
Baleen CLI Documentation

Release 0.3.1

Gabriel Somoza <gabriel@strategery.io>

October 09, 2015

1	User's Guide	3
1.1	Installation	3
1.2	Configuration	3
1.3	Usage	3
2	Developer's Guide	5
2.1	3rd Party Components	5
2.2	Commands	6
2.3	Comparator	7
2.4	Configuration	8
2.5	HelperSet	9
2.6	Providers	9
2.7	Repository	10
2.8	Storage	11
2.9	Timeline	11
2.10	Tutorial: Build Your Own Migrations Library	11
3	Reference	17
3.1	Baleen\Cli	17
4	Indices and Tables	53
	PHP Namespace Index	55

Welcome to Baleen CLI's documentation!

Baleen CLI is a command-line wrapper that uses Symfony Console and leverages the power of Baleen Migrations. Its primary purpose is to provide out-of-the-box functionality for Baleen Migrations. But thanks to its flexible architecture it can also be used as a starting point for your own migrations library.

Warning: This program is an early release and licensed under the MIT license. Please read the *LICENSE* file attached to the source-code before using this program to migrate sensitive data. And of course: always back up your data before migrating.

Contents:

1.1 Installation

Install with Composer:

```
composer install baleen/cli
```

1.2 Configuration

Then run the following command on the root directory of the project you want to migrate:

```
./vendor/bin/baleen init
```

This will create a configuration file called `.baleen.yml` and a storage file at `.baleen_versions`. You should put the config file under version control, but not the storage file.

Why a storage file instead of a database table? Baleen is database-agnostic. That's where YOU come in: you can customize Baleen CLI to use any other type of storage you can imagine. A local file is just the default we decided to ship with.

1.3 Usage

To see usage information and a list of available commands run `./vendor/bin/baleen` without any arguments.

1.3.1 Creating Migrations

Creating a migration is very easy, just run:

```
./vendor/bin/baleen create SomeOptionalTitle
```

A new migration file will be created inside your configured migrations folder.

Note that Baleen CLI uses PSR-4 standards for autoloading the generated migrations out of the configured migrations directory.

Advanced Baleen CLI users can overwrite the class generator to fit the needs of their project.

1.3.2 Adding Your Code

If you open a migration file you'll see it has two empty methods: `up()` and `down()`, which will be called when migrating upwards or downwards (respectively). You can put any code you want inside those methods - for example code to transform your database, an image, a document, etc. - anything you want to migrate!

Developer's Guide

Baleen CLI can perform migrations on its own, but it was also built as a framework that can help developers build their own migrations library without worrying about migrations boilerplate.

The purpose of this section is to show how developers can use the Baleen CLI framework in order to supply a migrations library for their own framework or application.

2.1 3rd Party Components

Baleen CLI uses a series of 3rd-party components to power its functionality. Here's a quick overview of the main components being used and how. Throughout the rest of the guide we will assume you know how those components work, so we recommend reading up on each component's documentation on an as-needed basis.

Main components (used directly by Baleen CLI) listed in alphabetical order. Their dependencies are not included in this list unless they're used directly as well.

2.1.1 Baleen\Migrations

Provides the domain model for migrations. Many of the architectural concepts from that library also apply to Baleen CLI. We recommend reading up on the Baleen Migrations documentation for a better understanding of how this library works. For example, you might want to know what the "Storage" is meant to do.

[Baleen Migrations on Github](#)

2.1.2 League\Container

Allows for easy customization of almost every aspect of the framework.

[League Container on Github](#)

2.1.3 League\Flysystem

A file-system abstraction library. Used e.g. to read/write config files.

[League Flysystem on Github](#)

2.1.4 League\Tactician

A simple library that implements the CommandBus pattern. More about it in the [Commands](#) section.

[League Tactician on Github](#)

2.1.5 Symfony\Config

A library that facilitates creating and validating configuration schemas. More about it in the [Configuration](#) section.

[Symfony Config on Github](#)

2.1.6 Symfony\Console

A widely-adopted library for interacting with the command-line.

[Symfony Console on Github](#)

2.2 Commands

Service Name: `Services::COMMANDS`

Symfony Console's commands are each of the actions that can be executed by an end-user of the migrations library. However we took a slightly different approach with commands than usual:

2.2.1 Commands vs Messages & Handlers

One of the things that we found annoying in other migration libraries is that almost everything had to be configured before you could even execute a basic library command in the console. For example, in some libraries you can't even get help about available commands without first configuring your database connection - which usually doesn't make much sense.

In order to solve that issue Baleen CLI decouples the Command definition from its execution (this might in fact be a good feature for future SymfonyConsole versions). This decoupling is facilitated by [League\Tactician](#) together with [League\Container](#) to gather application dependencies ONLY once they're actually needed. This allows end-users to query the help for your executable without having to configure a database connection, because the database connection will only be instantiated when its ACTUALLY needed.

Tip: All of this happens behind the scenes so you don't have to worry about it. Refer to the [BaseCommand](#) class for more information.

Since we're using a CommandBus library, we decided to disambiguate the meaning of Command as follows:

- Whenever we refer to Commands we mean [Symfony\Console](#) commands, and we'll mark them with **bold**. E.g. **status**, or **migrate**.
- But whenever we refer to Messages, we refer to a "command message" from the [CommandBus](#) pattern. These messages are classes whose name ends with "Message", e.g. "StatusMessage" or "MigrateMessage" - which correspond to the example commands on the previous item (they provide the command definition to [Symfony\Console](#)).

2.2.2 Customizing Commands

The list of available commands is configured and provided by *CommandsProvider*, which also allows for a flexible way to customize them:

1. Create your own commands provider class.
2. Have it extend *CommandsProvider*.
3. Add a public constructor and use it to customize the contents of the `$commands` protected property. E.g.:

```
/**
 * Example from Doctrine Migrations
 * __construct
 */
public function __construct()
{
    // custom message and custom handler
    $this->commands[Services::CMD_REPOSITORY_CREATE] = [
        'message' => CreateMessage::class,
        'handler' => CreateHandler::class,
    ];

    // customize message only - use default handler
    $this->commands[Services::CMD_TIMELINE_EXECUTE]['message'] = ExecuteMessage::class;

    // custom message and custom handler
    $this->commands[Services::CMD_TIMELINE_MIGRATE] = [
        'message' => MigrateMessage::class,
        'handler' => MigrateHandler::class,
    ];
    BaseCommandsProvider::__construct();
}
```

4. Overwrite the default commands provider by supplying a reference to your own inside `config/providers.php`.

In other words, adding or modifying commands is as simple as extending the default *CommandsProvider* and customizing the `$commands` property to point to the correct messages and handlers. The base *CommandsProvider* will do the rest of the work for you.

2.3 Comparator

Service Name: *Services::COMPARATOR*

The comparator is simply an invokable class that receives two Versions as arguments by implementing *ComparatorInterface*. The reason its a class is that we preferred to make sure we had control over the comparator's signature - if we had accepted closures we would have forced implementations to make many unnecessary parameter type-checks.

When invoked, the comparator must return an integer less than, equal to, or greater than 0 (zero) in order to indicate if the first version is earlier than, equal to, or later than the second version (respectively). By earlier/later we refer to the order in which the versions should be executed.

The Comparator is used at different points in the domain logic so its useful to have it as a service.

The default Comparator used by Baleen CLI is an instance of *DefaultComparator*. The service is registered by default in the *TimelineProvider*.

2.4 Configuration

Service Name: *Services::CONFIG*

The Baleen CLI framework allows you to easily define your configuration’s schema using *Symfony\Config* definitions. A default schema is provided that covers the basic structure.

The default configuration schema is as follows:

```
<?php
[
    // list of Providers to load
    'providers' => [
        'key' => 'My\Provider', // FQCN
        // ...
    ],
    // config for the Repository service
    'migrations' => [
        'directory' => '/path/to/migrations',
        'namespace' => 'Migrations\Namespace', // for autoloading
    ],
    // config for the Storage service
    'storage' => [
        'file' => '/path/to/storage.txt', // path to storage file
    ],
];
```

Throughout this tutorial we refer to specific configuration options using a dot (.) to indicate nesting levels. E.g. the configuration file location can be specified with the “storage.file” option.

Note: The “migrations” section in the configuration corresponds to the Repository. But we decided to name it “migrations” simply because that would be a more recognizable name for end-users who are not familiar with the nomenclature we use at Baleen.

2.4.1 Customizing the Configuration

If you’re building your own migrations library based on Baleen CLI you can easily customize the configuration definition and main configuration class to your needs. If you read any of the other guides most of the process should be familiar by now, but there are a couple of unique considerations - the main configuration file must:

- Provide default configuration values that comply with the *Definition*.
- Accept a configuration that’s compliant to the Definition.

Therefore if you change the configuration class, its likely you’ll have to change the Definition, and vice-versa.

Configuration Class

To customize the configuration class simply create a new class that implements *ConfigInterface* and adds any additional functionality you may need. Then provide the class in the container under the key mentioned at the top of this page.

A set of methods that’s important to mention is *ConfigInterface::getDefaults* and the *Config::get{SECTION}Defaults* family of methods (e.g. *Config::getProviderDefaults*). The main *getDefaults()* method should return a complete array of defaults, which is what will be used to create the end-user’s configuration file when the user runs the `config:init` command. The other methods in the family

are in charge of returning defaults for the specified section. This is not enforced by the interface, but its useful if you only want to customize the “storage” defaults but not the rest. See the next section for further explanation.

Another method that’s important to note is `ConfigInterface::getDefinition`. It must return an instance of `Symfony\Component\Config\Definition\ConfigurationInterface`.

Note: For more information about the methods declared in `ConfigInterface` refer to the referenced docs.

If the changes to your configuration class don’t affect the default definition then just return the default definition. Otherwise read the next section.

Definition Class

The definition class is simply a class that implements `Symfony\Component\Config\Definition\ConfigurationInterface`. So if you want to start your configuration from scratch just implement that interface on a clean class.

