Laravel - Forms

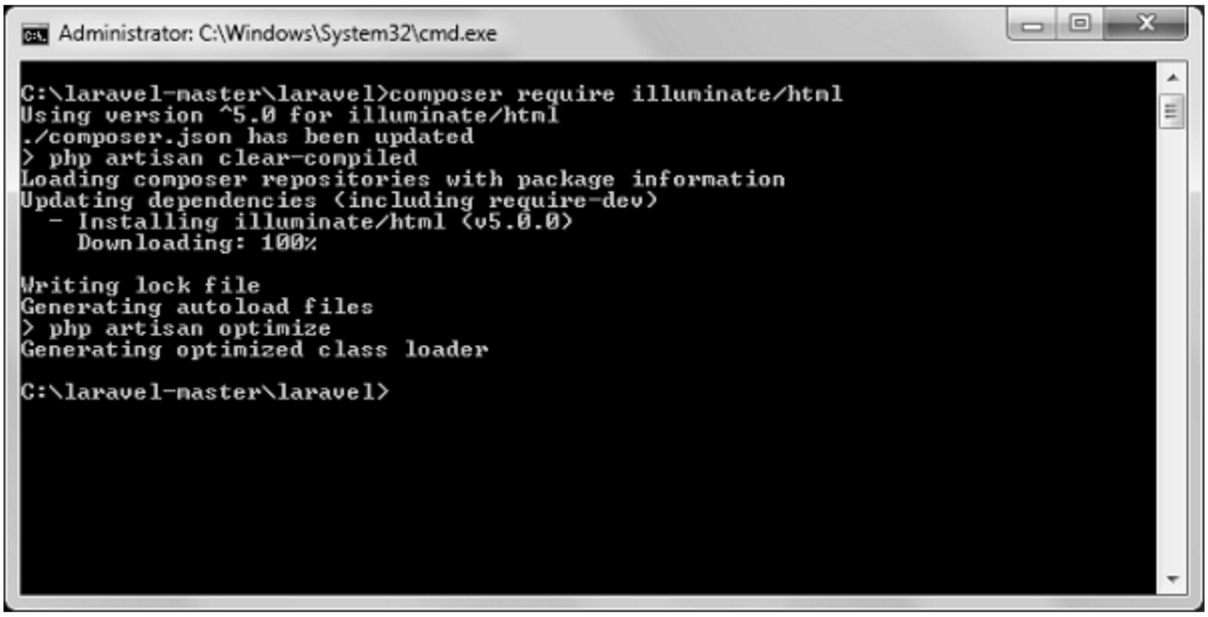
Laravel provides various in built tags to handle HTML forms easily and securely. All the major elements of HTML are generated using Laravel. To support this, we need to add HTML package to Laravel using composer.

Example 1

Step 1 − Execute the following command to proceed with the same.

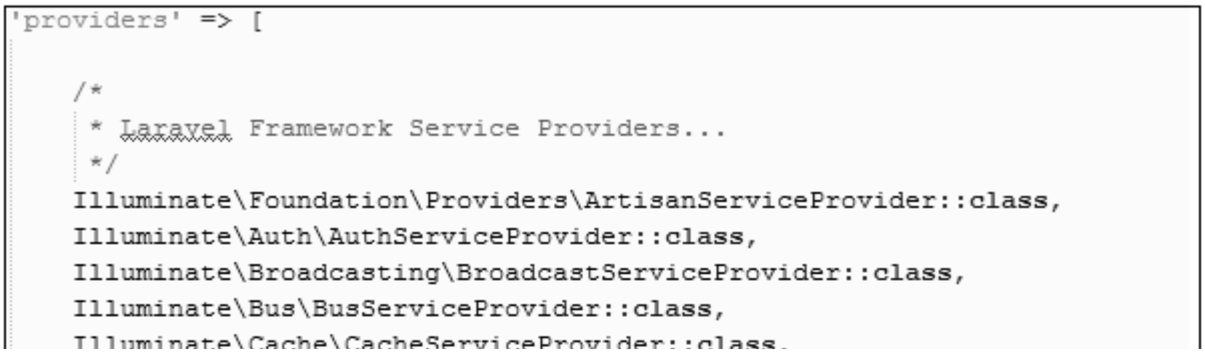
```
composer require illuminate/html
```

Step 2 − This will add HTML package to Laravel as shown in the following image.

