References

This cheat sheet references docume ntation directly from the Laravel website where more in-depth explan ations are provided.

```
https: //l ara vel.co m/d ocs /6.x /r outing
```

Basic Route Syntax

```
Route: :ve rb( '/uri', UserCo ntr oll er@ met hod);
```
Calls index method from UserCo ntr oller

```
Route: :ve rb( '/uri', $callb ack);
```
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 its Class

Basic verbs/ methods than can be called on Route:
get, post, put, patch, delete, options
```
https: //l ara vel.co m/a pi/ 6.x /Il lum ina -
te/ Con tra cts /Ro uti ng/ Reg ist rar.html
```

$uri points to the url

Example: localh os t : 80 80/ *user* ...
is the uri

More Route Methods

```
Route: :vi ew( '/w elc ome', 'welco me');
```
Shortcut if route only needs to a view and not a full route or controller

```
Route: :vi ew( '/w elc ome', 'welcome', ['name' =>
'Taylo r']);
```
Optional array of data may be passed as third arg

```
Route: :ma tch ([' get', 'post'], '/', function ()
{ // });
```
Responds to multiple HTTP verbs

More Route Methods (cont)

```
Route: :an y('/', function () { // });
```
Responds to all HTTP verbs using the any method:

```
Route: :re dir ect ('/ here', '/there', 301);
```
/here redirects to /there. 3rd param is optional. It overrides the default 302 status which can be verified using Chrome dev tools in the network tab

```
Route: :pe rma nen tRe dir ect ('/ here',
'/there');
```
Return a permanent 301 status code:

CSRF Protection

```
<form method="POST" action="/profile">
        @csrf
        ...
</f orm>
```
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: :ge t(' /us er/ {id }/{ name}', function
($id, $name) {
return 'User #'. $id . ' is ' . $name;
});
```
Route params are encased in {} braces. Use as many as needed
Route parameters are injected into route callbacks / contro llers based on order
Example: localh ost :80 80/ use r/1 234 5/Fred renders "User #12345 is Fred"

By jlampstack

cheatography.com/jlampstack/

Not published yet.
Last updated 28th November, 2019.
Page 1 of 2.

Sponsored by CrosswordCheats.com
Learn to solve cryptic crosswords!
Http://crosswordcheats.com

Route Parameters - Optional

Placing a ? mark after the param name makes it optional

Route: :ge t(' /us er/ {na me} /{i d?}', function ($name, $id = 'unknown') {
return $name . ' \'s user number is ' . $id;
});

Make sure to give the param a default value such as null or Fred

Example 1: http:/ /12 7.0.0.1 :8 000 /us er/ Fred`

Renders: Fred 's user number is unknown

Example 2: http:/ /12 7.0.0.1 :8 000 /us er/ Fre - d/123`

Renders: Fred 's user number is 123

Contraints

The where method is chained to the route and accepts the name of the parameter and a regular expression defining how the parameter should be constr ained

Example:

```
Route: :ge t(' use r/{ id}', function ($id) {
//
})->where('id', '[0-9]+');
```

Global Constraint

To set a route parameter to always be constr ained by a regular expres sion, use the pattern method in the boot method of RouteS erv ice Pro vid er.php file Use the pattern method in the boot method of RouteS erv ice Pro vid er.php

Example:

```
public function boot() { // Route: :pa tte -
rn( 'id', '[0-9]+'); parent ::b oot(); }
```

id parameter now must always consist of only numbers to execute, no matter which Route is using the id parameter name

Encoded Forwarded Slashes

The Laravel routing component allows all characters except /. You must explicitly allow / to be part of your placeh older using a where condition regular expression

```
Route: :ge t(' sea rch /{s ear ch}', function ($search) {
return $search;
})->wh ere ('s earch', '.*');
```

Named Routes

```
Route: :ge t(' use r/p rof ile', function () { //
})->na me( 'pr ofi le');
```