

# Sử dụng LibMan – công cụ hỗ trợ tải thư viện

Hướng dẫn tự học lập trình ASP.NET Core toàn tập > Sử dụng LibMan – công cụ hỗ trợ tải thư viện

Trong bài học này chúng ta sẽ học cách cấu hình và sử dụng LibMan – công cụ hỗ trợ làm việc với ASP.NET Core của Visual Studio để tải thư viện css/js cho client.

## NỘI DUNG CỦA BÀI [ Ấn ]

1. LibMan là gì?
2. Tạo project thử nghiệm
  - 2.1. Bật chức năng LibMan Manage Client-Side Libraries
3. Cài đặt thư viện BootStrap
  - 3.1. Option 1 – Sử dụng file cấu hình libman.json
  - 3.2. Option 2 – Sử dụng giao diện đồ họa
4. Cài đặt jQuery
5. Một số kỹ thuật khác với LibMan
  - 5.1. Xóa bỏ toàn bộ thư viện
  - 5.2. Xóa hoặc cập một thư viện
6. Kết luận

## LibMan là gì?

Khi phát triển ứng dụng web nói chung bạn sẽ luôn có nhu cầu cài đặt và sử dụng các bộ thư viện CSS và JavaScript thông dụng như BootStrap, jQuery.

Bạn có thể cài đặt các thư viện CSS và JavaScript theo kiểu thủ công (download-copy-paste) từ site của nhà cung cấp. Tuy nhiên cách thức này có rất nhiều bất cập. Hầu hết đều tránh sử dụng phương pháp này.

Cộng đồng đã phát triển những công cụ hỗ trợ cài đặt và quản lý các thư viện rất mạnh mẽ như NPM, YARN, WebPack. Tuy vậy chúng đều khá phức tạp khi cài đặt và sử dụng. Lý do là những công cụ này hướng tới hỗ trợ những người phát triển ứng dụng client-side/front-end (chạy trên trình duyệt) và do đó có rất nhiều tính năng và công cụ bổ trợ.

Đối với người mới bắt đầu tiếp cận ứng dụng web hoặc chỉ tập trung cho server-side (back-end), việc học sử dụng những công cụ này khá vất vả và mất nhiều thời gian công sức. Trong tình huống này, bạn chỉ cần một công cụ đơn giản giúp cài đặt các thư viện bạn cần một cách nhanh chóng để có thể tập trung vào công việc cần thiết.

**LibMan** (Library Manager) là một công cụ của Visual Studio hỗ trợ tải các thư viện CSS và JavaScript cho ứng dụng ASP.NET Core được xây dựng với mục đích như vậy. LibMan là một công cụ đơn giản, nhẹ và tiện lợi giúp tìm và tải các file thư viện từ các nguồn trên Internet về dự án.

Cần lưu ý rằng LibMan không phải là một hệ thống quản lý gói phần mềm như NPM, và cũng không phải là công cụ thay thế cho các hệ thống tương tự.

Chúng ta cùng thực hiện một ví dụ.

# Tạo project thử nghiệm

**Bước 1.** Tạo một empty project mới:

## Configure your new project

ASP.NET Core Web Application C# Linux macOS Windows Cloud Service Web

Project name

Location

Solution name

☒ Place solution and project in the same directory

## Create a new ASP.NET Core web application

.NET Core ASP.NET Core 3.1

**Empty**  
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

**API**  
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

**Web Application**  
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

**Web Application (Model-View-Controller)**  
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

**Authentication**

No Authentication

[Change](#)

**Advanced**

☐ Configure for HTTPS

☐ Enable Docker Support  
(Requires [Docker Desktop](#))

