],
  "BookAuthors": [
    {
      "BookID": 2,
      "AuthorID": 1
    }
  ]
}

```

#### 2.3.4. books/getbyidtoview/{id}

Request URL:

Request URL

<https://webapiptit.azurewebsites.net/api/books/getbyidtoview/2>

Object parameter:

Kết quả trả về dưới dạng JSON:

```

{
  "ID": 2,
  "Name": "Đi Chúc Của Chủ tịch Hồ Chí Minh",
  "PubDate": "2020-04-13T00:00:00",
  "Cost": 30000,
  "Retail": 24000,
  "BookCategoryName": "Văn hóa xã hội - Lịch sử",
  "PublisherName": "Nhà xuất bản Trẻ",
  "Images": [
    "https://newshop.vn/public/uploads/products/19709/di-chuc-cua-chu-tich-ho-chi-minh-bia-truoc.jpg"
  ],
  "AuthorNames": [
    "Hồ Chí Minh"
  ]
}

```

#### 2.3.5. books/update

Request URL:

### Request URL

`https://webapiptit.azurewebsites.net/api/books/update`

Object parameter:

Model	Example Value
<b>BookUpdateRequest {</b>	
<b>Id</b> (integer, optional),	
<b>Name</b> (string, optional),	
<b>PubDate</b> (string, optional),	
<b>Cost</b> (integer, optional),	
<b>Retail</b> (integer, optional),	
<b>CategoryID</b> (integer, optional),	
<b>PublisherID</b> (integer, optional),	
<b>BookImages</b> (Array[string], optional),	
<b>BookAuthors</b> (Array[integer], optional)	
<b>}</b>	

Kết quả trả về dưới dạng JSON nếu không thành công:

```
{
  "Success": "false",
  "Reason": "An error occurred while updating the entries. See the inner exception for details."
}
```

### 2.3.6. books/create

Request URL:

#### Request URL

`https://webapiptit.azurewebsites.net/api/books/create`

Object parameter:

Model	Example Value
<b>BookCreateRequest {</b>	
<b>Name</b> (string, optional),	
<b>PubDate</b> (string, optional),	
<b>Cost</b> (integer, optional),	
<b>Retail</b> (integer, optional),	
<b>CategoryID</b> (integer, optional),	
<b>PublisherID</b> (integer, optional),	
<b>BookImages</b> (Array[string], optional),	
<b>BookAuthors</b> (Array[integer], optional)	
<b>}</b>	

Kết quả trả về dưới dạng JSON nếu không thành công:

```
{
  "Success": "false",
  "Reason": "An error occurred while updating the entries. See the inner exception for details"
}
```

### 2.3.7. books/delete/{id}

Request URL:

Request URL

```
https://webapiptit.azurewebsites.net/api/books/delete/0
```

Object parameter:

Kết quả trả về dưới dạng JSON nếu không thành công:

```
{
  "Message": "An error has occurred."
}
```

## 2.4. BookAuthor APIs

### BookAuthor

Show/Hide | List Operations | Expand Operations

GET	/api/bookauthors/getallbyauthorid/{id}
GET	/api/bookauthors/getallbybookid/{id}
PUT	/api/bookauthors/update
POST	/api/bookauthors/create
DELETE	/api/bookauthors/delete/{bookid}/{authorid}

Cung cấp các API nội bộ nhằm lấy tất cả sách theo tác giả hay tất cả tác giả của sách bất kỳ.

## 2.5. BookCategory APIs

### BookCategory

Show/Hide | List Operations | Expand Operations

GET	/api/bookcategories/getall
GET	/api/bookcategories/getbyid/{id}
GET	/api/bookcategories/getallpublisherbook/{id}
PUT	/api/bookcategories/update
POST	/api/bookcategories/create
DELETE	/api/bookcategories/delete/{id}



BookCategory			
Method	Url patterm	Parameter	Description
GET	bookcategories/getall		Get all bookcategory
	bookcategories/getbyid/{id:int}	id	Get bookcategory by id
PUT	bookcategories/update	bookcategory	Update bookcategory. bookcategory(object): Entity want to update
POST	bookcategories/create	bookcategory	Create bookcategory. bookcategory(object): Entity want to create
DELETE	bookcategories/delete	id	Delete bookcategory. id(int): id of bookcategory want to delete

### 2.5.1. Bookcategories/getall

Request URL:

[Request URL](#)

```
https://webapiptit.azurewebsites.net/api/bookcategories/getall
```

Object parameter:

Kết quả trả về dưới dạng JSON:

