Named Routes and Route Variables

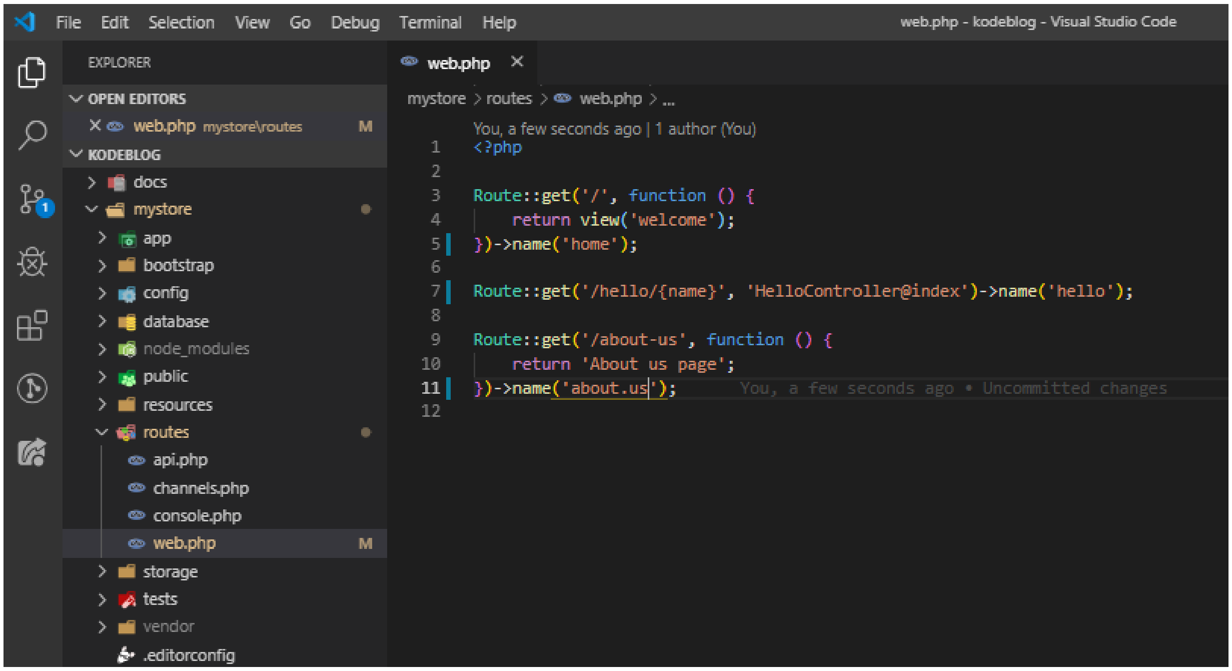
Named Routes

Description

- Routes are defined in a single place but used everywhere.
- The home page can have a link to products page.
- The about us page can also have the same link and so many places through out our application.
- Laravel allows us to name our routes so that we can use the route name instead of the actual URL.

Add Names on Existing Routes

- In VSC, open file `kodeblog\mystore\routes\web.php`
- Add names for each of the routes

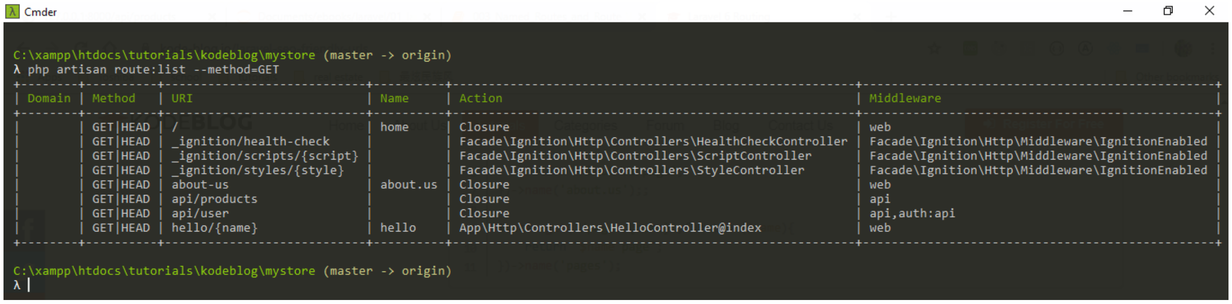


```php
<?php

Route::get('/', function () {
    return view('welcome');
})->name('home');

Route::get('/hello/{name}', 'HelloController@index')->name('hello');

Route::get('/about-us', function () {
    return 'About us page';
})->name('about.us');
```