Linux

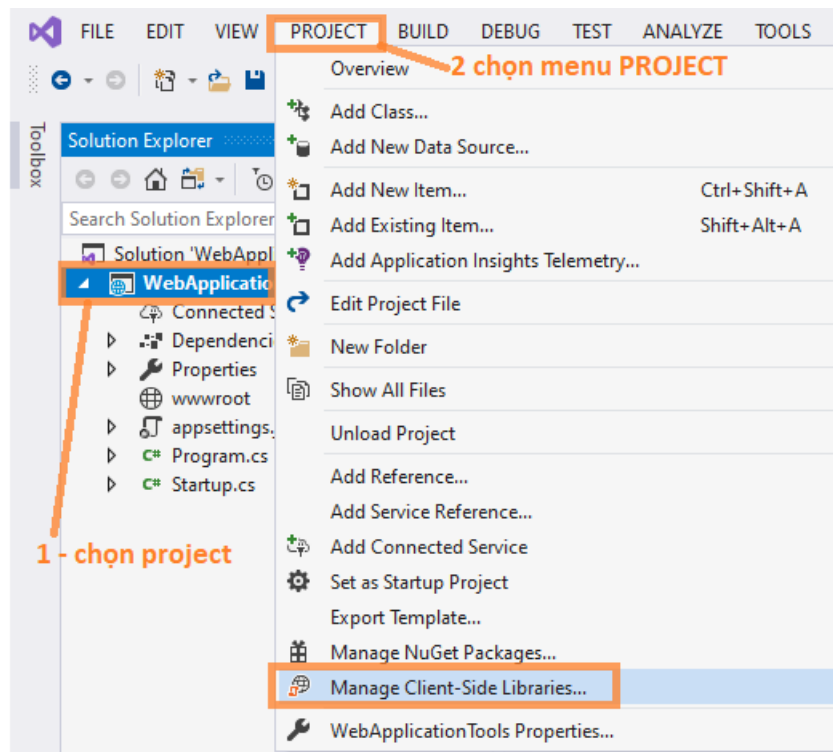
**Bước 2.** Cấu hình sử dụng StaticFiles middleware:

```
15. public void Configure(IApplicationBuilder app, IWebHostEnvironment env) {
16.     if (env.IsDevelopment()) {
17.         app.UseDeveloperExceptionPage();
18.     }
19.
20.     app.UseRouting();
21.     app.UseStaticFiles();
22.
23.     app.UseEndpoints(endpoints => {
24.         endpoints.MapGet("/", async context => {
25.             await context.Response.WriteAsync("Hello World!");
26.         });
27.     });
28. }
```

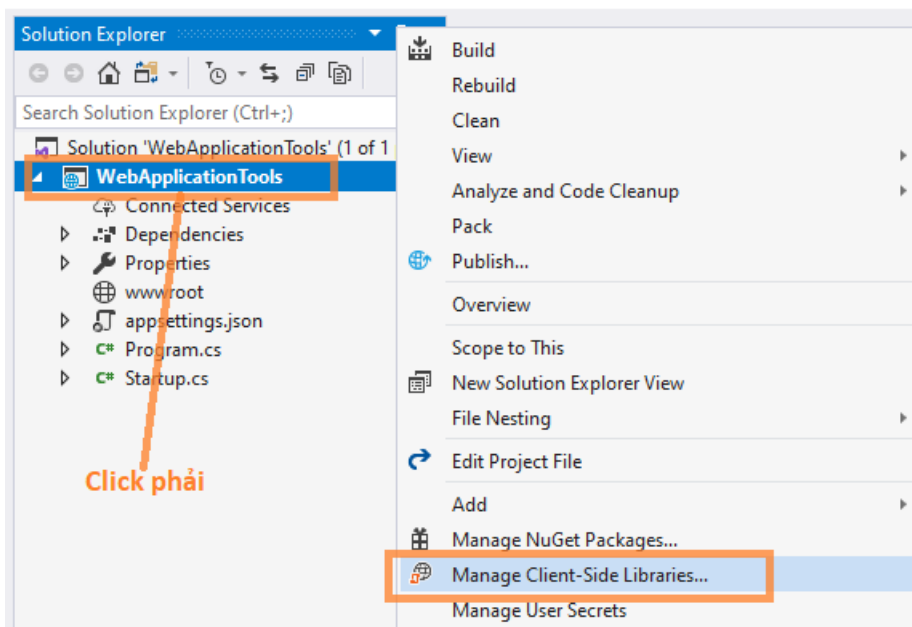
**Bước 3.** Tạo thư mục `wwwroot` trực thuộc project.

