

Laravel 大型项目系列教程（四）

一、前言

上节教程中实现了发布文章的功能,本节教程中将大概实现在首页和用户主页分页显示文章和标签列表、用户能够修改删除文章。

二、Let's go

1. 首页显示文章和标签列表

我们需要在首页显示文章和标签列表,修改 `views/index.blade.php` :

```
@extends('_layouts.default')

@section('main')
<div class="am-g am-g-fixed">
  <div class="am-u-md-8">
    @foreach ($articles as $article)
      <article class="blog-main">
        <h3 class="am-article-title blog-title">
          <a href="{{ URL::route('article.show', $article->id) }}">{{ $article->title }}</a>
        </h3>
        <h4 class="am-article-meta blog-meta">
          by <a href="#">{{ $article->user->nickname }}</a> posted on {{ $article->created_at->format('Y/m/d H:i') }} under
          @foreach ($article->tags as $tag)
            <a href="#" style="color: #fff;" class="am-badge am-badge-success am-radius">{{ $tag->name }}</a>
          @endforeach
        </h4>
        <div class="am-g">
          <div class="am-u-sm-12">
            @if ($article->summary)
              <p>{{ $article->summary }}</p>
            @endif
            <hr class="am-article-divider"/>
          </div>
        </div>
      </article>
    @endforeach
  </div>

  <div class="am-u-md-4 blog-sidebar">
    <br/>
    <div class="am-panel-group">
```

```

<section class="am-panel am-panel-default">
  <div class="am-panel-hd"><span class="am-icon-tags"></span> Tags</div>
  <ul class="am-list">
    @for ($i = 0, $len = count($tags); $i < $len; $i++)
      <li>
        <a href="#">{{ $tags[$i]->name }}
        @if ($i == 0)
          <span class="am-fr am-badge am-badge-danger am-round">{{ $tags[$i]->count }}</spa
n>
        @elseif ($i == 1)
          <span class="am-fr am-badge am-badge-warning am-round">{{ $tags[$i]->count }}</spa
n>
        @elseif ($i == 2)
          <span class="am-fr am-badge am-badge-success am-round">{{ $tags[$i]->count }}</sp
an>
        @else
          <span class="am-fr am-badge am-round">{{ $tags[$i]->count }}</span>
        @endif
      </a>
    </li>
  @endfor
</ul>
</section>
</div>
</div>
</div>
@stop

```

在 **custom.css** 中增加：

```

@media only screen and (min-width: 641px) {
  .blog-sidebar {
    font-size: 1.4rem;
  }
}
.blog-main {
  padding: 20px 0;}
.blog-title {
  margin: 10px 0 20px 0;}
.blog-meta {
  font-size: 14px;
  margin: 10px 0 20px 0;
  color: #222;}
.blog-meta a {
  color: #27ae60;}