```
[
  {
    "ID": 1,
    "Name": "Chính trị - pháp luật",
    "Alias": "chinh-tri-phap-luat",
    "Description": "none"
  },
  {
    "ID": 2,
    "Name": "Khoa học công nghệ - Kinh tế",
    "Alias": "khoa-hoc-cong-nghe-kinh-te",
    "Description": "none"
  },
  {
    "ID": 3,
    "Name": "Văn hóa xã hội - Lịch sử",
    "Alias": "van-hoa-xa-hoi-lich-su",
    "Description": "none"
  },
  {
    "ID": 4,
    "Name": "Văn học nghệ thuật",
    "Alias": "van-hoc-nghe-thuat",
    "Description": "none"
  },
  {
    "ID": 5,
    "Name": "Giáo trình",
    "Alias": "giao-trinh",

```

### 2.5.2. bookcategories/getbyid/{id:int}

Request URL:

Request URL

```
https://webapiptit.azurewebsites.net/api/bookcategories/getbyid/1
```

Object parameter:

Kết quả trả về dưới dạng JSON:

```
{
  "ID": 1,
  "Name": "Chính trị - pháp luật",
  "Alias": "chinh-tri-phap-luat",
  "Description": "none"
}
```

### 2.5.3. bookcategories/update

Request URL:

### Request URL

`https://webapiptit.azurewebsites.net/api/bookcategories/update`

Object parameter:

Model	Example Value
	<pre>{   "ID": 0,   "Name": "string",   "Alias": "string",   "Description": "string" }</pre>

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Update Category succeeded."
}
```

#### 2.5.4. bookcategories/create

Request URL:

### Request URL

`https://webapiptit.azurewebsites.net/api/bookcategories/create`

Object parameter:

Model	Example Value
	<pre>{   "ID": 0,   "Name": "string",   "Alias": "string",   "Description": "string" }</pre>

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Create category succeeded."
}
```

#### 2.5.5. bookcategories/delete

Request URL:

### Request URL

```
https://webapiptit.azurewebsites.net/api/bookcategories/delete/9
```

Object parameter:

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Delete category succeeded."
}
```

## 2.6. Publisher APIs

### Publisher

Show/Hide | List Operations | Expand Operations

GET	/api/publishers/getall
GET	/api/publishers/getbyid/{id}
PUT	/api/publishers/update
POST	/api/publishers/create
DELETE	/api/publishers/delete/{id}

Publisher				
Method	Url pattern	Parameter		Description
GET	publishers/getall			Lấy tất cả các nhà xuất bản
	publishers/getbyid/{id:int}	id	int	Lấy nhà xuất bản theo ID
PUT	publishers/update	publisher	Object	Cập nhật nhà xuất bản. publisher(object): nhà xuất bản muốn cập nhật
POST	publishers/create	publisher	Object	Create publisher. publisher(object): Entity want to create
DELETE	publishers/delete	id	int	Xóa nhà xuất bản. id(int): Id của nhà xuất bản muốn xóa

### 2.6.1. publishers/getall

Request URL:

### Request URL

`https://webapiptit.azurewebsites.net/api/publishers/getall`

Object parameter:

Kết quả trả về dưới dạng JSON:

```
[
  {
    "ID": 1,
    "Name": "Nhà xuất bản Kim Đồng",
    "Phone": "02439434730",
    "Address": "55 Quang Trung, Hai Bà Trưng, Hà Nội"
  },
  {
    "ID": 2,
    "Name": "Nhà xuất bản Trẻ",
    "Phone": "(028) 39316289",
    "Address": "161B Lý Chính Thắng, Phường 7, Quận 3, Thành phố Hồ Chí Minh"
  },
  {
    "ID": 3,
    "Name": "Nhà xuất bản Tổng hợp thành phố Hồ Chí Minh",
    "Phone": "(028) 38 225 340",
    "Address": "62 Nguyễn Thị Minh Khai, Phường Đa Kao, Quận 1, TP.HCM"
  },
  {
    "ID": 4,
    "Name": "Nhà xuất bản chính trị quốc gia sự thật",
    "Phone": "024.3822-1633",
    "Address": "Số 6/86 Duy Tân, Cầu Giấy, Hà Nội"
  },
  {
    "ID": 5,
    "Name": "Nhà xuất bản giáo dục",
    "Phone": "024.38220801",
  }
]
```

### 2.6.2. publishers/getbyid/{id:int}

Request URL:

### Request URL

`https://webapiptit.azurewebsites.net/api/publishers/getbyid/1`

Object parameter:

Kết quả trả về dưới dạng JSON:

```
{
  "ID": 1,
  "Name": "Nhà xuất bản Kim Đồng",
  "Phone": "02439434730",
  "Address": "55 Quang Trung, Hai Bà Trưng, Hà Nội"
}
```

### 2.6.3. publishers/update

Request URL:

**Request URL**

```
https://webapiptit.azurewebsites.net/api/publishers/update
```

Object parameter:

Model	Example Value
	<pre>{   "ID": 0,   "Name": "string",   "Phone": "string",   "Address": "string" }</pre>

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Update publisher succeeded."
}
```

### 2.6.4. publishers/create

Request URL:

**Request URL**

