**Laravel Mix công cụ xây dựng tổng hợp cho dự án Laravel**

3.392 Views [0](https://allaravel.com/laravel-tutorials/laravel-mix-cong-cu-xay-dung-tong-hop-cho-du-an-laravel/#respond) by [FirebirD](https://allaravel.com/author/kiendc/)

Trong một dự án phầm mềm, viết code chỉ là một trong những công việc cần làm, ngay cả với những website nhỏ, các công việc vẫn cần thực hiện như nén ảnh, tối ưu hóa các file css, javascript… giúp giảm kích thước lưu trữ và tối ưu băng thông. Việc làm này tạo ra trải nghiệm tốt cho người dùng do tốc độ tải trang nhanh chóng hơn, đặc biệt Google cũng rất thích điều này, Google sẽ đánh giá website của bạn tốt hơn và dẫn đến thứ hạng SEO cũng tốt hơn. Laravel Mix là một công cụ như vậy, nó giúp xây dựng file css từ các CSS pre-processing như Sass, Less dễ dàng hơn bao giờ hết. Laravel Mix giúp bạn giảm rất nhiều thời gian trong các công việc có thể tự động hóa được.

**Quản lý mã CSS, Javascript, tổng hợp và biên dịch tài nguyên Laravel Mix**

Các phiên bản từ Laravel 5.3 trở về trước Laravel sử dụng Laravel Elixir. Laravel Elixir xuất hiện lần đầu vào tháng 10 năm 2014, công cụ này cho phép tự động gắn kết các file và kiểm soát các tài nguyên liên quan dễ dàng. Laravel Elixir được xây dựng dựa trên Gulp (Một công cụ xây dựng hệ thống mã nguồn mở sử dụng để tự động hóa các tác vụ thông qua Gulp task). [Laravel Mix là cái tên mới từ phiên bản Laravel 5.4](https://allaravel.com/laravel-news/laravel-elixir-se-co-ten-moi-laravel-mix-tu-phien-ban-5-4/), nó được xây dựng dựa trên Webpack.js. Từ nãy đến giờ hơi dông dài chút, giờ chúng ta sẽ tiến hành cài đặt và sử dụng Laravel Mix trong những ví dụ thực tế giúp hiểu rõ hơn Laravel Mix là gì?

**Cài đặt và thiết lập Laravel Mix**

Trước tiên chúng ta cần cài đặt [Node.js](https://nodejs.org/en/) là môi trường Javascript runtime xây dựng dựa vào bộ engine Chrome’s V8 Javascript, nó sử dụng các mô hình xử lý theo sự kiện, mô hình non-blocking I/O do đó chạy cực nhẹ nhàng và hiệu quả. Node.js chứa bên trong một hệ sinh thái cung cấp các gói thư viện lập trình là [npm](https://www.npmjs.com/), có thể  thấy đây là một hệ sinh thái lớn nhất cung các thư viện lập trình mã nguồn mở.

Laravel sử dụng npm để quản lý các thư viện lập trình javascript, css của mình, do đó bạn cần cài đặt npm trước tiên. Do Node.js đã tích hợp npm vào bộ cài, do đó bạn cần cài đặt Node.js là đã có npm. [Tải bộ cài Node.js](https://nodejs.org/en/download/) về, lựa chọn phiên bản cho phù hợp với môi trường của mình, nó hỗ trợ các môi trường thông dụng hiện nay như Linux, OSX, Windows. Trong bài viết này tôi sử dụng môi trường Windows, sau khi cài đặt xong để kiểm tra phiên bản node.js và npm chúng ta sử dụng lệnh

c:\xampp\htdocs\laravel-test>node -v

v6.10.1

c:\xampp\htdocs\laravel-test>npm -v

3.10.10

Chú ý, vì phiên bản của npm được release thường xuyên hơn, do đó bạn nên cập nhật npm để có được phiên bản mới nhất, sử dụng câu lệnh cập nhật như sau:

c:\xampp\htdocs\laravel-test>npm install npm@latest -g

[ ..............] - loadRequestedDeps: sill install loadAllDepsIntoIdealTree

...

