PHPBU Manual

Edition for PHPBU 4.0.

Sebastian Feldmann

PHPBU Manual: Edition for PHPBU 4.0.

Sebastian Feldmann

Publication date 2025.06.22

Table of Contents

List of Tables

List of Examples

Chapter 1. Installing PHPBU

Requirements

PHPBU 4.0 requires PHP 5.4, but using the latest version of PHP is highly recommended. If you want to use the PHAR version the minimum PHP version is 5.5.

PHPBU requires the dom [http://php.net/manual/en/dom.setup.php] and json [http://php.net/manual/en/json.installation.php] extensions, which are normally enabled by default.

PHPBU also requires the spl [http://php.net/manual/en/spl.installation.php] extension. It is required by the PHP core since 5.3.0 and normally cannot be disabled.

The different backup sources may need their respective executable binary f.e. `mysqldump`.

PHP Archive (PHAR)

The easiest way to install PHPBU is to download a PHP Archive (PHAR) [http://php.net/phar] that has all required dependencies of PHPBU bundled in a single file.

If the Suhosin [http://suhosin.org/] extension is enabled, you need to allow execution of PHARs in your `php.ini`:

```
suhosin.executor.include.whitelist = phar
```

To globally install the PHAR:

```
$ wget http://phar.phpbu.de/phpbu.phar
$ chmod +x phpbu.phar
$ sudo mv phpbu.phar /usr/local/bin/phpbu
$ phpbu --version
phpbu x.y.z
```

You may also use the downloaded PHAR file directly:

```
$ wget https://phar.phpbu.de/phpbu.phar
$ php phpbu.phar --version
phpbu x.y.z
```

Composer

If you are using Composer [https://getcomposer.org/] to manage your dependencies, simply add `phpbu/phpbu` to your `composer.json` file.

```
{
    "require": {
        "phpbu/phpbu": "4.0.*"
    }
}
```

For a system-wide installation via Composer, just run:

```
composer global require "phpbu/phpbu=4.0.*"
```

Make sure you have `~/.composer/vendor/bin/` in your path.

Framework integrations

The following framework integrations are available:

`phpbu-laravel`	Integrates phpbu with the laravel artisan command line tool. See the section called “Laravel” for details.

Chapter 2. Configuration

To use PHPBU and create your backups you first have to create a configuration file.

The configuration of PHPBU is done with a simple XML file. You can validate your config files with the respective PHPBU schema definition found at schema.phpbu.de [http://schema.phpbu.de].

XML-Configuration

You should name your configuration file `phpbu.xml` or `phpbu.xml.dist` so you don't have to specify the path with the `--configuration` option.

The following skeleton is a good point to start and get into detail from here.

Example 2.1. XML-Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<phpbu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="http://schema.phpbu.de/4.0/phpbu.xsd">
  <backups>
    <backup>
      <!-- backup source -->
      <source type="mysqldump">
        <option name="databases" value="mydbname"/>
        <option name="user" value="user.name"/>
        <option name="password" value="topsecret"/>
      </source>
      <!-- where should the backup be stored -->
      <target dirname="backup/mysql"
              filename="mysqldump-%Y%m%d-%H%i.sql"
              compress="bzip2"/>
      <!-- check the created backup -->
      <check type="SizeMin" value="10M"/>
      <!-- cleanup the backup location
           as soon as all created backups exceed 500MB
           remove oldest backups -->
      <cleanup type="Capacity">
        <option name="size" value="500M"/>
      </cleanup>
    </backup>
  </backups>
</phpbu>
```

List of tags

<phpbu>

The XML root element with at least xmlns:xsi and xsi:noNamespaceSchemaLocation.

Table 2.1. <phpbu> attributes

Name	Values	Required	Default	Description
bootstrap	filename	no	null	Include a php file to extend PHPBU
verbose	true | false	no	false	Controls the output verbosity

Name	Values	Required	Default	Description
debug	true | false	no	false	Controls the debug output

<backup>

Configures a backup process. You can have as many of these as you need.

Table 2.2. <backup> attributes

Name	Values	Required	Default	Description
name	string	no	null	A custom name for your backup used in the PHP-BU output.
stopOnError	true | false	no	false	Tells if the execution of phpbu will be stopped if this backup fails.

<source>

Defines the data that is backed up.

Table 2.3. <source> attributes

Name	Values	Required	Default	Description
type	string	yes	-	Type of backup, Chapter 5, *Backup Sources* has a list of all supported source types.

<target>

Defines where the backup is stored.

Table 2.4. <target> attributes

Name	Values	Required	Default	Description
dirname	string	yes	-	Path to the directory where the backup is stored, absolute or relative to the config file.
filename	string	yes	-	Filename of the stored backup.
compress	string	no	null	Type of compressor to use to compress the backup.

<check>

Specifies what kind of tests should be performed to validate the backup.

Table 2.5. <check> attributes

Name	Values	Required	Default	Description
type	string	yes	-	Type of check, Chapter 6, *Backup Validation* has a list of all supported check types.
value	string	yes	-	Value the backup is getting checked against.

<crypt>

Specifies what kind of encryption should be used to encrypt the backup.

Table 2.6. <crypt> attributes

Name	Values	Required	Default	Description
type	string	yes	-	Type of encryption, Chapter 7, *Encryption* has a list of all supported crypt types.
skipOnFailure	true | false	no	false	You can still execute the encryption, even of some failure occurred.

<sync>

Copy the created backup to another location.

Table 2.7. <sync> attributes

Name	Values	Required	Default	Description
type	string	yes	-	Type of sync, Chapter 8, *Sync Backups* has a list of all supported sync types.
skipOnFailure	true | false	no	false	You can still execute the sync, even of some failure occurred.

<cleanup>

Tells phpbu which files to delete after a successful backup.

Table 2.8. <cleanup> attributes

Name	Values	Required	Default	Description
type	string	yes	-	Type of cleanup, Chapter 9,

Name	Values	Required	Default	Description
				Cleanup backups has a list of all supported cleanup types.
skipOnFailure	true | false	no	-	You can still execute the cleanup, even of some failure occurred.

<option>

A simple key value tag to configure <source>, <crypt>, <sync> and <cleanup>.

Table 2.9. <option> attributes

Name	Values	Required	Default	Description
name	string	yes	-	Option key.
value	string	yes	-	Option value.

JSON-Configuration

This is roughly the same skeleton as shown above at the XML-Section.

Example 2.2. JSON-Configuration

```
{
  "verbose": true,
  "logging": [
    {
      "type": "json",
      "target": "backup/json.log"
    }
  ],
  "backups": [
    {
      "source": {
        "type": "mysqldump",
        "options": {
          "databases": "mydbname",
          "user": "user.name",
          "password": "topsecret"
        }
      },
      "target": {
        "dirname": "backup",
        "filename": "mysql-%Y%m%d-%H%i.sql",
        "compress": "bzip2"
      },
      "checks": [
        {
          "type": "sizemin",
          "value": "10M"
        }
      ],
      "syncs": [
        {
          "type": "sftp",
```

```
          "options": {
            "host": "backup.example.com",
            "user": "user.name",
            "password": "topsecret",
            "path": "backup/someName"
          }
        }
      ],
      "cleanup": {
        "type": "Capacity",
        "options": {
          "size": "200M"
        }
      }
    }
  ]
}
```

Adapters

Adapters are used to include configuration settings from other configuration sources.

So with adapters you don't have to put your passwords into your phpbu configuration file. Instead you can use environment variables or .env configuration files.

And this ist how it works: You add an adapter to your phpbu configuration like shown in the examples below. You have to choose a `type` and a `name`. The name is completely up to you, but you are not allowed to use colons. With this done you can now reference your configured adapter for any option value in your phpbu configuration like this.