```
https://webapiptit.azurewebsites.net/api/publishers/create
```

Object parameter:

Model	Example Value
	<pre>{   "ID": 0,   "Name": "string",   "Phone": "string",   "Address": "string" }</pre>

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Create publisher succeeded."
}
```

#### 2.6.5. publishers/delete

Request URL:

Request URL

```
https://webapiptit.azurewebsites.net/api/publishers/delete/13
```

Object parameter:

Kết quả trả về dưới dạng JSON:

```
{
  "Success": "true",
  "Reason": "Delete publisher succeeded."
}
```

## CLIENT – Winform

Công cụ:

- Microsoft .NET Framework 4.6.1
- Microsoft Visual Studio + DevExpress 19.2.3.0
- Microsoft.AspNet.WebApi.Client NuGet package (Web API Client Libraries).

**Các công nghệ hỗ trợ chính:**

- Lập trình bất đồng bộ asynchronous C#
- Thư viện HttpClient để nhận và gửi yêu cầu đến server

**Chức năng chính:** Lấy dữ liệu về, thêm, xóa, sửa dữ liệu lên Server API dưới dạng JSON

**Các bước thực hiện chính:**

1. **Tải Package** - [Microsoft.AspNet.WebApi.Client](#) NuGet package (Web API Client Libraries).  
Mở Tools menu → NuGet Package Manager → Package Manager Console → Ở cửa sổ Package Manager Console nhập lệnh sau:

```
Install-Package Microsoft.AspNet.WebApi.Client
```

2. **Cài đặt HttpClient** – Tạo một thể hiện của lớp HttpClient và set up Base Address và DefaultRequestHeaders

```
3. private void frmMain_Load(object sender, EventArgs e)
4. {
```

```

5.         HttpClient client = HttpClients.GetInstance();
6.         // add base Url
7.         client.BaseAddress = new Uri($" https://webapiptit.azurewebsites.net/api/");
8.         client.DefaultRequestHeaders.Accept.Clear();
9.
10.        // Add an Accept header for JSON format.
11.        client.DefaultRequestHeaders.Accept.Add(
12.            new MediaTypeWithQualityHeaderValue("application/json"));
13.    }

```

### 3. Gửi Request đến Server

Có nhiều phương thức hỗ trợ nhưng trong đồ án sử dụng các phương thức sau đây:

#### GET: GetAsync

```

1. public async Task<Models.Author> GetAuthorAsync(int authorId)
2. {
3.     HttpClient client = HttpClients.GetInstance();
4.     Models.Author author = null;
5.     HttpResponseMessage response = await client.GetAsync("authors/getbyid/" + authorId);
6.     if (response.IsSuccessStatusCode)
7.     {
8.         author = await response.Content.ReadAsAsync<Models.Author>();
9.     }
10.    return author;
11. }

```

#### POST: PostAsJsonAsync

```

1. public async Task<bool> AddNewAuthorAsync(Models.Author author)
2. {
3.
4.     HttpClient client = HttpClients.GetInstance();
5.     HttpResponseMessage response = await client.PostAsJsonAsync("authors/create", author);
6.     if (!response.IsSuccessStatusCode)
7.     {
8.         return false; ;
9.     }
10.    return true;
11. }

```

#### PUT: PutAsJsonAsync

```

1. public async Task<bool> EditAuthorAsync(Models.Author author)
2. {
3.
4.     HttpClient client = HttpClients.GetInstance();
5.     HttpResponseMessage response = await client.PutAsJsonAsync("authors/update/", author);
6.
7.     if (!response.IsSuccessStatusCode)
8.     {
9.         return false; ;

```



```

10.     }
11.     return true;
12. }

```

## DELETE: DeleteAsync

```

1. public async Task<System.Net.HttpStatusCode> DeleteAuthorAsync(int authorId)
2. {
3.     HttpClient client = HttpClients.GetInstance();
4.     HttpResponseMessage response = await client.DeleteAsync("authors/delete/" + authorId);
5.     return response.StatusCode;
6. }

```

## 4. Nhận Response

Sử dụng phương thức **ReadAsStringAsync** trong gói [System.Net.Http](https://docs.microsoft.com/en-us/dotnet/api/system.net.http)

```

1. public async Task<List<Models.BookCategory>> GetAllBookCategoryAsync()
2. {
3.     HttpClient client = HttpClients.GetInstance();
4.
5.     HttpResponseMessage response = await client.GetAsync("bookcategories/getall");
6.     List<Models.BookCategory> listBookCategory = null;
7.     if (response.IsSuccessStatusCode)
8.     {
9.         listBookCategory = await response.Content.ReadAsStringAsync<List<Models.BookCategory>>();
10.
11.     }
12.
13.     return listBookCategory;
14. }

```

```

1. private async void LoadDataToGridView()
2. {
3.     var listBookCategory = await _BookCategoryRepository.GetAllBookCategoryAsync();
4.     gridControl1.DataSource = listBookCategory;
5. }

```

### • Để tăng Performance :

**HttpClient** chú tâm đến việc khởi tạo một lần và được sử dụng lại trong suốt vòng đời của một ứng dụng.

Các điều kiện sau đây có thể dẫn đến lỗi **SocketException**:

- Tạo một thể hiện của lớp **HttpClient** mới cho mỗi yêu cầu.
- Máy chủ chịu tải nặng.