+-- which@1.2.12

| `-- isexe@1.1.2

+-- wrappy@1.0.2

`-- write-file-atomic@1.3.1

Ok, chúng ta thử kiểm tra lại phiên bản npm

c:\xampp\htdocs\laravel-test>npm -v

4.4.4

Như vậy đã cập nhật được phiên bản npm mới nhất cho đến thời điểm hiện tại.

Cũng giống như [Composer](https://allaravel.com/laravel-tutorials/composer-cong-cu-quan-ly-thu-vien-php-tuyet-voi/) sử dụng composer.json, npm sử dụng file package.json để quản lý các thư viện đang sử dụng trong dự án.

{

"private": true,

"scripts": {

"dev": "node node\_modules/cross-env/bin/cross-env.js NODE\_ENV=development node\_modules/webpack/bin/webpack.js --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js",

"watch": "node node\_modules/cross-env/bin/cross-env.js NODE\_ENV=development node\_modules/webpack/bin/webpack.js --watch --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js",

"watch-poll": "node node\_modules/cross-env/bin/cross-env.js NODE\_ENV=development node\_modules/webpack/bin/webpack.js --watch --watch-poll --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js",

"hot": "node node\_modules/cross-env/bin/cross-env.js NODE\_ENV=development node\_modules/webpack-dev-server/bin/webpack-dev-server.js --inline --hot --config=node\_modules/laravel-mix/setup/webpack.config.js",

"production": "node node\_modules/cross-env/bin/cross-env.js NODE\_ENV=production node\_modules/webpack/bin/webpack.js --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js"

},

"devDependencies": {

"axios": "^0.15.3",

"bootstrap-sass": "^3.3.7",

"jquery": "^3.1.1",

"laravel-mix": "^0.8.1",

"lodash": "^4.17.4",

"vue": "^2.1.10"

}

}

Chúng ta thấy Laravel Mix đã được thiết lập sẵn trong file này, ngay cả Javascript framework Vue.js cũng đã được thiết lập. Thực hiện cài đặt các gói này bằng lệnh:

c:\xampp\htdocs\laravel-test>npm install --no-bin-links

[ ............] / fetchMetadata: verb afterAdd C:\Users\Admin\AppData\Roam

...

| | | +-- shellwords@0.1.0

| | | `-- which@1.2.14 deduped

| | +-- object-assign@4.1.1 deduped

| | `-- strip-ansi@3.0.1 deduped

| `-- webpack-stats-plugin@0.1.4

+-- lodash@4.17.4

`-- vue@2.2.5

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.0.0 (node\_modules\ch

okidar\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@

1.1.1: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}

)

–no-bin-links chỉ sử dụng trong môi trường Windows, để ngăn chặn việc tạo ra các symlink cho các file bin (trong Windows gọi là shortcut cho file thực thi).

**Chạy Laravel Mix**

Trong thư mục gốc dự án xây dựng bằng Laravel sẽ có một file là package.json, file này chính là file cấu hình cho các công cụ quản lý thư viện lập trình như npm hoặc yarn. Bạn nên tìm hiểu [kiến thức cơ bản về npm](https://allaravel.com/tutorials/quan-ly-thu-vien-lap-trinh/npm-la-cong-cu-gi/) và [cách sử dụng yarn](https://allaravel.com/tutorials/quan-ly-thu-vien-lap-trinh/yarn-la-gi-tai-sao-yarn-dang-thay-the-dan-npm/) vì có nhiều phần kiến thức liên quan. File package.json chứa các gói thư viên lập trình Javascript sẽ được cài đặt trong dự án và một số các kịch bản xây dựng như sau:

* dev
* watch
* hot
* production

Các kịch bản này được thực thi như sau:

npm run dev – Thực thi tất cả các tác vụ Mix một lần.