```

修改 **routes.php**：

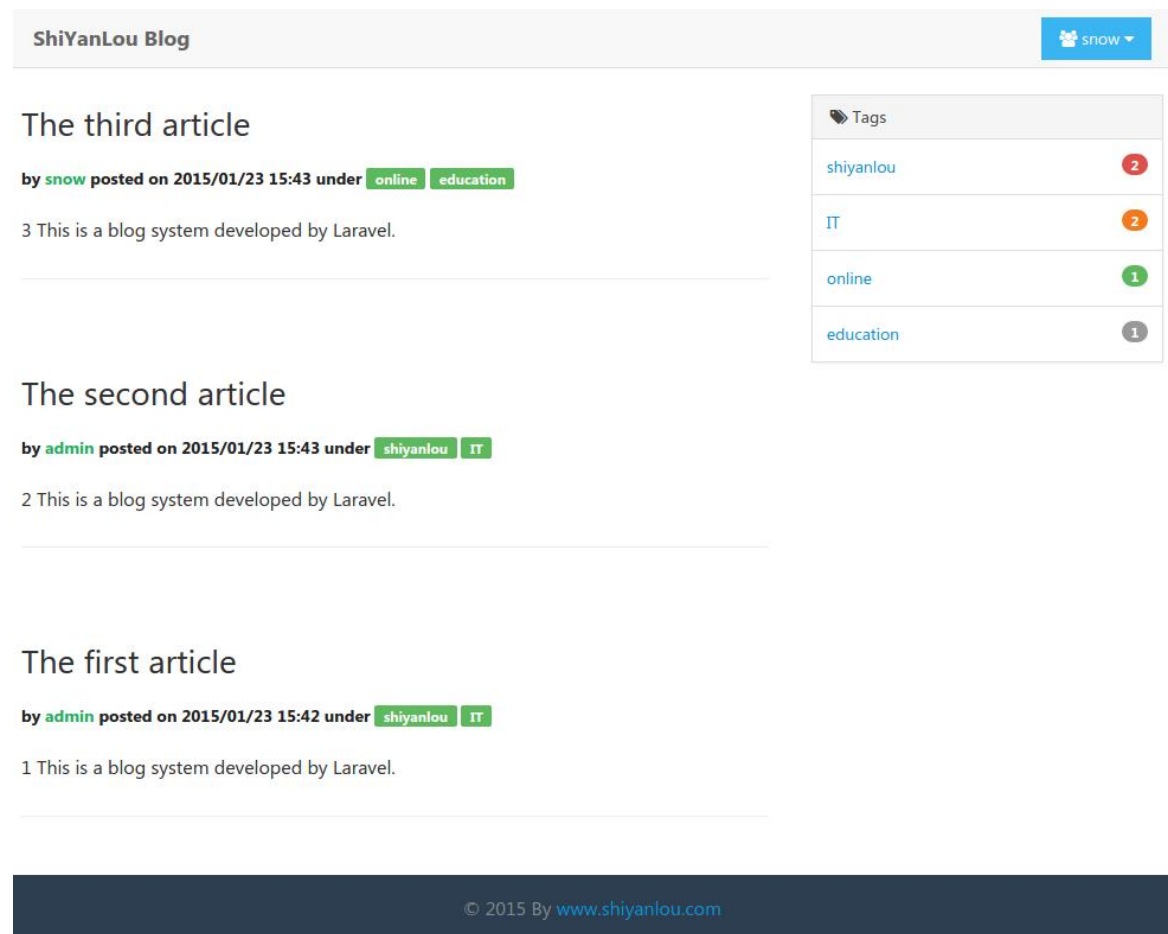
```

Route::get('/', function(){
    $articles = Article::with('user', 'tags')->orderBy('created_at', 'desc')->paginate(10);
    $tags = Tag::where('count', '>', '0')->orderBy('count', 'desc')->orderBy('updated_at', 'desc')->take(10)->get();
    return View::make('index')->with('articles', $articles)->with('tags', $tags);
});

```

上面 `Article::with()` 使用了预加载，可以减少查询次数。

发布几篇文章然后访问首页：



2. 实现用户主页

我们在发表文章后应该能在用户主页看到文章列表，并能对文章进行操作，先在导航栏 `nav.blade.php` 的 `@else` 上面添加一个按钮 `My Articles`：

```
<div class="am-topbar-right">
  <a href="{{ URL::to('user/'. Auth::id() . '/articles') }}" class="am-btn am-btn-primary am-topbar-btn a
m-btn-sm topbar-link-btn"><span class="am-icon-list"></span> My Articles</a>
</div>
```