Baleen’s configuration definition is broken down into multiple methods to make it easy to customize only certain sections of the configuration. For example, if all you need is to customize the “storage” section of the definition:

1. Create a new Definition class that extends `Definition`.
2. Override the `:php:meth:Definition::addStorageNode` method and supply your own rules for that node.
3. The method must return an instance of `Symfony\Component\Config\Definition\Builder\NodeDefinition`.

The Definition class is not offered as a service. Its only accessible through the main configuration class’ `ConfigInterface::getDefinition` function. So if you have to change the definition you will always have to override the main Config class as well.

2.5 HelperSet

Service Name: `Services::HELPERSET`

The `HelperSetProvider` supplies the `Application` with a `Symfony\Component\Console\Helper\HelperSet`. The individual helpers are also registered as services to allow for easier customization.

2.5.1 Customizing the HelperSet

If only the question or config helpers need to be customized then simply override their service. Otherwise its recommended to replace the `HelperSetProvider` class entirely in order to customize the HelperSet service.

2.6 Providers

Providers are a concept of `League\Container`: they’re simple classes that tell the container about a group of services they can provide. They only register those services in the container once at least one of those services is actually requested from the Container. Its important to keep that in mind.

Note: For more information about how Providers work refer to the `League\Container`: documentation.

Tip: Providers are central to Baleen CLI because they’re the main way services are registered into a container.

2.6.1 Bootstrap

The `bootstrap.php` file is the one that handles the initialization of the Container and adds providers to it. The sequence is as follows:

1. Container is initialized with a few default services.
2. The `ConfigProvider` gets registered as the first provider. This provider must provide the configuration service (named after the `Services::CONFIG` constant, which we'll refer to as the "Config service").
3. The Config service is immediately fetched from the container.
4. All providers supplied by the Config service are then added to the Container. At least one of them must supply the `Service::APPLICATION` service.
5. The Application service is fetched and the `Application::run()` method is executed.

2.6.2 Customizing Providers

Providers can be customized (added, overwritten, removed, etc.) several ways:

1. Directly in the `bin/bootstrap.php` file. Not recommended, unless the provider can't be configured.
2. By modifying the `config/providers.php` file. This is the recommended method if you're building your own migrations library (see the guide).
3. On the end-user's configuration file, under the "providers" section (similar to #2). This just give end-users the possibility to further customize the behaviour of the migrations library if desired.

2.7 Repository

Service Name: `Services::REPOSITORY`

The repository service must either implement `RepositoryInterface` directly or extend `AbstractRepository`, which provides some convenient boilerplate functionality.

The Repository Provider must also provide the `Services::REPOSITORY_FILESYSTEM` and `:php:const'Services::MIGRATION_FACTORY'` services.

The default implementation for this service is `DirectoryRepository`. And unless you need to load migrations for another source other than the filesystem, or from multiple folders at a time, the default implementation should cover the needs of most migration libraries.

2.7.1 Repository Filesystem

Service Name: `Services::REPOSITORY_FILESYSTEM`

This service must be a instance of `League\Flysystem\FilesystemInterface` and is used both to load the migrations and also to create them (the latter with the **migrations:create** command). Its root must therefore point to the folder where those migrations will be located.

By default the repository's folder can be customized by end-users by using the configuration option "migrations.directory".

2.7.2 Migration Factory

Service Name: `Services::MIGRATION_FACTORY`

This service must implement `FactoryInterface`. It's used to instantiate all migrations that are loaded by the repository, which is useful for example if the migration class needs certain dependencies to be injected.

The default implementation used for this service is `SimpleFactory`.

2.8 Storage

Service Name: `Services::STORAGE`

The Storage service must implement `StorageInterface`, and we recommend that it additionally extends `BaleenMigrationsStorageAbstractStorage`. For more information on how to implement `StorageInterface` please refer to the Baleen Migrations documentation.

The default Storage service implementation is the `FileStorage` class, which saves versions into a local file. The file's location can be specified using the configuration option "storage.file", and defaults to `.baleen_versions`.

2.9 Timeline

Service Name: `Services::TIMELINE`

This service provides an instance of `BaleenMigrationsTimeline` and sets its `Comparator`. Since this service almost exclusively relies on other services in the framework, migration libraries will hardly ever need to replace this service.

By default a special factory is used to instantiate the Timeline. The factory is an instance of `TimelineFactory`. When instantiating a Timeline, the factory creates a collection of available migrations by "enriching" the migrations available in the `Repository` service with the state data available from the `Storage` service. In the process, it also makes sure that there's no missing migration files in the Repository (e.g. a version that is reported as migrated in the Storage but that has no corresponding file in the Repository).

In Baleen CLI the default `TimelineProvider` provides also the `Services::COMPARATOR` service. Refer to the `Comparator` page for more information.

2.10 Tutorial: Build Your Own Migrations Library

This guide will provide a general overview of how you can build your own migrations library using the Baleen CLI framework.

We'll use Doctrine Migrations as inspiration for our examples, but keep in mind that you can create a migration library for virtually anything using Baleen.

2.10.1 Doctrine Intro

Since Doctrine Migrations will be our main example for the guide, its worth mentioning some basic things about how Doctrine works and what exactly we're set to accomplish here.

Doctrine provides applications with a flexible database abstraction layer. In other words its the interface between an application and one or more databases. Among other things, Doctrine provides an `ObjectManager` as one of the

entry points to most of the functionality. This “entry point” is an important concept to keep in mind when building a migration library.

What’s important under the scope of this guide is that the purpose of building a migrations library for Doctrine is to provide a mechanism to ensure the database schema - and potentially some data - remains consistent across different environments.

2.10.2 Overview

During the course of this tutorial we’ll accomplish the following key goals:

1. Create the skeleton for our migrations library based in Baleen CLI.
2. Create a Doctrine-specific [Storage](#) service.
3. Create a Doctrine-specific abstract migration class that can be extended by concrete migrations.
4. Link to the application: inject the application’s `ObjectManager` into that Migration class.
5. Customize the configuration schema to offer options to configure the Storage and other aspects of our new library.

2.10.3 Getting Started

The first step of creating your own migrations library is to copy the skeleton of Baleen CLI into a new folder. The skeleton consists of these files:

- `bin/baleen`: entry point for unix systems. All it does is include the bootstrap file.
- `bin/bootstrap.php`: bootstrap script for your application.
- `config/defaults.php`: default values for your configuration file.
- `config/providers.php`: contains a list of providers that will be loaded during bootstrap.
- `src/Application`: specifies you application’s name and version.

You can optionally also copy the test configuration files if you’d like to use a similar approach to testing.

Next initialize Composer (`composer init`) and include the following dependencies:

- `baleen/cli:dev-master`
- `doctrine/dbal`: if you’re building for another library then you’d reference that other library instead, of course.

Then rename `bin/baleen` to whatever makes sense for your new migrations library, e.g. `bin/migrations` - and add a reference to it in your `composer.json` file:

```
{
    "bin": {
        "bin/migrations"
    }
}
```

Finally, customize the `src/Application.php` file to use the name and version of your library.

After all of this you should already be able to test your binary by executing `bin/migrations` in your terminal and see a list of available commands.

At this point you might find it interesting that if you run any of those commands they will WORK, only that the functionality provided is the default Baleen CLI functionality. That’s because we haven’t customized any of the

providers, and therefore all of the providers (:file:config/providers.php) that are being loaded by the bootstrap file are still the default Baleen CLI providers.

2.10.4 Creating the Doctrine Storage Service

By default Baleen CLI stores versions into a `.baleen_versions` file. Since we want to version a database it would make sense to store the database version in the database itself. Doing that is easy. The goal for this step is to create a `DoctrineStorage` class that implements `Baleen\Migrations\Storage\AbstractStorage` and implements all missing functions (such as `doFetchAll()`, `saveCollection()`, etc.).

First create a `Version` entity for Doctrine. All it needs is an “id” field of the type “string”, unique and primary-key. That field will store the id of the version that has been migrated.

Note: The resulting entity class can be found in the Doctrine Migrations repository under `lib/Entity/Version`.

Then create the `DoctrineStorage` class. Extend `AbstractStorage` and implement the missing methods. You can use a constructor (or setters) to inject the dependencies it will need for its methods. For example you’ll want to inject the `ObjectManager`, a doctrine `Repository`, or possibly even both - depending on your needs. You can see a finished `DoctrineStorage` class in the same Github repository linked above under `lib/Storage/DoctrineStorage.php`.

Finally, all that’s left is to declare a service in the Container that will return `DoctrineStorage` under a predefined service name. The predefined service name can simply be referenced using the `Services::STORAGE` constant. Since a `Storage` object is meant to be stateless it can be declared as a singleton. Keep in mind the service factory must take care of injecting the Doctrine dependencies (`Object Manager` and/or `Entity Repository`) during instantiation.

To declare that service simply create a new `Service Provider` class and add it to the list of providers in `config/providers.php` - you can use any string as the key but “storage” is self-explanatory. You should also remove the reference to the default storage provider.

Note: The resulting `Storage Provider` class can be found in the Doctrine Migrations repository under file `lib/Provider/StorageProvider`, which works together with `lib/Provider/DoctrineProvider` to inject the Doctrine dependencies.

With the `DoctrineStorage` service in place would be able to use the migration commands just like you would with vanilla Baleen CLI, only that instead of saving migrated versions in a file they will be saved to a database. It won’t work just yet though.

2.10.5 Creating the Abstract Migration Class

Baleen Migrations provides a default migration class that all concrete migrations can extend: `Baleen\Migrations\Migration\SimpleMigration`. This simple, abstract migrations class extends `Baleen\Migrations\Migration\MigrationInterface` and also `Baleen\Migrations\Migration\Capabilities\OptionsAwareInterface` in order to provide “some” contextual information to the concrete migration implementations. However, in the context of Doctrine Migrations, `SimpleMigration` is not enough. We want concrete migrations to be able to easily access the `ObjectManager`, which we need to inject during or immediately after instantiation.