mpn run watch – Thực thi tất cả các tác vụ Mix và tiếp tục theo dõi các tài nguyên, nếu có sự thay đổi này đó sẽ tự động biên dịch lại tài nguyên, kịch bản này rất hay được dùng khi phát triển ứng dụng.

npm run hot – Thực thi tất cả các tác vụ Mix nhưng vẫn active và theo dõi sự thay đổi các tài nguyên, nếu có sự thay đổi, nó thực hiện cập nhật chỉ các module có sự thay đổi, sau đó nó báo cáo với ứng dụng về sự thay đổi và cập nhật mã trong trình duyệt mà không có tải lại tài nguyên.

npm run production – Thực thi tất cả các tác vụ Mix và tối ưu (minify) các file kết quả đầu ra, khi đó bạn có thể sử dụng các kết quả để triển khai ứng dụng trên máy chủ.

**Các xử lý và cài đặt trong Laravel Mix**

**Làm việc với CSS**

Trong thư mục gốc dự án chứa file webpack.mix.js là file thiết lập các entry point – là các điểm bắt đầu để Mix quét và thực hiện các kịch bản. Từ đây, Laravel Mix xác định được chính xác cách biên dịch các tài nguyên. Laravel Mix hỗ trợ rất nhiều các ngôn ngữ tiền xử lý CSS như Less, Sass, Stylus, PostCSS…

**Less, Sass, Stylus**

Laravel Mix có các phương thức less(), sass(), stylus() để thực hiện biên dịch các ngôn ngữ tiền xử lý CSS thành ngôn ngữ CSS thông thường mà trình duyệt hiểu được. Cú pháp thực hiện như sau:

mix.less('resources/assets/less/app.less', 'public/css');

Với câu lệnh này khi thực hiện build, nó sẽ biên dịch từ file resources/assets/less/app.less thành file public/css/app.css. Bạn có thể thay đổi tên file đầu ra bằng cách đưa đường dẫn đầy đủ cùng tên file bạn cần đặt:

mix.less('resources/assets/less/app.less', 'public/stylesheets/styles.css');

Để biên dịch nhiều file Less thực hiện chuỗi các tác vụ như sau:

mix.less('resources/assets/less/app.less', 'public/css')

.less('resources/assets/less/admin.less', 'public/css');

Cũng với cách tương tự bạn có thể thực hiện với các ngôn ngữ tiền xử lý CSS khác như Sass, Stylus bằng cách sử dụng mix.sass() hoặc mix.stylus và bằng cách thay các file với phần mở rộng là .scss, .sass, .styl

**CSS thông thường**

Nếu bạn muốn gộp nhiều file CSS thành một file duy nhất, bạn có thể sử dụng phương thức style:

mix.styles([

'public/css/vendor/normalize.css',

'public/css/vendor/videojs.css'

], 'public/css/all.css');

**Xử lý URL trong file CSS**

Laravel Mix được xây dựng dựa trên Webpack do đó chúng ta cần hiểu qua [Cách thức sử dụng Webpack](https://allaravel.com/tutorials/quan-ly-thu-vien-lap-trinh/huong-dan-webpack-co-ban/). Trong biên dịch CSS, Webpack sẽ tối ưu và rewrite lại các phương thức url() gọi trong các file stylesheet. Xem xét ví dụ sau:

.example {

background: url('../images/example.png');

}

Laravel Mix sẽ tìm file example.png và copy vào thư mục public/images và thực hiện rewrite url() như sau:

.example {

background: url(/images/example.png?d41d8cd98f00b204e9800998ecf8427e);

}

Nếu cấu trúc thư mục của bạn đã có sẵn và bạn không muốn thực hiện rewrite đường dẫn, thiết lập tùy chọn tắt tính năng này đi như sau:

mix.sass('resources/assets/app/app.scss', 'public/css')

.options({

processCssUrls: false

});

**Source maps**

Các file output sau khi đã được gộp và minify sẽ rất khó để debug, Source Maps giúp chúng ta ánh xạ code đã biên dịch này với các mã nguồn ban đầu giúp cho trình duyệt có thể debug chính xác các file ban đầu. Mặc định, source map được tắt đi nhưng có thể bật tính năng này bằng cách thực hiện phương thức mix.sourceMaps() trong webpack.mix.js:

mix.js('resources/assets/js/app.js', 'public/js')

.sourceMaps();

Nó cung cấp thêm thông tin ánh xạ giúp debug trên các công cụ phát triển trên trình duyệt, tuy nhiên nó cũng làm tăng dung lượng và hiệu năng thực thi với các file này.

