
Comodojo/zip Documentation

Release 3.0

Marco Giovinazzi

Feb 13, 2021

Contents:

1	Installation	3
1.1	Requirements	3
2	Zip usage	5
2.1	Basic operations	5
2.2	Zip file operations	6
2.3	Additional methods	9
2.4	Password protected zip files	9
3	ZipManager usage	11
3.1	Basic operations	11
3.2	Additional methods	13

This library simplifies and automates the management of zip files using the [PHP ZipArchive](#).

It includes the `\Comodojo\Zip\ZipManager` class to manage multiple Zip files at once (extract, add, remove, merge).

CHAPTER 1

Installation

First install composer, then:

```
composer require comodojo/zip
```

1.1 Requirements

To work properly, comodojo/zip requires PHP $\geq 7.4.0$.

The `\Comodojo\Zip\Zip` class is designed to streamline the management of a single Zip file.

2.1 Basic operations

2.1.1 Open a zip file

```
1 <?php namespace My\Namespace;  
2  
3 use \Comodojo\Zip\Zip;  
4  
5 $zip = Zip::open('file.zip');
```

2.1.2 Create zip file

```
1 <?php namespace My\Namespace;  
2  
3 use \Comodojo\Zip\Zip;  
4  
5 $zip = Zip::create('file.zip');
```

2.1.3 Check zip file

```
1 <?php namespace My\Namespace;  
2  
3 use \Comodojo\Zip\Zip;  
4  
5 $is_valid = Zip::check('file.zip'); // true in case of success
```

2.2 Zip file operations

2.2.1 Extract

To extract the whole content of the zip file:

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
6
7 // extract whole archive
8 $zip->extract('/path/to/uncompressed/files');
```

To select one or multiple files to extract from the archive:

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
6
7 // extract one file
8 $zip->extract('/path/to/uncompressed/files', 'file');
9
10 // extract multiple files
11 $zip->extract('/path/to/uncompressed/files', ['file1', 'file2']);
```

2.2.2 Add a file or a directory

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::create('file.zip');
6
7 $zip->add('/path/to/my/file');
8
9 $zip->add('/path/to/my/directory');
```

To add only the directory content (i.e. flattening files):

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::create('file.zip');
6
7 $zip->add('/path/to/my/directory', true);
```

To set a default path and add files with relative location (i.e. change the root folder):

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::create('file.zip');
```

(continues on next page)

(continued from previous page)

```

7  // move the path
8  $zip->setPath('/path/to/my');
9
10 // add relative files or directories
11 $zip->add('file')
12     ->add('directory');
```

2.2.3 Change the compression method

Note: This feature is available since comodojo/zip 3.0

To change the compression method while adding a file or a directory:

```

1  <?php namespace My\Namespace;
2
3  use \Comodojo\Zip\Zip;
4
5  $zip = Zip::create('file.zip');
6
7  // add a file specifying the compression method to use
8  // available methods:
9  // Zip::CM_DEFAULT
10 // Zip::CM_STORE
11 // Zip::CM_DEFLATE
12 $zip->add('/path/to/my/file', false, Zip::CM_DEFLATE);
```

Different files can have different compression methods, for example:

```

1  <?php namespace My\Namespace;
2
3  use \Comodojo\Zip\Zip;
4
5  $zip = Zip::create('file.zip');
6
7  $zip->add('/path/to/my/file_1', false, Zip::CM_DEFLATE)
8      ->add('/path/to/my/file_2', false, Zip::CM_STORE);
```

2.2.4 Add multiple files/directories

The `Zip::add()` method accepts an array in input to add more resources at once:

```

1  <?php namespace My\Namespace;
2
3  $zip = Zip::create('file.zip');
4
5  use \Comodojo\Zip\Zip;
6
7  $zip->add([
8      '/path/to/my/file1',
9      '/path/to/my/file2'
10 ]);
11
12 // the Zip::add() method can be chained too:
13 $zip->add('/path/to/my/file1')
14     ->add('/path/to/my/file2');
```

2.2.5 Delete a file or a directory

To delete a file or a directory from a Zip file:

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
6
7 $zip->delete('file');
```

Also the `Zip::delete()` method accepts an array in input to delete multiple files at once:

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
6
7 $zip->delete([
8     'file1',
9     'file2'
10 ]);
11
12 // the Zip::delete() method can be chained too:
13 $zip->delete('file1')
14     ->delete('file2');
```

2.2.6 List content of the file

The `Zip::listFiles()` method can be used to get the list of files in the zip archive as an array:

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
6 $zip->listFiles();
```

2.2.7 Count the number of elements

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
6 $elements = count($zip);
```

2.2.8 Close

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\Zip;
4
5 $zip = Zip::open('file.zip');
```

(continues on next page)

(continued from previous page)

```

6
7 // ...
8
9 $zip->close();

```

2.3 Additional methods

2.3.1 Set SkipMode

The `Zip::setSkipMode()` method can force the Zip class to skip hidden files while adding directories:

```

1 <?php namespace My\Namespace;
2
3 $zip = Zip::open('file.zip');
4
5 // set the skip mode
6 // available modes:
7 // Zip::SKIP_NONE (default)
8 // Zip::SKIP_HIDDEN
9 // Zip::SKIP_ALL
10 // Zip::SKIP_COMODOJO
11 $zip->setSkipped(Zip::SKIP_HIDDEN);
12
13 // get skip mode
14 $mode = $zip->getSkipped();

```

2.3.2 Change file mask (extract)

To change the file mask:

```

1 <?php namespace My\Namespace;
2
3 $zip = Zip::open('file.zip');
4
5 // set the file mask (default 777)
6 $zip->setMask(0644);
7
8 // get mask
9 $mask = $zip->getMask();

```

2.4 Password protected zip files

The `Zip::setPassword()` method can be used to set a password for the current zip.