In order to make that easy, Baleen Migrations offers the ability to specify a “Migration Factory”. This special factory must extend `Baleen\Migrations\Migration\Factory\FactoryInterface`, and in Baleen CLI the factory can simply be offered as a service named after the `Services::MIGRATION_FACTORY` constant. If you read the previous section you should already understand how to do that (hint: you’ll want to replace the “repository” provider).

Note: The resulting Doctrine-specific `MigrationFactory` class can be found in the Doctrine Migrations repository at `lib/Migration/MigrationFactory`. And the provider can be found at `lib/Providers/RepositoryProvider`.

As for the abstract migration class itself, which we'll simply name `AbstractMigration`, it will require Doctrine's `ObjectManager` on the constructor and make it available to concrete migrations as a protected property.

Note: An example of the abstract migration class can also be found in the Doctrine Migrations repository at `lib/Migration/AbstractMigration`.

2.10.6 Linking to the Application

With the `Storage` and `AbstractMigration` classes in place and their respective services properly configured, there's still one more thing to do before the migration commands can be executed: we have to find a way to pick up the application's Object Manager and inject it to both the `Storage` service and the `Migration Factory` service. Regardless of how we approach this, we'll need a bit of the end-user's help.

The way Doctrine typically does this kind of integration (that is: for its other CLI tools) is by allowing the user to load the application's Object Manager into a Console Helper (instance of `Symfony\Component\Console\Helper\Helper`) through a pre-defined integration file at the root of the project named `cli-config.php`. For more information about how the file can make the integration work refer to the Doctrine documentation.

So as far as we're concerned all we need to do is check if that file exists, load it, and then add the resulting Console Helper to our Application class. The code for that can go in the `bin/bootstrap.php` file or in a special provider. In DoctrineMigrations we opted for a more comprehensive approach, and the code for this particular integration approach can be found in `lib/Providers/HelperSetProvider.php`.

An alternative to this approach would be to offer users the possibility to configure access to the database inside the migration configuration file (the one created with the command `bin/migrations init`). We would then pick up the configuration and instantiate our own Object Manager based on it. But that approach requires more work from the end-user (e.g. maintain two separate doctrine configuration files for the same database), so it shouldn't be the primary way to integrate. In Doctrine Migrations we chose to offer end-users the ability to choose freely which of the two approaches they prefer.

Once you're able to access to the application's Object Manager from within the application you can make it a service in the container. Offering it as a service will make your life easier because you can then adjust the definitions of the `Storage` and `MigrationFactory` services to receive the Object Manager service during instantiation.

Note: Its worth mentioning that building a migrations tool for a generic doctrine-based project is harder than building a similar tool for an application framework (like Laravel, Wordpress or Magento, for example). The reason is that application frameworks tend to be convention-driven, which means the "entry point" to their database and their models is a well-known convention. That's not the case with configuraton-driven frameworks (like ZF2 or Symfony for example), which they tend to be harder to integrate with because their "entry point" is not a convention and is therefore essentially unknown until it gets configured by the user.

2.10.7 Custom Configuration Schema

If you followed the previous steps you'd notice that if at this point you issue the **config:init** command your generated configuration file will still have a "storage.file" option (used so the user can choose the name of the file used by `FileStorage` to store the list of migrated versions).

But we're not using file storage anymore - we're using Doctrine Storage - so that configuration option is obsolete. Let's remove it.

You can easily replace the default configuration class (*Config*) by simply creating a new configuration class that extends *ConfigInterface*. Then register that class in the Container (in a similar way as indicated in previous steps) using the *Services::CONFIG* constant. And finally provide the default values for the configuration file by implementing the *ConfigInterface::getDefaults* function.

You can then supply a different "definition" (refer to the `:php:ns:Symfony\Config` module for more information) by creating a new definition class and implementing the *ConfigInterface::getDefinition* function so that it returns your own definition. This custom definition would of course not include rules for the option we want to get rid of.

Next time you run the **config:init** command, the default values from your Config class will be validated against the new Definition, and then written into the file.

You can see examples of a custom configuration file and definition in the Doctrine Migrations repository inside the `lib/Config` folder.

2.10.8 Conclusion

In this guide we built a doctrine-specific migrations library that can be used by developers that use Doctrine in their project. Several other Baleen CLI features were not covered, but if you understood how you can use the Container and its providers to customize Baleen to your needs then this tutorial's ultimate goal has been accomplished, and you should be in good shape to get started on your own.

Tip: If you followed this guide please help us improve it by submitting a pull-request with any changes that you think might be useful for future readers.

3.1 Baleen\Cli

3.1.1 Baleen\Cli\Application

class Application

The entry point to Baleen CLI's commands.

constant VERSION

Version to show in the help / usage message.

property container

protected Container

The LeagueContainer instance used by Baleen CLI.

__construct (*\$commands*, *HelperSet \$helperSet*)

Parameters

- **\$commands** (*SymfonyComponentConsoleCommandCommand[]*) – Array of Commands available for the Application.
- **\$helperSet** (*HelperSet*) –

init (*\$commands*, *HelperSet \$helperSet*)

Parameters

- **\$commands** (*SymfonyComponentConsoleCommandCommand[]*) –
- **\$helperSet** (*HelperSet*) –

3.1.2 Baleen\Cli\BaseCommand

class BaseCommand

The base Command class used to build all the command definitions for the Application.

property container

protected Container

property commandBus

protected CommandBus

A reference to the CommandBus in charge of handling Messages.

property serviceAlias

protected string

property serviceClass

protected string

___**construct** (*ContainerInterface \$container, \$serviceAlias, \$serviceClass*)

Parameters

- **\$container** (*ContainerInterface*) – A reference to the Application’s Container.
- **\$serviceAlias** (*string*) – The key in the Container for the command that the instance of this class represents.
- **\$serviceClass** (*string*) – Needed in order to run certain checks against the class before instantiating it with the container. This helps us make those checks without triggering all the other services through the Container’s DI functionality.

getContainer ()

Returns Container

getCommandBus ()

getCommandBus.

Returns CommandBus

execute (*InputInterface \$input, OutputInterface \$output*)

Executes the current command by retrieving its associated Message from the Container, setting the Input and Output according to what was received as parameters, and finally passing that Message to the CommandBus for handling.

Parameters

- **\$input** (*InputInterface*) – An InputInterface instance
- **\$output** (*OutputInterface*) – An OutputInterface instance

Returns int|null null or 0 if everything went fine, or an error code

configure ()

Calls the message’s static “configure” public function passing \$this as argument to allow the message to configure the command.

3.1.3 Baleen\Cli\CommandBus

Baleen\Cli\CommandBus\AbstractMessage

class AbstractMessage

Class AbstractMessage.

property config

protected ConfigInterface

property input

protected InputInterface

property output

protected OutputInterface

property cliCommand

protected Command

getConfig()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

configure (*Command \$command*)

Configures a console command by setting name, description, arguments, etc.

Parameters

- **\$command** (*Command*) –

Baleen\Cli\CommandBus\Config

Baleen\Cli\CommandBus\Config\AbstractConfigMessage

class AbstractConfigMessage

Shared functionality for all configuration-related Messages.

property configStorage

protected ConfigStorage

property config

protected ConfigInterface

property input

protected InputInterface

property output

protected OutputInterface

property cliCommand

protected Command

getConfigStorage()

Returns ConfigStorage

setConfigStorage (*ConfigStorage \$configStorage*)

Parameters

- **\$configStorage** (*ConfigStorage*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

configure (*Command \$command*)

Configures a console command by setting name, description, arguments, etc.

Parameters

- **\$command** (*Command*) –

Baleen\Cli\CommandBus\Config\InitHandler

class InitHandler

Handles the config:init command.

handle (*InitMessage \$message*)

Handle an InitMessage. Creates an end-user configuration file using default values. If the file already exists it simply exists without doing anything.

Parameters

- **\$message** (*InitMessage*) –

Baleen\Cli\CommandBus\Config\InitMessage

class InitMessage

Message class for the config:init command.

property configStorage
protected ConfigStorage

property config
protected ConfigInterface

property input
protected InputInterface

property output
protected OutputInterface

property cliCommand
protected Command

configure (*Command \$command*)

Parameters

- **\$command** (*Command*) –

getConfigStorage ()

Returns ConfigStorage

setConfigStorage (*ConfigStorage \$configStorage*)

Parameters

- **\$configStorage** (*ConfigStorage*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\CLI\CommandBus\Config>StatusHandler**class StatusHandler**

Handles the config:status command.

property output

protected OutputInterface

property message

protected StatusMessage

handle (*StatusMessage \$message*)

Handles a StatusMessage, which prints the status of the migrations system in a developer-friendly format (inspired by “git status”).

Parameters

- **\$message** (*StatusMessage*) –

printPendingVersion (*Version \$version, \$style*)

Formats and prints a pending version with the given style.

Parameters

- **\$version** (*Version*) – The Version to print.
- **\$style** (*string*) – One of the STYLE_* constants.

getRelativePath (*\$from, \$to*)

Returns the relative path between two known paths.

Parameters

- **\$from** –
- **\$to** –

Returns string

printDiff (*\$versions, \$message, \$style = self::STYLE_INFO*)

Prints an array (group) of Versions all with the given style. If the array is empty then it prints nothing.

Parameters

- **\$versions** (*array*) –
- **\$message** (*string|string[]*) – Message(s) to print before the group of versions.
- **\$style** (*string*) – One of the STYLE_* constants.

splitDiff (*\$diff, \$comparator, Version \$head*)

Splits an array of Versions into two arrays and returns them. The first one contains all versions before the HEAD (latest migrated) Version, and the second one contains all Versions after HEAD. Head is never included in either of the arrays.

Parameters

- **\$diff** (*Version[]*) – The array of Versions that should be split.
- **\$comparator** (*callable*) – The comparator used to sort Versions.
- **\$head** (*Version*) – The HEAD version.

Returns array

Baleen\Cli\CommandBus\Config\StatusMessage**class StatusMessage**

Message class for the config:status command.

property repository

protected RepositoryInterface

property filesystem

protected Filesystem

property storage

protected StorageInterface

property comparator

protected callable

property configStorage

protected ConfigStorage

property config

protected ConfigInterface

property input

protected InputInterface

property output

protected OutputInterface

property cliCommand

protected Command

configure (*Command \$command*)

Configures a console command by setting name, description, arguments, etc.