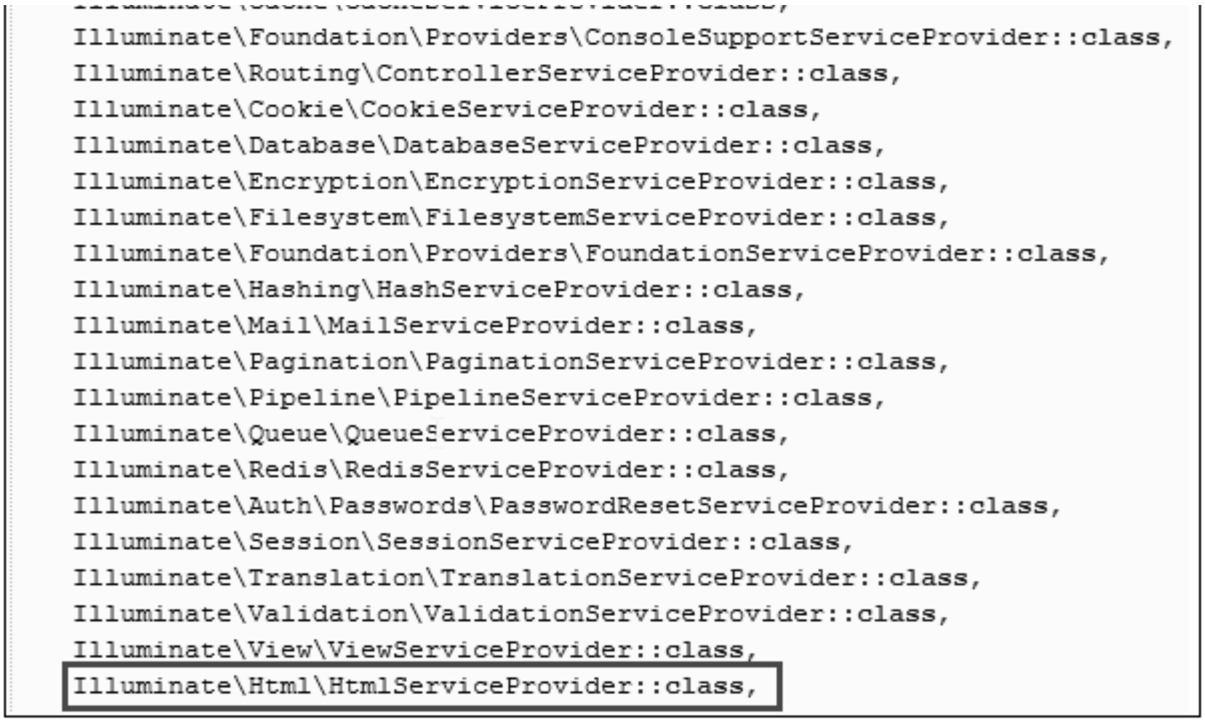


```
C:\laravel-master\laravel>composer require illuminate/html
Using version ^5.0 for illuminate/html
./composer.json has been updated
> php artisan clear-compiled
Loading composer repositories with package information
Updating dependencies (including require-dev)
  - Installing illuminate/html (v5.0.0)
    Downloading: 100%

Writing lock file
Generating autoload files
> php artisan optimize
Generating optimized class loader

C:\laravel-master\laravel>
```

Step 3 − Now, we need to add the package shown above to Laravel configuration file which is stored at **config/app.php.** Open this file and you will see a list of Laravel service providers as shown in the following image. Add HTML service provider as indicated in the outlined box in the following image.