```
adapter:my-chosen-adapter-name:path.to.value
```

The keyword `adapter` tells phpbu that this option value should be received from an adapter, followed by a `colon`, followed by your chosen `adapter name` so phpbu knows which adapter to use, followed by a `colon`, followed by the identifier for your value. The identifier could be the name of a environment variable or a "path" through an configuration array like this "db.mysql.username" completely depending on the adapter implementation.

For a list of available adapters and a more detailed documentation check out the adapter section.

Example 2.3. XML-Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<phpbu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="http://schema.phpbu.de/4.0/phpbu.xsd">
  <adapters>
    <adapter type="dotenv" name="environment">
      <option name="file" value=".env">
    </adapter>
  <adapters>
  <backups>
    <backup>
      <!-- backup source -->
      <source type="mysqldump">
        <option name="databases" value="mydbname"/>
        <option name="user" value="adapter:environment:DB_USER"/>
        <option name="password" value="adapter:environment:DB_PASSWORD"/>
      </source>
      <!-- where should the backup be stored -->
      <target dirname="backup/mysql"
              filename="mysqldump-%Y%m%d-%H%i.sql"/>
```

```
    </backup>
  </backups>
</phpbu>
```

Example 2.4. JSON-Configuration

```
{
  "verbose": true,
  "adapters": [
    {
      "type": "dotenv",
      "name": "environment",
      "options": {
        "file": ".env"
      }
    }
  ],
  "backups": [
    {
      "source": {
        "type": "mysqldump",
        "options": {
          "databases": "mydbname",
          "user": "adapter:environment:DB_USER",
          "password": "adapter:environment:DB_PASSWORD"
        }
      },
      "target": {
        "dirname": "backup",
        "filename": "mysql-%Y%m%d-%H%i.sql"
      }
    }
  ]
}
```

Schedule backups

Scheduling your backups is not part of phpbu. Nevertheless here is an example how to do it with Cron [https://de.wikipedia.org/wiki/Cron]. Adding one of the following lines to your crontab will execute your backup every morning at 3:10 AM.

Example 2.5. Crontab example

```
# +------------------------- min (0 - 59)
# |    +--------------------- hour (0 - 23)
# |    |    +---------------- day of month (1 - 31)
# |    |    |    +----------- month (1 - 12)
# |    |    |    |    +------ day of week (0 - 7) (Sunday=0 or 7)
# |    |    |    |    |
# *    *    *    *    *

# this requires phpbu to be in the cron users path
10     3    *    *    * phpbu --configuration=/var/www/my-backup/phpbu.xml

# alternatively you can specify the full path
#10     3    *    *    * /home/user/bin/phpbu.phar --configuration=/var/www/my-backup/ph
```

Chapter 3. Configuration Adapters

For a basic understanding why adapters are useful, have a look at the configuration documentation.

You can build your own configuration adapter. Want to know how? Have a look at the extending phpbu section

Environment adapter "env"

Adapter to use environment variables for option values.

Example 3.1. Adapter definition

```
<adapter type="env" name="myName"></adapter>
```

Example 3.2. Adapter reference

```
<option name="password" value="adapter:myName:SOME_ENV_VAR" />
```

Dotenv adapter "dotenv"

Adapter to use .env files to load environment variables.

Table 3.1. Options

Name	Required	Default	Description	
file	no	.env	Path to the .env file	

Example 3.3. Adapter definition

```
<adapter type="dotenv" name="myName">
    <option name="file" value="my-dot-env-dir/.env"/>
  </adapter>
```

Example 3.4. Adapter reference

```
<option name="password" value="adapter:myName:SOME_DOTENV_VAR" />
```

PHP config adapter "array"

Adapter to use php array config files.

Example 3.5. Example PHP config file

```
<?php return ['foo' => ['bar' => 'baz']];
```

Table 3.2. Options

Name	Required	Default	Description	
file	yes	-	Path to the php config file	

Example 3.6. Adapter definition

```
<adapter type="array" name="myName">
    <option name="file" value="my-config-dir/config.php"/>
  </adapter>
```

Example 3.7. Adapter reference

```
<option name="password" value="adapter:myName:foo.bar" />
```

Chapter 4. The Command-Line-Runner

If you have a configuration file named `phpbu.xml` in your current working directory you can execute phpbu like this

```
~$ phpbu
phpbu 4.0.0

Runtime:       PHP 7.1.0
Configuration: /home/user/phpbu.xml

OK (1 backup, 0 checks, 0 encryption, 0 syncs, 0 cleanups)
```

If you want to use a custom name for your configuration file you have to use the `--configuration` option.

```
~$ phpbu --configuration=MyConfig.xml
phpbu 4.0.0

Runtime:       PHP 7.0.2
Configuration: /home/user/MyConfig.xml

OK (1 backup, 0 checks, 0 encryption, 0 syncs, 0 cleanups)
```

Command-Line-Runner options

Here is the list of available options for the command line runner.

```
$ phpbu --help
phpbu 4.0.0

Usage: phpbu [option]

--bootstrap=<file>     A "bootstrap" PHP file that is included before the backup.
--configuration=<file> A phpbu xml config file.
--colors               Use colors in output.
--debug                Display debugging information during backup generation.
--limit=<subset>       Limit backup execution to a subset.
--simulate             Perform a trial run with no changes made.
-h, --help             Print this usage information.
-v, --verbose          Output more verbose information.
-V, --version          Output version information and exit.
```

`--bootstrap` A "bootstrap" PHP file that is included before executing the backup.

`--configuration` Path to the phpbu config file to use. See Chapter 2, *Configuration* for more details.

`--colors` Use colors in output.

`--debug` Display debugging information during backup generation.

`--limit` Limit backup execution to a subset. See the limit section for details.

`--simulate` Execute phpbu without actually executing any backup, check, encryption, sync or cleanup task. See the simulation section for details.

Simulation

In order to check what actions phpbu would perform, phpbu supports a simulation mode where no backup is actually created and no sync and cleanup tasks are executed. In simulation mode, phpbu will show you, what it would have done, if it would not have been a simulation run. You will get detailed output, which enables you to track every action phpbu would have performed.

This is a great way to test if the correct backup commands will be executed and if all the right backups will be cleaned up.

Warning

If phpbu has to use any kind of credentials executing a binaries, like for example the `mysql-dump` command, this credentials will maybe displayed in the simulation output.

If you are extending phpbu, you can support simulation as well. You can find detailed information on this in Chapter 11, *Extending PHPBU*.

Limit

If you don't want to execute all of your configured backups you can define a subset of backups by using the limit option.

All you have to to, is to set the limit option to a comma separated list of backup names (nameA,nameB,nameC).

Warning

If you don't define explicit backup names, phpbu will use the source type as backup name as fallback.

Chapter 5. Backup Sources

You can configure multiple Backups in one configuration file. Each backup is represented by a `<backup>` Tag. To configure the data that is stored use the `<source>` Tag.

Table 5.1. Supported types of backup sources:

Type	Description
arangodump	Backup ArangoDB with the arangodump command line tool.
mongodump	Backup MongoDB with the mongodump command line tool.
mysqldump	Backup MySQL with the mysqldump command line tool.
pgdump	Backup PostgresSQL with the pg_dump command line tool.
redis	Backup redis with the redis-cli command line tool.
rsync	Collect files via the rsync command line tool.
tar	Backup a directory with the tar command line tool.
xtrabackup	Backup MySQL with the percona xtrabackup command line tool.

ArangoDB

Backup a ArangoDB database with arangodump.

Table 5.2. arangodump-Options

Name	Value	Required	Default	Description
endpoint	string	no	tcp://localhost:8529	URI to your ArangoDB endpoint.
username	string	no	null	Username for the ArangoDB authentication.
password	string	no	null	Password for the ArangoDB authentication.
disableAuthentication	boolean	no	false	Disable the ArangoDB authentication.
database	string	no	_system	Database to backup.
dumpData	boolean	no	true	Dump database data.
includeSystemCollections	boolean	no	false	Dump system collections as well.

Name	Value	Required	Default	Description
collections	string	no	all	List of collections to dump.
pathToArango-Dump	string	no		Custom arango-dump executable location.

Example 5.1. arangodump XML example

```
<!-- source arangodump -->
<source type="arangodump">
  <option name="database" value="myDatabase" />
  <option name="dumpData" value="true" />
  <option name="collections" value="myCollection1,myCollection2" />
</source>
```

Example 5.2. arangodump JSON example

```
{
  "type": "arangodump",
  "options": {
    "database": "myDatabase",
    "dumpData": "true",
    "collections": "myCollection1,myCollection2"
  }
}
```

MongoDB

Backup a MongoDB database with mongodump.

Table 5.3. mongodump-Options

Name	Value	Required	Default	Description
host	string	no	localhost	Hostname of your MongoDB-Server.
user	string	no	OS-User	Username for the MongoDB authentication.
password	string	no	null	Password for the MongoDB authentication.
authentication-Database	string	no	null	MongoDB database that holds the user credentials.
databases	string	no	all	List of databases you want to backup, all by default.
excludeCollections	string	no	null	List of collections you don't want to backup.
excludeCollectionsWithPrefix	string	no	null	List of collection prefixes to exclude

Name	Value	Required	Default	Description
				matching collections from backup.
pathToMongodump	string	no		Custom mongodump executable location.

Example 5.3. mongodump XML example

```
<!-- source mongodump -->
<source type="mongodump">
  <option name="databases" value="myDatabase"/>
  <option name="user" value="user.name"/>
  <option name="password" value="topsecret"/>
  <option name="excludeCollections" value="collectionFoo,collectionBar"/>
</source>
```

Example 5.4. mongodump JSON example

```
{
  "type": "mongodump",
  "options": {
    "databases": "myDatabase",
    "user": "user.name",
    "password": "topsecret",
    "excludeCollections": "collectionFoo,collectionBar"
  }
}
```

MySQL - mysqldump

Backup a MySQL database with mysqldump.

Table 5.4. mysqldump-Options

Name	Value	Required	Default	Description
host	string	no	localhost	Hostname of your MySQL-Server.
user	string	no	OS-User	Username for the MySQL authentication.
password	string	no	null	Password for the MySQL authentication.
databases	string	no	all	List of databases you want to backup, all by default.
ignoreTables	string	no	null	List of tables you don't want to backup.
filePerTable	true | false	no	false	Create directory with separate files for each table.
structureOnly	string	no	null	List of tables where only the ta-

Name	Value	Required	Default	Description
				ble structure will be stored. Can not be used with *filePerTable*
hexBlob	true | false	no	false	Use the --hex-blob option to dump blog fields in hex.
lockTables	true | false	no	null	Use the --lock-tables option to lock tables during the dump.
singleTransaction	true | false	no	null	Use the --single-transaction option to add BEGIN SQL statement before dumping data from server.
extendedInsert	true | false	no	false	Use multiple-row INSERT syntax. This results in a smaller dump file and speeds up inserts when the file is reloaded.
pathToMysqldump	string	no		Custom mysqldump executable location.

Example 5.5. mysqldump XML example

```
<!-- source mysqldump -->
<source type="mysqldump">
  <option name="databases" value="myDatabase"/>
  <option name="user" value="user.name"/>
  <option name="password" value="topsecret"/>
  <option name="ignoreTable" value="tableFoo,tableBar"/>
  <option name="structureOnly" value="logTable1,logTable2"/>
</source>
```

Example 5.6. mysqldump JSON example

```
{
  "type": "mysqldump",
  "options": {
    "databases": "myDatabase",
    "user": "user.name",
    "password": "topsecret",
    "ignoreTable": "tableFoo,tableBar",
    "structureOnly": "logTable1,logTable2"
  }
}
```

MySQL - xtrabackup

Backup a MySQL database with percona xtrabackup.

Table 5.5. xtrabackup-options

Name	Value	Required	Default	Description
dataDir	string	no		MySQL data directory.
host	string	no	localhost	Host to connect to.
user	string	no	OS-User	User to connect with.
password	string	no	false	Password to authenticate the user.
databases	string	no	-	List of databases to backup, all by default.
include	string	no	-	Pattern to include databases and tables to the backup e.g. `^mydatabase[.]mytable`.
pathToXtraBackup	string	no		Custom xtrabackup executable location.

Example 5.7. xtrabackup XML example

```
<!-- source xtrabackup -->
<source type="xtrabackup">
  <option name="databases" value="myDB1,myDB2" />
</source>
```

Example 5.8. xtrabackup JSON example

```
{
  "type": "xtrabackup",
  "options": {
    "databases": "myDB1,myDB2"
  }
}
```

pgdump

Backup PostgresSQL databases using the pg_dump command line tool.

Table 5.6. pgdump-Options

Name	Value	Required	Default	Description
host	string	no	-	Host to connect to.
port	integer	no	-	Port to use to connect to the PostgresSQL server.
user	string	no	-	User to use to connect.
password	string	no	-	Password to use to authenticate.

Name	Value	Required	Default	Description
database	string	yes	-	Database to back-up.
schemaOnly	bool	no	false	Dump only schema information.
dataOnly	string	no	false	Dump no schema information.
schemas	string	no	-	List of schemas to dump, separated by comma.
excludeSchemas	string	no	-	List of schemas to exclude, separated by comma.
tables	string	no	-	List of tables to backup, separated by comma.
excludeTables	string	no	-	List of tables to exclude, separated by comma.
excludeTableData	string	no	-	List of tables where data is not dumped, separated by comma.
noPermissions	bool	no	false	Don't dump any permission schemas.
noOwner	bool	no	false	Don't dump any owner changing statements.
format	string	no	plain	Dump format, plain, custom, directory.
encoding	string	no	-	Set the dump encoding.
pathToPgdump	string	no		Custom pgdump executable location.

Example 5.9. pgdump XML example

```
<!-- source pgdump -->
<source type="pgdump">
  <option name="database" value="myDstabase"/>
  <option name="user" value="myName"/>
  <option name="password" value="topSecret"/>
</source>
```

Example 5.10. pgdump JSON example

```
{
  "type": "pgdump",
  "options": {
```

```
    "database": "myDatabase",
    "user": "myName",
    "password": "topSecret",
  }
}
```

redis

Backup redis data structure store using redis-cli.

Table 5.7. redis-Options

Name	Value	Required	Default	Description
pathToRedisData	string	yes	-	Path to the redis data .rdb file.
port	integer	no	6379	Port to use to connect to the redis server.
password	string	no	-	Password to use to authenticate.
pathToRedisCli	string	no		Custom redis-cli executable location.

Example 5.11. redis XML example

```
<!-- source redis -->
<source type="redis">
  <option name="pathToRedisData" value="/var/lib/redis/6379/dump.rdb"/>
</source>
```

Example 5.12. redis JSON example

```
{
  "type": "redis",
  "options": {
    "pathToRedisData": "/var/lib/redis/6379/dump.rdb"
  }
}
```

rsync

Backup files via the rsync command line tool.

Table 5.8. rsync-Options

Name	Value	Required	Default	Description
path	string	yes	-	Path to file/directory to sync.
host	string	no	-	Host to connecto to.
user	string	no	-	User to use to authenticate.

Name	Value	Required	Default	Description
delete	string	no	false	Remove deleted files locally as well.
isDirSync	string	no	false	Keep the directory instead of archiving it at every phpbu execution.
pathToRsync	string	no		Custom rsync executable location.

Example 5.13. redis XML example

```
<!-- source rsync -->
<source type="rsync">
  <option name="path" value="/tmp/foo"/>
</source>
```

Example 5.14. redis JSON example

```
{
  "type": "rsync",
  "options": {
    "path": "/tmp/foo"
  }
}
```

Directories

Backup directories using tar.

Table 5.9. tar-Options

Name	Value	Required	Default	Description
path	string	yes	-	Path to the source directory.
exclude	string	no	-	Comma separated list of paths to exclude.
ignoreFailedRead	boolean	no	false	Ignore changing source directory.
compressProgram	string	no	-	Set custom compression program f.e. lbzip2.
removeSourceDir	boolean	no	false	Remove the directory after compression.
pathToTar	string	no		Custom tar executable location.

Example 5.15. tar XML example

```
<!-- source tar -->
```

```
<source type="tar">
  <option name="path" value="htdocs/uploads"/>
</source>
```

Example 5.16. tar JSON example

```
{
  "type": "tar",
  "options": {
    "path": "htdocs/uploads"
  }
}
```

Chapter 6. Backup Validation

You can validate your backup after creation with some simple Checks.

Table 6.1. Available checks:

Type	Description
SizeMin	Checks if the backups is at least as big as the configured size.
SizeDiffPreviousPercent	Checks if your backup doesn't differ more then a configured value from the previous backup.

Minimal size

Example 6.1. SizeMin XML example

```
<!-- check min size -->
<check type="SizeMin" value="30M"/>
```

Example 6.2. SizeMin JSON example

```
{"type": "SizeMin", "value": "30M"}
```

Compare to previous Backups

Example 6.3. SizeDiffPreviousPercent XML example

```
<!-- check size against previous backups -->
<check type="SizeDiffPreviousPercent" value="10"/>
```

Example 6.4. SizeDiffPreviousPercent JSON example

```
{"type": "SizeDiffPreviousPercent", "value": "10"}
```

Chapter 7. Encryption

PHPBU can encrypt your backup using `openssl` or `mcrypt`.

Hint

If the choice is yours, I strongly recommend using openssl. Here [https://paragonie.com/blog/2015/05/if-you-re-typing-word-mcrypt-into-your-code-you-re-doing-it-wrong] is a good article explaining why.

Table 7.1. Available crypts:

Type	Description
mcrypt	Encrypts your backup with the mcrypt command line tool.
openssl	Encrypts your backup with the openssl command line tool.

mcrypt

Table 7.2. mcrypt options

Name	Value	Required	Default	Description
algorithm	string	yes	-	Algorithm to use to encrypt the backup.
key	string	yes	-	Secret key to use for encryption.

Example 7.1. mcrypt XML example

```
<!-- encryption -->
<crypt type="mcrypt">
  <option name="algorithm" value="blowfish"/>
  <option name="key" value="mySecretKey"/>
</crypt>
```

Example 7.2. mcrypt JSON example

```
{
  "type": "mcrypt",
  "options": {
    "algorithm": "blowfish",
    "key": "mySecretKey"
  }
}
```

openssl

Please be sure to use *only* `password` or `certFile`.

Table 7.3. openssl options

Name	Value	Required	Default	Description
password	string	yes	-	Encrypt backup with 'openssl enc -pass...'.
certFile	string	yes	-	Encrypt backup with 'openssl smime ... myCert.pem'.
algorithm	string	yes	-	Algorithm to use to encrypt the backup. *Attention* you have to specify different algorithm names for using `password` and using a `certfile`.
keepUncrypted	boolean	no	false	Don't remove the uncrypted backup.
pathToOpenSSL	string	no	-	Used to specify a special path to the openssl command.

Example 7.3. openssl XML example using password encryption

```
<!-- encryption -->
<crypt type="openssl">
  <option name="password" value="mySecretPassword"/>
  <option name="algorithm" value="aes-256-cbc"/>
</crypt>
```

Example 7.4. openssl JSON example using password encryption

```
{
  "type": "openssl",
  "options": {
    "password": "mySecretPassword",
    "algorithm": "aes-256-cbc"
  }
}
```

Example 7.5. openssl XML example using SSL cert encryption

```
<!-- encryption -->
<crypt type="openssl">
  <option name="certFile" value="ssl/MyCert.pem"/>
  <option name="algorithm" value="aes256"/>
</crypt>
```

Example 7.6. openssl JSON example using SSL cert encryption

```
{
  "type": "openssl",
  "options": {
```

```
    "certFile": "ssl/MyCert.pem",
    "algorithm": "aes256"
  }
}
```

Create a private key and a certificate pem file

To encrypt your backups with a cert file you have to create a *private key* and a *certificate pem file*.

```
$ openssl req -x509 -new -days 100000 -key private.pem -out certificate.pem
```

The created `certificate.pem` is used to encrypt your backups and should be referenced in your phpbu configuration as `certFile`. The `private.pem` file is used to decrypt your backups.

Decrypt

Decrypt a cert file encoded backup

```
$ openssl smime -decrypt -aes256 -inform DER \
  -in backup.tar.bz2.enc \
  -out backup.tar.bz2 \
  -inkey private.pem
```

Decrypt a password encoded backup

```
$ openssl enc -d -a -aes-256-cbc \
  -in backup.tar.bz2.enc \
  -out backup.tar.bz2 \
  -pass pass:mySecretPassword
```

Chapter 8. Sync Backups

It's best practice to not store all you backups locally or not only storing them locally. With PHPBU you can copy your backups to different locations.

Caution

If you are syncing your backup to a cloud service it's advised to encrypt your backup beforehand. Especially if your backup contains personal identifiable information [https://en.wikipedia.org/wiki/Personally_identifiable_information].

Table 8.1. Supported types of syncs:

Type	Description
amazon	Copy your backup to your amazon s3 account.
copy.com	Copy your backup to a copy account.
dropbox	Copy your backup to a dropbox account.
rsync	Sync your backups with rsync.
sftp	Copy your backup to a server via SFTP.
SoftLayer Object Storage	Sync your backup to a Softlayer Object Storage account.

Amazon S3

Sync your backup to an amazon S3 account.

If you are not using the PHAR Version you have to require `"aws/aws-sdk-php": "2.7.*"` in your composer file.

Table 8.2. amazons3-Options

Name	Value	Required	Default	Description
key	string	yes	-	The amazon s3 key.
secret	string	yes	-	The amazon S3 secret.
bucket	string	yes	-	A bucket where to store the backup.
region	string	yes	-	The region where the bucket is located e.g. 'eu-central-1'.
path	string	no	/	Path where to store the backup in your bucket.
useMultiPartUpload	boolean	no	false	Use Amazon S3 MultiPartUpload functionality.

Example 8.1. amazons3 XML example

```
<!-- sync amazon s3 -->
<sync type="amazons3">
```

```
  <option name="key" value="myAwsKey"/>
  <option name="secret" value="myAwsSecret"/>
  <option name="bucket" value="backup"/>
  <option name="region" value="eu-central-1"/>
  <option name="path" value="/some/dir"/>
</sync>
```

Example 8.2. amazons3 JSON example

```
{
  "type": "amazons3",
  "options": {
    "key": "myAwsKey",
    "secret": "myAwsSecret",
    "bucket": "backup",
    "region": "eu-central-1",
    "path": "/some/dir",
    "useMultiPartUpload": "true"
  }
}
```

Dropbox

Sync your backup to a dropbox account.

If you are not using the PHAR Version you have to require `"dropbox/dropbox-sdk": "1.1.*"` in your composer file.

Table 8.3. dropbox-Options

Name	Value	Required	Default	Description
token	string	yes	-	The dropbox authentication token. Go to www.dropbox.com/developers/apps [https://www.dropbox.com/developers/apps] Create your app Choose: dropbox api app files and datastore yes provide some app name "my-dropbox-app" generate access token to authenticate connection to your dropbox
path	string	yes	-	Directory where to store the backup in your dropbox account.

Example 8.3. dropbox XML example

```
<!-- sync dropbox -->
```

```
<sync type="dropbox">
  <option name="token" value="myCrazyLongApiTokenThatIGotFromDropbox"/>
  <option name="path" value="/some/dir"/>
</sync>
</section>
```

Example 8.4. dropbox JSON example

```
{
  "type": "dropbox",
  "options": {
    "token": "myCrazyLongApiTokenThatIGotFromDropbox",
    "path": "/some/dir"
  }
}
```

Rsync

Sync your backup via the rsync command.

Table 8.4. rsync-Options

Name	Value	Required	Default	Description
path	string	yes	-	The remote path where your backups should be synced to.
host	string	yes	-	The remote host where your backups should be synced to.
user	string	no	OS-User	The user to connect to the remote host. Make sure the user is able to connect without entering a password otherwise you get prompted for a password while executing phpbu.
dirsync	boolean	no	false	Sync the target directory instead of syncing only the target file.
delete	boolean	no	false	Add the --delete option to the rsync call, so locally deleted files will be deleted remotely as well.
exclude	string	no	-	List of files to exclude from the sync sepa-

Name	Value	Required	Default	Description
				rated by ":" e.g. "*.suffix:foo.bar".
args	string	no	-	Advanced mode to use completely custom options. "rsync {args}". WARNING: phpbu is not escaping this in any way so use with caution! Use %TARGET_FILE% and %TARGET_DIR% as reference to your created backup.

Example 8.5. rsync XML example

```
<!-- sync rsync -->
<sync type="rsync">
  <option name="path" value="/backup/some/path"/>
  <option name="dirsync" value="true"/>
  <option name="host" value="backup.example.com"/>
  <option name="user" value="backup"/>
</sync>
```

Example 8.6. rsync JSON example

```
{
  "type": "rsync",
  "options": {
    "path": "/backup/some/path",
    "dirsync": "true",
    "host": "backup.example.com",
    "user": "backup"
  }
}
```

SFTP / FTP

Copy your backup to another server via SFTP or FTP.

If you want to use SFTP and you are not using the PHAR version you have to require `"phpseclib/phpseclib": "2.0.*@dev"` in your composer file. For this to work you have to set the `minimum-stability` to `dev`.

If you want to use the FTP-Sync your PHP has to be compiled with `--enable-ftp`.

Table 8.5. (s)ftp-Options

Name	Value	Required	Default	Description
host	string	yes	-	The host you want do copy your backups to.

Name	Value	Required	Default	Description
user	string	yes	-	The user you want to connect with.
password	string	no	-	The password to authenticate the user.
path	string	yes	-	The remote path where to copy the backup.

Example 8.7. sftp XML example

```
<!-- sync sftp -->
<sync type="sftp">
  <option name="host" value="backup.example.com"/>
  <option name="user" value="user.name"/>
  <option name="password" value="topsecret"/>
  <option name="path" value="backup/someName"/>
</sync>
```

Example 8.8. sftp JSON example

```
{
  "type": "sftp",
  "options": {
    "host": "backup.example.com",
    "user": "user.name",
    "password": "topsecret",
    "path": "backup/someName"
  }
}
```

If you want to use plain FTP just use `ftp` as type, options stay the same.

Softlayer Object Storage

Sync your backup to a Softlayer account.

If you are not using the PHAR Version you have to require `"softlayer/objectstorage": "dev-master"` in your composer file. For this to work you have to set the `minimum-stability` to `dev`.

Table 8.6. softlayer-Options

Name	Value	Required	Default	Description
user	string	yes	-	The user you want to connect with.
secret	string	no	-	The api key to authenticate the user.
host	string	yes	-	The host you want do copy your backups to.
container	string	yes	-	The Object Storage Container where to put the backup.

Name	Value	Required	Default	Description
path	string	yes	-	The remote path where to copy the backup.

Example 8.9. softlayer XML example

```
<!-- sync softlayer object storage -->
<sync type="softlayer">
  <option name="user" value="user.name"/>
  <option name="secret" value="topsecret"/>
  <option name="host" value="some.softlayer.domain.com"/>
  <option name="container" value="backup"/>
  <option name="path" value="/backup/someName"/>
</sync>
```

Example 8.10. softlayer JSON example

```
{
  "type": "softlayer",
  "options": {
    "user": "user.name",
    "secret": "topsecret",
    "host": "some.softlayer.domain.com",
    "container": "backup",
    "path": "/backup/someName"
  }
}
```

Chapter 9. Cleanup backups

You can remove old backups to limit the space your backups consume on your local hard drive. It's important to note, that only *local* backups will be cleaned up. So whatever kind of `Sync` you are using. `Cleanup` won't touch your remote files.

If you want to remove your synced backups as well, currently the only way to do this is to use the Rsync sync with it's `delete` option.

Table 9.1. Supported types of cleanups:

Type	Description
Capacity	Define the size your backups should occupy. If your backups start to exceed this value PHPBU will start to delete backups, starting with the oldest ones.
Outdated	Remove backups older than a configured time-value. For example "2W" to delete all backup older than two weeks.
Quantity	Keep only a configured amount of backups.

Cleanup by capacity

Remove oldest backups if all created backups exceed the configured limit.

Table 9.2. Capacity-Options

Name	Value	Required	Default	Description
size	string	yes	-	Space your backups are allowed to occupy. e.g. 500M for 500 Megabytes or 1G for one Gigabyte.
deleteTarget	boolean	no	false	If you don't want to keep any backups locally, just specify a size of "0B", set this to "true" and even the current backup will be deleted.

Example 9.1. capacity XML example

```
<!-- cleanup capacity: keep 250 Megabytes of backups -->
<cleanup type="capacity">
  <option name="size" value="250M"/>
  <option name="deleteTarget" value="false"/>
</cleanup>
```

Example 9.2. capacity JSON example

```
{
```

```
  "type": "capacity",
  "options": {
    "size": "250M",
    "deleteTarget": "false"
  }
}
```

Cleanup by date

Remove backups that are older than a configured value.

Table 9.3. Outdated-Options

Name	Value	Required	Default	Description
older	string	yes	-	Time to keep backups. e.g. 1m for one month or 20d for twenty days.

Example 9.3. outdated XML example

```
<!-- cleanup outdated: keep backups for 2 weeks -->
<cleanup type="outdated">
  <option name="older" value="2W"/>
</cleanup>
```

Example 9.4. outdated JSON example

```
{
  "type": "Outdated",
  "options": {
    "older": "2W"
  }
}
```

Cleanup by quantity

Keep only a configured amount of backups.

Table 9.4. Quantity-Options

Name	Value	Required	Default	Description
amount	string	yes	-	The amount of backups to keep.

Example 9.5. quantity XML example

```
<!-- cleanup quantity: keep the last 20 backups -->
<cleanup type="quantity">
  <option name="amount" value="20"/>
</cleanup>
```

Example 9.6. quantity JSON example

```
{
```

```
  "type": "quantity",
  "options": {
    "amount": "20"
  }
}
```

Chapter 10. Logging

JSON

You can create a simple json logfile with PHPBU's json logger.

Example 10.1. json XML example

```
<!-- create a json logfile -->
<log type="json" target="backup/log.json"/>
```

Example 10.2. json JSON example

```
{"type": "json", "target": "backup/log.json"}
```

E-Mail

PHPBU uses SwiftMailer to send backup reports to one or more configured E-Mail-Addresses. If you just want to get notified if something went wrong, you can use the `sendOnlyOnError` option to not send any E-Mail on a successful backup.

Table 10.1. Mail-Options

Name	Value	Required	Default	Description
recipients	string	yes	-	List of emails separated by semi-colon.
sendOnlyOnError	string	no	false	Sending reports only if something goes wrong.
subject	string	no	PHPBU Back-up Report from $HOSTNAME	E-Mail subject of the backup report.
transport	string	yes	-	SwiftMailer Transport that's used to send the E-Mail (mail, send-mail, smtp, null).
sender.mail	string	phpbu@hostname	-	The sender E-Mail-Address
sender.name	string	no	OS-User	The sender name.
sender.name	string	no	OS-User	The sender name.
sendmail options				
sendmail.path	string	no	-	Path to local send-mail binary.
sendmail.options	string	no	-	Sendmail options (e.g. -bs).
SMTP options				
smtp.port	string	no	587	Port to use to connect to SMTP server.

Name	Value	Required	Default	Description
smtp.host	string	no	-	SMTP hostname.
smtp.username	string	no	-	The SMTP login.
smtp.password	string	no	-	The SMTP password.
smtp.encryption	string	no	-	Type of encryption (e.g. ssl).

Example 10.3. mail XML example

```
<!-- use mail to keep up to date with you backup creation -->
<log type="mail">
  <option name="transport" value="mail"/>
  <option name="recipients" value="user.name@example.com"/>
</log>
```

Example 10.4. mail JSON example

```
{
  "type": "mail",
  "options": {
     "transport": "mail",
     "recipients": "user.name@example.com"
   }
}
```

Chapter 11. Extending PHPBU

You can extend PHPBU with your own Source, Check, Crypt, Sync, Cleanup, Adapter and Logger implementations.

All you have to do is to register your implementation at the PHPBU Factory. You can do this by adding you own bootstrap file.

```
phpbu --bootstrap=extend.php
```

Your extend.php file could look something like this:

Example 11.1. Extend bootstrap

```
<?php
  // make your class accessible
  require 'Acme/MySource.php';
  // register your class as a source
  try {
      phpbu\App\Factory::register('source', 'mysource', '\\Acme\\MySource');
  } catch (Exception $e) {
      die($e->getMessage());
  }
```

For this to work your class Acme\MySource hast to implement the `phpbu\App\Backup\Source` Interface. After registering your Source you can use `mysource` as source-type within your XML or JSON configuration like this.

Example 11.2. Custom Source XML example

```
<source type="mysource"></source>
```

Example 11.3. Custom Source JSON example

```
{"type": "mysource"}
```

If the alias is already taken this will throw an Exception. You can force phpbu to overwrite any previously registered class using the $force Parameter. Even the phpbu sources could be replaced with custom implementations.

Example 11.4. Overwrite default sources

```
<?php
// make your class accessible
require 'Acme/MyMysqldump.php';
// register your class as source, override already registered class
phpbu\App\Factory::register('source', 'mysqldump', '\\Acme\\MyMysqldump', true);
```

Executing and simulating

If you are extending phpbu you have two options. The first is to simply implement the execution interface. This way you can execute your own Source, Check, Crypt, Sync or Cleanup tasks. The second option is to implement the respective `Simulator` interface. Choosing this option you not only have to implement the execution part but rather implement a simulation part as well.

Simulation is supported out of the box for every implementation, but choosing option two enables you to add viable information about the actions you perform in your code within a simulation run.

Table 11.1. Types to register:

Type	Interface for execution only	Interface for execution and simulation
source	phpbu\App\Backup\Source	phpbu\App\Backup\Source \Simulator
check	phpbu\App\Backup\Check	phpbu\App\Backup\Check\Simulator
crypt	phpbu\App\Backup\Crypt	phpbu\App\Backup\Crypt
sync	phpbu\App\Backup\Sync	phpbu\App\Backup\Sync\Simulator
cleanup	phpbu\App\Backup\Cleanup	phpbu\App\Backup\Cleanup \Simulator
adapter	phpbu\App\Adapter	-
logger	phpbu\App\Backup\Logger	-

Create a custom Source

A minimal custom Source class could look like this.

Example 11.5. Create a custom source

```
<?php
namespace Acme;

use \phpbu\App\Backup\Result;
use \phpbu\App\Backup\Source;
use \phpbu\App\Backup\Target;

class MySource implements Source
{
    /**
     * Some data that has to be configured.
     */
    private $someConfigValue;

    /**
     * Here you should validate the configuration and setup your class.
     *
     * @param array $conf
     */
    public function setup(array $conf)
    {
        // here you are getting all of your configured options
        // $conf[ <option name> ] = <option value>
        $this->someConfigValue = $conf['foo']
    }

    /**
     * In here you should create your backup.
     *
     * @return \phpub\App\Backup\Source\Status
     * @throws \phpbu\App\Backup\Source\Exception
     */
    public function backup(Target $target, Result $result)
    {
        // use these methods to store the backup at the configured location
        // $target->getPath()
```

```
        // $target->getPathname()
        // $target->getFilename()
        // to know if the backup should be compressed use
        // $target->shouldBeCompressed()
        // to get the compression settings use
        // $target->getCompressor()
        // if you want to log some debug information use
        // $result->debug('some message');
        // anything bad happens throw a \phpbu\App\Exception
        // return a backup status
        // this is important so the runner knows if it still has to compress the backup
        // return Status::create()->uncompressed($pathToUncompressedFile);
    }
}
```

Create a custom Check

Custom Check implementation example.

Example 11.6. Create a custom check

```
<?php
namespace Acme;

use \phpbu\App\Backup\Check;
use \phpbu\App\Backup\Collector;
use \phpbu\App\Backup\Target;

class MyCheck implements Check
{
    /**
     * Perform your check.
     */
    public function pass(Target $target, $value, Collector $collector)
    {
        // use Target to refer to the created backup
        // $target->getPathname()
        // $value is the configured value
        // <check type="mycheck" value="myvalue"/>
        // $value = "myvalue"
        // use the collector to get a list of previous backups array<splFileInfo>
        // $history = $collector->getBackups()
        // throw a \phpbu\App\Backup\Check\Exception if something bad is happening
        // finally return if the check passes or not
        return $boolean;
    }
}
```

Create a custom Sync

Implementing a custom Sync.

Example 11.7. Create a custom sync

```
<?php
namespace Acme;

use \phpbu\App\Backup\Sync;
use \phpbu\App\Backup\Target;
```

```
class MySync implements Sync
{
    /**
     * Some data that has to be configured.
     */
    private $someConfigValue;

    /**
     * Here you should validate the configuration and setup your class.
     */
    public function setup(array $conf)
    {
        // here you are getting all of your configured options
        // $conf[ <option name> ] = <option value>
        $this->someConfigValue = $conf['foo']
    }

    /**
     * Execute your sync.
     */
    public function sync(Target $target, Result $result)
    {
        // use Target to refer to the created backup
        // $target->getPathname()
        // execute your sync
        // if you want to log some debug information use
        // $result->debug('some message');
        // throw a \phpbu\App\Backup\Sync\Exception if something is going wrong
    }
}
```

Create a custom Cleaner

Implementing a custom cleaner.

Example 11.8. Create a custom cleaner

```
<?php
namespace Acme;

use \phpbu\App\Backup\Collector;
use \phpbu\App\Backup\Sync;
use \phpbu\App\Backup\Target;

class MyCleaner implements Sync
{
    /**
     * Some data that has to be configured.
     */
    private $someConfigValue;

    /**
     * Here you should validate the configuration and setup your class.
     */
    public function setup(array $conf)
    {
        // here you are getting all of your configured options
        // $conf[ <option name> ] = <option value>
        $this->someConfigValue = $conf['foo']
    }

    /**
```

```
     * Execute your cleanup.
     */
    public function cleanup(Target $target, Collector $collector, Result $result)
    {
        // use Target to refer to the created backup
        // $target->getPathname()
        // use the collector to get a list of previous backups array<splFileInfo>
        // delete files matching your cleanup rules
        // $history = $collector->getBackups()
        // if you want to log some debug information use
        // $result->debug('some message');
        // throw a \phpbu\App\Backup\Cleaner\Exception if something is going wrong
    }
}
```

Create a custom Adapter

Custom Adapter implementation example.

Example 11.9. Create a custom adapter

```
<?php
namespace Acme;

use \phpbu\App\Adapter;

class MyAdapter implements Adapter
{
    /**
     * Setup the adapter.
     *
     * @param  array $conf
     * @return void
     */
    public function setup(array $conf)
    {
        // here you will get all configured adapter options
    }

    /**
     * Return a value for a given path.
     *
     * @param  string $path
     * @return string
     */
    public function getValue($path)
    {
        // path is everything behind the second colon
        // adapter:name:$path
        return 'some_value';
    }
}
```

Create a custom Logger

List of available events:

- phpbu.debug
- phpbu.app_start

- phpbu.app_end
- phpbu.backup_start
- phpbu.backup_failed
- phpbu.backup_end
- phpbu.crypt_start
- phpbu.crypt_failed
- phpbu.crypt_end
- phpbu.check_start
- phpbu.check_failed
- phpbu.check_end
- phpbu.sync_start
- phpbu.sync_failed
- phpbu.sync_skipped
- phpbu.sync_end
- phpbu.cleanup_start
- phpbu.cleanup_failed
- phpbu.cleanup_skipped
- phpbu.cleanup_end

Implementing a custom logger.

Example 11.10. Create a custom logger

```
<?php
namespace Acme;

use phpbu\App\Event;
use phpbu\App\Log\Logger;

class MyLogger implements Logger
{
    /**
     * Logger interface requires 'getSubscribedEvents' method.
     * Define the events you want to get notified about.
     */
    public static function getSubscribedEvents()
    {
        return array(
            'phpbu.debug'   => 'onDebug',
            'phpbu.app_end' => 'onPhpbuEnd',
        );
    }

    /**
     * Logger interface requires 'setup' method to configure your logger.
     * Here you should validate the configuration and setup your class.
```

```
     */
    public function setup(array $conf)
    {
        // here you are getting all of your configured options
        // $conf[ <option name> ] = <option value>
    }

    /**
     * Handle the debug event.
     */
    public function onDebug(Event\Debug $event)
    {
        // use $event->getMessage() to get the debug message
    }

    /**
     * Handle phpbuEnd event
     */
    public function onPhpbuEnd(Event\App\End $event)
    {
        // handle the phpbuEnd
        // use $event->getResult() to get the application result
    }
}
```

Summary

Example 11.11. Example file structure

```
acme/
├── backup/
├── src/
|   ├── bootstrap.php
|   ├── MyCheck.php
|   ├── MyCleaner.php
|   ├── MyCrypt.php
|   ├── MySource.php
|   └── MySync.php
|
├── phpbu.phar
└── phpbu.xml.dist
```

Example 11.12. Example bootstrap.php

```
<?php
// use your autoloader or require your classes manually
// so phpbu can find them while executing you backup
// register your class as source, you may override already registered classes
phpbu\App\Factory::register('source', 'mysource', '\\Acme\\MySource');
phpbu\App\Factory::register('check', 'mycheck', '\\Acme\\MyCheck');
phpbu\App\Factory::register('cleanup', 'mycleanup', '\\Acme\\MyCleaner');
```

Example 11.13. Example phpbu.xml.dist

```
<?xml version="1.0" encoding="UTF-8"?>
<phpbu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="http://schema.phpbu.de/4.0/phpbu.xsd">
  <backups>
    <backup>
      <!-- source -->
      <source type="mysource">
```

```
        <option name="foo" value="bar"/>
      </source>
      <!-- where should the backup be stored -->
      <target dirname="backup"
              filename="mysource-%Y%m%d-%H%i.backup"
              compress="bzip2"/>
      <!-- check the created backup -->
      <check type="mycheck" value="100"/>
      <!-- cleanup the backup location -->
      <cleanup type="mycleanup">
        <option name="foo" value="bar"/>
      </cleanup>
    </backup>
  </backups>
</phpbu>
```

Chapter 12. Framework integrations

Laravel

The `laravel-phpbu` package integrates `phpbu` into the laravel framework so you can use the `artisan` command to execute your backups and a laravel style php configuration to set them up.

Requirements

- PHP 5.5
- Laravel 5.*
- phpbu 4.0.*

Installation

Use composer to install the package.

```
composer require "phpbu/phpbu-laravel"
```

Add the package ServiceProvider to your `config/app.php` configuration file.

```
'providers' => [
    /*
    * phpbu Backup Service Providers...
    */
    phpbu\Laravel\ServiceProvider::class,
];
```

Finally use the laravel `artisan` command to create a configuration skeleton by publishing the package.

```
php artisan vendor:publish --provider="phpbu\Laravel\ServiceProvider"
```

After publishing the ServiceProvider a `phpbu.php` configuration file is created in your config directory.

Configuration

There are two ways of configuring your backup.

1. Use the created `phpbu.php` configuration file
2. Use a standard `phpbu.xml` or `phpbu.json` configuration file

The main difference between those two options is the implemented feature set.

While the phpbu configuration files give you full access to all features of phpbu the laravel configuration file currently does not support all features of phpbu f.e. encryption.

Using the laravel configuration

The main concept of configuring backups stays the same. A backup consists of a source and a target. The source represents the data you want to backup. The target represents the location where you want

to store your backup. You can add checks, cleanups and syncs to a backup to validate, cleanup or sync your backup to another location.

In the laravel configuration file you can specify two types of backups, `directories` and `databases` (MySQL, PostgreSQL).

Directories

Define the directory you want to backup and the target location where to store your backup.

```
'directories' => [
        [
            'source' => [
                'path'    => storage_path('app'),
                'options' => [],
            ],
            'target' => [
                'dirname'     => storage_path('/backup/app'),
                'filename'    => 'app-%Y%m%d-%H%i.tar',
                'compression' => 'bzip2',
            ]
        ]
    ],
```

Databases

Just define the laravel database connection configured in your `databases.php` configuration you want to backup and you are done. You can overwrite every setting defined with the specific option f.e. `username` or `password` in case you have to use different credentials to backup your database.

You can use every option the mysqldump source supports f.e. to exclude some tables. Just use the *option-name* as index and the *option-value* as value.

```
'databases' => [<emphasise>
        [
            'source' => [
                'connection' => 'mysql',
                'options'    => []
            ],
            'target' => [
                'dirname'     => storage_path('backup/db'),
                'filename'    => 'dump-%Y%m%d-%H%i.sql',
                'compression' => 'bzip2',
            ]
        ],
    ],
```

Adding Check, Sync and Cleanup

Add *check*, *sync* and *cleanup* definitions to your backup configuration.

```
'check' => [
                'type'  => 'SizeMin',
                'value' => '10B',
            ],
            'sync' => [
              'filesystem' => 's3',
              'path'       => '/backups/db'
            ],
            'cleanup' => [
                'type'    => 'quantity',
```

```
                'options' => [
                    'amount' => '20'
                ]
            ]
```

For a list of available `checks` and `cleanups` check the corresponding documentation sections.

To sync your backups you can use every filesystem defined in your laravel `filesystems.php` configuration file.

Using a phpbu configuration file

If you want to use a phpbu configuration file your laravel phpbu.php configuration should look something like this.

```
<?php
return [
    /*
    |--------------------------------------------------
    | phpbu configuration
    |--------------------------------------------------
    |
    | Path to a phpbu configuration xml or json file.
    */

    'phpbu' => base_path('phpbu.xml'),
];
```

For this to work you have to setup a valid phpbu.xml configuration file in your project root.

Remember that all paths in a phpbu configuration have to to absolute or relative to the specified phpbu configuration file.

Usage

To execute the backup use the `artisan` command line tool.

```
php artisan phpbu:backup
```

To execute a dry-run without making any changes.

```
php artisan phpbu:backup --phpbu-simulate
```

Appendix A. Index

Index

S

T

V

X

Appendix B. Copyright

```
THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF
SUCH TERMS AND CONDITIONS.

1. Definitions

   a. "Adaptation" means a work based upon the Work, or upon the
      Work and other pre-existing works, such as a translation,
      adaptation, derivative work, arrangement of music or other
      alterations of a literary or artistic work, or phonogram or
      performance and includes cinematographic adaptations or any
      other form in which the Work may be recast, transformed, or
      adapted including in any form recognizably derived from the
      original, except that a work that constitutes a Collection
      will not be considered an Adaptation for the purpose of this
      License. For the avoidance of doubt, where the Work is a
      musical work, performance or phonogram, the synchronization of
      the Work in timed-relation with a moving image ("synching")
      will be considered an Adaptation for the purpose of this
      License.

   b. "Collection" means a collection of literary or artistic works,
      such as encyclopedias and anthologies, or performances,
      phonograms or broadcasts, or other works or subject matter
      other than works listed in Section 1(f) below, which, by
      reason of the selection and arrangement of their contents,
      constitute intellectual creations, in which the Work is
      included in its entirety in unmodified form along with one or
      more other contributions, each constituting separate and
      independent works in themselves, which together are assembled
      into a collective whole. A work that constitutes a Collection
      will not be considered an Adaptation (as defined above) for
      the purposes of this License.

   c. "Distribute" means to make available to the public the
      original and copies of the Work or Adaptation, as appropriate,
      through sale or other transfer of ownership.

   d. "Licensor" means the individual, individuals, entity or
      entities that offer(s) the Work under the terms of this License.

   e. "Original Author" means, in the case of a literary or artistic
      work, the individual, individuals, entity or entities who
      created the Work or if no individual or entity can be
      identified, the publisher; and in addition (i) in the case of
      a performance the actors, singers, musicians, dancers, and
      other persons who act, sing, deliver, declaim, play in,
      interpret or otherwise perform literary or artistic works or
      expressions of folklore; (ii) in the case of a phonogram the
      producer being the person or legal entity who first fixes the
      sounds of a performance or other sounds; and, (iii) in the
      case of broadcasts, the organization that transmits the
      broadcast.

   f. "Work" means the literary and/or artistic work offered under
      the terms of this License including without limitation any
      production in the literary, scientific and artistic domain,
      whatever may be the mode or form of its expression including
      digital form, such as a book, pamphlet and other writing; a
      lecture, address, sermon or other work of the same nature; a
      dramatic or dramatico-musical work; a choreographic work or
      entertainment in dumb show; a musical composition with or
      without words; a cinematographic work to which are assimilated
      works expressed by a process analogous to cinematography; a
      work of drawing, painting, architecture, sculpture, engraving
```

or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

```
      i. Non-waivable Compulsory License Schemes. In those
         jurisdictions in which the right to collect royalties
         through any statutory or compulsory licensing scheme cannot
         be waived, the Licensor reserves the exclusive right to
         collect such royalties for any exercise by You of the
         rights granted under this License;

      ii. Waivable Compulsory License Schemes. In those
          jurisdictions in which the right to collect royalties
          through any statutory or compulsory licensing scheme can
          be waived, the Licensor waives the exclusive right to
          collect such royalties for any exercise by You of the
          rights granted under this License; and,

      iii. Voluntary License Schemes. The Licensor waives the right
           to collect royalties, whether individually or, in the
           event that the Licensor is a member of a collecting
           society that administers voluntary licensing schemes, via
           that society, from any exercise by You of the rights
           granted under this License.

The above rights may be exercised in all media and formats whether
now known or hereafter devised. The above rights include the right
to make such modifications as are technically necessary to exercise
the rights in other media and formats. Subject to Section 8(f), all
rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly
   made subject to and limited by the following restrictions:

   a. You may Distribute or Publicly Perform the Work only under the
      terms of this License. You must include a copy of, or the
      Uniform Resource Identifier (URI) for, this License with every
      copy of the Work You Distribute or Publicly Perform. You may
      not offer or impose any terms on the Work that restrict the
      terms of this License or the ability of the recipient of the
      Work to exercise the rights granted to that recipient under
      the terms of the License. You may not sublicense the Work. You
      must keep intact all notices that refer to this License and to
      the disclaimer of warranties with every copy of the Work You
      Distribute or Publicly Perform. When You Distribute or
      Publicly Perform the Work, You may not impose any effective
      technological measures on the Work that restrict the ability
      of a recipient of the Work from You to exercise the rights
      granted to that recipient under the terms of the License. This
      Section 4(a) applies to the Work as incorporated in a
      Collection, but this does not require the Collection apart
      from the Work itself to be made subject to the terms of this
      License. If You create a Collection, upon notice from any
      Licensor You must, to the extent practicable, remove from the
      Collection any credit as required by Section 4(b), as
      requested. If You create an Adaptation, upon notice from any
      Licensor You must, to the extent practicable, remove from the
      Adaptation any credit as required by Section 4(b), as requested.

   b. If You Distribute, or Publicly Perform the Work or any
      Adaptations or Collections, You must, unless a request has
      been made pursuant to Section 4(a), keep intact all copyright
      notices for the Work and provide, reasonable to the medium or
      means You are utilizing: (i) the name of the Original Author
      (or pseudonym, if applicable) if supplied, and/or if the
      Original Author and/or Licensor designate another party or
      parties (e.g., a sponsor institute, publishing entity,
```

journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

   a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

   b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

   a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

   b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

   c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

   d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

   e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

   f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be

```
      enforced according to the corresponding provisions of the
      implementation of those treaty provisions in the applicable
      national law. If the standard suite of rights granted under
      applicable copyright law includes additional rights not
      granted under this License, such additional rights are deemed
      to be included in the License; this License is not intended to
      restrict the license of any rights under applicable law.

Creative Commons is not a party to this License, and makes no
warranty whatsoever in connection with the Work. Creative Commons
will not be liable to You or any party on any legal theory for any
damages whatsoever, including without limitation any general,
special, incidental or consequential damages arising in connection
to this license. Notwithstanding the foregoing two (2) sentences,
if Creative Commons has expressly identified itself as the Licensor
hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the
Work is licensed under the CCPL, Creative Commons does not authorize
the use by either party of the trademark "Creative Commons" or any
related trademark or logo of Creative Commons without the prior
written consent of Creative Commons. Any permitted use will be in
compliance with Creative Commons' then-current trademark usage
guidelines, as may be published on its website or otherwise made
available upon request from time to time. For the avoidance of
doubt, this trademark restriction does not form part of this
License.

Creative Commons may be contacted at http://creativecommons.org/.

======================================================================
```