Parameters

- **\$command** (*Command*) –

getRepository ()

Returns RepositoryInterface

setRepository (*RepositoryInterface \$repository*)**Parameters**

- **\$repository** (*RepositoryInterface*) –

getFilesystem ()

Returns Filesystem

setFilesystem (*Filesystem \$filesystem*)**Parameters**

- **\$filesystem** (*Filesystem*) –

getStorage ()

Returns StorageInterface

setStorage (*StorageInterface \$storage = null*)**Parameters**

- **\$storage** (*StorageInterface*) –

getComparator ()

 Returns callable

setComparator (\$comparator)

 Parameters

- **\$comparator** (*callable*) –

getConfigStorage ()

 Returns ConfigStorage

setConfigStorage (ConfigStorage \$configStorage)

 Parameters

- **\$configStorage** (*ConfigStorage*) –

getConfig ()

setConfig (ConfigInterface \$config)

 Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (InputInterface \$input)

 Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (OutputInterface \$output)

 Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

 Returns Command

setCliCommand (Command \$cliCommand)

 Parameters

- **\$cliCommand** (*Command*) –

Baleen\Cli\CommandBus\Factory

Baleen\Cli\CommandBus\Factory\DefaultFactory

class DefaultFactory
 Class DefaultFactory

create (\$class)
 create

 Parameters

- **\$class** –

Returns AbstractMessage

Baleen\Cli\CommandBus\Factory\MessageFactoryInterface

interface MessageFactoryInterface

create (*\$class*)

create

Parameters

- **\$class** –

Returns AbstractMessage

Baleen\Cli\CommandBus\MessageInterface

interface MessageInterface

Interface MessageInterface.

configure (*Command \$command*)

Configures a console command by setting name, description, arguments, etc.

Parameters

- **\$command** (*Command*) –

getConfig ()

getConfig.

Returns ConfigInterface

setConfig (*ConfigInterface \$config*)

setConfig.

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

getInput.

Returns InputInterface

setInput (*InputInterface \$input*)

setInput.

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

getOutput.

Returns OutputInterface

setOutput (*OutputInterface \$output*)

setOutput.

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()
getCliCommand.

Returns Command

setCliCommand (*Command \$command*)
setCliCommand.

Parameters

- **\$command** (*Command*) –

Baleen\CLI\CommandBus\Repository

Baleen\CLI\CommandBus\Repository\AbstractRepositoryListHandler

class AbstractRepositoryListHandler

Class AbstractRepositoryListHandler.

outputVersions (*LinkedVersions \$versions, OutputInterface \$output*)

Parameters

- **\$versions** (*LinkedVersions*) –
- **\$output** (*OutputInterface*) –

getCollection (*RepositoryInterface \$repository, \$comparator = null*)
getCollection.

Parameters

- **\$repository** (*RepositoryInterface*) –
- **\$comparator** –

Returns LinkedVersions

Baleen\CLI\CommandBus\Repository\AbstractRepositoryMessage

class AbstractRepositoryMessage

Class AbstractRepositoryMessage.

property repository
protected RepositoryInterface

property filesystem
protected Filesystem

property comparator
protected callable

property config
protected ConfigInterface

property input
protected InputInterface

property output
protected OutputInterface

property cliCommand
protected Command

getRepository()
Returns RepositoryInterface

setRepository (RepositoryInterface \$repository)

Parameters

- **\$repository** (RepositoryInterface) –

getFileSystem()
Returns Filesystem

setFileSystem (Filesystem \$filesystem)

Parameters

- **\$filesystem** (Filesystem) –

getComparator()
Returns callable

setComparator (\$comparator)

Parameters

- **\$comparator** (callable) –

getConfig()

setConfig (ConfigInterface \$config)

Parameters

- **\$config** (ConfigInterface) –

getInput()

setInput (InputInterface \$input)

Parameters

- **\$input** (InputInterface) –

getOutput()

setOutput (OutputInterface \$output)

Parameters

- **\$output** (OutputInterface) –

getCliCommand()
Returns Command

setCliCommand (Command \$cliCommand)

Parameters

- **\$cliCommand** (Command) –

configure (Command \$command)
Configures a console command by setting name, description, arguments, etc.

Parameters

- **\$command** (*Command*) –

Baleen\Cli\CommandBus\Repository\CreateHandler

class **CreateHandler**

Class CreateHandler.

handle (*CreateMessage \$command*)
handle.

Parameters

- **\$command** (*CreateMessage*) –

Returns false|string

generate (*\$className, \$namespace = null*)

Parameters

- **\$className** (*string*) –
- **\$namespace** (*string*) –

Returns ClassGenerator

writeClass (*ClassGenerator \$class, Filesystem \$filesystem, \$destinationDir*)
Function writeClass.

Parameters

- **\$class** (*ClassGenerator*) –
- **\$filesystem** (*Filesystem*) –
- **\$destinationDir** –

Returns string|false

Baleen\Cli\CommandBus\Repository\CreateMessage

class **CreateMessage**

Class CreateMessage.

property repository
protected RepositoryInterface

property filesystem
protected Filesystem

property comparator
protected callable

property config
protected ConfigInterface

property input
protected InputInterface

property output
protected OutputInterface

property cliCommand
protected Command

configure (*Command \$command*)

Parameters

- **\$command** (*Command*) –

getRepository ()

Returns RepositoryInterface

setRepository (*RepositoryInterface \$repository*)

Parameters

- **\$repository** (*RepositoryInterface*) –

getFileSystem ()

Returns Filesystem

setFileSystem (*Filesystem \$filesystem*)

Parameters

- **\$filesystem** (*Filesystem*) –

getComparator ()

Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\CLI\CommandBus\Repository\LatestHandler

class LatestHandler

Class LatestHandler.

handle (*LatestMessage \$command*)
handle.

Parameters

- **\$command** (*LatestMessage*) –

outputVersions (*LinkedVersions \$versions, OutputInterface \$output*)

Parameters

- **\$versions** (*LinkedVersions*) –
- **\$output** (*OutputInterface*) –

getCollection (*RepositoryInterface \$repository, \$comparator = null*)
getCollection.

Parameters

- **\$repository** (*RepositoryInterface*) –
- **\$comparator** –

Returns *LinkedVersions*

Baleen\CLI\CommandBus\Repository\LatestMessage

class LatestMessage

Class ListMessage.

property repository
protected *RepositoryInterface*

property filesystem
protected *Filesystem*

property comparator
protected callable

property config
protected *ConfigInterface*

property input
protected *InputInterface*

property output
protected *OutputInterface*

property cliCommand
protected *Command*

configure (*Command \$command*)
configure.

Parameters

- **\$command** (*Command*) –

getRepository()

Returns RepositoryInterface

setRepository (*RepositoryInterface \$repository*)

Parameters

- **\$repository** (*RepositoryInterface*) –

getFileSystem()

Returns Filesystem

setFileSystem (*Filesystem \$filesystem*)

Parameters

- **\$filesystem** (*Filesystem*) –

getComparator()

Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

getConfig()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\CLI\CommandBus\Repository\ListHandler

class ListHandler

Class ListHandler.

handle (*ListMessage \$command*)
handle.

Parameters

- **\$command** (*ListMessage*) –

outputVersions (*LinkedVersions \$versions, OutputInterface \$output*)

Parameters

- **\$versions** (*LinkedVersions*) –
- **\$output** (*OutputInterface*) –

getCollection (*RepositoryInterface \$repository, \$comparator = null*)
getCollection.

Parameters

- **\$repository** (*RepositoryInterface*) –
- **\$comparator** –

Returns *LinkedVersions*

Baleen\CLI\CommandBus\Repository\ListMessage

class ListMessage

Class ListMessage.

property repository
protected *RepositoryInterface*

property filesystem
protected *Filesystem*

property comparator
protected callable

property config
protected *ConfigInterface*

property input
protected *InputInterface*

property output
protected *OutputInterface*

property cliCommand
protected *Command*

configure (*Command \$command*)
configure.

Parameters

- **\$command** (*Command*) –

getRepository()
Returns RepositoryInterface

setRepository (*RepositoryInterface \$repository*)

Parameters

- **\$repository** (*RepositoryInterface*) –

getFileSystem()
Returns Filesystem

setFileSystem (*Filesystem \$filesystem*)

Parameters

- **\$filesystem** (*Filesystem*) –

getComparator()
Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

getConfig()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand()
Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\Cli\CommandBus\Storage

Baleen\Cli\CommandBus\Storage\AbstractStorageMessage

class **AbstractStorageMessage**

Class AbstractStorageMessage.

property storage

protected StorageInterface

property comparator

protected callable

property config

protected ConfigInterface

property input

protected InputInterface

property output

protected OutputInterface

property cliCommand

protected Command

getStorage ()

Returns StorageInterface

setStorage (*StorageInterface \$storage = null*)

Parameters

- **\$storage** (*StorageInterface*) –

getComparator ()

Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

configure (*Command \$command*)

Configures a console command by setting name, description, arguments, etc.

Parameters

- **\$command** (*Command*) –

Baleen\Cli\CommandBus\Storage\LatestHandler

class LatestHandler

Class LatestHandler.

handle (*LatestMessage \$command*)

handle.

