



## دورة مجانية لتعلم 5 Laravel بالعربي

الكاتب والمبرمج: Safaa Al-Hayali

تاريخ النشر: 4/17/15

الحقوق محفوظة لموقع Safaaalhayali.com

# بسم الله الرحمن الرحيم

## دورة تعلم Laravel 5

قررت عمل دورة تعليمية منهجية لتعلم الLaravel Framework التي تعتبر احدث Framework للPHP لعدم توافر مصادر كافية لها على الأنترنت ولصعوبة التعامل معها, الدورة مؤلفة من 12 جزء أبتداءً من الجزء الأول لتعلم كيف تحمل وتنصب الLaravel الى غاية مرحلة الDeployment, المدة لأنتهاء الدورة 3 اسابيع وفي كل اسبوع 4 اجزاء.

في هذا الكتاب سأنشر الجزء الثامن لهذه الدورة وباقي الأجزاء تستطيع مشاهدتها على الموقع التالي:

[www.safaaalhayali.com/search/label/Laravel5](http://www.safaaalhayali.com/search/label/Laravel5)

سبب نشرها على شكل اجزاء في الموقع اعلاه, لأن في كل جزء ربما تسأل او تستفسر عن موضوع معين فلهذا لسهولة التعلم جعلتها على شكل مقالات وفي كل مقالة يوجد مربع حوار للتفاعل معي ومع باقي المطورين.....

وبالتوفيق لكم في تعلم الLaravel 5

## الجزء 8: استخدام ال Form component

في هذا الجزء سنتعلم كيف نستخدم ال Form component طريقة تنزيله وتهيئته والعمل معه وإضافة حقول داخل ال Form وكيف نعمل ال Form Model Binding اي يمكن تضمين الكائن مع ال Form وإضافة معلوماته داخل حقول ال Form بتطبيق عملي على صفحة تعديل الخبر edit

ال Laravel ذكية لتمييز بين ال parameter المزدودة في الدالة, حيث ال order غير مهم مثلاً:

عند إنشاء توجيه لل URL التالية في ملف ال routes.php:

```
get('news/{slug}', 'NewsContorller@show');
```

وفي ملف ال NewsController.php في دالة ال show تلاحظ ترتيب قيم ال parameter غير مهم:

```
Public function show($slug , news $news){}
```

```
Public function show(news $news , $slug){}
```

وكذلك من مميزات ال laravel امكانية عمل method injection للكائن, كما تلاحظ في الدالتين اعلاه استخدمنا ال Eloquent Model الذي هو `news $news` ك parameter , وكذلك سنستخدمه عند انشاء دالة جديدة لتعديل الأخبار اي سيصبح لدينا ثلاث دوال تستخدم نفس ال ('News') ال Eloquent Model عن طريق عمل Method Injection للدالة

بهذا الشكل:

```
Public function index(news $news){}
```

```
Public function show(news $news , $slug){}
```

```
Public function edit(news $news , $slug){}
```

يمكن تقليص الكود عن طريق انشاء دالة انشائية لل Class نفسه, بدل استدعاء ال ('News') ال Eloquent Model ثلاث مرات , سنستدعيه مرة واحدة فقط في دالة ال `__construct` كالتالي:

```
Private $news;
```

```
Public function index(news $news){
```

```
    $this->news = $news
```

```
}
```

وبهذا سنحذف ال Eloquent Model('News') من الدوال الثلاث وسنشير له فقط داخل الدالة لأنه سبق وتم انشاءه في دالة ال \_\_construct كالأتي:

```
Public function index(){
$this->news->get();
}
Public function show($slug){
$this->news->whereSlug($slug)->first();
}
Public function edit($slug){
$this->news->whereSlug($slug)->first();
}
```

والآن سأقوم بإنشاء دالة لتعديل الأخبار, اول شيء يجب عمل توجيه لتلك الدالة في ملف routes.php:

```
get('news/{slug}/edit', 'NewsContorller@edit');
```

وبعدها ال routes.php سيقوم بتوجيهنا لدالة edit الموجودة في ملف NewsController , سأنشأ تلك الدالة كالأتي:

```
Public function edit($slug){
$news = $this->news->whereSlug($slug)->first();
return view('news.edit', compact('news'));
}
```

وال Controller بدوره سيسترجع الواجهة edit مع تمرير كائن ال news المراد التعديل عليه.