Trích: <https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client>

Vì vậy để tránh lỗi đáng tiếc xảy ra được nêu trên thì phải có cách nào đó để có thể tạo instance của lớp **HttpClient** một lần và sử dụng nó cho những lần sau, có một cách phổ biến là dùng mẫu thiết kế **singleton** trong **design pattern** để giải quyết việc này.

Code:

```
1. class HttpClients
2. {
3.     private volatile static HttpClient uniqueInstance;
4.     private static object syncRoot = new Object();
5.
6.     private HttpClients()
7.     { }
8.
9.     public static HttpClient GetInstance()
10.    {
11.
12.        if (uniqueInstance == null)
13.        {
14.            lock (syncRoot)
15.            {
16.                if (uniqueInstance == null)
17.                {
18.                    uniqueInstance = new HttpClient();
19.                }
20.            }
21.        }
22.        return uniqueInstance;
23.    }
24. }
```

- Giao diện chính của ứng dụng:

Book - Book Api Client

Book Management

Publisher

Author

Book

Book Category

MENU

Book

ID	Name	Pub Date	Cost	Retail	Book Category Name	Publisher Name
1	Bộ Sách Hồ Chí Minh - ...	Mon 4 13 2020	148000	118000	Chính trị – pháp luật	Nhà xuất bản Trẻ
2	Di Chúc Của Chủ Tịch ...	Mon 4 13 2020	30000	24000	Văn hóa xã hội – Lịch sử	Nhà xuất bản Trẻ
3	Nhật Ký Trong Tủ	Mon 4 13 2020	35000	32000	Văn hóa xã hội – Lịch sử	Nhà xuất bản giáo dục
4	Đề Mèn Phiêu Lưu Ký (...)	Mon 4 13 2020	120000	110000	Truyện - tiểu thuyết	Nhà xuất bản Kim Đồng
5	Tô Hoài - Tuyển Tập T...	Mon 4 13 2020	98000	78000	Truyện - tiểu thuyết	Nhà xuất bản Hội Nhà văn
6	Hai Khôi Tĩnh	Mon 4 13 2020	42000	34000	Truyện - tiểu thuyết	Nhà xuất bản Hội Nhà văn
7	Chút Phận Linh Đỉnh	Mon 4 13 2020	106000	96000	Truyện - tiểu thuyết	Nhà xuất bản Hội Nhà văn
8	10 Vạn Câu Hỏi Vì Sao ...	Mon 4 13 2020	55000	44000	Thiếu nhi	Nhà xuất bản Thanh niên
9	10 Vạn Câu Hỏi Vì Sao ...	Mon 4 13 2020	50000	40000	Thiếu nhi	Nhà xuất bản Thanh niên
10	10 Vạn Câu Hỏi Vì Sao ...	Mon 4 13 2020	65000	60000	Thiếu nhi	Nhà xuất bản Thanh niên
11	Hướng Dẫn Thực Hành...	Mon 4 13 2020	120000	110000	Khoa học công nghệ – Kinh tế	Nhà xuất bản lao động
12	Kế Toán Tài Chính	Mon 4 13 2020	95000	76000	Khoa học công nghệ – Kinh tế	Nhà xuất bản giáo dục
13	Người Tị Nạn	Mon 4 13 2020	195000	156000	Tâm lý - tâm linh - tôn giáo	Nhà xuất bản Hội Nhà văn
14	Từ Điển Thành Ngữ Tụ...	Mon 4 13 2020	189000	151000	Giáo trình	Nhà xuất bản Tổng hợp thành phố Hồ Chí Minh

Author - Book Api Client

Book Management

Publisher

Author

Book

Book Category

MENU

Book Author

Action

ADD NEW

DELETE

EDIT

RELOAD

LIST ALL OF AUTHOR

ID	Full Name	Address	Books
1	Hồ Chí Minh	Làng Sen, xã Kim Liên, huyện Nam Đàn, tỉnh Nghệ An	
2	Nguyễn Hữu Phước	Xã Khánh Hội, huyện U Minh, Cà Mau	
3	Ngô Minh Văn	778/12 Nguyen Kiem, Ward 4, Phu Nhuan Dist, Ho Chi Minh City	
4	Ngọc Linh	17 St. 8, Ward 7, Go Vap Dist, Ho Chi Minh City	
5	Lê Văn Thông	8 Hoang Van Thu Street, Phu Nhuan District, Ho Chi Minh City	
6	PGS.TS.Võ Văn Nhị	352 Lang Thuong Street, Dong Da District, Hanoi	
7	Tô Hoài	136/36 Nguyen Thi Tan Street, Ward 2, District 8, Ho Chi Minh City	
8	Hồ Biểu Chánh	220/42/16 Nguyen Xi St., Ward 26, Binh Thanh Dist, Ho Chi Minh City	
9	Khánh Linh	304/65 Ho Van Hue, Ward 9, Phu Nhuan Dist, Ho Chi Minh	
10	Gia Bảo	Str. 10, Song Than 1 IZ, Di An Townlet, Binh Duong	