2.4.1 Extract a password protected zip

```

1 <?php namespace My\Namespace;
2
3 $zip = Zip::open('file.zip');
4
5 // set the zip password
6 $zip->setPassword('FordPerfect')
7     ->extract('/destination/folder');

```

2.4.2 Create a password protected zip file

Note: This feature is available since comodojo/zip 3.0

To create a password protected Zip file, once a password is set for the archive, each file should be flagged as encrypted, using one of the available encryption method.

```
1  <?php namespace My\Namespace;
2
3  $zip = Zip::create('file.zip');
4
5  // set the zip password
6  $zip->setPassword('FordPerfect')
7      ->add('file', false, Zip::CM_DEFAULT, Zip::EM_AES_128);
```

Different files can have different encryption methods, for example:

```
1  <?php namespace My\Namespace;
2
3  $zip = Zip::create('file.zip');
4
5  // set the zip password and the encryption method
6  // available methods:
7  // Zip::EM_NONE (default)
8  // Zip::EM_AES_128
9  // Zip::EM_AES_192
10 // Zip::EM_AES_256
11 $zip->setPassword('FordPerfect')
12     ->add('file_1', false, Zip::CM_DEFAULT, Zip::EM_AES_128)
13     ->add('file_2', false, Zip::CM_DEFAULT, Zip::EM_AES_256);
```

CHAPTER 3

ZipManager usage

The `\Comodojo\Zip\ZipManager` class is designed to manage, extract and combine multiple zip files (`\Comodojo\Zip\Zip` objects).

3.1 Basic operations

3.1.1 Start the Manager and register Zips

```
1  <?php namespace My\Namespace;
2
3  use \Comodojo\Zip\{
4      Zip,
5      ZipManager
6  };
7
8  // init manager
9  $manager = new ZipManager();
10
11 // register an existing zip file
12 $manager->addZip(Zip::open('/path/to/my/file1.zip'));
13 // register a new zip file
14 $manager->addZip(Zip::create('/path/to/my/file3.zip'));
```

3.1.2 Zip management

```
1  <?php namespace My\Namespace;
2
3  use \Comodojo\Zip\{
4      Zip,
5      ZipManager
6  };
7
8  // init manager
9  $manager = new ZipManager();
10 $zip_1_id = $manager->addZip(Zip::open('/path/to/my/file1.zip'));
```

(continues on next page)

(continued from previous page)

```
11 $zip_2_id = $manager->addZip(Zip::open('/path/to/my/file2.zip'));
12
13 // get a list of registered zips as array
14 $list = $manager->listZips();
15
16 // remove a zip
17 $manager->removeZipById($zip_1_id);
18
19 // get a Zip
20 $zip = $manager->getZip($zip_1_id);
```

3.1.3 Add files to all zips

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\{
4     Zip,
5     ZipManager
6 };
7
8 // init manager
9 $manager = new ZipManager();
10
11 // register existing zips
12 $manager->addZip(Zip::open('/path/to/my/file1.zip'));
13 $manager->addZip(Zip::open('/path/to/my/file2.zip'));
14
15 // add a file to all zips
16 $manager->add('/path/to/my/file');
```

3.1.4 Extract zips

```
1 <?php namespace My\Namespace;
2
3 use \Comodojo\Zip\{
4     Zip,
5     ZipManager
6 };
7
8 // init manager
9 $manager = new ZipManager();
10
11 // register existing zips
12 $manager->addZip(Zip::open('/path/to/my/file1.zip'));
13 $manager->addZip(Zip::open('/path/to/my/file2.zip'));
14
15 // separate content in folders
16 $extract = $manager->extract('/path/to/uncompressed/files', true);
17
18 // use a single folder
19 $extract = $manager->extract('/path/to/uncompressed/files', false);
20
21 // extract single file
22 $extract = $manager->extract('/path/to/uncompressed/files', false, 'file');
23
24 // extract multiple files
25 $extract = $manager->extract('/path/to/uncompressed/files', false, ['file1', 'file2
  ↳']);
```

3.1.5 Merge zips

```

1  <?php namespace My\Namespace;
2
3  use \Comodojo\Zip\{
4      Zip,
5      ZipManager
6  };
7
8  // init manager
9  $manager = new ZipManager();
10
11 // register existing zips
12 $manager->addZip(Zip::open('/path/to/my/file1.zip'));
13 $manager->addZip(Zip::open('/path/to/my/file2.zip'));
14
15 // separate content in folders
16 $manager->merge('/path/to/output/file.zip', true);
17
18 // flatten files
19 $manager->merge('/path/to/output/file.zip', false);

```

3.1.6 Close zips

```

1  <?php namespace My\Namespace;
2
3  use \Comodojo\Zip\{
4      Zip,
5      ZipManager
6  };
7
8  // init manager
9  $manager = new ZipManager();
10
11 // register existing zips
12 $manager->addZip(Zip::open('/path/to/my/file1.zip'));
13 $manager->addZip(Zip::open('/path/to/my/file2.zip'));
14
15 $manager->close();

```

3.2 Additional methods

3.2.1 Change file mask (extract)

```

1  <?php namespace My\Namespace;
2
3  $manager = new ZipManager();
4
5  // set the file mask (default 777)
6  $manager->setMask(0644);
7
8  // get mask
9  $mask = $manager->getMask();

```

3.2.2 Change the base path

```
1  <?php namespace My\Namespace;
2
3  $manager = new ZipManager();
4
5  // set the base path
6  $manager->setPath('/path/to/files');
7
8  // get base path
9  $path = $manager->getPath();
```