## Bật chức năng LibMan Manage Client-Side Libraries

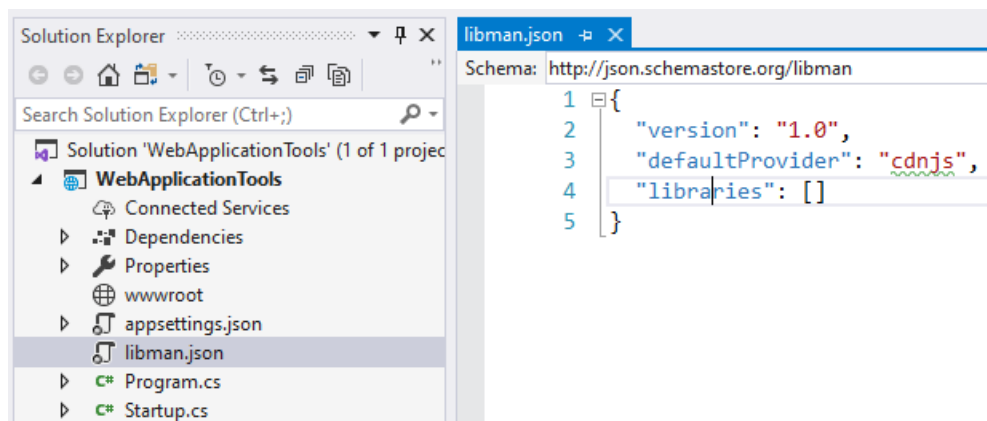
**Bước 4.** Chọn chức năng "Manage Client-Side Libraries theo một trong hai cách sau:



Hoặc



Khi này file libman.json với nội dung như sau sẽ được thêm vào project.



Đây là file cấu hình của LibMan.

## Cài đặt thư viện BootStrap

---

Bạn có hai cách làm việc với LibMan:

### Option 1 – Sử dụng file cấu hình libman.json

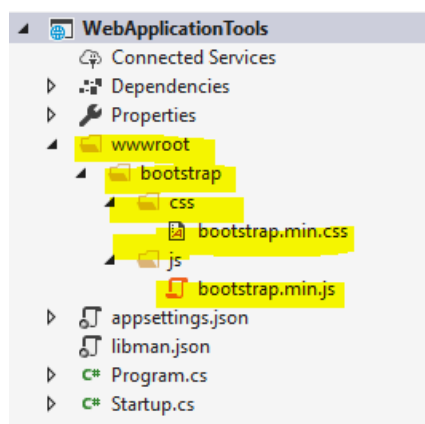
Nhập trực tiếp thông tin vào libman.json:

```
1. {  
2.   "version": "1.0",  
3.   "defaultProvider": "cdnjs",  
4.   "libraries": [  
5.     {  
6.       "library": "twitter-bootstrap@4.4.0",  
7.       "destination": "wwwroot/bootstrap",  
8.       "files": [ "css/bootstrap.min.css", "js/bootstrap.min.js" ]  
9.     }  
10.  ]  
11. }
```

Bạn yên tâm, Visual Studio hỗ trợ nhắc code cho file này rất tốt. Bạn chỉ cần nhớ và nhập vài chữ cái đầu tiên là đủ. Còn lại bạn có thể chọn từ danh sách hỗ trợ của Intellisense.

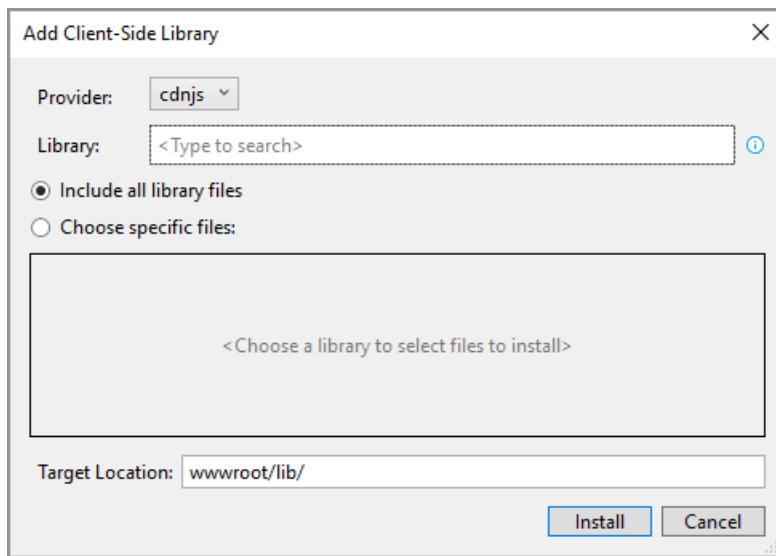
Trong ví dụ trên bạn yêu cầu tải thư viện bootstrap 4.4.0 về máy (vào thư mục wwwroot/bootstrap). Tuy nhiên, bạn chỉ muốn tải hai file bootstrap.min.css (trong thư mục css) và bootstrap.min.js (trong thư mục js).

Ngay khi lưu file libman.json, LibMan sẽ tải các file cần thiết vào đúng thư mục bạn chỉ định:

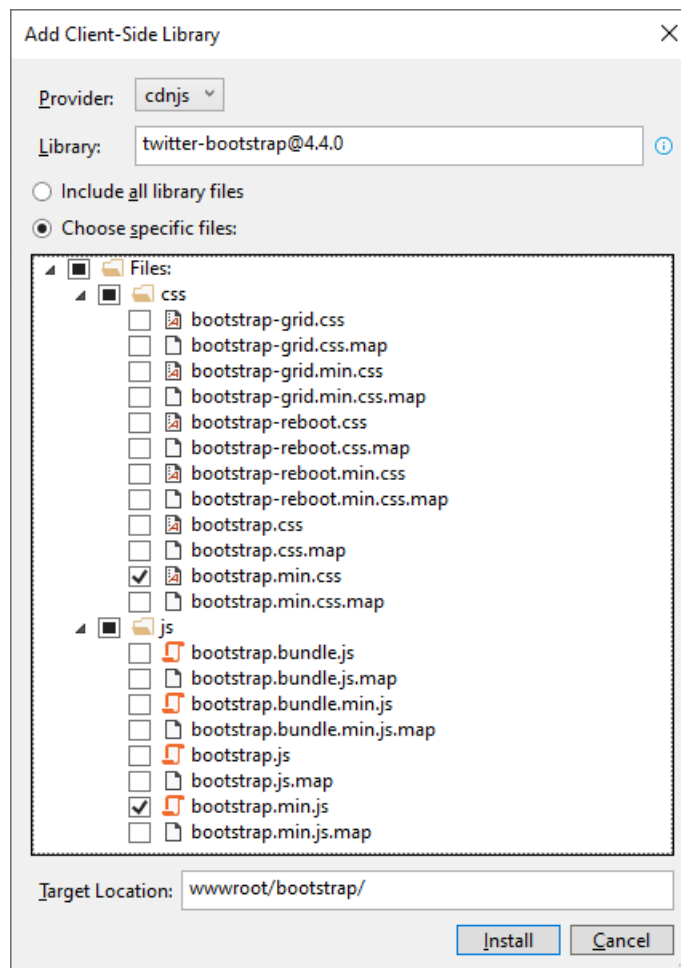


### Option 2 – Sử dụng giao diện đồ họa

Click phải vào project => Add => Client Side Library để mở cửa sổ sau:



Nhập tên thư viện cần tải và tích chọn file tương ứng vào ô textbox Library:



Cả hai cách cho cùng một kết quả. Những gì bạn chọn ở giao diện đồ họa cũng được đưa vào file libman.json.

