课堂笔记：

==============================================

一、路由

1.1 别名

Route::get('/admin/user',['as'=>'au','uses'=>function(){

//通过别名创建url

route('别名'，['参数']);

//通过字符串创建url

url('字符串'，['参数']);

}]);

1.2 csrf 保护

你的应用程序不受到 跨网站请求伪造 攻击

使用form表单创建post

<?php echo csrf\_field(); ?>

{{ csrf\_field() }} 推荐

使用ajax创建post提交

1. head头部添加： <meta name="csrf-token" content="{{ csrf\_token() }}">

2. 设置ajax 头部：

$.ajaxSetup({

headers: {

'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')

}

});

二、中间件

作用： 任务验证

1. 创建 php artisan make:middleware xxxMiddlewar

2. 编写中间件任务逻辑

// 检测session

if($request->session()->has('login')){

return $next($request);//通过 执行下一个请求

}else{

// 登录页面

return redirect('/admin/login');

}

3. 注册中间件

E:\wamp\www\lamp\_204\27\_laravel\web\app\Http\Kernel.php

添加： 'login'=>\App\Http\Middleware\LoginMiddlewar::class,

4. 使用

Route::get()->middleware('login');//对单个路由使用

//对一组路由使用

Route::group(['middleware'=>"login"],function(){

});

三、控制器

2.1 创建控制器

php artisan make:controller AbcController

2.2 指定控制器

基本

Route::get('/index','IndexController@index');

Route::post('/index','IndexController@index');

隐式

Route::controller('/index','IndexController');

注意：

方法名称前面加上访问方式　get post

资源

Route::resource('/index','IndexController');

由资源控制器处理的行为

动词 路径 行为（方法） 路由名称

=================================================================

GET /photos index photos.index

GET /photos/create create photos.create

POST /photos store photos.store

GET /photos/{id} show photos.show

GET /photos/{id}/edit edit photos.edit

PUT/PATCH /photos/{id} update photos.update

DELETE /photos/{id} destroy photos.destroy

一、请求

1.1 基本信息获取操作

请求路径

$request->path()

检查路径

$request->is('admin/\*')

获取完整url

$request -> url()

获取请求的方式

$request->method();

检测当前请求的方式

$request->isMethod('post')

获取请求参数

$name = $request->input('name');

默认值设置

$request->input('name', 'Sally');

检查值是否存在

$request->has('name')

获取所有参数

$request -> all()

提取部分

$request->only(['username', 'password']);

获取指定

$request->except(['credit\_card']);

除了当前给定

1.2 闪存操作

将所有的请求参数写入闪存中

$request->flash()

案例：

用户名：<input type="text" name='username' value="{{ old('username') }}">

将部分参数写入闪存中

$request->flashOnly('title','price')

将某些值去除后存到闪存中

$request->flashExcept('\_token');

简便使用

return back()->withInput();

1.3 文件上传

$request -> hasFile() ;检测是否有文件上传

$file = $request -> file() ; 创建文件上传对象

$file -> move() ；执行文件上传

$file ->getClientOriginalExtension()　获取文件后缀

1.4 cookie and session

cookie

设置

\Cookie::queue('name','iloveyou',10);设置cookie的时间为10分钟

return response('haha')->withCookie('uid',10,10);

读取

$request->cookie('name');

\Cookie::get('name');

session

// 获取 session 中的一条数据...

$value = session('key');

// 写入一条数据至 session 中...

session(['key' => 'value']);

二、响应

显示模板

return view('goods.add',[数据]);

返回json

return response()->json(['a'=>100,'b'=>2000]);

下载文件

return response()->download('文件url地址','文件的名称');

页面跳转

return redirect('/goods/add');

共享数据

E:\wamp\www\lamp\_204\28\_laravel\web\app\Providers\AppServiceProvider.php

public function boot()

{

$commons\_cates\_data = ['name1'=>'用户电器','name2'=>'电脑/办公'];

//共享数据

view()->share('commons\_cates\_data', $commons\_cates\_data);

}

三、模版语法

模板的默认存放位置

resources/views

使用变量

{{$username}}

使用注释

{{-- 内容 --}}

使用函数

{{ time() }}

设置默认值

{{ $str or 'aaaaaaa' }}

显示html代码

{!! $name !!}

流程控制

@if (count($records) === 1)

I have one record!

@elseif (count($records) > 1)

I have multiple records!

@else

I don't have any records!

@endif

循环控制

@for ($i = 0; $i < 10; $i++)

The current value is {{ $i }}

@endfor

@foreach ($users as $user)

<p>This is user {{ $user->id }}</p>

@endforeach

四、mysql操作

查询

DB::select('select \* from users where active = ?', [1]);

插入

DB::insert('insert into users (id, name) values (?, ?)', [1, 'Dayle']);

修改

DB::update('update users set votes = 100 where name = ?', ['John']);

删除

DB::delete('delete from users',[]);

==============================

作业 敲3遍

扩展：

模块