Show Results

- In `cmder`

  ```
  php artisan route:list --method=GET
  ```



```
C:\xampp\htdocs\tutorials\kodeblog\mystore (master -> origin)
λ php artisan route:list --method=GET
+--------+----------+----------------------------+----------+--------------------------------------------------------+-------------------------------------------------+
| Domain | Method   | URI                        | Name     | Action                                                 | Middleware                                      |
+--------+----------+----------------------------+----------+--------------------------------------------------------+-------------------------------------------------+
|        | GET|HEAD | /                          | home     | Closure                                                | web                                             |
|        | GET|HEAD | _ignition/health-check     |          | Facade\Ignition\Http\Controllers\HealthCheckController | Facade\Ignition\Http\Middleware\IgnitionEnabled |
|        | GET|HEAD | _ignition/scripts/{script} |          | Facade\Ignition\Http\Controllers\ScriptController      | Facade\Ignition\Http\Middleware\IgnitionEnabled |
|        | GET|HEAD | _ignition/styles/{style}   |          | Facade\Ignition\Http\Controllers\StyleController       | Facade\Ignition\Http\Middleware\IgnitionEnabled |
|        | GET|HEAD | about-us                   | about.us | Closure                                                | web                                             |
|        | GET|HEAD | api/products               |          | Closure                                                | api                                             |
|        | GET|HEAD | api/user                   |          | Closure                                                | api,auth:api                                    |
|        | GET|HEAD | hello/{name}               | hello    | App\Http\Controllers\HelloController@index             | web                                             |
+--------+----------+----------------------------+----------+--------------------------------------------------------+-------------------------------------------------+

C:\xampp\htdocs\tutorials\kodeblog\mystore (master -> origin)
λ |
```

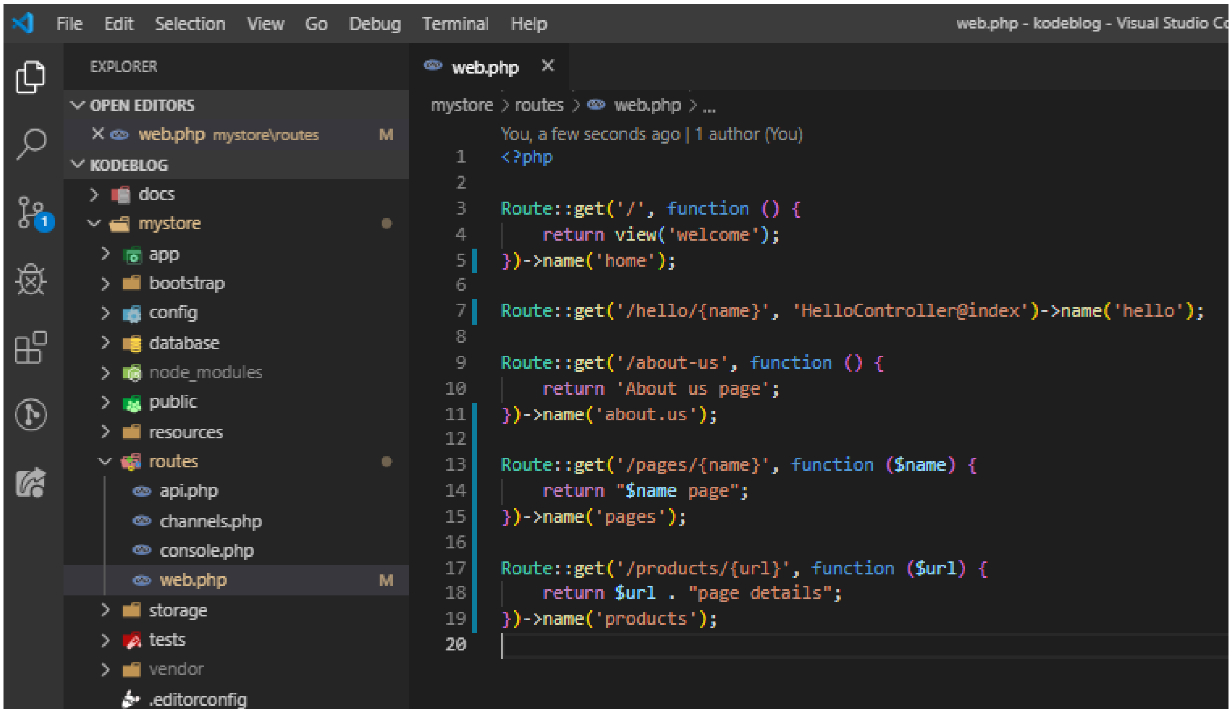
Route Parameters

Description

- Laravel allows us to define routes that accept parameters that are passed as arguments to functions that respond to route requests.
- For example. we can have a products page and a user can pass in macbook as a parameter argument to get the details about a product with URL `macbook` .
- Parameters are defined using curly brackets

