

vfs://Stream

Frank Kleine

Frank Kleine

Author of vfsStream

Software Architect
at 1&1 Internet

 @bovigo



vfs://Stream

A stream wrapper
to mock PHP's filesystem functions
into thinking they work with the real file system.

Stream wrapper?

Allows you to implement your own protocol handlers and streams for use with all other filesystem functions (such as `fopen()`, `fread()` etc.)

quoted from <http://php.net/manual/en/class.streamwrapper.php>

- Create a class with predefined methods
- Register class as handler for a protocol
- Class will be instantiated when a file pointer to a resource of this protocol is opened
- Filesystem function calls result in method calls on the instance of your class

Stream wrapper practice: implementation

```
class MyStreamWrapper {  
    private $read = false;  
  
    public function stream_open(...) { return true; }  
  
    public function stream_stat() { return []; }  
  
    public function stream_read($count) {  
        if (!$this->read) {  
            $this->read = true;  
            return 'Hello world!';  
        }  
  
        return false;  
    }  
  
    public function stream_eof() { return $this->read; }  
}
```

```
stream_wrapper_register(  
    'my',  
    MyStreamWrapper::class  
);  
  
var_dump(file_get_contents('my://hello'));  
  
// string(12) "Hello world!"
```




File system based class

```
class FileSystemCache {
    public function __construct($dir) {
        $this->dir = $dir;
    }

    public function store($key, $data) {
        if (!file_exists($this->dir)) {
            mkdir($this->dir, 0700, true);
        }

        file_put_contents($this->dir.'/'.$key, serialize($data));
    }
}
```

The Test: storing the data correctly

```
public function testStoresDataInFile() {  
  
    $cache = new FileSystemCache(__DIR__ . '/cache');  
  
    $cache->store('example', ['bar' => 303]);  
  
    $this->assertEquals(  
  
        ['bar' => 303],  
  
        unserialize(file_get_contents(__DIR__ . '/cache/example'))  
  
    );  
  
}
```

The Test: creation of directory

```
public function testCreatesDirectoryIfNotExists() {  
    $cache = new FileSystemCache(__DIR__ . '/cache');  
    $cache->store('example', ['bar' => 303]);  
    $this->assertFileExists(__DIR__ . '/cache');  
}
```

```
public function setUp() {  
    if (file_exists(__DIR__ . '/cache/example')) {  
        unlink(__DIR__ . '/cache/example');  
    }  
  
    if (file_exists(__DIR__ . '/cache')) {  
        rmdir(__DIR__ . '/cache');  
    }  
  
}
```

```
private function clean() {  
    if (file_exists(__DIR__ . '/cache/example')) {  
        unlink(__DIR__ . '/cache/example');  
    }  
  
    if (file_exists(__DIR__ . '/cache')) {  
        rmdir(__DIR__ . '/cache');  
    }  
  
}  
  
public function setUp() { $this->clean(); }  
public function tearDown() { $this->clean(); }
```

```
public function setUp() {  
    $this->root = vfsStream::setup();  
}
```



```
public function testCreatesDirectoryIfNotExists() {  
  
    $cache = new FileSystemCache(  
  
        $this->root->url() . '/cache'  
  
    );  
  
    $cache->store('example', ['bar' => 303]);  
  
    $this->assertTrue($this->root->hasChild('cache'));  
  
}
```

Test... better

```
public function testStoresDataInFile() {  
  
    $cache = new FileSystemCache($this->root->url().'/cache');  
  
    $cache->store('example', ['bar' => 303]);  
  
    $this->assertTrue($this->root->hasChild('cache/example'));  
  
    $this->assertEquals(  
  
        ['bar' => 303],  
  
        unserialize(  
  
            $this->root->getChild('cache/example')->getContent()  
  
        )  
  
    );  
  
}
```

Adding permissions

```
class FileSystemCache {  
    public function __construct($dir, $permissions = 0700) {  
        $this->dir = $dir;  
        $this->permissions = $permissions;  
    }  
  
    public function store($key, $data) {  
        if (!file_exists($this->dir)) {  
            mkdir($this->dir, $this->permissions, true);  
        }  
  
        file_put_contents(  
            $this->dir . '/' . $key,  
            serialize($data)  
        );  
    }  
}
```

Test permissions

```
public function testDirectoryIsCreatedWith0700ByDefault()  
{  
  
    $cache = new FileSystemCache(__DIR__ . '/cache');  
  
    $cache->store('example', ['bar' => 303]);  
  
    $this->assertEquals(  
  
        40700,  
  
        decoct(fileperms(__DIR__ . '/cache'))  
  
    );  
  
}
```

Test permissions

```
public function testDirectoryIsCreatedWithProvidedPermissions()
{

    umask(0);

    $cache = new FileSystemCache(__DIR__ . '/cache', 0770);

    $cache->store('example', ['bar' => 303]);

    $this->assertEquals(

        40770, decoct(fileperms(__DIR__ . '/cache'))

    );

}
```