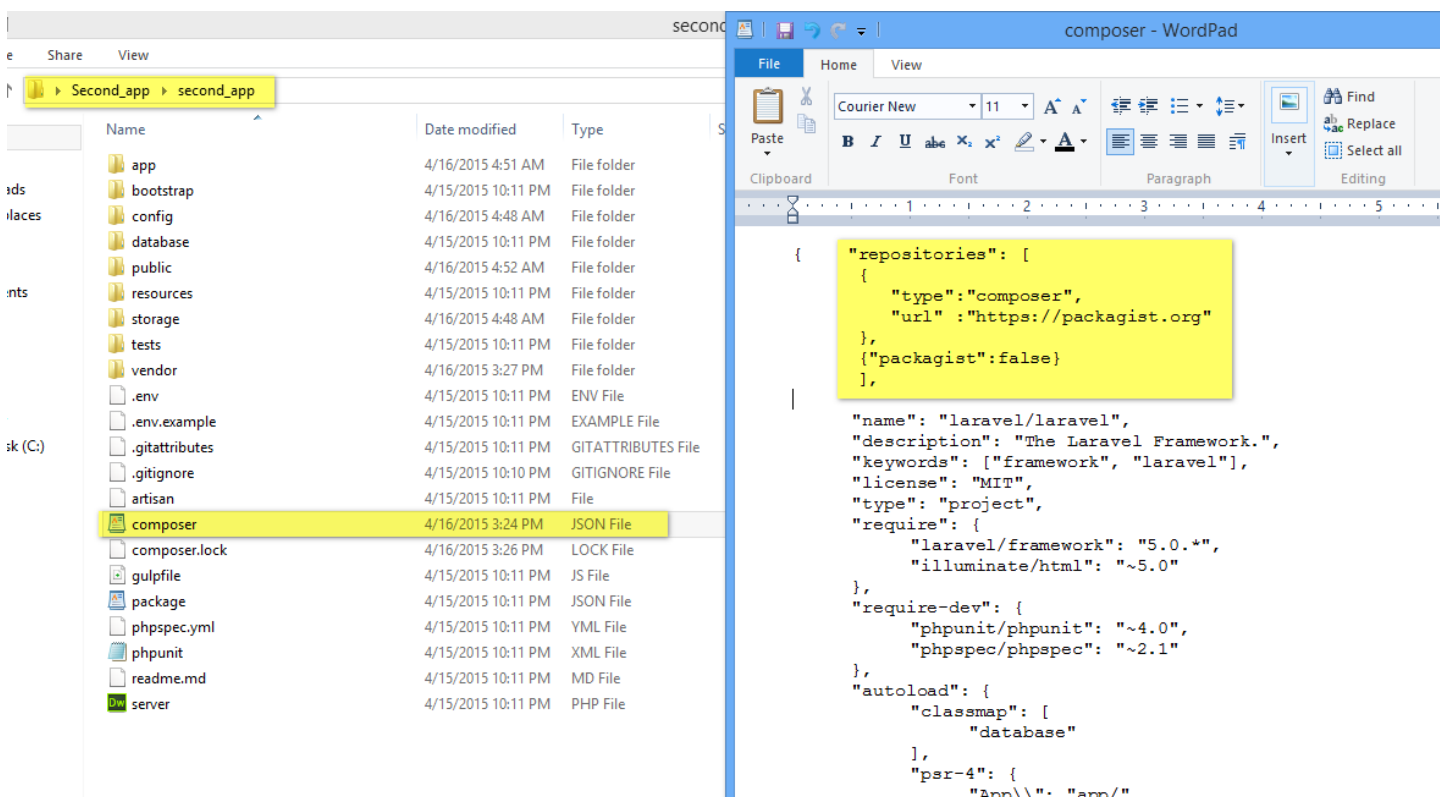
والآن للجزء الهام لهذا اليوم, انشاء ال Form الذي عن طريقه سنعدل بيانات الخبر, يمكن ذلك بالطريقة العادية والمباشرة عند طريق كتابة كود ال HTML الخاص بال Form , ولكن بما اننا نستعمل Laravel يمكن استخدام عنصر ال Form الذي

سيسهل طريقة كتابة الكود, قبل استخدام ال Form Component يتوجب علينا ادخال ال Package الخاص بذلك العنصر وتهيئته للعمل معه,

لأدراج ذلك العنصر عن طريق استخدام ال Composer:

**ملاحظة:** قبل كل شيء يوجد خطأ باستخدام ال Composer في انظمة ال Windows عند طلب ال Package من <http://packgist.org> يحدث فشل بفتح الأتصال عن طريق http الا عن طريق https يمكن تعديل ذلك بكتابة الكود التالي في ملف ال composer.json الموجود في ملف المشروع:

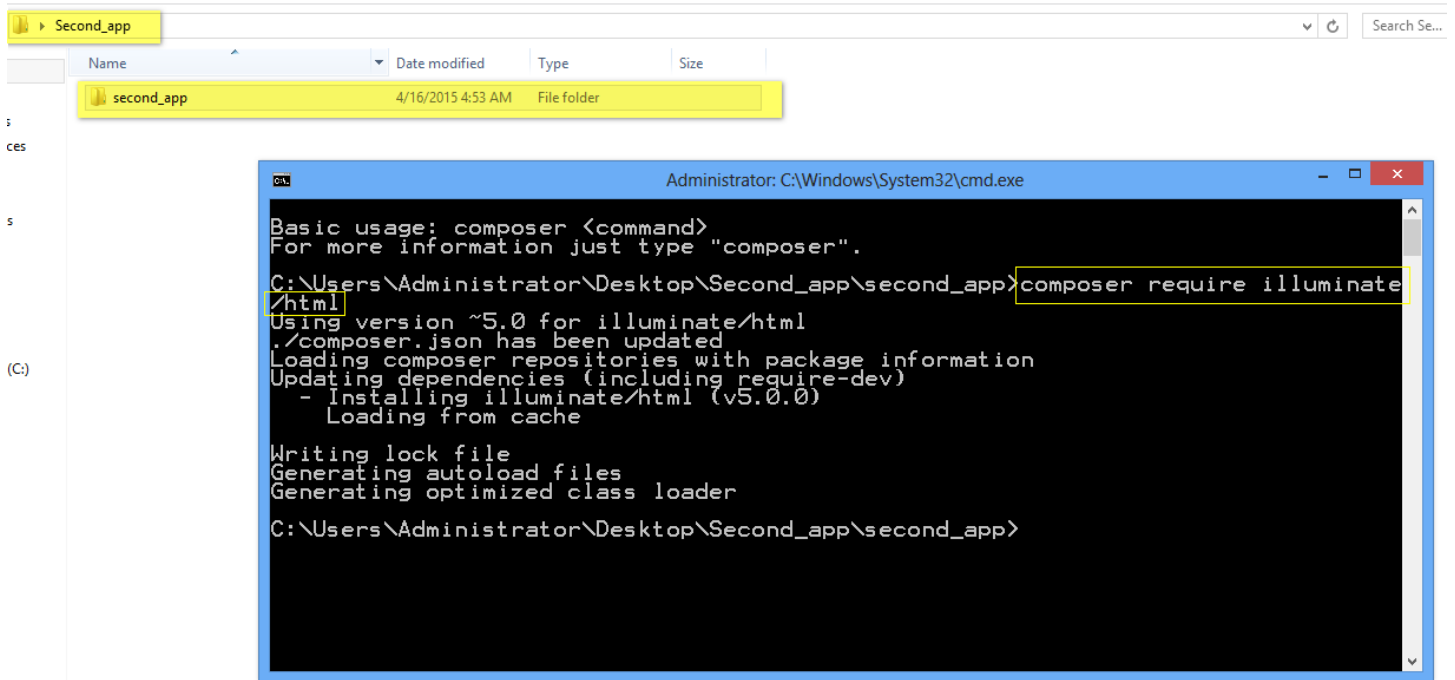
```
"repositories": [
  {
    "type": "composer",
    "url" : "https://packgist.org"
  },
  {"packagist": false}
],
```



وبعد ذلك يمكنك طلب ال Form component عن طريق كتابة الأمر التالي في نافذة الأوامر وبأستخدام ال composer:

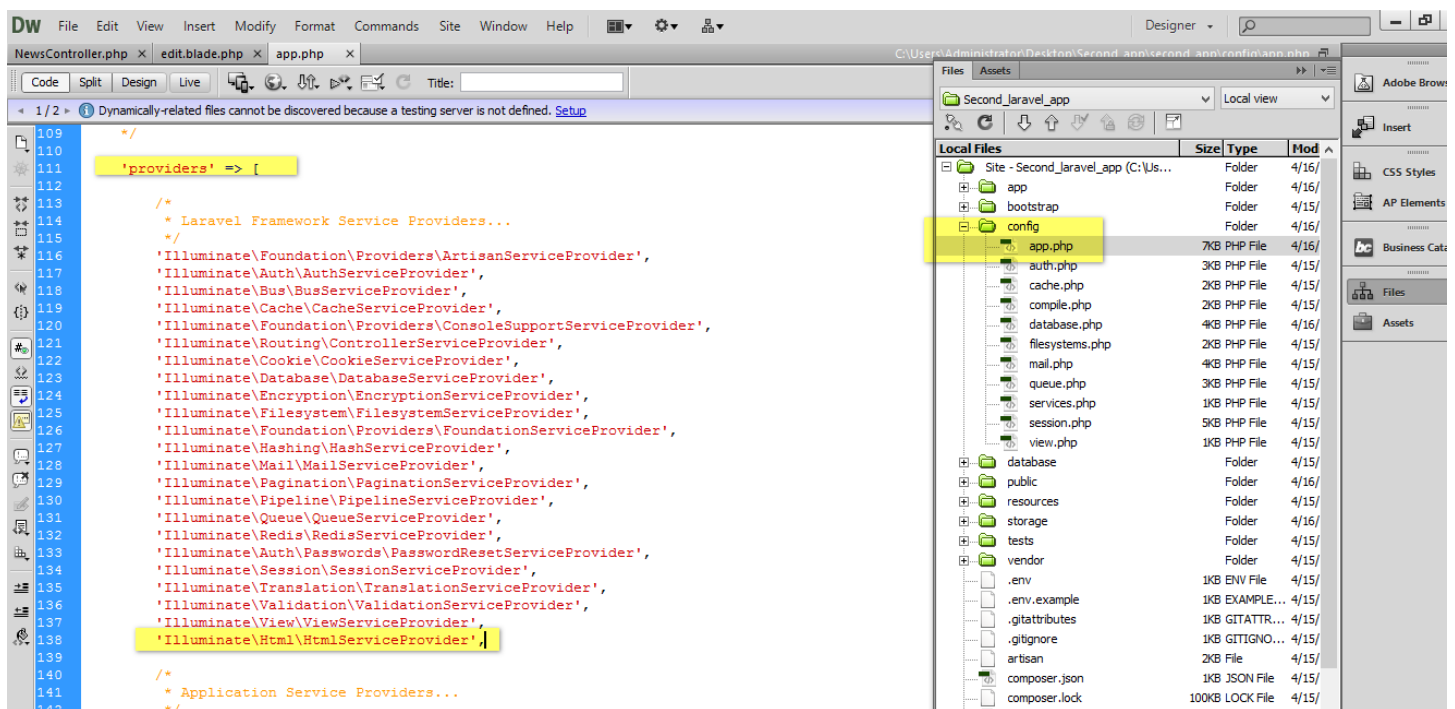
```
C:\Users\Administrator\Desktop\Second_app\second_app>c  
omposer require illuminate  
/html
```

