
site-analyzer Documentation

Release stable

Mar 19, 2019

Contents

1	Features	3
2	Installation	5
3	Usage	7
3.1	Reporting/Displaying the stored information	7
3.2	Main features	7
4	Examples	9
4.1	Example 1: time analysis	9
4.2	Example 2: user analysis	9
4.3	Example 3: path analytics	9
4.4	Example 4: a/b test	9
4.5	Example 5: Profile Analysis	9
4.6	Example 6: User-Time profile	10

SiteAnalyzer is a package to analyze PHP sites. With this package, it is possible to count pages, understand user behavior, analyze the interaction of the site, and perform A/B testings. SiteAnalyzer can be installed into any PHP in seconds without affecting the business logic of the application and customized according to the needs. SiteAnalyzer makes use of machine learning algorithms and statistics to analyze a site and create meaningful reports.

CHAPTER 1

Features

- Click counter
- User profile analysis
- A/B testing
- Site dynamics analysis

CHAPTER 2

Installation

SiteAnalyzer can be installed through [Composer](<https://getcomposer.org>):

```
`shell $ composer require rdorado/site-analyzer `
```

Or modify your composer.json requirements:

““**json**

```
    “require”: { “rdorado/site-analyzer”: “^0.0.1”  
    }
```

““ and update your project:

```
`shell $ composer update `
```


Count all the views/pages you want to observe by importing the `SiteAnalyzer` class and then using the `count` static method:

```
`php use SiteAnalyzer\SiteAnalyzer; SiteAnalyzer::count();`
```

You can also include other options to be stored on the database. That depends on the kind of reportings or analysis you want to perform:

```
`php $options = ['id' = $myid]; SiteAnalyzer::count($options);`
```

3.1 Reporting/Displaying the stored information

Once you have started to count the page hits, different sort of reports can be displayed. For example, a very basic report is the number of hits per page stored on the database:

```
`php $data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );`
```

3.2 Main features

3.2.1 Counting

The core functionality of `SiteAnalyzer` is to store counts. Every time the static method `SiteAnalyzer::count()` is called it stores the url associated

```
`php use SiteAnalyzer\SiteAnalyzer; SiteAnalyzer::count();`
```


4.1 Example 1: time analysis

```
$data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );
```

4.2 Example 2: user analysis

```
$data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );
```

4.3 Example 3: path analytics

```
$data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );
```

4.4 Example 4: a/b test

```
$data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );
```

4.5 Example 5: Profile Analysis

```
$data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );
```

4.6 Example 6: User-Time profile

```
$data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );  
### Page profiling $data = SiteAnalyzer::getStats(); print( SiteAnalyzer::transform($data, "html") );
```