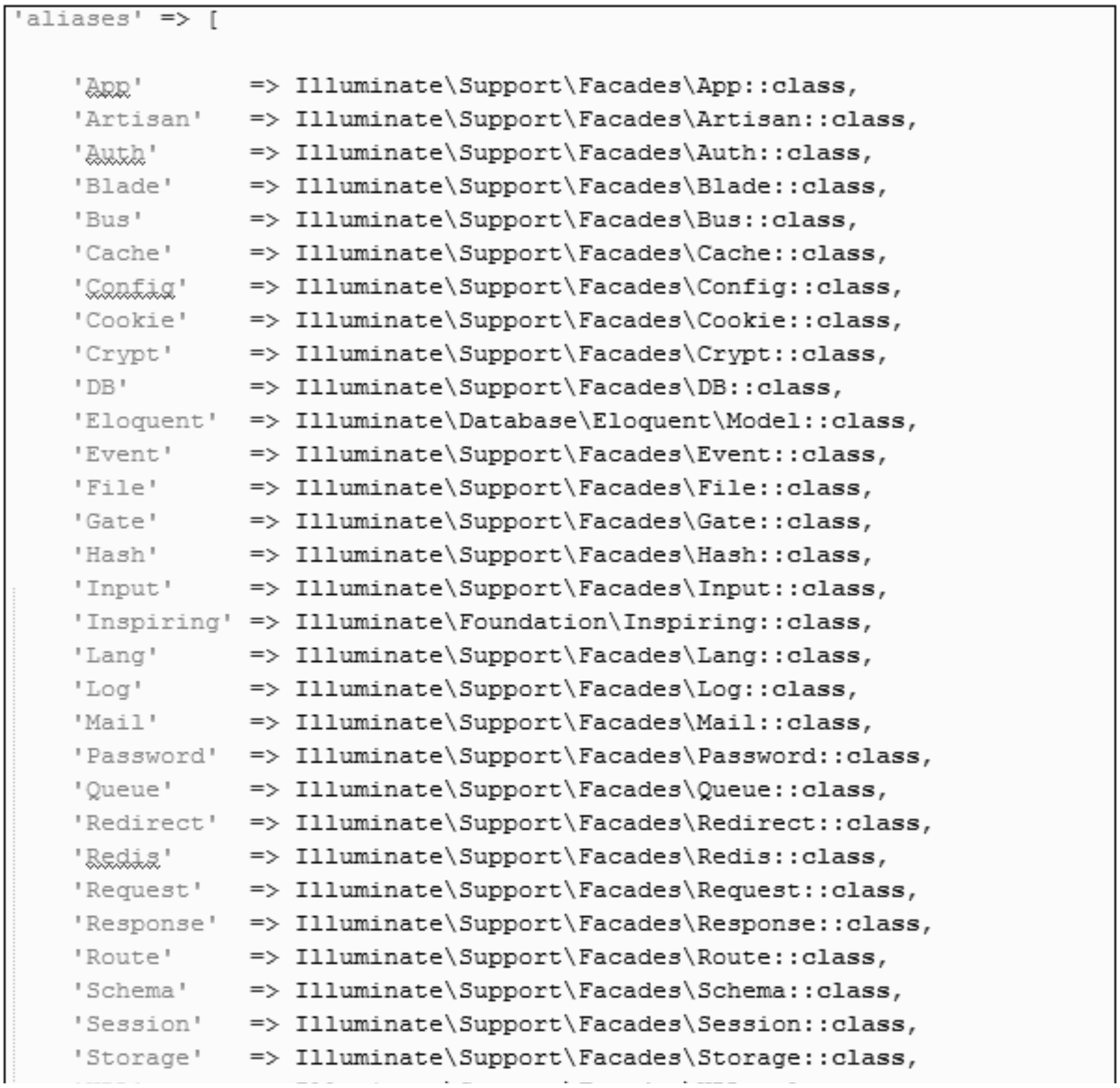


```
'providers' => [

    /*
     * Laravel Framework Service Providers...
     */
    Illuminate\Foundation\Providers\ArtisanServiceProvider::class,
    Illuminate\Auth\AuthServiceProvider::class,
    Illuminate\Broadcasting\BroadcastServiceProvider::class,
    Illuminate\Bus\BusServiceProvider::class,
    Illuminate\Cache\CacheServiceProvider::class,
```

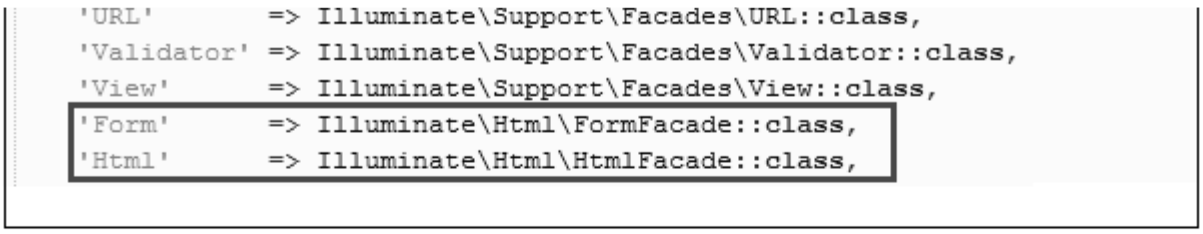


```
Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
Illuminate\Routing\ControllerServiceProvider::class,
Illuminate\Cookie\CookieServiceProvider::class,
Illuminate\Database\DatabaseServiceProvider::class,
Illuminate\Encryption\EncryptionServiceProvider::class,
Illuminate\Filesystem\FilesystemServiceProvider::class,
Illuminate\Foundation\Providers\FoundationServiceProvider::class,
Illuminate\Hashing\HashServiceProvider::class,
Illuminate\Mail\MailServiceProvider::class,
Illuminate\Pagination\PaginationServiceProvider::class,
Illuminate\Pipeline\PipelineServiceProvider::class,
Illuminate\Queue\QueueServiceProvider::class,
Illuminate\Redis\RedisServiceProvider::class,
Illuminate\Auth\Passwords\PasswordResetServiceProvider::class,
Illuminate\Session\SessionServiceProvider::class,
Illuminate\Translation\TranslationServiceProvider::class,
Illuminate\Validation\ValidationServiceProvider::class,
Illuminate\View\ViewServiceProvider::class,
Illuminate\Html\HtmlServiceProvider::class,
```

Step 4 − Add aliases in the same file for HTML and Form. Notice the two lines indicated in the outlined box in the following image and add those two lines.



```
'aliases' => [


    'App'       => Illuminate\Support\Facades\App::class,
    'Artisan'   => Illuminate\Support\Facades\Artisan::class,
    'Auth'      => Illuminate\Support\Facades\Auth::class,
    'Blade'     => Illuminate\Support\Facades\Blade::class,
    'Bus'       => Illuminate\Support\Facades\Bus::class,
    'Cache'     => Illuminate\Support\Facades\Cache::class,
    'Config'    => Illuminate\Support\Facades\Config::class,
    'Cookie'    => Illuminate\Support\Facades\Cookie::class,
    'Crypt'     => Illuminate\Support\Facades\Crypt::class,
    'DB'        => Illuminate\Support\Facades\DB::class,
    'Eloquent'  => Illuminate\Database\Eloquent\Model::class,
    'Event'     => Illuminate\Support\Facades\Event::class,
    'File'      => Illuminate\Support\Facades\File::class,
    'Gate'      => Illuminate\Support\Facades\Gate::class,
    'Hash'      => Illuminate\Support\Facades\Hash::class,
    'Input'     => Illuminate\Support\Facades\Input::class,
    'Inspiring' => Illuminate\Foundation\Inspiring::class,
    'Lang'      => Illuminate\Support\Facades\Lang::class,
    'Log'       => Illuminate\Support\Facades\Log::class,
    'Mail'      => Illuminate\Support\Facades\Mail::class,
    'Password'  => Illuminate\Support\Facades\Password::class,
    'Queue'     => Illuminate\Support\Facades\Queue::class,
    'Redirect'  => Illuminate\Support\Facades\Redirect::class,
    'Redis'     => Illuminate\Support\Facades\Redis::class,
    'Request'   => Illuminate\Support\Facades\Request::class,
    'Response'  => Illuminate\Support\Facades\Response::class,
    'Route'     => Illuminate\Support\Facades\Route::class,
    'Schema'    => Illuminate\Support\Facades\Schema::class,
    'Session'   => Illuminate\Support\Facades\Session::class,
    'Storage'   => Illuminate\Support\Facades\Storage::class,
```


```
'URL'       => Illuminate\Support\Facades\URL::class,
'Validator' => Illuminate\Support\Facades\Validator::class,
'View'      => Illuminate\Support\Facades\View::class,
'Form'      => Illuminate\Html\FormFacade::class,
'Html'      => Illuminate\Html\HtmlFacade::class,
```