ملاحظة: يجب تطبيق الأمر اعلاه على الملف الذي يحتوي على ملفات المشروع لأن ال Form component سيدرج ضمن مجلد ال Vendor



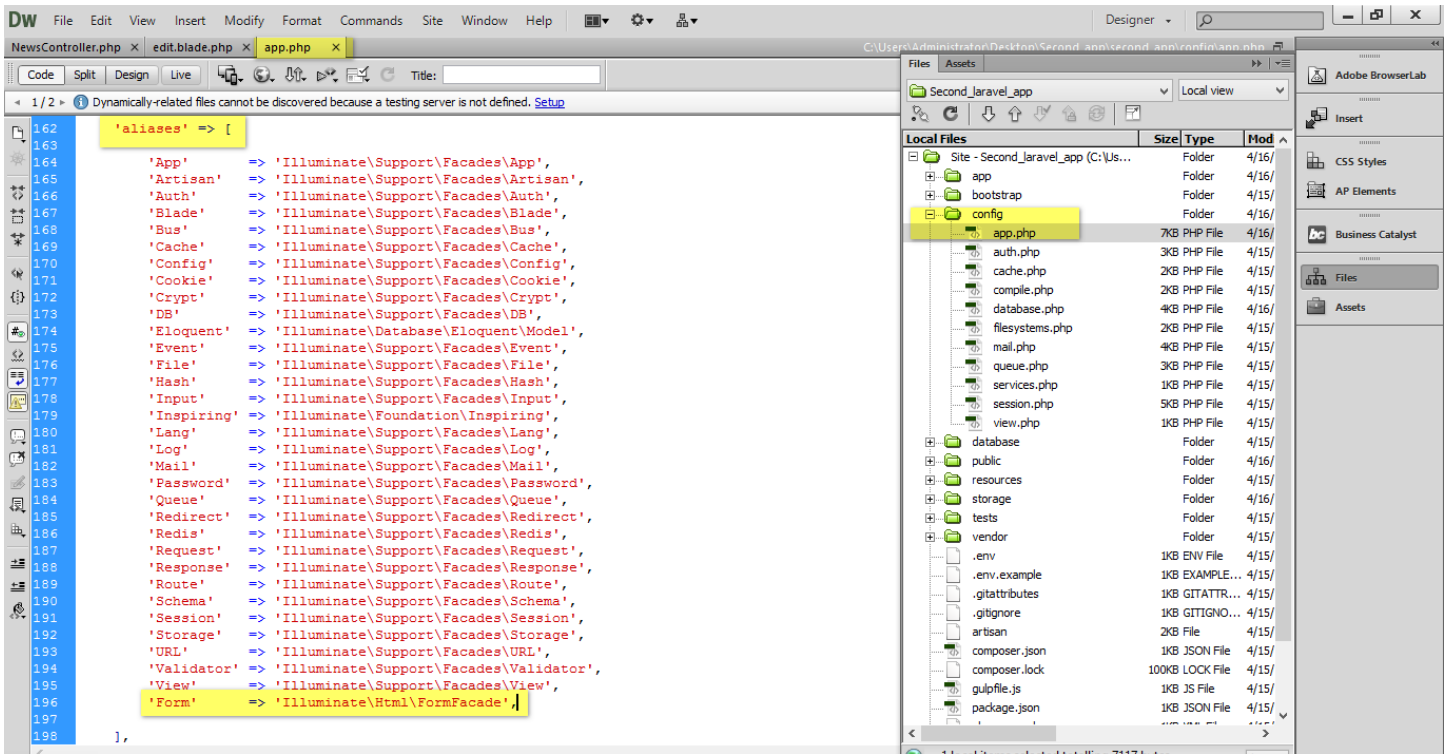
الخطوة الأخيرة الواجب عملها قبل استخدام ال Form component في ال Views وهي تهيئة ذلك العنصر للأستخدام عن طريق اضافة الأسطر البرمجية التالية في ملف config\app.php: ضمن ال Service Providers ادرج الكود التالي:

```
'Illuminate\Html\HtmlServiceProvider',
```



وضمن الأسماء المستعارة aliases للClasses ادخل الكود البرمجي التالي:

```
'Form' => 'Illuminate\Html\FormFacade',
```