Considering Windows

```
public function testDirectoryIsCreatedWithProvidedPermissions() {  
  
    umask(0);  
  
    $cache = new FileSystemCache(__DIR__ . '/cache', 0770);  
  
    $cache->store('example', ['bar' => 303]);  
  
    if (DIRECTORY_SEPARATOR === '\\') {  
  
        $this->assertEquals(40777, decoct(fileperms(__DIR__ . '/cache')));  
  
    } else {  
  
        $this->assertEquals(40770, decoct(fileperms(__DIR__ . '/cache')));  
  
    }  
  
}
```


Doing it right

```
public function testDirectoryIsCreatedWithProvidedPermissions() {  
  
    $cache = new FileSystemCache(  
  
        $this->root->url() . '/cache', 0770  
  
    );  
  
    $cache->store('example', ['bar' => 303]);  
  
    $this->assertEquals(  
  
        0770,  
  
        $this->root->getChild('cache')->getPermissions()  
  
    );  
  
}
```

Interlude: file permissions for delete

```
→ cache touch example
→ cache chmod 000 example
→ cache ls -l
total 0
----- 1 mikey staff 0B 23 0kt 08:26 example
→ cache rm example
→ cache ls -l
→ cache touch example
→ cache ls -l
total 0
-rw-r--r-- 1 mikey staff 0B 23 0kt 08:27 example
→ cache cd ..
→ test chmod 555 cache
→ test cd cache
→ cache ls -l
total 0
-rw-r--r-- 1 mikey staff 0B 23 0kt 08:27 example
→ cache rm example
rm: example: Permission denied
```

Error handling

```
class FileSystemCache {  
    ...  
    public function store($key, $data) {  
        if (!file_exists($this->dir)) {  
            mkdir($this->dir, $this->permissions, true);  
        }  
        if (false === @file_put_contents($this->dir . '/' . $key,  
serialize($data))) {  
            throw new \Exception('Failure to store ' . $key . ': '  
error_get_last()['message']);  
        }  
        return true;  
    }  
}
```

Testing error handling

```
/**
 * @test
 * @expectedException \Exception
 * @expectedExceptionMessage failed to open stream
 */
public function throwsExceptionWhenFailureOccurs() {
    vfsStream::newFile('example', 0000)
        ->withContent('notoverwritten')
        ->at($this->root);

    $cache = new FileSystemCache($this->root->url());
    $cache->store('example', ['bar' => 303]);
}
```

Alternative: pretend full disc

```
/**
 * @test
 * @expectedException Exception
 * @expectedExceptionMessage possibly out of free disk space
 */
public function throwsExceptionWhenDiskFull() {
    vfsStream::setQuota(10);
    $cache = new FileSystemCache($this->root->url());
    $cache->store('example', ['bar' => 303]);
}
```

Config files

```
public function get($id) {  
    if (!isset($this->properties()[$id])) {  
        if (!$this->hasFallback()) {  
            throw new \Exception(...);  
        }  
        $id = 'default';  
    }  
    if (!isset($this->properties()[$id]['dsn'])) {  
        throw new \Exception(...);  
    }  
    return $this->properties()[$id];  
}
```


Config files, continued

```
protected function properties() {  
    if (null === $this->dbProperties) {  
        $propertiesFile = $this->configPath . '/' . $this->descriptor . '.ini';  
        if (!file_exists($propertiesFile) || !is_readable($propertiesFile)) {  
            throw new \Exception(...);  
        }  
        $propertyData = @parse_ini_file($propertiesFile, true);  
        if (false === $propertyData) {  
            throw new \Exception(...);  
        }  
        $this->dbProperties = $propertyData;  
    }  
    return $this->dbProperties;  
}
```

Test with different config files

```
public function setUp() {  
    $root = vfsStream::setup();  
    $this->configFile = vfsStream::newFile($filename)->at($root);  
    $this->propertyBasedConfig = new PropertyBasedDbConfig($root->url());  
}
```

```
public function testReturnsConfigWhenPresentInFile() {  
    $this->configFile->setContent(' [foo]  
dsn="mysql:host=localhost;dbname=foo" ');  
    $this->assertEquals(  
        ['dsn' => 'mysql:host=localhost;dbname=foo'],  
        $this->propertyBasedConfig->get('foo')  
    );  
}
```

Test with different config files

```
public function testReturnsDefaultWhenNotPresentInFileButDefaultConfigured()
{
    $this->configFile->setContent( '[default]
dsn="mysql:host=localhost;dbname=example" ');

    $this->assertEquals(
        [ 'dsn' => 'mysql:host=localhost;dbname=example' ],
        $this->propertyBasedConfig->get( 'foo' )
    );
}

public function testThrowsExceptionWhenNotPresentInFile() {
    $this->configFile->setContent( '[bar]
dsn="mysql:host=localhost;dbname=example" ');

    $this->propertyBasedConfig->get( 'foo' );
}
```