Create Two Routes with Parameter

- In VSC, open file `kodeblog\mystore\routes\web.php`
- Add new route `/pages/{name}`
- Add new route `/products/{url}`



```php
<?php

Route::get('/', function () {
    return view('welcome');
})->name('home');

Route::get('/hello/{name}', 'HelloController@index')->name('hello');

Route::get('/about-us', function () {
    return 'About us page';
})->name('about.us');

Route::get('/pages/{name}', function ($name) {
    return "$name page";
})->name('pages');

Route::get('/products/{url}', function ($url) {
    return $url . "page details";
})->name('products');

```

Test Online

- In `cmder` , start server
- In browser, access

```
http://127.0.0.1:8000/pages/contact-us
http://127.0.0.1:8000/pages/about-us
http://127.0.0.1:8000/pages/services
http://127.0.0.1:8000/products/apple
http://127.0.0.1:8000/products/pen
```

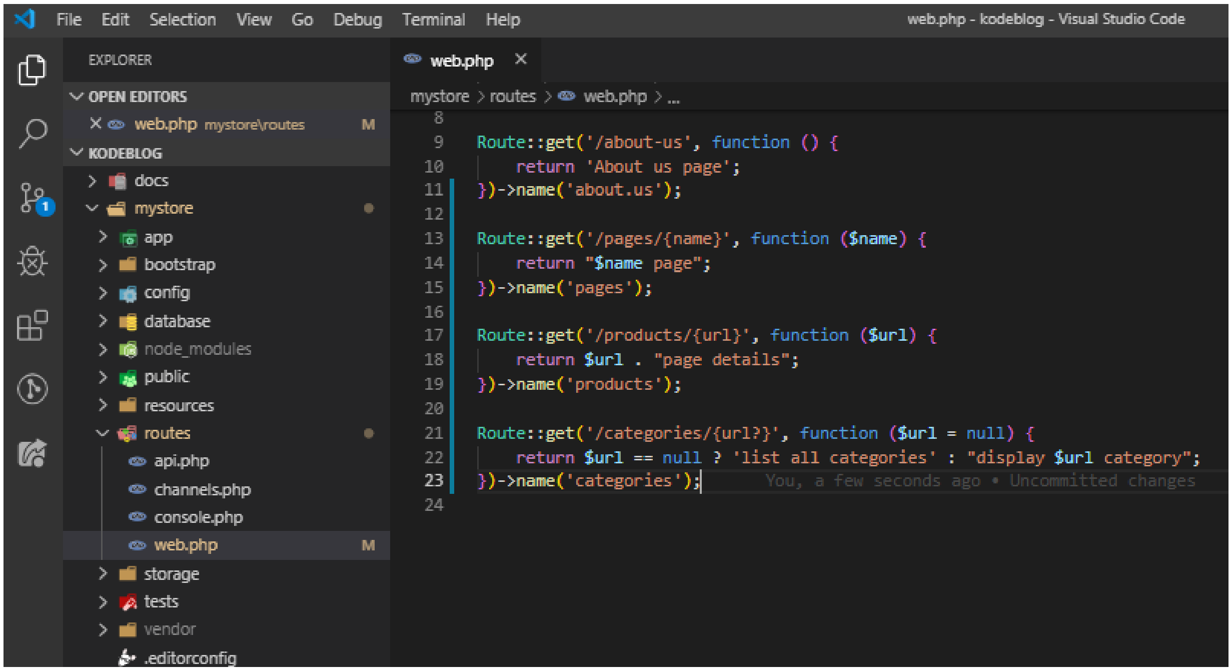
Optional Parameters

Description

- Optional parameters are parameters that can be skipped when calling the route.
- For example
  - We can define a single route to handle both all categories listing and a single category.
  - If the category URL is not passed in as an argument then we list everything else we list details for a single product.

Create a Route with Optional Paramenters

- In VSC, open file `kodeblog\mystore\routes\web.php`
- Add route `/categories/{url?}`



```php

Route::get('/about-us', function () {
    return 'About us page';
})->name('about.us');

Route::get('/pages/{name}', function ($name) {
    return "$name page";
})->name('pages');

Route::get('/products/{url}', function ($url) {
    return $url . "page details";
})->name('products');

Route::get('/categories/{url?}', function ($url = null) {
    return $url == null ? 'list all categories' : "display $url category";
})->name('categories');

```

Test Online

- In `cmder` , start server
- In browser, access

```
http://127.0.0.1:8000/categories
http://127.0.0.1:8000/categories/computers
```

Commit Step

In [ ]: