/////////////////////////////////////////////////////////////////////////////////////////////////////////////

// aConnectorInstance.inc

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

<?php

abstract class aConnectorInstance

{

public $mdbRes = null;

public $status = false;

public function \_\_construct(string $\_uri, int $\_port, string $\_login = '', string $\_pwd = '')

{

try {

$manager = new MongoDB\Driver\Manager('mongodb://' . $\_uri . ':' . $\_port);

$command = new MongoDB\Driver\Command(['ping' => 1]);

$results = $manager->executeCommand('admin', $command);

var\_export(current($results->toArray()));

echo PHP\_EOL;

$this->mdbRes = $manager;

$this->status = true;

} catch (\MongoDB\Driver\Exception\ConnectionException |

\MongoDB\Driver\Exception\InvalidArgumentException |

\MongoDB\Driver\Exception\RuntimeException |

\MongoDB\Driver\Exception\Exception |

Throwable $e) {

$msg = sprintf('%s@%d-->', basename(\_\_FILE\_\_), \_\_LINE\_\_);

echo $msg . 'An exception was trapped connection to mongo stand-alone instance' . PHP\_EOL;

echo $e->getMessage() . PHP\_EOL;

}

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

// aMongoToolbox.inc

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

<?php

try {

require\_once "aConnectorInstance.inc";

} catch (ParseError $p) {

$msg = sprintf("%s@%d-->", basename(\_\_FILE\_\_), \_\_LINE\_\_);

echo $msg . 'Parse error encountered!' . PHP\_EOL;

echo $p->getMessage() . PHP\_EOL;

exit(1);

}

abstract class aMongoToolbox extends aConnectorInstance

{

public function \_\_construct(string $\_uri, int $\_port, string $\_login = '', string $\_pwd = '')

{

parent::\_\_construct($\_uri, $\_port, $\_login, $\_pwd);

if (!$this->status) {

echo 'Unable to connect to the mongoDB instance' . PHP\_EOL;

return;

}

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

// cUserManager.inc

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

<?php

try {

require\_once "aMongoToolbox.inc";

} catch (ParseError $p) {

$msg = sprintf("%s@%d-->", basename(\_\_FILE\_\_), \_\_LINE\_\_);

echo $msg . 'Parse error encountered!' . PHP\_EOL;

echo $p->getMessage() . PHP\_EOL;

exit(1);

}

class cUserManager extends aMongoToolbox

{

public function \_\_construct(array $\_mongoConnectData)

{

list($uri, $port, $login, $pass) = $\_mongoConnectData;

parent::\_\_construct($uri, $port, $login, $pass);

if (!$this->status) return;

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

// testMongo.php

/////////////////////////////////////////////////////////////////////////////////////////////////////////////

<?php

try {

include "cUserManager.inc";

} catch (ParseError $p) {

$msg = sprintf("%s@%d-->", basename(\_\_FILE\_\_), \_\_LINE\_\_);

echo $msg . 'Parse error encountered!' . PHP\_EOL;

echo $p->getMessage() . PHP\_EOL;

exit(1);

}

$data = [

$uri = '192.168.1.57',

$port = 27017,

$login = '',

$pass = ''

];

try {

$toolBox = new cUserManager($data);

} catch (TypeError $t) {

$msg = sprintf("%s@%d-->", basename(\_\_FILE\_\_), \_\_LINE\_\_);

echo $msg . 'Caught TypeError: ' . PHP\_EOL;

echo $t->getMessage() . PHP\_EOL;

exit(1);

}

if ($toolBox->status)

echo 'Successfully Connected to stand-alone mongoDB!!!' . PHP\_EOL;