

References

This cheat sheet references documentation directly from the [Laravel 6.x Routing](https://laravel.com/docs/6.x/routing) documentation.

<https://laravel.com/docs/6.x/routing>

Basic Route Syntax

```
Route::verb( 'uri', UserController@method);
```

Calls index method from UserController

```
Route::verb( 'uri', $callback);
```

Uses callback function

Basic Route Syntax Explained

Route is a Laravel Class

verb is a static method called on the Route Class. Static methods use :: scope resolution operator to point to the class.

Basic verbs/ methods that can be called on Route:

get, post, put, patch, delete, options

<https://laravel.com/api/6.x/Illuminate/Contracts/Routing/Registrar.html>

\$uri points to the url

Example: localhost:8080/user ... user is the controller

More Route Methods

```
Route::view( 'welcome', 'welcome');
```

Shortcut if route only needs to a view and not a full controller

```
Route::view( 'welcome', 'welcome', ['name' => 'Taylor']);
```

Optional array of data may be passed as third argument

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

Responds to multiple HTTP verbs

```
Route::any('/', function () { // });
```

Responds to all HTTP verbs using the any method:

```
Route::redirect( '/ here', '/there', 301);
```

/here redirects to /there. 3rd param is optional. It overrides the default 302 status which can be verified with curl. The where method is chained to the route and accepts an array of key-value pairs.

```
Route::permanentRedirect( '/ here', '/there');
```

Return a permanent 301 status code:

CSRF Protection

CSRF protection is enabled by default in Laravel. It prevents cross-site request forgery attacks.

```
@csrf
```

```
...
```

```
</form>
```

Any HTML form pointing to POST, PUT, or DELETE routes that are defined in the web routes file should include a CSRF token field as a security layer or the request will be rejected.

Route Parameters - Required

Used to capture a segment of URI within a route.

```
Route::get( '/user/{id}/{name}', function ($id, $name) {
    return 'User #'. $id . ' is ' . $name;
});
```

Static methods use :: scope resolution operator to point to the class.

Route params are encased in {} braces. Use as many as you need.

Route parameters are injected into route callbacks

<https://laravel.com/api/6.x/Illuminate/Contracts/Routing/Registrar.html>

Example: localhost:8080/user/12345/Fred renders

Route Parameters - Optional

Placing a ? mark after the param name makes it optional.

```
Route::get( '/user/{name}/{id?}', function ($name, $id) {
    return $name . 's user number is ' . $id;
});
```

Route::view('welcome', 'welcome');

Make sure to give the param a default value such as 'Taylor'.

Example 1: http://127.0.0.1:8000/user/Fred`

Renders: Fred 's user number is unknown

Example 2: http://127.0.0.1:8000/user/Fred/12345`

Renders: Fred 's user number is 123

Constraints

Route::where('id', '^[0-9]+\$');

Example:

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

```
//
```

Global Constraint

To set a route parameter to always be constrained by a regular expression, use the pattern method in the boot() method in routes.php

Example:

```
public function boot() {
    // Route: :pattern('id', '[0-9]+'); parent::boot()
    // id parameter now must always consist of only numbers to execute, no matter which Route is using the id parameter
```

Encoded Forwarded Slashes

The Laravel routing component allows all characters except /. You must explicitly allow / to be part of your routes.

```
Route::get('search/{search}', function ($search) {
    return $search;
})->where('search', '.*');
```

Named Routes

```
Route::get('user/profile', function () {
    // ...
})->name('profile');
```



By **jlampstack**
cheatography.com/jlampstack/

Not published yet.
 Last updated 28th November, 2019.
 Page 2 of 2.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>