修改 `home.blade.php`：

```
@extends('_layouts.default')

@section('main')<div class="am-g am-g-fixed blog-g-fixed">
  <div class="am-u-sm-12">
    <table class="am-table am-table-hover am-table-striped ">
      <thead>
```

```

<tr>
  <th>Title</th>
  <th>Tags</th>
  @if ($user->id == Auth::id())
    <th>Managment</th>
  @endif
</tr>
</thead>
<tbody>
  @foreach ($articles as $article)
    <tr>
      <td><a href="{{ URL::route('article.show', $article->id) }}">{{ $article->title }}</a></td>
      <td>
        @foreach ($article->tags as $tag)
          <span class="am-badge am-badge-success am-radius">{{ $tag->name }}</span>
        @endforeach
      </td>
      @if ($user->id == Auth::id())
        <td>
          <a href="{{ URL::to('article/'. $article->id . '/edit') }}" class="am-btn am-btn-xs am-btn-primary"><span class="am-icon-pencil"></span> Edit</a>
          {{ Form::open(array('url' => 'article/' . $article->id, 'method' => 'DELETE', 'style' => 'display: inline;')) }}
          <button type="button" class="am-btn am-btn-xs am-btn-danger" id="delete{{ $article->id }}"><span class="am-icon-remove"></span> Delete</button>
          {{ Form::close() }}
        </td>
      @endif
    </tr>
  @endforeach
</tbody>
</table>
</div></div><div class="am-modal am-modal-confirm" tabindex="-1" id="my-confirm">
<div class="am-modal-dialog">
  <div class="am-modal-bd">
  </div>
  <div class="am-modal-footer">
    <span class="am-modal-btn" data-am-modal-cancel>No</span>
    <span class="am-modal-btn" data-am-modal-confirm>Yes</span>
  </div>
</div></div><script>
$(function() {
  $('[id^=delete]').on('click', function() {
    $('.am-modal-bd').text('Sure you want to delete it?');
    $('#my-confirm').modal({
      relatedTarget: this,
      onConfirm: function(options) {
        $(this.relatedTarget).parent().submit();
      },
      onCancel: function() {
      }
    });
  });
});
</script>
@stop

```

先添加一个 **UserController** :

```
$ php artisan generate:controller UserController
```

在 `UserController.php` 中增加：

```
public function articles(User $user){
    return View::make('home')->with('user', $user)->with('articles', Article::with('tags')->where('user_id',
    '=', $user->id)->orderBy('created_at', 'desc')->get());
}
```

在 `routes.php` 中增加：

```
Route::get('user/{user}/articles', 'UserController@articles');
```

并修改原来的 `Route::get('home')`：

```
Route::get('home', array('before' => 'auth', function(){
    return View::make('home')->with('user', Auth::user())->with('articles', Article::with('tags')->where('user_id', '=', Auth::id())->orderBy('created_at', 'desc')->get());
}));
```

现在当用户登录或点击 `My Articles` 按钮后会跳转到用户主页显示文章列表，并且点击标题时能跳转到显示文章内容页面：



用户主页完成了，另外当在首页和文章内容页面点击作者时也能跳转到相应用户的主页，但是如果不是本用户是没有操作权限的，给 `views/index.blade.php` 中的作者增加链接地址：

```
<a href="{{ URL::to('user' . $article->user->id . '/articles') }}">{{ $article->user->nickname }}</a>
```

给 `articles/show.blade.php` 中的作者增加链接地址：

```
<a href="{{ URL::to('user' . $article->user->id . '/articles') }}" style="cursor: pointer;">{{ $article->user->nickname }}</a>
```

现在点击这两个超链接的时候就能跳转到相应用户的主页了，但是没有操作权限：

ShiYanLou Blog Users		My Articles	admin
Title	Tags	There is no edit and delete button	
The third article	online education		

© 2015 By www.shiyanlou.com

3. 首页分页显示文章

当文章很多时，我们就要分页显示了，Laravel 已经为我们实现好了[分页](#)逻辑，但它默认的是 Bootstrap 的样式，由于我们使用 AmazeUI，所以需要自定义表示器。先在 `app` 目录创建一个名为 `Blog` 的文件夹，这个文件夹中主要放置我们自己写的扩展类，在其中新建一个名为 `PaginationPresenter.php` 的文件，修改：

```
class PaginationPresenter extends Illuminate\Pagination\Presenter {

    public function getActivePageWrapper($text)
    {
        return '<li class="am-active"><a href="">'.$text.'</a></li>';
    }

    public function getDisabledTextWrapper($text)
    {
        return '<li class="am-disabled"><a href="">'.$text.'</a></li>';
    }

    public function getPageLinkWrapper($url, $page, $rel = null)
    {
        return '<li><a href="'.$url.'">'.$page.'</a></li>';
    }
}
```

完成之后这个类还不能被找到，需要在 `composer.json` 中的 `autoload classmap` 中增加 `"app/Blog"`，然后执行：

```
$ composer dump-autoload
```