Edit

ID

1

Full Name

Hồ Chí Minh

Address

Làng Sen, xã Kim Liên, huyện Nam Đàn, tỉnh Nghệ An

SAVE

CANCEL

## CLIENT – Android

- I. Các công cụ và thư viện cần dùng:
  - Android Studio.
  - Máy ảo NoxPlayer.
  - Thư viện **retrofit2** thao tác với Api.
  - Thư viện **gson** chuyển từ JSON qua Class.
  - Thư viện **coroutines** hỗ trợ gọi Api bất đồng bộ.
- II. Các bước thực hiện:
  1. Implementation các thư viện cần thiết vào file **build.gradle**.

```
implementation 'com.google.code.gson:gson:2.8.6'

implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-core:1.3.0'
implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.3.0'

implementation 'com.squareup.retrofit2:retrofit:2.8.1'
implementation 'com.squareup.retrofit2:converter-gson:2.8.1'
implementation 'com.jakewharton.retrofit:retrofit2-kotlin-coroutines-adapter:0.9.2'
```

2. Tạo một interface để thực hiện giao tiếp.

```
package com.ptit.bookapi.utils

import com.google.gson.GsonBuilder
import com.jakewharton.retrofit2.adapter.kotlin.coroutines.CoroutineCallAdapterFactory
import com.ptit.bookapi.models.*
import kotlinx.coroutines.Deferred
import okhttp3.ResponseBody
import retrofit2.Response
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.*

const val BASE_URL = "https://webapiptit.azurewebsites.net/api/"

const val GET_BOOK_LIST: String = "books/getall"
const val GET_BOOK_DETAIL: String = "books/getbyid/{book_id}"
const val POST_BOOK: String = "books/create"
const val PUT_BOOK: String = "books/update"
const val DELETE_BOOK: String = "books/delete/{book_id}"

const val GET_CATEGORY_LIST: String = "bookcategories/getall"

const val GET_AUTHOR_LIST: String = "authors/getall"
const val GET_AUTHOR_DETAIL: String = "authors/getbyid/{author_id}"
const val POST_AUTHOR: String = "authors/create"
```

```

const val PUT_AUTHOR: String = "authors/update"
const val DELETE_AUTHOR: String = "authors/delete/{author_id}"

const val GET_PUBLISHER_LIST: String = "publishers/getall"

interface ApiService {

    @GET(GET_BOOK_DETAIL)
    fun bookDetailAsync(
        @Path("book_id") bookId: Int
    ): Deferred<Response<Book>>

    @GET(GET_AUTHOR_DETAIL)
    fun authorDetailAsync(
        @Path("author_id") authorId: Int
    ): Deferred<Response<Author>>

    @GET(GET_BOOK_LIST)
    fun bookListAsync(): Deferred<MutableList<Book>>

    @DELETE(DELETE_BOOK)
    fun bookDeleteAsync(
        @Path("book_id") bookId: Int
    ): Deferred<Response<ResponseBody>>

    @DELETE(DELETE_AUTHOR)
    fun authorDeleteAsync(
        @Path("author_id") authorId: Int
    ): Deferred<Response<ResponseBody>>

    @GET(GET_CATEGORY_LIST)
    fun bookCategoryListAsync(): Deferred<List<BookCategory>>

    @GET(GET_AUTHOR_LIST)
    fun authorListAsync(): Deferred<MutableList<Author>>

    @GET(GET_PUBLISHER_LIST)
    fun publisherListAsync(): Deferred<List<Publisher>>

    @POST(POST_BOOK)
    fun bookPostAsync(@Body book: BookToPost):
Deferred<Response<ResponseBody>>

    @POST(POST_AUTHOR)
    fun authorPostAsync(@Body author: Author):
Deferred<Response<ResponseBody>>

    @PUT(PUT_BOOK)
    fun bookPutAsync(@Body book: BookToPut): Deferred<Response<ResponseBody>>

    @PUT(PUT_AUTHOR)
    fun authorPutAsync(@Body author: Author): Deferred<Response<ResponseBody>>

    companion object{
        operator fun invoke(): ApiService {