Large files

Large files

```
$root      = vfsStream::setup();

$largeFile = vfsStream::newFile('large.txt')

                ->withContent(LargeFileContent::withGigabytes(100))

                ->at($root);

var_dump(filesize($largeFile->url())); // int(107374182400)
```

Setup with a predefined directory structure

```
public function example() {  
    $structure = [  
        'examples' => [  
            'test.php' => 'some text content',  
            'other.php' => 'Some more text content',  
            'Invalid.csv' => 'Something else',  
        ],  
        'an_empty_folder' => [],  
        'badlocation.php' => 'some bad content',  
        '[Foo]' => 'a block device'  
    ];  
  
    $root = vfsStream::setup('root', null, $structure);  
  
    $this->assertTrue($root->hasChild('examples/test.php'));  
}
```

Setup by copy from real file system

```
public function example() {  
  
    $root = vfsStream::setup();  
  
    vfsStream::copyFromFileSystem(__DIR__ . '/../..', $root);  
  
    $this->assertTrue(  
  
        $root->hasChild(  
  
            'part06/SetupByCopyFromFileSystem.php'  
  
        )  
  
    );  
  
}
```

Using a visitor for assertions

```
public function example() {  
    $structure = [...];  
  
    vfsStream::setup('root', null, $structure);  
  
    // some operation which changes the structure here  
  
    $this->assertEquals(  
        ['root' => $changedStructure],  
        vfsStream::inspect(  
            new vfsStreamStructureVisitor()  
        )->getStructure()  
    );  
}
```


Inspecting the virtual file system

```
public function example() {  
  
    $structure = ['examples' => ['test.php'      => 'some text content',  
                                'other.php'     => 'Some more text content',  
                                'Invalid.csv' => 'Something else',  
                                ],  
                'an_empty_folder' => [],  
                'badlocation.php' => 'some bad content',  
                '[Foo]'           => 'a block device'  
    ];  
  
    vfsStream::setup('root', null, $structure);  
  
    vfsStream::inspect(new vfsStreamPrintVisitor());  
  
}
```

- root
 - examples
 - test.php
 - other.php
 - Invalid.csv
 - an_empty_folder
 - badlocation.php
 - [Foo]

Limitations: complete list

github.com/mikey179/vfsStream/wiki/Known-Issues

Limitations: not implemented

```
stream_set_blocking()
```

```
stream_set_timeout()
```

```
stream_set_write_buffer()
```

`chdir()`

`realpath()`

`SplFileInfo::getRealPath()`

`link()`

`symlink()`

`readlink()`

`linkinfo()`

`tempnam()`

HHVM :-/

```
191 $ php --version
192 HipHop VM 3.6.6 (rel)
193 Compiler: tags/HHVM-3.6.6-0-gead6875c956d4bd5e906ebea899c71c3b8a7182d
194 Repo schema: 5034239b4e72c991556d7e651946e7fe89d8c2f4
195 $ composer --version
196 Composer version 1.0-dev (40c14709f79f9d7ea35ac969cfbd7f41beb525bb) 2016-04-05 12:27:16
197
▶ 198 $ composer install install.1 6.87s
▶ 262 $ composer require satooshi/php-coveralls install.2 22.46s
302 $ vendor/bin/phpunit --coverage-text 22.05s
303 PHPUnit 4.8.10 by Sebastian Bergmann and contributors.
304
305 Runtime:      HHVM 3.6.6
306 Configuration: /home/travis/build/mikey179/vfsStream/phpunit.xml.dist
307 Warning:      The Xdebug extension is not loaded
308               No code coverage will be generated.
309
310 F..... 63 / 397 ( 15%)
311 ..... 126 / 397 ( 31%)
312 ..... 189 / 397 ( 47%)
313 ..... 252 / 397 ( 63%)
314 .....S.....SS...S..... 315 / 397 ( 79%)
315 ..... 378 / 397 ( 95%)
316 ....S.....
317
318 Time: 21.77 seconds, Memory: 24.10Mb
319
320 There was 1 failure:
321
322 1) /home/travis/build/mikey179/vfsStream/src/test/phpt/bug71287.phpt
323 Failed asserting that format description matches text.
324 --- Expected
325 +++ Actual
326 @@ @@
327 -Warning: file_put_contents(): Only 7 of 9 bytes written, possibly out of free disk space in %s on line %d
328 +
329
330 FAILURES!
331 Tests: 397, Assertions: 1342, Failures: 1, Skipped: 5.
```

More info

vfs.bovigo.org

github.com/mikey179/vfsStream-examples

[@bovigo](#)

Happy testing