Step 5 − Now everything is setup. Let’s see how we can use various HTML elements using Laravel tags.

Opening a Form

```
{{ Form::open(array('url' => 'foo/bar')) }}
  //
{{ Form::close() }}
```

Generating a Label Element

```
echo Form::label('email', 'E-Mail Address');
```

Generating a Text Input

```
echo Form::text('username');
```

Specifying a Default Value

```
echo Form::text('email', 'example@gmail.com');
```

Generating a Password Input

```
echo Form::password('password');
```

Generating a File Input

```
echo Form::file('image');
```

Generating a Checkbox Or Radio Input

```
echo Form::checkbox('name', 'value');
echo Form::radio('name', 'value');
```

Generating a Checkbox Or Radio Input That Is Checked

```
echo Form::checkbox('name', 'value', true);
echo Form::radio('name', 'value', true);
```

Generating a Drop-Down List

```
echo Form::select('size', array('L' => 'Large', 'S' => 'Small'));
```

Generating A Submit Button

```
echo Form::submit('Click Me!');
```

Example 2

Step 1 − Copy the following code to create a view called

resources/views/form.php.

resources/views/form.php

```
<html>
   <body>
      
      <?php
         echo Form::open(array('url' => 'foo/bar'));
            echo Form::text('username','Username');
            echo '<br/>';
            
            echo Form::text('email', 'example@gmail.com');
            echo '<br/>';
            echo Form::password('password');
            echo '<br/>';

            echo Form::checkbox('name', 'value');
            echo '<br/>';

            echo Form::radio('name', 'value');
            echo '<br/>';

            echo Form::file('image');
            echo '<br/>';

            echo Form::select('size', array('L' => 'Large', 'S' => '
            echo '<br/>';

            echo Form::submit('Click Me!');
         echo Form::close();
      ?>

   </body>
</html>
```

Step 2 − Add the following line in **app/Http/routes.php** to add a route for view form.php

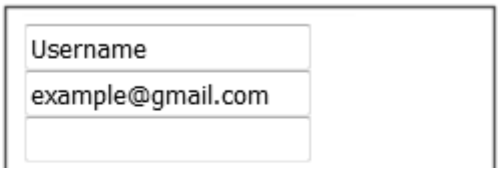
app/Http/routes.php

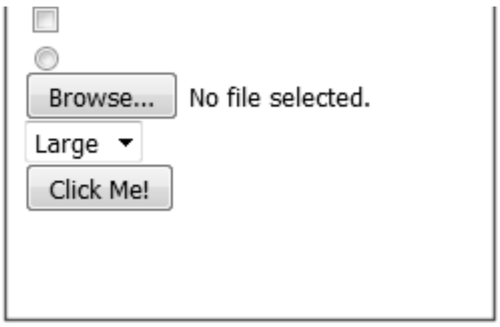
```
Route::get('/form',function() {
   return view('form');
});
```

Step 3 − Visit the following URL to see the form.

```
http://localhost:8000/form
```

Step 4 − The output will appear as shown in the following image.


Username
example@gmail.com


Browse...
No file selected.
Large
Click Me!