```

```

        val gsonFormat = GsonBuilder()
            .setDateFormat("yyyy-MM-dd'T'HH:mm:ss")
            .create()

        val retrofit = Retrofit.Builder()
            .addConverterFactory(GsonConverterFactory.create(gsonFormat))
            .baseUrl(BASE_URL)
            .addCallAdapterFactory(CoroutineCallAdapterFactory())
            .build()

        return retrofit.create(ApiService::class.java)
    }
}

```

3. Tạo một object class thực thi các service phía trên và trả về kết quả dưới dạng đọc được.

```

package com.ptit.bookapi.utils

import com.ptit.bookapi.models.*
import kotlinx.coroutines.runBlocking

object ApiImpl {
    private val apiService: ApiService = ApiService()
}

```

a. GET

```

fun bookListAsync(): MutableList<Book> = runBlocking{
    apiService.bookListAsync().await()
}

```

b. GET by Id

```

fun bookDetailAsync(bookId: Int): Book? = runBlocking{
    val deferred = apiService.bookDetailAsync(bookId).await()
    if(deferred.isSuccessful){
        deferred.body()
    }else{
        null
    }
}

```

c. POST

```

fun bookPost(book: BookToPost): CustomResponse = runBlocking {
    val response = apiService.bookPostAsync(book).await()
    if(response.isSuccessful)
        CustomResponse(response.isSuccessful, response.code(),
            response.body())
    CustomResponse(response.isSuccessful, response.code(),
        response.errorBody())
}

```

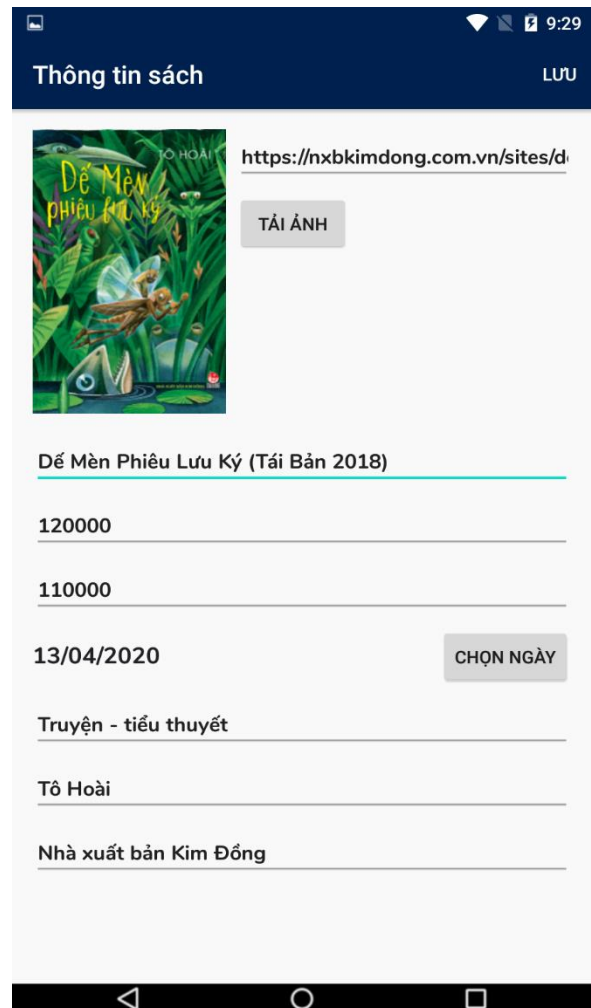
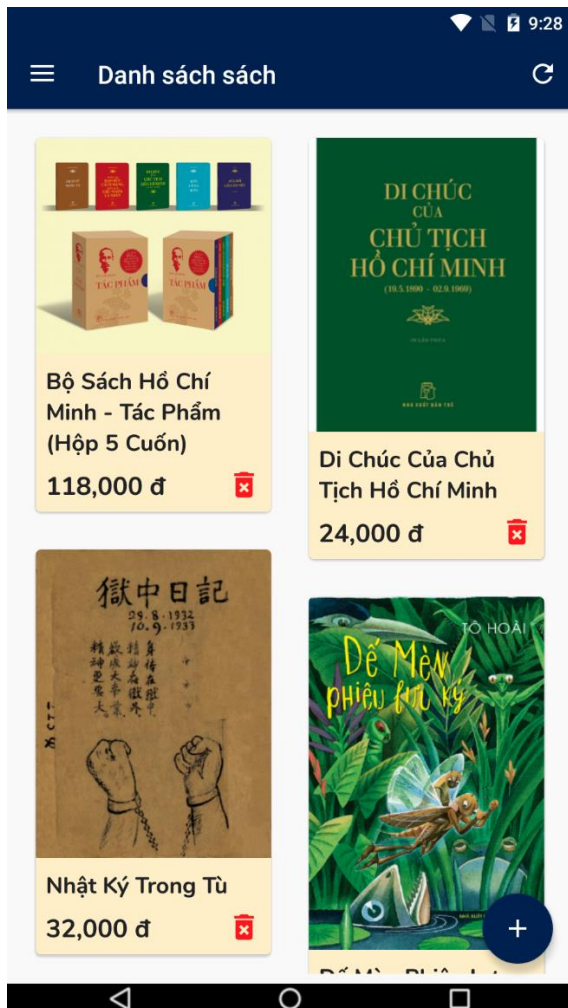
d. PUT

```
fun bookPut(book: BookToPut): CustomResponse = runBlocking {
    val response = apiService.bookPutAsync(book).await()
    if(response.isSuccessful)
        CustomResponse(response.isSuccessful, response.code(),
            response.body())
    CustomResponse(response.isSuccessful, response.code(),
        response.errorBody())
}
```