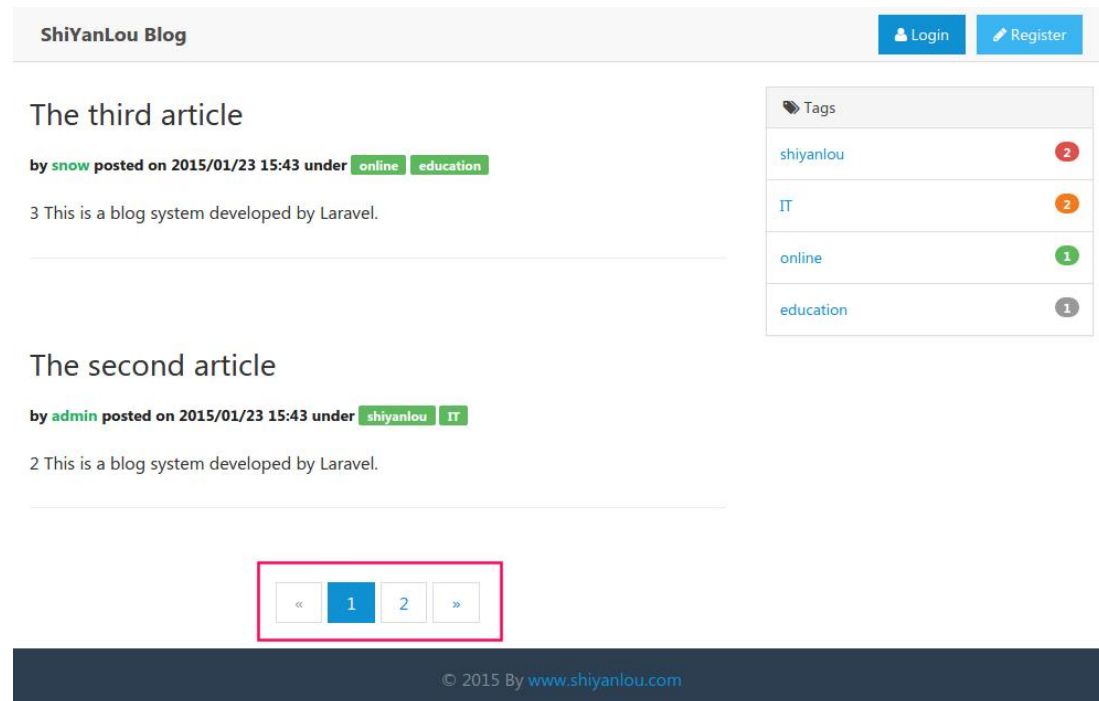
这样这个类就能被找到了，现在创建分页链接的视图：

```
$ php artisan generate:view pagination
```

修改 `pagination.blade.php`：

```
<ul class="am-pagination am-pagination-centered">
    {{ with(new PaginationPresenter($paginator))->render() }}
</ul>
```

完成后修改 `app/config/view.php` 中的 `pagination` 的值为 `paqination`，在 `routes.php` 中 `Route::get('/')` 内 `paginate()` 的参数就是指定每页显示的数量，由于我文章比较少，暂时把它设为 2，最后在 `views/index.blade.php` 中文章显示之后添加 `{{ $articles->links() }}`，现在访问首页就会看到如下分页链接了：



4.修改文章的视图

这步要实现在用户主页能够修改文章了，自己的文章只能自己或者管理员修改，在 `ArticleController.php` 中增加一个过滤器：

```
public function canOperation($route, $request){
    if (!(Auth::user()->is_admin or Auth::id() == Article::find(Route::input('article'))->user_id))
    {
        return Redirect::to('/');
    }
}
```

上面的 `Route::input('article')` 可以获得路由参数，这里就是文章的 `id` 值，然后在构造函数中添加使用过滤器，再添加一个 `csrf` 过滤器：

```
$this->beforeFilter('csrf', array('only' => array('store', 'update', 'destroy')));$this->beforeFilter('@canOperation', array('only' => array('edit', 'update', 'destroy')));
```

