
PhpStratum for MySQL and MariaDB

Jan 26, 2023

Contents

1	Further Reading	3
2	Table of Contents	5
2.1	Getting Started	5
2.2	Installing & Uninstalling PhpStratum	5
2.3	An Example	6
2.4	The PhpStratum Config File	6
2.5	The PhpStratum Program	8
2.6	License	9

PhpStratum is a set of PHP packages for loading stored procedures into your application's database and invoking those stored procedures conveniently from your application using an automatically generated wrapper class. Currently, PhpStratum is available for the following database management systems:

- [MySQL and MariaDB](#)
- [SQLite](#)

This is the documentation of PhpStratum for MySQL and MariaDB. This documentation is self contained and doesn't require any knowledge from any other documentation.

CHAPTER 1

Further Reading

You have the following options for further reading:

- Continue reading this documentation for MySQL and MariaDB.
- Reading a broader documentation about the concepts of [Stratum Projects](#) in other programming languages than PHP.
- Reading the general documentation of [PhpStratum](#).
- Reading the documentation of PhpStratum for [SQLite](#). This documentation is self contained and doesn't require any knowledge from any other documentation.

2.1 Getting Started

In this chapter you will learn how to install PhpStratum, loading stored routines, and generating the data layer. The data layer is a class for conveniently invoking your stored routines in your PHP code.

2.1.1 Installing PhpStratum

The preferred way to install PhpStratum is using [composer](#):

```
composer require setbased/php-stratum-mysql
composer require setbased/php-stratum
```

The first line installs the MySQL and MariaDB backend and the second line installs the frontend (a.k.a the CLI) of PhpStratum.

2.1.2 Running PhpStratum

You can run PhpStratum from the command line:

```
./vendor/bin/stratum
```

If you have set `bin-dir` in the `config` section in `composer.json` you must use a different path.

2.2 Installing & Uninstalling PhpStratum

2.2.1 Installing PhpStratum

The preferred way to install PhpStratum is using [composer](#):

```
composer require setbased/php-stratum-mysql
composer require setbased/php-stratum
```

The first line installs the MySQL and MariaDB backend and the second line installs the frontend (a.k.a the CLI) of PhpStratum.

Running PhpStratum

You can run PhpStratum from the command line:

```
./vendor/bin/stratum
```

If you have set `bin-dir` in the `config` section in `composer.json` you must use a different path.

For example:

```
{
  "config": {
    "bin-dir": "bin/"
  }
}
```

then you can run PhpStratum from the command line:

```
./bin/stratum
```

2.2.2 Uninstalling PhpStratum

Remove PhpStratum from your project with `composer`:

```
composer remove setbased/php-stratum-mysql
composer remove setbased/php-stratum
```

2.3 An Example

2.4 The PhpStratum Config File

This chapter is the specification of the stratum config file.

For most projects the stratum config file must added to the VCS and distributed to the production environment of your project (unless you have some other mechanism for loading stored routines into your production database).

The stratum config file is a JSON, XML, YML or INI file and consist out of five sections which we discuss in detail in the sections below. Throughout this manual we are using INI file.

2.4.1 The stratum section

The `stratum` section is mandatory and holds one variable only:

backend (mandatory) The fully qualified name of the class that implements the required backend. The backend must be a child class of `SetBased\Stratum\Backend\Backend`. The backend of PhpStratum for MySQL and MariaDB is implemented in `SetBased\Stratum\MySql\Backend\MySqlBackend`.

2.4.2 The database section

The `database` section is mandatory and holds the parameters to connect to the MySQL or MariaDB instance.

host (mandatory) The host where the MySQL or MariaDB instance is running.

user (mandatory) The user name. The user must have the *create routine* privilege.

password (mandatory) The password of *user*.

database (mandatory) The schema (database) with your application tables.

port (optional) The port number for connecting to the MySQL or MariaDB instance. Default value is 3306.

2.4.3 The constants section

The `constants` section is optional and holds the variables for constants based on column sizes and constants defined in PHP available in the sources of stored routines.

columns (mandatory) A plain text file with table column names and their corresponding constant names.

class (mandatory) The class defining constants.

prefix (mandatory) The default prefix to use when generating a constant name based on the size of a column.

2.4.4 The loader section

The `loader` section is mandatory and holds the variables for loading your stored routines into the database.

sources (mandatory) Holds the pattern with sources of stored routines. This variable can have two forms:

- A wildcard pattern like `lib/psql/**/* .psql`.
- A filename like `file:stratum-sources.txt`. The filename is relative to the directory of the stratum config file. The file holds a wildcard pattern per line.

metadata (mandatory) The path to the file where the metadata of stored routines must be stored in JSON format. This metadata is used to determine which stored routines must be loaded or removed from the database and provide information for the wrapper generator.

sql_mode (mandatory) The SQL mode under which the stored routines must run.

character_set (mandatory) The (default) character set under which the stored routine must run.

collate (mandatory) The (default) collate under which the stored routine must run.

Note: the SQL mode, character set, and collate under which a stored routine runs is determined when the stored routine is defined and not inherited from the session that calls the stored routine.

2.4.5 The wrapper section

The wrapper section is optional and holds variables for generating the wrapper class (a.k.a. the data layer).

parent_class (mandatory) The fully qualified name of the parent class of the wrapper class. PhpStratum for MySQL and MariaDB provides class `SetBased\Stratum\MySql\MySqlDataLayer`.

mangler_class (optional) The fully qualified name of the class for converting the name of a stored routine into the name of the method for calling the stored routine. This class must implement `SetBased\Stratum\Middle\NameMangler\NameMangler`. For converting stored routines with underscores use class `\SetBased\Stratum\Middle\NameMangler\PsrNameMangler`.

wrapper_class (mandatory) The fully qualified name of the generated wrapper class.

wrapper_file (mandatory) The path the file where to source of the wrapper class must be stored.

strict_types (optional, default 1) If 1 the wrapper class will enable strict mode. If 0 strict mode is disabled.

2.5 The PhpStratum Program

The PhpAudit program is a [Symfony console application](#) with five additional commands:

- `constants`
- `crud`
- `loader`
- `stratum`
- `wrapper`

We discuss the additional commands in the sections below.

Some commands provide additional output when in verbose mode (`-v`). All commands show the queries been executed in very verbose mode (`-vv`).

```
$ ./vendor/bin/stratum
stratum 6.0.0

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
                       --ansi          Force ANSI output
                       --no-ansi       Disable ANSI output
  -n, --no-interaction  Do not ask any interactive question
  -v|vv|vvv, --verbose  Increase the verbosity of messages: 1 for normal output,
                       2 for more verbose output and 3 for debug

Available commands:
  constants  Generates constants based on database IDs
  crud       An interactive command for generating stored routines for CRUD operations
  help       Displays help for a command
  list       Lists commands
  loader     Generates the routine wrapper class
  stratum    Runs the constants, loader, and wrapper commands
  wrapper    Generates a class with wrapper methods for calling stored routines

$
```

2.5.1 The constants command

2.5.2 The crud command

2.5.3 The loader command

2.5.4 The stratum command

2.5.5 The wrapper command

2.6 License

The project is licensed under the [MIT license](#).