**Chú ý:** Class CustomResponse sẽ chứa 3 thuộc tính:

- success: Request có success không
- code: Response code
- responseBody: Response từ server

### III. Giao diện Demo



# CLIENT - REACTJS

## Công cụ:

- ReactJs, React Hooks
- Visual Studio Code
- Microsoft.AspNet.WebApi.Client NuGet package (Web API Client Libraries).

## Các công nghệ hỗ trợ chính:

- Lập trình bất đồng bộ async JavaScript
- Thư viện HttpClient để nhận và gửi yêu cầu đến server

**Chức năng chính:** Lấy dữ liệu về, thêm, xóa, sửa dữ liệu lên Server API dưới dạng JSON

## Các bước thực hiện chính:

### 14. Dùng Yarn để cài đặt các gói

Vào thư mục chứa Project mở Terminal gõ lệnh

```
yarn
```

### 15. Thực thi các Service các các hàm

**GET:**

```
const getBooks = async () => {  
  try {  
    const response = await fetch("/api/books/getall");  
    const jsonData = await response.json();  
    setBooks(jsonData);  
  } catch (e) {  
    console.error("err", e.message);  
  }  
};
```



## POST:

```
const onSubmitForm = async (
  e,
  Name,
  Cost,
  Retail,
  CategoryID,
  PublisherID,
  BookImages,
  BookAuthors
) => {
  e.preventDefault();
  const pubDate = new Date().toISOString();
  try {
    const body = {
      Name: Name,
      PubDate: pubDate,
      Cost: Cost,
      Retail: Retail,
      CategoryID: CategoryID,
      PublisherID: PublisherID,
      BookImages: [BookImages],
      BookAuthors: BookAuthors,
    };
    const response = await fetch("/api/books/create", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify(body),
    });
    getBooks();
    history.push("/");
  } catch (err) {
    console.error(err.message);
  }
};
```

## UPDATE:

```
const updateBook = async (
  e,
  id,
  Name,
  Cost,
  Retail,
  CategoryID,
  PublisherID,
  BookImages,
  BookAuthors
) => {
  e.preventDefault();
  const pubDate = new Date().toISOString();
  try {
    debugger;
    const body = {
      ID: id,
      Name: Name,
      PubDate: pubDate,
      Cost: Cost,
      Retail: Retail,
      CategoryID: CategoryID,
      PublisherID: PublisherID,
      BookImages: [BookImages],
      BookAuthors: BookAuthors,
    };
    const response = await fetch("/api/books/update", {
      method: "PUT",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify(body),
    });
    getBooks();
    history.push("/");
  } catch (e) {
    console.error(e.messages);
  }
};
```

## DELETE:

```
const deleteBook = async (id) => {
  try {
    const deleteBook = await fetch(`/api/books/delete/${id}`, {
      method: "DELETE",
    });
    setBooks(books.filter((book) => book.ID !== id));
  } catch (e) {
    console.error(e.message);
  }
};
```

## 16. Demo Web:

### Danh Sách Book:

#	Book Name	Public Date	Cost	Edit	Delete
1	Bộ Sách Hồ Chí Minh - Tác Phẩm (Hộp 5 Cuốn)	2020-04-13T00:00:00	148000	Edit	Delete
2	Di Chúc Của Chủ tịch Hồ Chí Minh	2020-04-13T00:00:00	30000	Edit	Delete
3	Nhật Ký Trong Tủ	2020-04-13T00:00:00	35000	Edit	Delete
4	Để Mèn Phiêu Lưu Ký (Tái Bản 2018)	2020-04-13T00:00:00	120000	Edit	Delete
5	Tô Hoài - Tuyển Tập Truyện Ngắn Hay	2020-04-13T00:00:00	98000	Edit	Delete
6	Hai Khối Tinh	2020-04-13T00:00:00	42000	Edit	Delete
7	Chút Phận Linh Đình	2020-04-13T00:00:00	106000	Edit	Delete
8	10 Vạn Câu Hỏi Vì Sao - Khám Phá Thế Giới Vi Sinh Vật	2020-04-13T00:00:00	55000	Edit	Delete
9	10 Vạn Câu Hỏi Vì Sao - Khám Phá Thế Giới Thực Vật	2020-04-13T00:00:00	50000	Edit	Delete
10	10 Vạn Câu Hỏi Vì Sao - Khám Phá Thế Giới Động Vật	2020-04-13T00:00:00	65000	Edit	Delete

## Chi Tiết Sách:

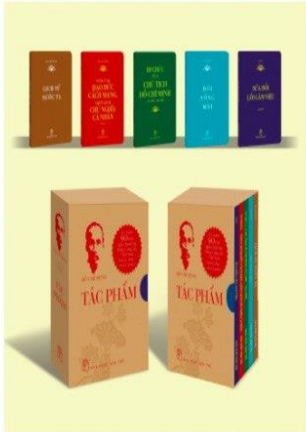
Book Manager

localhost:3000/book/1

Apps Facebook Messenger Instagram Code Learning Pluralsight Hệ thống quản trị... Đăng ký môn học... 568Play Portal Font Awesome Other Bookmarks

Home Books Add

### Bộ Sách Hồ Chí Minh - Tác Phẩm (Hộp 5 Cuốn)



Tác giả: Hồ Chí Minh

XUẤT BẢN: BỘ SÁCH HỒ CHÍ MINH - TÁC PHẨM (HỘP 5 CUỐN)

**Price: 148000 đ**

Danh Mục

Chính trị – pháp luật

## Xóa Sách:

Book Manager

localhost:3000/books

Apps Facebook Messenger Instagram Code Learning Pluralsight Hệ thống quản trị... Đăng ký môn học... 568Play Portal Font Awesome Other Bookmarks

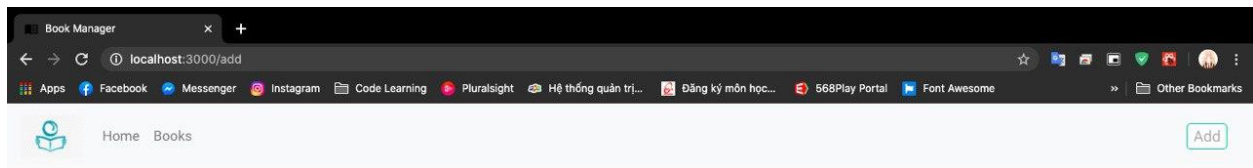
Home Books Deleted!

#	Book Name	Public Date	Cost	Edit	Delete
1	Bộ Sách Hồ Chí Minh - Tác Phẩm (Hộp 5 Cuốn)	2020-04-13T00:00:00	148000	Edit	Delete
2	Di Chúc Của Chủ tịch Hồ Chí Minh	2020-04-13T00:00:00	30000	Edit	Delete
3	Nhật Ký Trong Tủ	2020-04-13T00:00:00	35000	Edit	Delete
4	Đế Mèn Phiêu Lưu Ký (Tái Bản 2018)	2020-04-13T00:00:00	120000	Edit	Delete
5	Tô Hoài - Tuyển Tập Truyện Ngắn Hay	2020-04-13T00:00:00	98000	Edit	Delete
6	Hai Khối Tinh	2020-04-13T00:00:00	42000	Edit	Delete
7	Chút Phận Linh Đình	2020-04-13T00:00:00	106000	Edit	Delete
8	10 Vạn Câu Hỏi Vì Sao - Khám Phá Thế Giới Vi Sinh Vật	2020-04-13T00:00:00	55000	Edit	Delete
9	10 Vạn Câu Hỏi Vì Sao - Khám Phá Thế Giới Thực Vật	2020-04-13T00:00:00	50000	Edit	Delete
10	10 Vạn Câu Hỏi Vì Sao - Khám Phá Thế Giới Động Vật	2020-04-13T00:00:00	65000	Edit	Delete

1

2

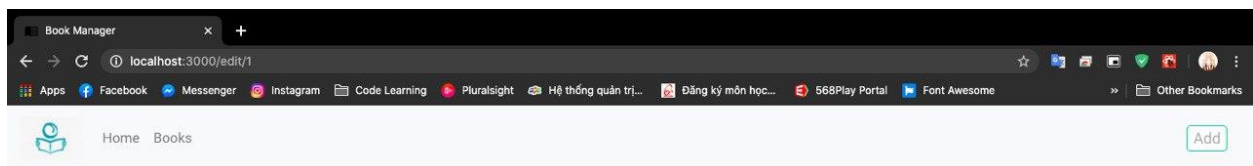
## Thêm Sách:



**Book Name**

  
**Cost** **Retail**  
   
**Categories**  
  
**Publishers**  
  
**File**  
  
**Author**  
  
  
  
  

## Sửa Sách:



**Book Name**

  
**Cost** **Retail**  
   
**Categories**  
  
**Publishers**  
  
**File**  
  
**Author**