**Làm việc với Javascript**

Laravel Mix cung cấp rất nhiều tính năng để làm việc với Javascript như biên dịch ECMAScript 2015 hay còn gọi là ES6, đóng gói module, minify đầu ra, kết hợp nhiều file thuần Javascript… Nó làm rất tốt công việc của mình mà không cần cấu hình nào:

mix.js('resources/assets/js/app.js', 'public/js');

Với một dòng code như thế này bạn đã có thể có sử dụng những ưu thế của Mix như sau:

* Sử dụng cú pháp của ES6
* Module hóa.
* Biên soạn các file .vue (Khi dự án sử dụng Vue.js)
* Minify với các đầu ra khi sử dụng kịch bản npm run production.

**Tách biệt các thư viện Javascript của bên thứ ba**

Có một vấn đề là các thư viện Javascript của bên thứ ba cũng được đóng gói cùng thư viện do bạn viết ra, trong hai loại thư viên này thì thư viện bên thứ ba rất ít khi thay đổi trong khi code bạn viết thì thay đổi thường xuyên hơn dẫn đến mỗi khi thay đổi, trình duyệt lại tải lại toàn bộ. Việc tách thư viên bên thứ ba ra thành file riêng giúp trình duyệt chỉ tải lại những gì cần thiết, giảm băng thông và tăng tốc độ xử lý file.

Mix cung cấp phương thức extract() cho việc này:

mix.js('resources/assets/js/app.js', 'public/js')

.extract(['vue'])

Phương thức extract() chấp nhận đầu vào là một mảng các thư viện bên thứ ba mà bạn muốn đóng gói thành file vendor.js. Với đoạn mã trên, Laravel Mix sẽ sinh ra ba file:

* public/js/manifest.js: webpack manifest runtime.
* public/js/vendor.js: chứa các thư viện bên thứ ba.
* public/js/app.js: chứa code do bạn viết.

Trong file HTML bạn cũng phải đưa vào các file Javascript này theo thứ tự:

<script src="/js/manifest.js"></script>

<script src="/js/vendor.js"></script>

<script src="/js/app.js"></script>

**React**

Mix có thể tự động cài đặt plugin Babel cần thiết cho React. Để xử lý React chỉ đơn giản thay phương thức js() thành react():

mix.react('resources/assets/js/app.jsx', 'public/js');

Laravel Mix tải và cài đặt plugin babel-preset-react trong trường hợp này.

**File cấu hình Webpack riêng**

Như [Hướng dẫn cơ bản của Webpack](https://allaravel.com/tutorials/quan-ly-thu-vien-lap-trinh/huong-dan-webpack-co-ban/), webpack sử dụng file webpack.config.js để thiết lập các cấu hình cho việc đóng gói thư viện. Laravel Mix đã tạo sẵn một file webpack.config.js với các thiết lập từ trước giúp bạn có thể thực thi luôn một cách nhanh chóng. Nhưng đôi khi bạn cũng cần thêm vào các kịch bản để làm khác đi, ví dụ thay vì sử dụng Sass bạn có thể dùng preprocessor Stylus. Trong tình huống này bạn có hai cách để thực hiện:

**Cách 1: sử dụng phương thức webpackConfig()**

Phương thức webpackConfig() giúp bạn thêm vào các cấu hình cho Webpack, nó không yêu cầu bạn phải copy và duy trì file webpack.config.js. Tham số của webpackConfig() là một đối tượng với các tùy chọn cấu hình như Tài liệu hướng dẫn Webpack.

mix.webpackConfig({

resolve: {

modules: [

path.resolve(\_\_dirname, 'vendor/laravel/spark/resources/assets/js')

]

}

});

**Cách 2: sử dụng file webpack.config.js**

Nếu bạn muốn tự mình cấu hình Webpack thì có thể copy file node\_modules/laravel-mix/setup/webpack.config.js vào thư mục gốc của dự án. Sau đó, thiết lập –config trong file package.json trỏ đến file webpack.config.js của bạn.