创建修改文章的视图：

```
$ php artisan generate:view articles.edit
```

修改 `articles/edit.blade.php`：

```

@extends('_layouts.default')

@section('main')<div class="am-g am-g-fixed">
  <div class="am-u-sm-12">
    <h1>Edit Article</h1>
    <hr/>
    @if ($errors->has())
      <div class="am-alert am-alert-danger" data-am-alert>
        <p>{{ $errors->first() }}</p>
      </div>
    @endif
    {{ Form::model($article, array('url' => URL::route('article.update', $article->id), 'method' => 'PUT',
'class' => "am-form")) }}
    <div class="am-form-group">
      {{ Form::label('title', 'Title') }}
      {{ Form::text('title', Input::old('title')) }}
    </div>
    <div class="am-form-group">
      {{ Form::label('content', 'Content') }}
      {{ Form::textarea('content', Input::old('content'), array('rows' => '20')) }}
    <p class="am-form-help">
      <button id="preview" type="button" class="am-btn am-btn-xs am-btn-primary"><span clas
s="am-icon-eye"></span> Preview</button>
    </p>
  </div>
  <div class="am-form-group">
    {{ Form::label('tags', 'Tags') }}
    {{ Form::text('tags', Input::old('tags')) }}
    <p class="am-form-help">Separate multiple tags with a comma ","</p>
  </div>
  <p><button type="submit" class="am-btn am-btn-success">
    <span class="am-icon-pencil"></span> Modify</button>
  </p>
  {{ Form::close() }}
</div></div>
<div class="am-popup" id="preview-popup">
  <div class="am-popup-inner">
    <div class="am-popup-hd">
      <h4 class="am-popup-title"></h4>
      <span data-am-modal-close
        class="am-close">&times;</span>
    </div>
    <div class="am-popup-bd">
    </div>
  </div></div><script>
$(function() {
  $('#preview').on('click', function() {
    $('#am-popup-title').text($('#title').val());
    $.post('preview', {'content': $('#content').val()}, function(data, status) {
      $('#am-popup-bd').html(data);
    });
    $('#preview-popup').modal();
  });
});</script>
@stop

```


在 `routes.php` 中增加：

```
Route::post('article/{id}/preview', array('before' => 'auth', 'uses' => 'ArticleController@preview'));
```

这是为了修改文章时能够预览文章。

在 `ArticleController.php` 中修改：

```
public function edit($id){
    $article = Article::with('tags')->find($id);
    $tags = "";
    for ($i = 0, $len = count($article->tags); $i < $len; $i++) {
        $tags .= $article->tags[$i]->name . ($i == $len - 1 ? " : ',';";
    }
    $article->tags = $tags;
    return View::make('articles.edit')->with('article', $article);
}
```

现在在用户主页点击修改文章时会跳转到修改页面：

ShiYanLou Blog

My Articles

snow

Edit Article

Title

The third article

Content

```
##1.Introduction

3 This is a blog system developed by Laravel.
##2.Test

**code**:
...
function hello()
{
    return 'Welcome to ShiYanLou!';
}
...

**link**:

[www.shiyanlou.com](www.shiyanlou.com)

**image**:

![[image]](http://anything-about-doc.qiniudn.com/laravel-blog/24.png)
```

Preview

Tags

online,education

Separate multiple tags with a comma ","

Modify

© 2015 By www.shiyanlou.com

5.修改文章

在 `ArticleController.php` 添加修改文章的业务逻辑：

```
public function update($id){
    $rules = [
        'title' => 'required|max:100',
        'content' => 'required',
        'tags' => array('required', 'regex:/^\w+$|^(w+,)+w+$/',
    ];
    $validator = Validator::make(Input::all(), $rules);
    if ($validator->passes()) {
        $article = Article::with('tags')->find($id);
        $article->update(Input::only('title', 'content'));
        $resolved_content = Markdown::parse(Input::get('content'));
        $article->resolved_content = $resolved_content;
        $tags = array_unique(explode(',', Input::get('tags')));
        if (str_contains($resolved_content, '<p>')) {
            $start = strpos($resolved_content, '<p>');
            $length = strpos($resolved_content, '</p>') - $start - 3;
            $article->summary = substr($resolved_content, $start + 3, $length);
        } elseif (str_contains($resolved_content, '<h>')) {
            $start = strpos($resolved_content, '<h>');
            $length = strpos($resolved_content, '</h>') - $start - 4;
            $article->summary = substr($resolved_content, $start + 4, $length);
        }
        $article->save();
        foreach ($article->tags as $tag) {
            if (($index = array_search($tag->name, $tags)) !== false) {
                unset($tags[$index]);
            } else {
                $tag->count--;
                $tag->save();
                $article->tags()->detach($tag->id);
            }
        }
        foreach ($tags as $tagName) {
            $tag = Tag::whereName($tagName)->first();
            if (!$tag) {
                $tag = Tag::create(array('name' => $tagName));
            }
            $tag->count++;
            $article->tags()->save($tag);
        }
        return Redirect::route('article.show', $article->id);
    } else {
        return Redirect::route('article.edit', $id)->withInput()->withErrors($validator);
    }
}
```

