
Road Trip Documentation

Release 0.0.1

Steven Liebrecht

May 19, 2018

Table of Contents

1	Overview	3
1.1	Requirements	3
1.2	Installation	3
1.3	License	3
2	Quickstart	5
2.1	Basic example	5
2.2	Setting options and prefixes	5
2.3	Loading multiple collections	6
2.4	Loading collections from files	6
2.5	Caching routes	6
3	Route	7
3.1	Properties	7
3.2	Public Functions	7
3.3	Property Documentation	7
3.3.1	collection : RouteCollection	7
3.4	Function Documentation	7
3.4.1	Route::__construct(RouteCollection \$collection, string \$method, string \$path, \$handler) . .	7
4	RouteCollection	9
4.1	Properties	9
4.2	Public Functions	9
4.3	Property Documentation	9
4.4	Function Documentation	9
5	Router	11
5.1	Properties	11
5.2	Public Functions	11
5.3	Property Documentation	11
5.4	Function Documentation	11

Note: This documentation is not yet finished, and the code is not finished either, so actually nothing is finished and I should work harder and now I'm going to do something productive instead of writing this useless note.

1.1 Requirements

- PHP 7.1 or higher
- If you use Apache, you will need the `mod_rewrite` module.
- You will need to redirect all requests that do not lead to existing files to your index file, most of the times this is the `index.php` file in your public root.

1.2 Installation

The recommended way to install Road Trip is with [Composer](#).

You can install Road Trip by using the Composer CLI.

```
composer require stevenliebregt/road-trip:^1.0
```

Alternatively you can add the dependency to your existing `composer.json` file.

```
{
    "require": {
        "stevenliebregt/road-trip": "^1.0"
    }
}
```

After installing it, you need to require Composer's autoloader in your code.

1.3 License

This project is licensed using the [MIT license](#).

Copyright (c) 2018 Steven Liebrecht

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This page provides a quick introduction to Road Trip, and shows some examples. If you have not installed Road Trip yet, please head over to the [overview](#) page.

2.1 Basic example

This basic example will get you started right away.

```
<?php
use StevenLiebregt\RoadTrip\RouteCollection;
use StevenLiebregt\RoadTrip\Router;

$router = new Router();

$collection = new RouteCollection();
$collection->get('/products', 'ProductController.index');
$collection->post('/products', 'ProductController.create');

$router->addCollection($collection);
$router->compile();

$match = $router->match('THE_CURRENT_REQUEST_METHOD', 'THE_CURRENT_REQUEST_URI');
```

2.2 Setting options and prefixes

This example will show you how to set prefixes for path, names and handlers, and how you can set options.

```
<?php
```

(continues on next page)

(continued from previous page)

```
use StevenLiebregt\RoadTrip\RouteCollection;
use StevenLiebregt\RoadTrip\Router;

$router = new Router();

$collection = new RouteCollection();
$collection->setPathPrefix('/api');

$collection->get('/foo', 'Foo.Action'); // The prefix resolves this to `/api/foo`.

$router->addCollection($collection);
$router->compile();

$match = $router->match('THE_CURRENT_REQUEST_METHOD', 'THE_CURRENT_REQUEST_URI');
```

2.3 Loading multiple collections

TODO

2.4 Loading collections from files

TODO

2.5 Caching routes

TODO

3.1 Properties

TODO

3.2 Public Functions

TODO

3.3 Property Documentation

3.3.1 collection : RouteCollection

Explanation

3.4 Function Documentation

3.4.1 Route::__construct(RouteCollection \$collection, string \$method, string \$path, \$handler)

Explanation about each parameter

4.1 Properties

TODO

4.2 Public Functions

TODO

4.3 Property Documentation

TODO

4.4 Function Documentation

TODO

5.1 Properties

TODO

5.2 Public Functions

TODO

5.3 Property Documentation

TODO

5.4 Function Documentation

TODO