## Cài đặt jQuery

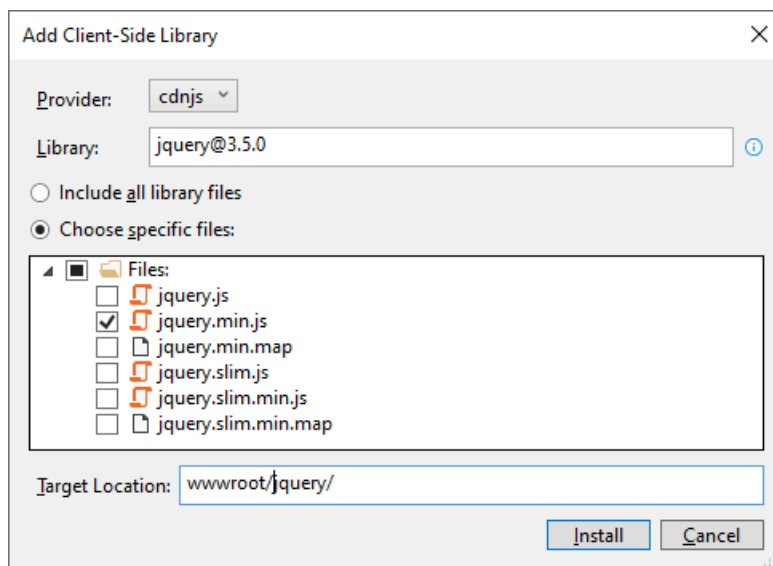
Do Bootstrap yêu cầu phải có thêm jQuery, hãy thực hiện thao tác tương tự như trên để cài đặt jQuery vào thư mục wwwroot/lib:

```

1. {
2.   "version": "1.0",
3.   "defaultProvider": "cdnjs",
4.   "libraries": [
5.     {
6.       "library": "twitter-bootstrap@4.4.0",
7.       "destination": "wwwroot/bootstrap/",
8.       "files": [
9.         "js/bootstrap.min.js",
10.        "css/bootstrap.min.css"
11.      ]
12.    },
13.    {
14.      "library": "jquery@3.5.0",
15.      "destination": "wwwroot/jquery/",
16.      "files": [
17.        "jquery.min.js"
18.      ]
19.    }
20.  ]
21. }

```

Hoặc bằng giao diện đồ họa:

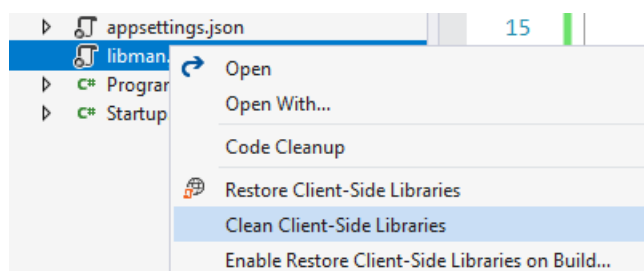


## Một số kỹ thuật khác với LibMan

Sau đây là một số kỹ thuật khác khi làm việc với LibMan.

### Xóa bỏ toàn bộ thư viện

Click phải chuột vào file libman.json và chọn Clean Client-Side Libraries:



Thao tác này sẽ xóa bỏ tất cả các file thư viện đã tải về project nhưng giữ nguyên file cấu hình libman.json.

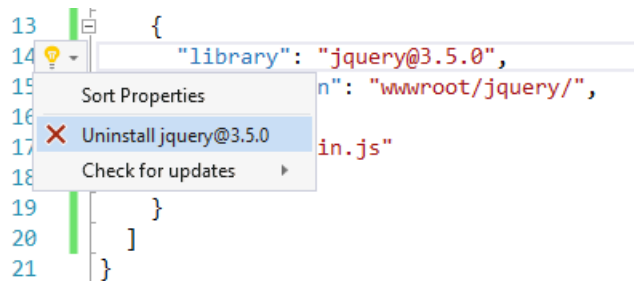
Nếu bạn mở file libman.json và ấn Ctrl + S (để lưu file lại), tất cả các thư viện sẽ được tải lại.

Tính năng này rất có ích nếu bạn vô tình làm hỏng các file thư viện và cần khôi phục lại.

## Xóa hoặc cập một thư viện

Nếu bạn chỉ muốn xóa một thư viện cụ thể nào đó, hãy mở file libman.json và đặt con trỏ chuột vào mục chứa thông tin của thư viện đó.

Sau đó bật menu Quick Action bằng cách ấn vào nút hình bóng đèn hoặc tổ hợp phím Ctrl + . (dấu chấm).



Cách nhanh chóng hơn cả để xóa thư viện chính là xóa mục tương ứng trong file cấu hình!

Nếu muốn cập nhật thư viện bạn thực hiện tương tự như thao tác xóa bỏ một thư viện nhưng chọn mục Check for updates.

## Kết luận

Trong bài học này bạn đã làm quen với LibMan – công cụ hỗ trợ tải các thư viện CSS và JavaScript cho ứng dụng Asp.net Core. Đây là một công cụ đơn giản nhưng rất tiện lợi giúp chúng ta cài đặt những bộ thư viện cơ bản cần thiết cho quá trình học và phát triển ứng dụng.

- + Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
  - + Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
  - + Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
- Cảm ơn bạn!