Laravel II
Posted March 25th, 2024

Basic attempt to do a simple crud application.
1. So you can install laravel globally
<code>
composer global require laravel/installation
</code>
This installs the laravel binary in ${HOME}/.config/composer/vendor/bin
Its useful to put this as a separate user. I chose composer
<code>
useradd composer
su - composer
composer global require laravel/installation
chmod 777 /home/composer
</code>
So other users can use the laravel installation binary, add the location to
Path. edit /etc/profile.d/local.sh
<code>
export PATH=$PATH:/home/composer/.config/composer/vendor/bin
</code>

2. starter kits. So following a tutorial for laravel, and as usual its out of data. So to create a project:
<code>
laravel new posty
</code>
And now it asks me for a "starter kit"
None, Breeze, Jetstream.
I chose None

And now it asks me the testing framework
Pest, or PHPUnit
I chose Pest - it was the default
And now whether I want to initialize a git repository

And now it asks me for what database to use
Mysql, Mariadb, PostgresSQl, SQLite, SQL Server

3. php artisan lists all the various things you can
do with artisan e.g. laravel

4. So blade The laravel default template engine.
  First thing is to be introduced to Layouts....
  a. php artisan make:component Layout
    // this generates a Layout component with
    // resources/views/controller/layout.blade.php
   b. Here is a sample layout controller (resoure/views/controller/layout.blade.php
   <code>
   <pre>
   <html>
 <head>
   <title>{{ $title ?? 'Example Website' }}</title>
   <link rel="stylesheet" href="{{ asset('/css/app.css') }}">
 </head>
 <body>
   <nav>
     <h3>Welcome to my website</h3>
     <hr>
   </nav>
   {{ $slot }}
   <footer>
     <hr />
     © 2023 example.com
   </footer>
 </body>
</html>
</pre>
   </code>

c. So a content php file is called a view.
   each file view is named viewname.blade.php
  example welcome.blade.php

d. Sample welcome.blade.php
  <code>

<pre>
<x-layout>
<x-slot name="title">
Home | Example Website
</x-slot>

<h1>Hello World!</h1>
Lorem ipsum dolor sit amet consectetur adipiscing elit. Hic, aut?
<button class="btn">Get Started</button>

</x-layout>
</pre>
</code>
the magic x-layout tag tells the template to use the layout component
the x-slot tag lets you pass alternate content to the layout
the code {{ $title ?? 'Example Website' }} says to show the variable title if it exists otherwise "Example Website"
The x-slot allows you to overlayout that.

5. tailwindcss. Install via npm
<code>
npm install tailwindcss
</code>
So... an aside, you will see warnings on npm if you are pulling from the http vs https registry. You can configure your registry in npm

<code>
npm config set registry https://registry.npmjs.org/
</code>

6. I use emacs... but all the samples use code visual studio
Download from https://code.visualstudio.com/Download

7. models, controllers and crud oh my.

So by default the laravel resource code for controllers doesn't handle POSTs well.

So - you make a model
php artisan make:model Product
php artisan make:controller ProductController --resource

So if you use
Router::resource("product", ProductController::class);
you can't just use POST from your html form post.
from the docs its supposed to be PUT method, but PUT method makes the fields passed as a url - which means you hit a limitation method on requests where the content is over a certain limit.

So vendor/laravel/framework/src/Illuminate/Routing/ResourceRegistrar.php line 493, add "POST" in the methods that can match.

So now you can use POST. really stupid.. I ended up having to chance down where it matches the code in laravel. 2 1/2 hours later.

Unbelievable the number of simple projects that purport to show a simple crud example in laravel. Half of them do not have the routing information provided. Some clearly are from previous versions of laravel.

It would be really nice if it wrote the actual functions in the controller when you do --resource