默认路由文件路径  **routes/web.php**

#### 响应多个 HTTP 方法的路由

可以使用 match 方法，也可以使用 any 方法注册一个实现响应所有 HTTP 的请求的路由：

Route::match(['get', 'post'], '/', function () {

//

});

Route::any('foo', function () {

//

});

**CSRF 保护**

任何指向 web 中 POST, PUT 或 DELETE 路由的 HTML 表单请求都应该包含一个 CSRF 令牌，否则，这个请求将会被拒绝。更多的关于 CSRF 的说明在 [CSRF 说明文档](https://learnku.com/docs/laravel/5.4/csrf)：

<form method="POST" action="/profile">

{{ csrf\_field() }}

...

</form>

## 路由参数

### 必选路由参数

当然，有时我们需要在路由中捕获一些 URL 片段。例如，我们需要从 URL 中捕获用户的 ID ，我们可以这样定义路由参数：

Route::get('user/{id}', function ($id) {

return 'User '.$id;

});

也可以根据需要在路由中定义多个参数：

Route::get('posts/{post}/comments/{comment}', function ($postId, $commentId) {

//

});

路由的参数通常都会被放在 {} 内，并且参数名只能为字母，当运行路由时，参数会通过路由闭包来传递。

**注意：** 路由参数不能包含 - 字符。请用下划线 (\_) 替换。

### 可选路由参数

声明路由参数时，如需指定该参数为可选，可以在参数后面加上 ? 来实现，但是相应的变量必须有默认值：

Route::get('user/{name?}', function ($name = 'John') {

return $name;

});

### 正则表达式约束

你可以使用 where 方法来规范你的路由参数格式。where 方法接受参数名称和定义参数约束规则的正则表达式：

Route::get('user/{name}', function ($name) {

//

})->where('name', '[A-Za-z]+');

Route::get('user/{id}', function ($id) {

//

})->where('id', '[0-9]+');

Route::get('user/{id}/{name}', function ($id, $name) {

//

})->where(['id' => '[0-9]+', 'name' => '[a-z]+']);