Parameters

- **\$command** (*LatestMessage*) –

Baleen\Cli\CommandBus\Storage\LatestMessage

class LatestMessage

Class ListMessage.

property storage

protected StorageInterface

property comparator

protected callable

property config

protected ConfigInterface

property input

protected InputInterface

property output

protected OutputInterface

property cliCommand

protected Command

configure (*Command \$command*)

Parameters

- **\$command** (*Command*) –

getStorage ()

Returns StorageInterface

setStorage (*StorageInterface \$storage = null*)

Parameters

- **\$storage** (*StorageInterface*) –

getComparator ()

Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\Cli\CommandBus\Timeline

Baleen\Cli\CommandBus\Timeline\AbstractTimelineCommand

class AbstractTimelineCommand

Class AbstractTimelineCommand.

property storage

protected StorageInterface

property timeline

protected Timeline

property config

protected ConfigInterface

property input
protected InputInterface

property output
protected OutputInterface

property cliCommand
protected Command

configure (*Command \$command*)

Parameters

- **\$command** (*Command*) –

getStorage ()

Returns StorageInterface

setStorage (*StorageInterface \$storage = null*)

Parameters

- **\$storage** (*StorageInterface*) –

getTimeline ()

Returns Timeline

setTimeline (*Timeline \$timeline*)

Parameters

- **\$timeline** (*Timeline*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\Cli\CommandBus\Timeline\ExecuteHandler

class **ExecuteHandler**

Class ExecuteHandler.

handle (*ExecuteMessage \$command*)

Parameters

- **\$command** (*ExecuteMessage*) –

Baleen\Cli\CommandBus\Timeline\ExecuteMessage

class **ExecuteMessage**

Class ExecuteMessage.

property storage

protected *StorageInterface*

property timeline

protected *Timeline*

property config

protected *ConfigInterface*

property input

protected *InputInterface*

property output

protected *OutputInterface*

property cliCommand

protected *Command*

configure (*Command \$command*)

Parameters

- **\$command** (*Command*) –

getStorage ()

Returns *StorageInterface*

setStorage (*StorageInterface \$storage = null*)

Parameters

- **\$storage** (*StorageInterface*) –

getTimeline ()

Returns *Timeline*

setTimeline (*Timeline \$timeline*)

Parameters

- **\$timeline** (*Timeline*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\Cli\CommandBus\Timeline\MigrateHandler

class MigrateHandler

Class MigrateHandler.

property progress

protected ProgressBar

property saveChanges

protected bool

property strategies

protected array

property trackProgress

protected bool

property output

protected OutputInterface

property command

protected MigrateMessage

handle (*MigrateMessage \$command*)

handle.

Parameters

- **\$command** (*MigrateMessage*) –

attachEvents (*OutputInterface \$output, EventDispatcherInterface \$dispatcher*)

Parameters

- **\$output** (*OutputInterface*) –

- **\$dispatcher** (*EventDispatcherInterface*) –

saveVersionListener (*MigrationEvent \$event*)
saveVersionListener.

Parameters

- **\$event** (*MigrationEvent*) –

onMigrationBefore (*MigrationEvent \$event*)

Parameters

- **\$event** (*MigrationEvent*) –

onMigrationAfter (*MigrationEvent \$event*)
onMigrationAfter.

Parameters

- **\$event** (*MigrationEvent*) –

onCollectionBefore (*CollectionEvent \$event*)
onCollectionBefore.

Parameters

- **\$event** (*CollectionEvent*) –

onCollectionAfter ()
onCollectionAfter.

getStrategyOption (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

Returns string

Baleen\CLI\CommandBus\Timeline\MigrateMessage

class MigrateMessage

Class MigrateMessage.

property storage

protected StorageInterface

property timeline

protected Timeline

property config

protected ConfigInterface

property input

protected InputInterface

property output

protected OutputInterface

property cliCommand

protected Command

configure (*Command \$command*)

Parameters

- **\$command** (*Command*) –

getStorage ()

Returns StorageInterface

setStorage (*StorageInterface \$storage = null*)

Parameters

- **\$storage** (*StorageInterface*) –

getTimeline ()

Returns Timeline

setTimeline (*Timeline \$timeline*)

Parameters

- **\$timeline** (*Timeline*) –

getConfig ()

setConfig (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

getInput ()

setInput (*InputInterface \$input*)

Parameters

- **\$input** (*InputInterface*) –

getOutput ()

setOutput (*OutputInterface \$output*)

Parameters

- **\$output** (*OutputInterface*) –

getCliCommand ()

Returns Command

setCliCommand (*Command \$cliCommand*)

Parameters

- **\$cliCommand** (*Command*) –

Baleen\Cli\CommandBus\Util

Baleen\Cli\CommandBus\Util\ComparatorAwareInterface

interface ComparatorAwareInterface

getComparator ()

Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

Baleen\CLI\CommandBus\Util\ComparatorAwareTrait

trait ComparatorAwareTrait

property comparator
protected callable

getComparator ()

Returns callable

setComparator (*\$comparator*)

Parameters

- **\$comparator** (*callable*) –

Baleen\CLI\CommandBus\Util\ConfigStorageAwareInterface

interface ConfigStorageAwareInterface

Interface ConfigStorageAwareInterface.

getConfigStorage ()
getConfigStorage.

Returns mixed

setConfigStorage (*ConfigStorage \$configStorage*)
setConfigStorage.

Parameters

- **\$configStorage** (*ConfigStorage*) –

Returns mixed

Baleen\CLI\CommandBus\Util\ConfigStorageAwareTrait

trait ConfigStorageAwareTrait

Class ConfigStorageAwareTrait.

property configStorage
protected ConfigStorage

getConfigStorage ()

Returns ConfigStorage

setConfigStorage (*ConfigStorage \$configStorage*)

Parameters

- **\$configStorage** (*ConfigStorage*) –

Baleen\CLI\CommandBus\Util\FilesystemAwareInterface**interface FilesystemAwareInterface****getFilesystem()****Returns** Filesystem**setFilesystem** (*Filesystem \$filesystem*)**Parameters**

- **\$filesystem** (*Filesystem*) –

Baleen\CLI\CommandBus\Util\FilesystemAwareTrait**trait FilesystemAwareTrait**

Class FilesystemAwareTrait.

property filesystem

protected Filesystem

getFilesystem()**Returns** Filesystem**setFilesystem** (*Filesystem \$filesystem*)**Parameters**

- **\$filesystem** (*Filesystem*) –

Baleen\CLI\CommandBus\Util\RepositoryAwareInterface**interface RepositoryAwareInterface****getRepository()****Returns** RepositoryInterface**setRepository** (*RepositoryInterface \$repository*)**Parameters**

- **\$repository** (*RepositoryInterface*) –

getFilesystem()**Returns** Filesystem**setFilesystem** (*Filesystem \$filesystem*)**Parameters**

- **\$filesystem** (*Filesystem*) –

Baleen\CLI\CommandBus\Util\RepositoryAwareTrait

trait RepositoryAwareTrait

Class RepositoryAwareTrait.

property repository

protected RepositoryInterface

property filesystem

protected Filesystem

getRepository ()

Returns RepositoryInterface

setRepository (RepositoryInterface \$repository)

Parameters

- **\$repository** (*RepositoryInterface*) –

getFileSystem ()

Returns Filesystem

setFileSystem (Filesystem \$filesystem)

Parameters

- **\$filesystem** (*Filesystem*) –

Baleen\CLI\CommandBus\Util\StorageAwareInterface

interface StorageAwareInterface

Interface StorageAwareInterface.

getStorage ()

Returns StorageInterface

setStorage (StorageInterface \$storage = null)

Parameters

- **\$storage** (*StorageInterface*) –

Baleen\CLI\CommandBus\Util\StorageAwareTrait

trait StorageAwareTrait

property storage

protected StorageInterface

getStorage ()

Returns StorageInterface

setStorage (StorageInterface \$storage = null)

Parameters

- **\$storage** (*StorageInterface*) –

Baleen\Cli\CommandBus\Util\TimelineAwareInterface**interface TimelineAwareInterface**

Class AbstractTimelineCommand.

getTimeline()**Returns** Timeline**setTimeline** (*Timeline \$timeline*)**Parameters**

- **\$timeline** (*Timeline*) –

Baleen\Cli\CommandBus\Util\TimelineAwareTrait**trait TimelineAwareTrait****property timeline**

protected Timeline

getTimeline()**Returns** Timeline**setTimeline** (*Timeline \$timeline*)**Parameters**

- **\$timeline** (*Timeline*) –

3.1.4 Baleen\Cli\Config**Baleen\Cli\Config\Config****class Config**

Class Config.

__construct (*\$config = []*, *\$defaults = true*)**Parameters**

- **\$config** (*array*) –
- **\$defaults** –

getDefaults()**Returns** array**getMigrationDefaults()**

Default values for the migrations section.

Returns array**getStorageDefaults()**

Default values for the storage section.

Returns array

getProviderDefaults ()

Default values for the providers section.

Returns array

getProviders ()

getProviders.

Returns array

getMigrationsDirectoryPath ()

Returns string

getMigrationsDirectory ()

Returns mixed

getMigrationsNamespace ()

Returns string

getStorageFilePath ()

Returns string

getStorageFile ()

Returns mixed

getConfigFilePath ()

Returns string

getFileName ()

Returns string

toArray ()

Returns array

getDefinition ()

getDefinition.

Returns Definition

getCleanArray ()

Returns a clone of itself but only with settings that can be configured by the end-user.

Returns array

Baleen\CLI\Config\ConfigInterface

interface ConfigInterface

Interface ConfigInterface provides a common interface to be extended for applications based on this framework. The resulting class will provide the config configClass (see Symfony's ConfigurationInterface).

getDefaults ()

Returns an array of default values.

Returns array

toArray ()

Returns the entire configuration as an array,.

Returns array

getFileName ()
Returns the default configuration file name.
Returns string

getDefinition ()
Returns an instance of the configuration definition.
Returns ConfigurationInterface

getCleanArray ()
Returns an array only with settings that can be configured by the end-user.
Returns array

Baleen\Cli\Config\ConfigStorage

class ConfigStorage
Class ConfigStorage.

property projectFileSystem
protected FileSystemInterface

property localConfigStack
protected array

property processor
protected Processor

property configClass
protected string

property definition
protected ConfigurationInterface

property defaultFileName
protected string

__construct (*\$configClass, FileSystemInterface \$projectFileSystem, \$localConfigStack = []*)
ConfigStorage constructor.

Parameters

- **\$configClass** –
- **\$projectFileSystem** (*FileSystemInterface*) –
- **\$localConfigStack** –

load (*\$configFileName = null*)

Parameters

- **\$configFileName** (*string*) – The path to the consumer’s config file (eg .baleen.yml) relative to the project filesystem

Returns Config

write (*ConfigInterface \$config*)

Parameters

- **\$config** (*ConfigInterface*) –

Returns bool

isInitialized (*ConfigInterface \$config*)

Returns whether the specified configuration has an existing user-facing config file.

Parameters

- **\$config** (*ConfigInterface*) –

Returns bool

Baleen\CLI\Config\Definition

class Definition

Class Definition.

getConfigTreeBuilder ()

addMigrationsNode ()

addMigrationsNode.

Returns SymfonyComponentConfigDefinitionBuilderNodeDefinition

addStorageNode ()

addStorageNode.

Returns SymfonyComponentConfigDefinitionBuilderNodeDefinition

3.1.5 Baleen\CLI\Exception

Baleen\CLI\Exception\CliException

class CliException

Class CliException.

property message

protected

property code

protected

property file

protected

property line

protected

__clone ()

__construct (*\$message, \$code, \$previous*)

Parameters

- **\$message** –
- **\$code** –
- **\$previous** –

__wakeup ()

getMessage ()

getCode ()

```
getFile()  
getLine()  
getTrace()  
getPrevious()  
getTraceAsString()  
__toString()
```

3.1.6 Baleen\Cli\Provider

Baleen\Cli\Provider\ApplicationProvider

class ApplicationProvider

Class ApplicationProvider.

property provides

protected

register()

Use the register method to register items with the container via the protected `$this->container` property or the `getContainer` method from the ContainerAwareTrait.

Baleen\Cli\Provider\CommandsProvider

class CommandsProvider

Class CommandsProvider.

property provides

protected

property commands

protected array

__construct()

register()

Use the register method to register items with the container via the protected `$this->container` property or the `getContainer` method from the ContainerAwareTrait.

Baleen\Cli\Provider\ConfigProvider

class ConfigProvider

Class ConfigProvider.

property provides

protected

register()

Use the register method to register items with the container via the protected `$this->container` property or the `getContainer` method from the ContainerAwareTrait.

Baleen\Cli\Provider\HelperSetProvider

```
class HelperSetProvider  
    Class HelperSetProvider.  
  
    property provides  
        protected  
  
    register ()
```

Baleen\Cli\Provider\RepositoryProvider

```
class RepositoryProvider  
    Class RepositoryProvider.  
  
    property provides  
        protected  
  
    register ()
```

Baleen\Cli\Provider\Services**interface Services**

This interface contains constants for the names of services in the Service Container. Its useful in order to: A) reduce the coupling between classes for service providers (since they can use a single interface to reference services) and B) provide an easy way to override certain services for libraries that use Baleen CLI as their framework.

```
constant CONFIG  
    Reference to the Config service  
  
constant CONFIG_STORAGE  
    Reference to the ConfigStorage service  
  
constant BALEEN_BASE_DIR  
    Reference to Baleen CLI's base directory  
  
constant COMMAND_BUS  
    Reference to the CommandBus service  
  
constant COMMANDS  
    Reference to an array of available commands  
  
constant CMD_CONFIG_INIT  
    Reference to the config:init command  
  
constant CMD_CONFIG_STATUS  
    Reference to the config:status command  
  
constant CMD_TIMELINE_EXECUTE  
    Reference to the timeline:execute command  
  
constant CMD_TIMELINE_MIGRATE  
    Reference to the timelien:migrate command  
  
constant CMD_REPOSITORY_CREATE  
    Reference to the repository:create command  
  
constant CMD_REPOSITORY_LATEST  
    Reference to the repository:latest command
```

constant CMD_REPOSITORY_LIST
Reference to the repository:list command

constant CMD_STORAGE_LATEST
Reference to the storage:latest command

constant APPLICATION
Reference to the Symfony Console Application instance

constant AUTOLOADER
Reference to the Composer autoloader

constant HELPERSET
Reference to the Symfony Console HelperSet to be used for the Application

constant HELPERSET_QUESTION
Reference to the Question console helper

constant HELPERSET_CONFIG
Reference to the Config console helper

constant REPOSITORY
Reference to the Repository service

constant REPOSITORY_FILESYSTEM
Reference to the Filesystem to be used for the Repository service

constant MIGRATION_FACTORY
Reference to the factory to be used to instantiate Migrations

constant STORAGE
Reference to the Storage service

constant TIMELINE
Reference to the Timeline service

constant COMPARATOR
Reference to the Comparator service

Baleen\Cli\Provider\StorageProvider

class StorageProvider
Class StorageProvider.

property provides
protected

register()

Baleen\Cli\Provider\TimelineProvider

class TimelineProvider
Class TimelineProvider.

property provides
protected

register()

Indices and Tables

- `genindex`

b

Baleen\Cli\Config, 8
Baleen\Cli\Provider, 9
Baleen\Migrations, 5

l

League\Container, 5
League\Flysystem, 5
League\Tactician, 6

s

Symfony\Config, 6
Symfony\Console, 6

Symbols

__clone() (CliException method), 48
 __construct() (Application method), 17
 __construct() (BaseCommand method), 18
 __construct() (CliException method), 48
 __construct() (CommandsProvider method), 49
 __construct() (Config method), 45
 __construct() (ConfigStorage method), 47
 __toString() (CliException method), 49
 __wakeup() (CliException method), 48

A

AbstractConfigMessage (class), 19
 AbstractMessage (class), 18
 AbstractRepositoryListHandler (class), 26
 AbstractRepositoryMessage (class), 26
 AbstractStorageMessage (class), 34
 AbstractTimelineCommand (class), 36
 addMigrationsNode() (Definition method), 48
 addStorageNode() (Definition method), 48
 Application (class), 17
 Application::VERSION (class constant), 17
 ApplicationProvider (class), 49
 attachEvents() (MigrateHandler method), 39

B

Baleen\Cli\Config (namespace), 8
 Baleen\Cli\Provider (namespace), 6–9
 Baleen\Migrations (namespace), 5
 BaseCommand (class), 17

C

cliCommand (AbstractConfigMessage property), 19
 cliCommand (AbstractMessage property), 18
 cliCommand (AbstractRepositoryMessage property), 26
 cliCommand (AbstractStorageMessage property), 34
 cliCommand (AbstractTimelineCommand property), 37
 cliCommand (CreateMessage property), 28
 cliCommand (ExecuteMessage property), 38
 cliCommand (InitMessage property), 21

cliCommand (LatestMessage property), 30, 35
 cliCommand (ListMessage property), 32
 cliCommand (MigrateMessage property), 40
 cliCommand (StatusMessage property), 23
 CliException (class), 48
 code (CliException property), 48
 command (MigrateHandler property), 39
 commandBus (BaseCommand property), 17
 commands (CommandsProvider property), 49
 CommandsProvider (class), 49
 comparator (AbstractRepositoryMessage property), 26
 comparator (AbstractStorageMessage property), 34
 comparator (ComparatorAwareTrait property), 42
 comparator (CreateMessage property), 28
 comparator (LatestMessage property), 30, 35
 comparator (ListMessage property), 32
 comparator (StatusMessage property), 23
 ComparatorAwareInterface (interface), 41
 ComparatorAwareTrait (trait), 42
 config (AbstractConfigMessage property), 19
 config (AbstractMessage property), 18
 config (AbstractRepositoryMessage property), 26
 config (AbstractStorageMessage property), 34
 config (AbstractTimelineCommand property), 36
 Config (class), 45
 config (CreateMessage property), 28
 config (ExecuteMessage property), 38
 config (InitMessage property), 21
 config (LatestMessage property), 30, 35
 config (ListMessage property), 32
 config (MigrateMessage property), 40
 config (StatusMessage property), 23
 configClass (ConfigStorage property), 47
 ConfigInterface (interface), 46
 ConfigProvider (class), 49
 configStorage (AbstractConfigMessage property), 19
 ConfigStorage (class), 47
 configStorage (ConfigStorageAwareTrait property), 42
 configStorage (InitMessage property), 20
 configStorage (StatusMessage property), 23
 ConfigStorageAwareInterface (interface), 42

ConfigStorageAwareTrait (trait), [42](#)
 configure() (AbstractConfigMessage method), [20](#)
 configure() (AbstractMessage method), [19](#)
 configure() (AbstractRepositoryMessage method), [27](#)
 configure() (AbstractStorageMessage method), [35](#)
 configure() (AbstractTimelineCommand method), [37](#)
 configure() (BaseCommand method), [18](#)
 configure() (CreateMessage method), [29](#)
 configure() (ExecuteMessage method), [38](#)
 configure() (InitMessage method), [21](#)
 configure() (LatestMessage method), [30](#), [35](#)
 configure() (ListMessage method), [32](#)
 configure() (MessageInterface method), [25](#)
 configure() (MigrateMessage method), [40](#)
 configure() (StatusMessage method), [23](#)
 container (Application property), [17](#)
 container (BaseCommand property), [17](#)
 create() (DefaultFactory method), [24](#)
 create() (MessageFactoryInterface method), [25](#)
 CreateHandler (class), [28](#)
 CreateMessage (class), [28](#)

D

DefaultFactory (class), [24](#)
 defaultFileName (ConfigStorage property), [47](#)
 Definition (class), [48](#)
 definition (ConfigStorage property), [47](#)

E

execute() (BaseCommand method), [18](#)
 ExecuteHandler (class), [38](#)
 ExecuteMessage (class), [38](#)

F

file (CliException property), [48](#)
 filesystem (AbstractRepositoryMessage property), [26](#)
 filesystem (CreateMessage property), [28](#)
 filesystem (FilesystemAwareTrait property), [43](#)
 filesystem (LatestMessage property), [30](#)
 filesystem (ListMessage property), [32](#)
 filesystem (RepositoryAwareTrait property), [44](#)
 filesystem (StatusMessage property), [23](#)
 FilesystemAwareInterface (interface), [43](#)
 FilesystemAwareTrait (trait), [43](#)

G

generate() (CreateHandler method), [28](#)
 getCleanArray() (Config method), [46](#)
 getCleanArray() (ConfigInterface method), [47](#)
 getCliCommand() (AbstractConfigMessage method), [20](#)
 getCliCommand() (AbstractMessage method), [19](#)
 getCliCommand() (AbstractRepositoryMessage method), [27](#)

getCliCommand() (AbstractStorageMessage method), [35](#)
 getCliCommand() (AbstractTimelineCommand method), [37](#)
 getCliCommand() (CreateMessage method), [29](#)
 getCliCommand() (ExecuteMessage method), [39](#)
 getCliCommand() (InitMessage method), [21](#)
 getCliCommand() (LatestMessage method), [31](#), [36](#)
 getCliCommand() (ListMessage method), [33](#)
 getCliCommand() (MessageInterface method), [26](#)
 getCliCommand() (MigrateMessage method), [41](#)
 getCliCommand() (StatusMessage method), [24](#)
 getCode() (CliException method), [48](#)
 getCollection() (AbstractRepositoryListHandler method), [26](#)
 getCollection() (LatestHandler method), [30](#)
 getCollection() (ListHandler method), [32](#)
 getCommandBus() (BaseCommand method), [18](#)
 getComparator() (AbstractRepositoryMessage method), [27](#)
 getComparator() (AbstractStorageMessage method), [34](#)
 getComparator() (ComparatorAwareInterface method), [41](#)
 getComparator() (ComparatorAwareTrait method), [42](#)
 getComparator() (CreateMessage method), [29](#)
 getComparator() (LatestMessage method), [31](#), [36](#)
 getComparator() (ListMessage method), [33](#)
 getComparator() (StatusMessage method), [24](#)
 getConfig() (AbstractConfigMessage method), [20](#)
 getConfig() (AbstractMessage method), [18](#)
 getConfig() (AbstractRepositoryMessage method), [27](#)
 getConfig() (AbstractStorageMessage method), [34](#)
 getConfig() (AbstractTimelineCommand method), [37](#)
 getConfig() (CreateMessage method), [29](#)
 getConfig() (ExecuteMessage method), [38](#)
 getConfig() (InitMessage method), [21](#)
 getConfig() (LatestMessage method), [31](#), [36](#)
 getConfig() (ListMessage method), [33](#)
 getConfig() (MessageInterface method), [25](#)
 getConfig() (MigrateMessage method), [41](#)
 getConfig() (StatusMessage method), [24](#)
 getConfigFilePath() (Config method), [46](#)
 getConfigStorage() (AbstractConfigMessage method), [19](#)
 getConfigStorage() (ConfigStorageAwareInterface method), [42](#)
 getConfigStorage() (ConfigStorageAwareTrait method), [42](#)
 getConfigStorage() (InitMessage method), [21](#)
 getConfigStorage() (StatusMessage method), [24](#)
 getConfigTreeBuilder() (Definition method), [48](#)
 getContainer() (BaseCommand method), [18](#)
 getDefaults() (Config method), [45](#)
 getDefaults() (ConfigInterface method), [46](#)
 getDefinition() (Config method), [46](#)
 getDefinition() (ConfigInterface method), [47](#)

getFile() (CliException method), [48](#)
 getFileName() (Config method), [46](#)
 getFileName() (ConfigInterface method), [47](#)
 getFileSystem() (AbstractRepositoryMessage method), [27](#)
 getFileSystem() (CreateMessage method), [29](#)
 getFileSystem() (FilesystemAwareInterface method), [43](#)
 getFileSystem() (FilesystemAwareTrait method), [43](#)
 getFileSystem() (LatestMessage method), [31](#)
 getFileSystem() (ListMessage method), [33](#)
 getFileSystem() (RepositoryAwareInterface method), [43](#)
 getFileSystem() (RepositoryAwareTrait method), [44](#)
 getFileSystem() (StatusMessage method), [23](#)
 getInput() (AbstractConfigMessage method), [20](#)
 getInput() (AbstractMessage method), [19](#)
 getInput() (AbstractRepositoryMessage method), [27](#)
 getInput() (AbstractStorageMessage method), [34](#)
 getInput() (AbstractTimelineCommand method), [37](#)
 getInput() (CreateMessage method), [29](#)
 getInput() (ExecuteMessage method), [39](#)
 getInput() (InitMessage method), [21](#)
 getInput() (LatestMessage method), [31](#), [36](#)
 getInput() (ListMessage method), [33](#)
 getInput() (MessageInterface method), [25](#)
 getInput() (MigrateMessage method), [41](#)
 getInput() (StatusMessage method), [24](#)
 getLine() (CliException method), [49](#)
 getMessage() (CliException method), [48](#)
 getMigrationDefaults() (Config method), [45](#)
 getMigrationsDirectory() (Config method), [46](#)
 getMigrationsDirectoryPath() (Config method), [46](#)
 getMigrationsNamespace() (Config method), [46](#)
 getOutput() (AbstractConfigMessage method), [20](#)
 getOutput() (AbstractMessage method), [19](#)
 getOutput() (AbstractRepositoryMessage method), [27](#)
 getOutput() (AbstractStorageMessage method), [34](#)
 getOutput() (AbstractTimelineCommand method), [37](#)
 getOutput() (CreateMessage method), [29](#)
 getOutput() (ExecuteMessage method), [39](#)
 getOutput() (InitMessage method), [21](#)
 getOutput() (LatestMessage method), [31](#), [36](#)
 getOutput() (ListMessage method), [33](#)
 getOutput() (MessageInterface method), [25](#)
 getOutput() (MigrateMessage method), [41](#)
 getOutput() (StatusMessage method), [24](#)
 getPrevious() (CliException method), [49](#)
 getProviderDefaults() (Config method), [45](#)
 getProviders() (Config method), [46](#)
 getRelativePath() (StatusHandler method), [22](#)
 getRepository() (AbstractRepositoryMessage method), [27](#)
 getRepository() (CreateMessage method), [29](#)
 getRepository() (LatestMessage method), [31](#)
 getRepository() (ListMessage method), [32](#)

getRepository() (RepositoryAwareInterface method), [43](#)
 getRepository() (RepositoryAwareTrait method), [44](#)
 getRepository() (StatusMessage method), [23](#)
 getStorage() (AbstractStorageMessage method), [34](#)
 getStorage() (AbstractTimelineCommand method), [37](#)
 getStorage() (ExecuteMessage method), [38](#)
 getStorage() (LatestMessage method), [35](#)
 getStorage() (MigrateMessage method), [41](#)
 getStorage() (StatusMessage method), [23](#)
 getStorage() (StorageAwareInterface method), [44](#)
 getStorage() (StorageAwareTrait method), [44](#)
 getStorageDefaults() (Config method), [45](#)
 getStorageFile() (Config method), [46](#)
 getStorageFilePath() (Config method), [46](#)
 getStrategyOption() (MigrateHandler method), [40](#)
 getTimeline() (AbstractTimelineCommand method), [37](#)
 getTimeline() (ExecuteMessage method), [38](#)
 getTimeline() (MigrateMessage method), [41](#)
 getTimeline() (TimelineAwareInterface method), [45](#)
 getTimeline() (TimelineAwareTrait method), [45](#)
 getTrace() (CliException method), [49](#)
 getTraceAsString() (CliException method), [49](#)

H

handle() (CreateHandler method), [28](#)
 handle() (ExecuteHandler method), [38](#)
 handle() (InitHandler method), [20](#)
 handle() (LatestHandler method), [30](#), [35](#)
 handle() (ListHandler method), [32](#)
 handle() (MigrateHandler method), [39](#)
 handle() (StatusHandler method), [22](#)
 HelperSetProvider (class), [50](#)

I

init() (Application method), [17](#)
 InitHandler (class), [20](#)
 InitMessage (class), [20](#)
 input (AbstractConfigMessage property), [19](#)
 input (AbstractMessage property), [18](#)
 input (AbstractRepositoryMessage property), [26](#)
 input (AbstractStorageMessage property), [34](#)
 input (AbstractTimelineCommand property), [36](#)
 input (CreateMessage property), [28](#)
 input (ExecuteMessage property), [38](#)
 input (InitMessage property), [21](#)
 input (LatestMessage property), [30](#), [35](#)
 input (ListMessage property), [32](#)
 input (MigrateMessage property), [40](#)
 input (StatusMessage property), [23](#)
 isInitialized() (ConfigStorage method), [47](#)

L

LatestHandler (class), [30](#), [35](#)
 LatestMessage (class), [30](#), [35](#)

League\Container (namespace), [5](#)
 League\Flysystem (namespace), [5](#)
 League\Tactician (namespace), [6](#)
 line (CliException property), [48](#)
 ListHandler (class), [32](#)
 ListMessage (class), [32](#)
 load() (ConfigStorage method), [47](#)
 localConfigStack (ConfigStorage property), [47](#)

M

message (CliException property), [48](#)
 message (StatusHandler property), [22](#)
 MessageFactoryInterface (interface), [25](#)
 MessageInterface (interface), [25](#)
 MigrateHandler (class), [39](#)
 MigrateMessage (class), [40](#)

O

onCollectionAfter() (MigrateHandler method), [40](#)
 onCollectionBefore() (MigrateHandler method), [40](#)
 onMigrationAfter() (MigrateHandler method), [40](#)
 onMigrationBefore() (MigrateHandler method), [40](#)
 output (AbstractConfigMessage property), [19](#)
 output (AbstractMessage property), [18](#)
 output (AbstractRepositoryMessage property), [26](#)
 output (AbstractStorageMessage property), [34](#)
 output (AbstractTimelineCommand property), [37](#)
 output (CreateMessage property), [28](#)
 output (ExecuteMessage property), [38](#)
 output (InitMessage property), [21](#)
 output (LatestMessage property), [30](#), [35](#)
 output (ListMessage property), [32](#)
 output (MigrateHandler property), [39](#)
 output (MigrateMessage property), [40](#)
 output (StatusHandler property), [22](#)
 output (StatusMessage property), [23](#)
 outputVersions() (AbstractRepositoryListHandler method), [26](#)
 outputVersions() (LatestHandler method), [30](#)
 outputVersions() (ListHandler method), [32](#)

P

printDiff() (StatusHandler method), [22](#)
 printPendingVersion() (StatusHandler method), [22](#)
 processor (ConfigStorage property), [47](#)
 progress (MigrateHandler property), [39](#)
 projectFileSystem (ConfigStorage property), [47](#)
 provides (ApplicationProvider property), [49](#)
 provides (CommandsProvider property), [49](#)
 provides (ConfigProvider property), [49](#)
 provides (HelperSetProvider property), [50](#)
 provides (RepositoryProvider property), [50](#)
 provides (StorageProvider property), [51](#)
 provides (TimelineProvider property), [51](#)

R

register() (ApplicationProvider method), [49](#)
 register() (CommandsProvider method), [49](#)
 register() (ConfigProvider method), [49](#)
 register() (HelperSetProvider method), [50](#)
 register() (RepositoryProvider method), [50](#)
 register() (StorageProvider method), [51](#)
 register() (TimelineProvider method), [51](#)
 repository (AbstractRepositoryMessage property), [26](#)
 repository (CreateMessage property), [28](#)
 repository (LatestMessage property), [30](#)
 repository (ListMessage property), [32](#)
 repository (RepositoryAwareTrait property), [44](#)
 repository (StatusMessage property), [23](#)
 RepositoryAwareInterface (interface), [43](#)
 RepositoryAwareTrait (trait), [44](#)
 RepositoryProvider (class), [50](#)

S

saveChanges (MigrateHandler property), [39](#)
 saveVersionListener() (MigrateHandler method), [39](#)
 serviceAlias (BaseCommand property), [17](#)
 serviceClass (BaseCommand property), [18](#)
 Services (interface), [50](#)
 Services::APPLICATION (class constant), [51](#)
 Services::AUTOLOADER (class constant), [51](#)
 Services::BALEEN_BASE_DIR (class constant), [50](#)
 Services::CMD_CONFIG_INIT (class constant), [50](#)
 Services::CMD_CONFIG_STATUS (class constant), [50](#)
 Services::CMD_REPOSITORY_CREATE (class constant), [50](#)
 Services::CMD_REPOSITORY_LATEST (class constant), [50](#)
 Services::CMD_REPOSITORY_LIST (class constant), [50](#)
 Services::CMD_STORAGE_LATEST (class constant), [51](#)
 Services::CMD_TIMELINE_EXECUTE (class constant), [50](#)
 Services::CMD_TIMELINE_MIGRATE (class constant), [50](#)
 Services::COMMAND_BUS (class constant), [50](#)
 Services::COMMANDS (class constant), [50](#)
 Services::COMPARATOR (class constant), [51](#)
 Services::CONFIG (class constant), [50](#)
 Services::CONFIG_STORAGE (class constant), [50](#)
 Services::HELPERSET (class constant), [51](#)
 Services::HELPERSET_CONFIG (class constant), [51](#)
 Services::HELPERSET_QUESTION (class constant), [51](#)
 Services::MIGRATION_FACTORY (class constant), [51](#)
 Services::REPOSITORY (class constant), [51](#)
 Services::REPOSITORY_FILESYSTEM (class constant), [51](#)
 Services::STORAGE (class constant), [51](#)

- Services::TIMELINE (class constant), [51](#)
- setCliCommand() (AbstractConfigMessage method), [20](#)
- setCliCommand() (AbstractMessage method), [19](#)
- setCliCommand() (AbstractRepositoryMessage method), [27](#)
- setCliCommand() (AbstractStorageMessage method), [35](#)
- setCliCommand() (AbstractTimelineCommand method), [37](#)
- setCliCommand() (CreateMessage method), [29](#)
- setCliCommand() (ExecuteMessage method), [39](#)
- setCliCommand() (InitMessage method), [21](#)
- setCliCommand() (LatestMessage method), [31, 36](#)
- setCliCommand() (ListMessage method), [33](#)
- setCliCommand() (MessageInterface method), [26](#)
- setCliCommand() (MigrateMessage method), [41](#)
- setCliCommand() (StatusMessage method), [24](#)
- setComparator() (AbstractRepositoryMessage method), [27](#)
- setComparator() (AbstractStorageMessage method), [34](#)
- setComparator() (ComparatorAwareInterface method), [41](#)
- setComparator() (ComparatorAwareTrait method), [42](#)
- setComparator() (CreateMessage method), [29](#)
- setComparator() (LatestMessage method), [31, 36](#)
- setComparator() (ListMessage method), [33](#)
- setComparator() (StatusMessage method), [24](#)
- setConfig() (AbstractConfigMessage method), [20](#)
- setConfig() (AbstractMessage method), [19](#)
- setConfig() (AbstractRepositoryMessage method), [27](#)
- setConfig() (AbstractStorageMessage method), [34](#)
- setConfig() (AbstractTimelineCommand method), [37](#)
- setConfig() (CreateMessage method), [29](#)
- setConfig() (ExecuteMessage method), [38](#)
- setConfig() (InitMessage method), [21](#)
- setConfig() (LatestMessage method), [31, 36](#)
- setConfig() (ListMessage method), [33](#)
- setConfig() (MessageInterface method), [25](#)
- setConfig() (MigrateMessage method), [41](#)
- setConfig() (StatusMessage method), [24](#)
- setConfigStorage() (AbstractConfigMessage method), [19](#)
- setConfigStorage() (ConfigStorageAwareInterface method), [42](#)
- setConfigStorage() (ConfigStorageAwareTrait method), [42](#)
- setConfigStorage() (InitMessage method), [21](#)
- setConfigStorage() (StatusMessage method), [24](#)
- setFilesystem() (AbstractRepositoryMessage method), [27](#)
- setFilesystem() (CreateMessage method), [29](#)
- setFilesystem() (FilesystemAwareInterface method), [43](#)
- setFilesystem() (FilesystemAwareTrait method), [43](#)
- setFilesystem() (LatestMessage method), [31](#)
- setFilesystem() (ListMessage method), [33](#)
- setFilesystem() (RepositoryAwareInterface method), [43](#)
- setFilesystem() (RepositoryAwareTrait method), [44](#)
- setFilesystem() (StatusMessage method), [23](#)
- setInput() (AbstractConfigMessage method), [20](#)
- setInput() (AbstractMessage method), [19](#)
- setInput() (AbstractRepositoryMessage method), [27](#)
- setInput() (AbstractStorageMessage method), [34](#)
- setInput() (AbstractTimelineCommand method), [37](#)
- setInput() (CreateMessage method), [29](#)
- setInput() (ExecuteMessage method), [39](#)
- setInput() (InitMessage method), [21](#)
- setInput() (LatestMessage method), [31, 36](#)
- setInput() (ListMessage method), [33](#)
- setInput() (MessageInterface method), [25](#)
- setInput() (MigrateMessage method), [41](#)
- setInput() (StatusMessage method), [24](#)
- setOutput() (AbstractConfigMessage method), [20](#)
- setOutput() (AbstractMessage method), [19](#)
- setOutput() (AbstractRepositoryMessage method), [27](#)
- setOutput() (AbstractStorageMessage method), [34](#)
- setOutput() (AbstractTimelineCommand method), [37](#)
- setOutput() (CreateMessage method), [29](#)
- setOutput() (ExecuteMessage method), [39](#)
- setOutput() (InitMessage method), [21](#)
- setOutput() (LatestMessage method), [31, 36](#)
- setOutput() (ListMessage method), [33](#)
- setOutput() (MessageInterface method), [25](#)
- setOutput() (MigrateMessage method), [41](#)
- setOutput() (StatusMessage method), [24](#)
- setRepository() (AbstractRepositoryMessage method), [27](#)
- setRepository() (CreateMessage method), [29](#)
- setRepository() (LatestMessage method), [31](#)
- setRepository() (ListMessage method), [33](#)
- setRepository() (RepositoryAwareInterface method), [43](#)
- setRepository() (RepositoryAwareTrait method), [44](#)
- setRepository() (StatusMessage method), [23](#)
- setStorage() (AbstractStorageMessage method), [34](#)
- setStorage() (AbstractTimelineCommand method), [37](#)
- setStorage() (ExecuteMessage method), [38](#)
- setStorage() (LatestMessage method), [35](#)
- setStorage() (MigrateMessage method), [41](#)
- setStorage() (StatusMessage method), [23](#)
- setStorage() (StorageAwareInterface method), [44](#)
- setStorage() (StorageAwareTrait method), [44](#)
- setTimeline() (AbstractTimelineCommand method), [37](#)
- setTimeline() (ExecuteMessage method), [38](#)
- setTimeline() (MigrateMessage method), [41](#)
- setTimeline() (TimelineAwareInterface method), [45](#)
- setTimeline() (TimelineAwareTrait method), [45](#)
- splitDiff() (StatusHandler method), [22](#)
- StatusHandler (class), [22](#)
- StatusMessage (class), [23](#)
- storage (AbstractStorageMessage property), [34](#)
- storage (AbstractTimelineCommand property), [36](#)
- storage (ExecuteMessage property), [38](#)
- storage (LatestMessage property), [35](#)
- storage (MigrateMessage property), [40](#)

storage (StatusMessage property), [23](#)
storage (StorageAwareTrait property), [44](#)
StorageAwareInterface (interface), [44](#)
StorageAwareTrait (trait), [44](#)
StorageProvider (class), [51](#)
strategies (MigrateHandler property), [39](#)
Symfony\Config (namespace), [6](#)
Symfony\Console (namespace), [6](#)

T

timeline (AbstractTimelineCommand property), [36](#)
timeline (ExecuteMessage property), [38](#)
timeline (MigrateMessage property), [40](#)
timeline (TimelineAwareTrait property), [45](#)
TimelineAwareInterface (interface), [45](#)
TimelineAwareTrait (trait), [45](#)
TimelineProvider (class), [51](#)
toArray() (Config method), [46](#)
toArray() (ConfigInterface method), [46](#)
trackProgress (MigrateHandler property), [39](#)

W

write() (ConfigStorage method), [47](#)
writeClass() (CreateHandler method), [28](#)