**Copy file và thư mục**

Laravel Mix cung cấp phương thức copy() để thực hiện việc copy file và thư mục đến một vị trí nào đó trong dự án. Tính năng này rất hữu ích khi bạn muốn copy file hoặc thư mục từ node\_modules vào thư mục public.

mix.copy('node\_modules/foo/bar.css', 'public/css/bar.css');

Để duy trì cấu trúc thư mục khi copy một thư mục Mix khuyến cáo bạn sử dụng phương thức copyDirectory() thay cho phương thức copy():

mix.copyDirectory('assets/img', 'public/img');

**Đánh phiên bản file đầu ra**

Đây là một tính năng cực kỳ hữu ích của Laravel Mix, chúng ta cùng xem xét bối cảnh sau. Các trình duyệt đều lưu cache các file tải về trong một khoảng thời gian, do đó trong khoảng thời gian này không có cách nào bắt trình duyệt phải tải lại các file cần thiết do các file này có tên giống với file phiên bản trước. Để giải quyết vấn đề này Laravel Mix có phương thức version(), phương thức này hoạt động bằng cách thêm vào một chuỗi là kết quả hàm băm nội dung file.

mix.js('resources/assets/js/app.js', 'public/js')

.version();

Khi thực hiện build, các file đầu ra sẽ có dạng app.hash\_string.js, và mỗi lần hash\_string sẽ thay đổi nếu nội dung file thay đổi, vậy làm cách nào để chèn vào mã HTML, Laravel hỗ trợ phương thức mix() giúp bạn giải quyết việc đó.

<link rel="stylesheet" href="{{ mix('/css/app.css') }}">

Vì file được đánh phiên bản không cần thiết trong quá trình phát triển ứng dụng nên chúng ta cần thiết lập chỉ tạo ra khi thực hiện npm run production như sau:

mix.js('resources/assets/js/app.js', 'public/js');

if (mix.config.inProduction) {

mix.version();

}

**Thông báo hoạt động**

Mặc định, mỗi khi một tác vụ Mix hoàn thành, một thông báo sẽ được Hệ điều hành đưa ra, để tắt thông báo này bạn có thể thêm vào phương thức disableNotifications() như sau:

mix.disableNotifications();

vào trong file webpack.mix.js

**Các thư viện mặc định trong Laravel Mix**

Laravel Mix thiết lập danh sách thư viện mặc định trong package.json, bạn có thể mở ra và thấy một số các thư viện như sau:

* [Axios](https://www.npmjs.com/package/axios): thư viện giúp tạo ra các HTTP request, là sự thay thế tốt cho vue-resources.
* [Bootstrap Sass](https://www.npmjs.com/package/bootstrap-sass): sử dụng cho file app.scss đưa vào các mã style cho Bootstrap.
* [jQuery](https://www.google.com.vn/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwitxtqaqu7TAhVBsJQKHRO0AXMQFggkMAA&url=https%3A%2F%2Fjquery.com%2F&usg=AFQjCNFnz7C6MAXGLm7pVcOD_LrOjJUUiA&sig2=zsNh6ujbyLbx2wdRH4-YRQ): thư viện javascript nổi tiếng giúp thao tác với DOM dễ dàng.
* [Lodash](https://lodash.com/): Cung cấp các hàm thao tác với dữ liệu.
* [Vue.js v2](https://vuejs.org/): framework javascript dần thay thế React, Angular làm nền tảng cho fontend.