这部分较难的是对 `Tag` 的处理，可能我的方法不是最好的。

这样就能真正的实现修改了：

The third article(Modify)

Author: snow Datetime: 2015/01/26 10:51

Tags: [online](#) [itedu](#)

1.Introduction

3 This is a blog system developed by Laravel.

2.Test

code:

```
function hello()
{
    return 'Welcome to ShiYanLou!';
}
```

link:

www.shiyanlou.com

image:

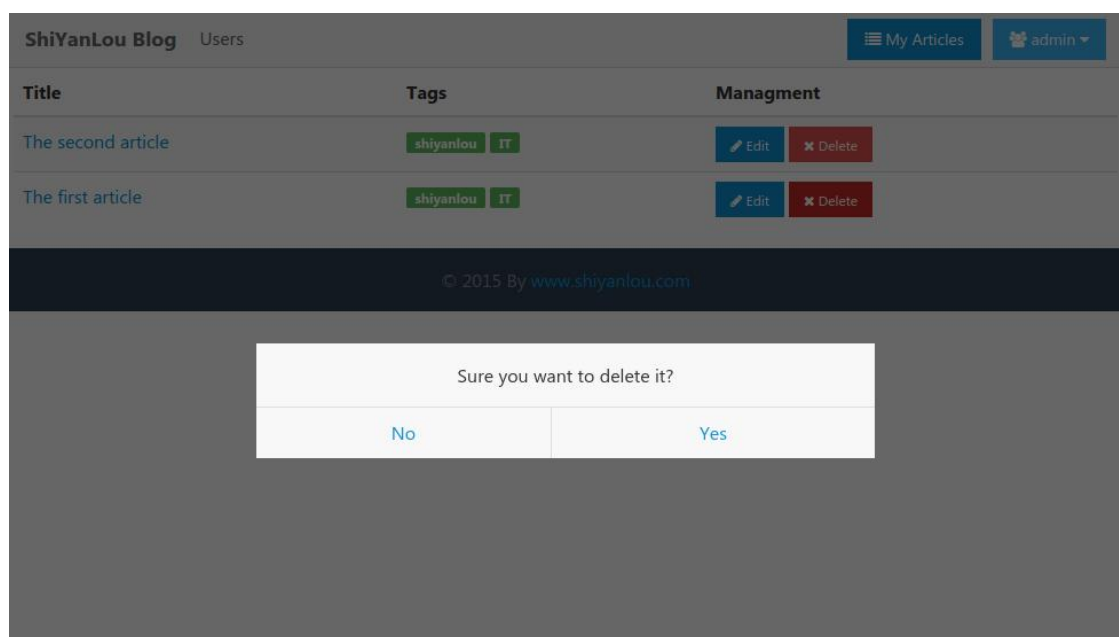


6.删除文章

在 `ArticleController.php` 中增加：

```
public function destroy($id){
    $article = Article::find($id);
    foreach ($article->tags as $tag) {
        $tag->count--;
        $tag->save();
        $article->tags()->detach($tag->id);
    }
    $article->delete();
    return Redirect::to('home');
}
```

我们这里删除文章其实是软删除，它在数据库中还是存在的。



当点击 **Yes** 后会发现文章被删除了。

7.小结

本节教程完成了在首页和用户主页显示文章和标签列表，用户能够管理自己的文章，但只在首页实现了分页显示文章，你可以自己实现在用户主页也分页显示，在删除文章确认提示的时候，你可以加上要删除文章的标题，这样用户体验会更好，你完全可以按你的想法进行实现。这节就到此结束了，我们已经实现了管理员用户管理，下节就将完善管理员模块，实现文章和标签管理。

别忘记还有最后的代码下载：

```
$ git clone https://github.com/shiyanlou/laravel-blog-4.git
```

本文详细出处：<http://www.shiyanlou.com/courses/123>