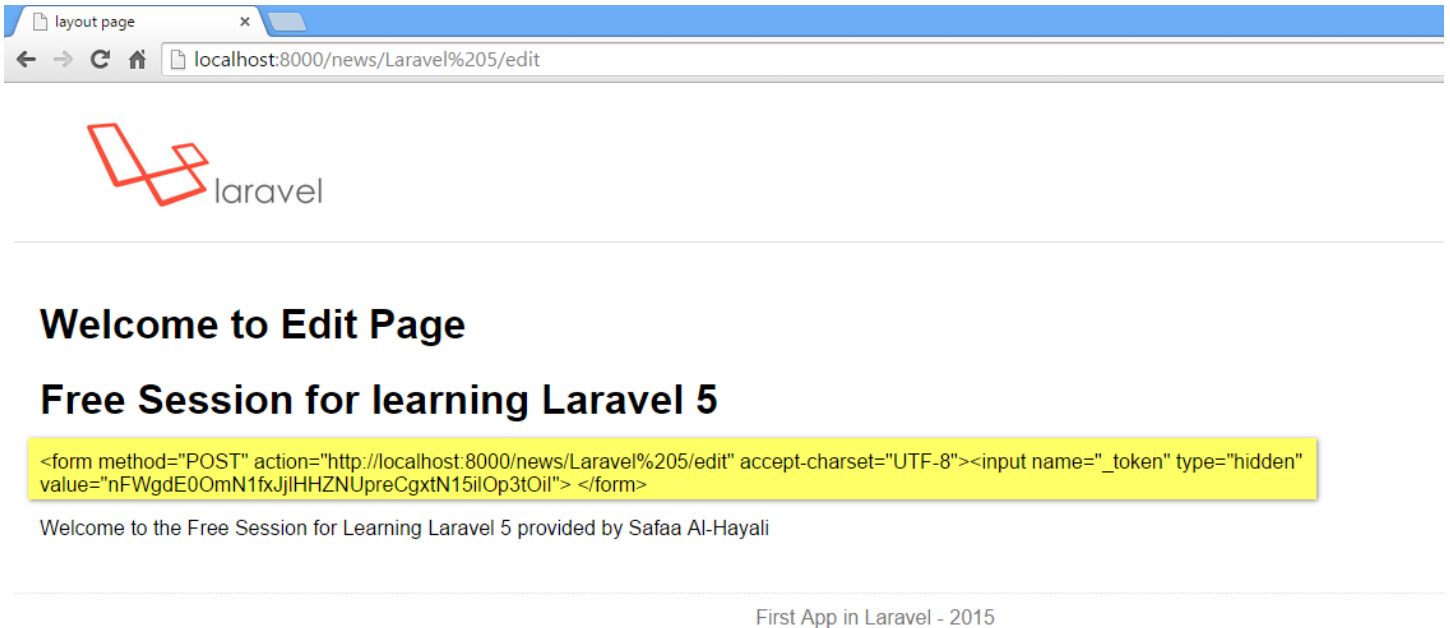
وبذلك انتهى تحميل ال Form Component ووقت تهيئته للعمل معه 😊

في الواجهة الخاصة بتعديل الأخبار edit.blade.php يمكن استخدام ال Form component بأبسط صورة كالآتي:

```
{{Form::open()}}
```

```
{{Form::close()}}
```

ولكن عند زيارة صفحة تعديل لأحد الأخبار يظهر التالي!!!



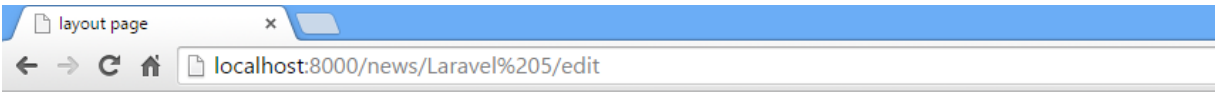
السبب لأن الأقواس المزدوجة تعمل على الهروب من HTML TAG لغرض امني , لكن في حالتنا هذه نريد ال HTML TAG أن تعمل وليس الهروب منها , يمكن ذلك بكتابة الكود اعلاه بالشكل التالي:

```
{!! Form::open() !!}
```

```
{!! Form::close() !!}
```

حيث استبدلت احد الأقواس بعلامتين تعجب معناها انا اعرف ماذا افعل لا تهرب من ال HTML TAG 😊  
النتيجة كالتالي:





## Welcome to Edit Page

## Free Session for learning Laravel 5

Welcome to the Free Session for Learning Laravel 5 provided by Safaa Al-Hayali

First App in Laravel - 2015

السؤال المحير الآن اين الFORM!؟

الجواب: بما ان الForm لا يحتوي على Fields مثل Text و الTextarea وغيرها فلا يتم اظهار شيء ولكن ماخلف الصفحة الكود موجود للForm:

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>layout page</title>
6 <link rel="stylesheet" href="http://localhost:8000/css/style.css" type="text/css"/>
7 </head>
8
9 <body>
10
11 <header>
12 
13 </header>
14
15 <div id="content">
16
17 <h1>Welcome to Edit Page</h1>
18 <h1>Free Session for learning Laravel 5</h1>
19 <form method="POST" action="http://localhost:8000/news/Laravel%205/edit" accept-charset="UTF-8"><input name="_token" type="hidden" value="nFwgdE00mN1fxJj1hHZMUpreCgxtN15i10p3toII">
20 </form>
21 <p>Welcome to the Free Session for Learning Laravel 5 provided by Safaa Al-Hayali</p>
22
23
24 </div>
25
26 <footer>First App in Laravel - 2015</footer>
27
28 </body>
29 </html>
30
```

والآن لأدراج العناصر في الForm مثل الtext والSubmit وغيرها يمكنك معرفة ذلك من الFormBuilder.php الموجود في المسار التالي:

C:\Users\Administrator\Desktop\Second\_app\second\_app\vendor\illuminate\html

لندرج بعض العناصر الرئيسية لتعديل احد اخبار الموقع:

نحتاج الى Text field لتعديل اسم الخبر title

نحتاج الى Textarea field لتعديل محتوى الخبر content

نحتاج الى Submit button لحفظ التغيير

يوجد شيء جداً مهم يجب معرفته وهو كيفية نضمن الكائن الذي يتوجب تعديله في الForm وهذا يسمى ب Form Model binding , الفائدة منه نستطيع من الكائن ادراج قيمه في الحقول,

والكود البرمجي لعمل ذلك في غاية السهولة فقط اشر الى استخدام الmodel:

```
{!! Form::model($news) !!}
```

**ملاحظة:** في الكود اعلاه سيتم تضمين الكائن في الForm وكذلك سيفتح ال Tag الخاص بعمل الForm

كما حددنا سابقاً الحقول المراد ادراجها في الForm

- Text field for title

```
{!! Form::text('title') !!}
```

- Textarea field for content

```
{!! Form::textarea('content') !!}
```

- Submit button

```
{!! Form::submit('Update') !!}
```

والكود البرمجي النهائي سيكون بالشكل التالي:

The screenshot shows the Adobe Dreamweaver interface. The main editor window displays the code for a Blade template named `edit.blade.php`. The code is as follows:

```
1 @extends('layout')
2
3
4
5 @section('content')
6 <h1>Welcome to Edit Page</h1>
7 <h1>{{ $news->title }}</h1>
8 {!! Form::model($news) !!}
9 {!! Form::text('title') !!}
10 <br />
11 {!! Form::textarea('content') !!}
12 <br />
13 {!! Form::submit('Update') !!}
14 <br />
15 {!! Form::close() !!}
16 <p>{{ $news->content }}</p>
17
18 @stop
```

The code is highlighted in yellow. The file explorer on the right shows the project structure for `Second_laravel_app`. The `news` folder is selected, and the `edit.blade.php` file is highlighted. The file explorer table is as follows:

Local Files	Size	Type	Mod
Site - Second_laravel_app (C:\Us...		Folder	4/16/
app		Folder	4/16/
bootstrap		Folder	4/15/
config		Folder	4/16/
database		Folder	4/15/
public		Folder	4/16/
resources		Folder	4/15/
assets		Folder	4/15/
lang		Folder	4/15/
views		Folder	4/16/
auth		Folder	4/15/
emails		Folder	4/15/
errors		Folder	4/15/
news		Folder	4/16/
edit.blade.php	1KB	PHP File	4/16/
index.blade.php	1KB	PHP File	4/9/2
show.blade.php	1KB	PHP File	4/9/2
vendor		Folder	4/15/
app.blade.php	3KB	PHP File	4/15/
home.blade.php	1KB	PHP File	4/15/
layout.blade.php	1KB	PHP File	4/6/2
welcome.blade.php	1KB	PHP File	4/15/
storage		Folder	4/16/
tests		Folder	4/15/
vendor		Folder	4/15/
.env	1KB	ENV File	4/15/
.env.example	1KB	EXAMPLE...	4/15/
.gitattributes	1KB	GITATTR...	4/15/
.gitignore	1KB	GITIGNO...	4/15/
artisan	2KB	File	4/15/

ماذا سيحدث عن زيارة صفحة تعديل احد الأخبار؟! لنجرب ذلك:



## Welcome to Edit Page

Free Session for learning Laravel 5

**\$news->title  
name = 'title'**

Free Session for learning Lar

Welcome to the Free Session for Learning Laravel 5  
provided by Safaa Al-Hayali

**\$news->content  
name='content'**

Update

Welcome to the Free Session for Learning Laravel 5 provided by Safaa Al-Hayali

First App in Laravel - 2015

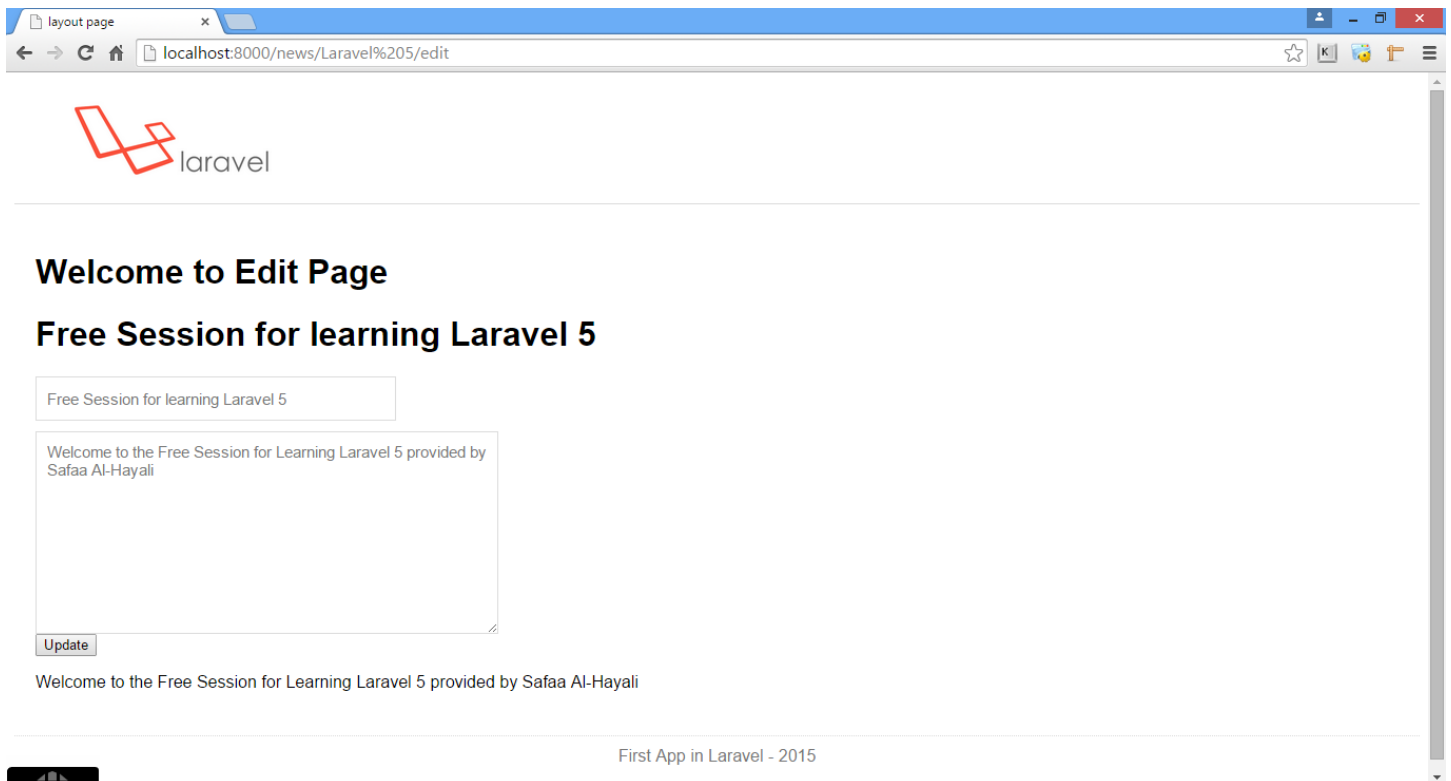
سنرى بأن الحقول ملئت بصورة تلقائية بخصائص الكائن الذي ضمن داخل الForm والسبب لأن خصائص الكائن له تسميه حقول نفس تسمية الحقول name في الForm

في اي حقل من حقول الForm مثل الText اول قيمة تكون هي اسم الحقل, ثاني قيمة تكون قيمة الحقل لاحظ اذا كتبت قيمة معينة ستستبدل قيمة الكائن بهذه القيمة في هذه الحالة اجعلها null , ثالث قيمة خصائص properties او سمات الHTML 5 الattributes مثلأ يمكن اضافة class لهذا الحقل في القيمة الثالثة كالآتي:

```
{!! Form::text('title',null,['class'=>'form_txt']) !!}  
{!! Form::textarea('content',null,['class'=>'form_txtA']) !!}
```

قمت بتعديل الStyle :

```
.form_txt, .form_txtA{  
    padding: 10px;  
    border: 1px solid #ddd;  
    font-family:'Arial','Helvetica';  
    color:#777;  
    font-size:15px;  
    display:block;  
}  
.form_txt{  
    width: 320px;  
    height: 20px;  
    margin-bottom: 10px;  
}  
.form_txt:focus, .form_txtA:focus{  
color:#000;  
}
```



بقى لدينا شيء واحد فقط وهو الى اين سنتجه اذا ضغطنا على زر update!؟

طبعاً في ال Form component ال URL الافتراضية ستكون موجهة لصفحة ذاتها وبطريقة ال POST

لتعديل ال URL مثلاً عند الضغط على زر ال update اريد توجيهي للخبر ذاته مع تحديث الخبر المعدل ونوع الطريقة request method تكون PATCH

الجواب سيكون في غاية السهولة فقط عدل قيمتين عند تضمين ال Model مع ال Form كالأتي:

```
{!!Form::model($news, ['url' => 'news/' . $news->slug, 'method' => 'PATCH']) !!}
```

حيث ان ال URL تشير الى الخبر ذاته كالأتي: /news/{slug} هذا التوجيه موجود داخل ملف ال routes.php ولكن طريقة طلبه Request Method تكون GET وليس PATCH!؟ حيث اذا ضغطت على الزر سيظهر خطأ عن وجود توجيه ال URL مع طريقة طلبه ب PATCH الغير موجودة في ال routes.php

الحل بأضافة هذا التوجيه في ملف ال routes.php كالأتي:

```
patch('news/{slug}', 'NewsController@update');
```

ملاحظة: المتصفحات لا تعرف ال Request method من نوع patch والlaravel يعمل على انشاء حقل خفي مع إعطاء قيمة الPATCH كالتالي:

```
<input name="_method" type="hidden" value="PATCH">
```

والآن تم انشاء التوجيه وعمل الForm بقي شيء واحد وهو عمل الدالة في الNewsController.php التي ستطلب قيم الكائن الجديدة من الحقول وتخزينها كقيم جديدة لذلك الكائن.

سأقوم بإنشاء الدالة في ملف NewsController وسأسترجع قيمة توضح ان توجيهي صحيح عند الضغط على زر update:

```
public function update($slug)
{
    return 'yes it\'s working !!';
}
```

والآن لمعرفة كيفية طلب القيم من حقول الForm فقط للتوضيح سأستخدم دالة الdebug :

Die in dump dd():

```
public function update($slug)
{
    dd(\Request::all());
    dd(\Request::input());
    dd(\Request::get('title'));
}
```

حيث عند استخدام ال request class مع دالة الall ستسترجع كل القيم داخل الform بمصفوفة KEY => VALUE

وعند استخدام ال request class مع دالة الinput ستسترجع كل قيم الإدخال كما بمصفوفة KEY => VALUE

وعند استخدام ال request class مع دالة الget ستسترجع قيمة واحدة فقط المحدد اسمها name

الغرض من استخدام الـ Request class اعلاه هو لمعرفة كيفية استخدامه لأسترجاع القيم من الـ form في الدالة التالي،  
والآن لنكمل كود دالة الـ update في الـ NewsController لاحظ عند الضغط على الـ URL المحددة تكون بهذا الشكل:

`News/{slug}`

قيمة الـ slug قد تم تمريرها للـ routes.php والذي بدوره اما ان يقوم بتحويلها لدالة الـ update مباشرة او قد نعمل Route model binding ونسترجع الكائن مباشرة ولكن للسهولة سأقوم بتحويل تلك القيمة مباشرة اي ان في دالة الـ update يجب تحديد parameter تستقبل الـ slug وايضاً عمل method injection للـ Request class الذي سيسترجع قيم من الـ Form وبهذا يصبح لدينا قيمتين داخل دالة الـ update

```
public function update($slug, Request $request)
{
}
```

ثاني شيء نسترجع الكائن حسب قيمة الـ Slug والذي سنعدل حقوله لاحقاً:

```
$news = $this->news->whereSlug($slug)->first();
```

ملاحظة: لا تنسى عمل الـ \_\_construct:

```
private $news;
public function __construct(news $news)
{
    return $this->news = $news;
}
```



والجزء الأخير هو بتعديل كل حقل للكائن حسب المسترجع عن طريق الكائن \$request كالاتي وبعدها تحويلنا لل news :page

```
$news->title = $request->get('title');  
$news->content = $request->get('content');  
$news->save();  
  
return redirect('news');
```

يصبح الكود النهائي لدالة الupdate:

```
public function update($slug, Request $request)  
{  
    $news = $this->news->whereSlug($slug)->first();  
    $news->title = $request->get('title');  
    $news->content = $request->get('content');  
    $news->save();  
  
    return redirect('news');
```

Free Sessions for learning Laravel 5

Welcome to the Free Session for Learning [Laravel](#) 5 provided by [Safaa Al-Hayali](#)

I think [Laravel](#) 5 part 8 is little advance?! Yes Or No?! :(

Update

Welcome to the Free Session for Learning Laravel 5 provided by Safaa Al-Hayali

First App in Laravel - 2015

Free Sessions for learning Laravel 5

Welcome to the Free Session for Learning Laravel 5 provided by Safaa Al-Hayali I think [Laravel](#) 5 part 8 is little advance?! Yes Or No?! :(

First App in Laravel - 2015

كما لاحظت في دالة الupdate قمت بتحديد كل حقل من الكائن \$news مع تجديده بقيمة مسترجعة من حقل الForm يوجد طريقة ادراج كل قيم الForm مع قيم الكائن عن طريق استخدام دالة fill:

```
$news->fill(['title'=>$request->get('title'),'content'=>$request->get('content')]);
```

## MassAssignmentException ولكن عن عمل تحديث لخبر معين سيظهر خطأ

سبب هذا الخطأ اننا نمرر كل الحقول دفعة واحدة لقاعدة البيانات وال Laravel تحمينا من هذا النوع من الأختراقات الذي قد يكون مصحوب بكود استرجاع بيانات او وضع بيانات حساسة!!!

ولحل ذلك قم بتحديد اي من الحقول التي ستعتبر fillable اي قابلة للملأ عن طريق اضافة الحقول التي تعتبر fillable في ملف news eloquent model:

```
protected $fillable = ['title','content'];
```

**ملاحظة:** اذا قم بتحديد الحقل title في الnews eloquent model ك fillable field اذاً باقي المعلومات التي سيدرجها المستخدم سيتم اهمالها فقط الحقل الذي مع اسم title سيتم ادراجه.

وكذلك يوجد طريقة اسهل لكتابة الكود اعلاه وهي:

```
$news->fill($request->input())->save();
```

وبهذا سيختصر كود دالة الupdate ليصبح بهذا الشكل:

```
public function update($slug, Request $request)
{
    $news = $this->news->whereSlug($slug)->first();
    $news->fill($request->input())->save();
    return redirect('news');
}
```

والآن ننهي الجزء لهذا اليوم بأضافة كود التحقق في واجهة الshow.blade.php ما اذا كان الكائن \$news يحتوي على معلومات واذا لم يحتوي على معلومات لا يتم اظهار كود ال HTML حوله:

```
@if($news->content)
```

```
<article class="news_content"> {!! nl2br($news->content) !!}
</article>
```

```
@endif
```

وكذلك سأضيف رابطين داخل اي صفحة اي خبر في show.blade.php:

```
<a href="/news">Back to News</a><br />
```

```
<a href="/news/{{ $news->slug }}/edit">Edit this News</a>
```

