

To check your php communicates with MongoDB well. Create a php file contains the following code

```
<?php

// connect to mongodb
$host="localhost:27017";
$userdb="dbtesting";

try{
    $manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");
    if ($manager)
    {
        $command = new MongoDB\Driver\Command(['create' => 'phptest']); /* create database (dbtesting) and collection (phptest) without Documents */
        $cursor = $manager->executeCommand($userdb, $command);
        echo "<pre>";
        var_dump($cursor);
        echo "</pre>";
    }
}

catch(Exception $e){
    echo "<center><h1>Doesn't work</h1></center>";
    print_r($e);
    exit;
}

if ($host){
    // select a database
echo "Database ". $userdb. " selected";
echo "<center><h1>Good connection</h1></center>";
}

?>
```

Save the file as **dbtest.php** in the folder xampp\htdocs. Call the file **dbtest.php** in the browser to check if it is working or not

## Example Drop database using MongoDB\Driver\Manager and PHP

We can use the MongoDB\Driver\Manager and PHP to drop a database using the following code.

Copy the code to a PHP file and save it as **DropDatabase.php** in the folder xampp\htdocs. Call the file **DropDatabase.php** in the browser to check if it drops the database or not

```
<?php

$host = "localhost:27017";
$userdb = "dbtesting"; // Name of the database to drop

try {
    // Create a new Manager instance
    $manager = new MongoDB\Driver\Manager("mongodb://{$host}");

    // Define the dropDatabase command
    $command = new MongoDB\Driver\Command(["dropDatabase" => 1]);

    // Execute the command
    $result = $manager->executeCommand($userdb, $command);

    echo "Database '{$userdb}' dropped successfully.";

} catch (MongoDB\Driver\Exception\Exception $e) {
    echo "Failed to drop database: ", $e->getMessage();
}

?>
```

## Example Create Collection using MongoDB\Driver\Manager and PHP

As seen before, to connect to a mongodb database mongodb driver manager is used. All database connections are maintained by mongodb driver manager. Database connection is not created on instantiation of a mongodb driver manager object. Driver manager takes parameter server address localhost with a port 27017.

We can use the MongoDB\Driver\Manager and PHP to create Collection.

```
<?php

$host="localhost:27017";
$userdb="Bookshop";

try{

    $manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");

    if ($manager) {
        $command = new MongoDB\Driver\Command(['create' => 'DatabaseBooks']); /* create database
(Bookshop) and collection (DatabaseBooks) without Documents */

        $cursor = $manager->executeCommand($userdb, $command);
        echo "<pre>";
    }
}catch(Exception $e){
    echo "<center><h1>Doesn't work</h1></center>";
    print_r($e);
    exit;
}

if ($host)
echo "<center><h1>Database Created Successfully</h1></center>";

?>
```

Save the file as **CreateCollection.php** in the folder xampp\htdocs. Call the file **CreateCollection.php** in the browser to check if it is working or not

## Creating a document

You can create a document by inserting it into a collection using PHP code as follows.

```
<?php

$host="localhost:27017";
$userdb="Bookshop";
$database=$userdb.".DatabaseBooks"; /* this is the database with collection */
$document = array(
    "_id" => "derickr",
    "name" => "Derick Rethans",
    "articles" => array(
        array(
            "title" => "Profiling PHP Applications",
            "url" => "http://derickrethans.nl/talks/profiling-
phptourlille11.pdf",
            ),
        array(
            "title" => "Xdebug",
            "url" => "http://derickrethans.nl/talks/xdebug-phpbcn11.pdf",
            )
        )
    );
try{

$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");
if ($manager) {
$bulk = new MongoDB\Driver\BulkWrite;

$bulk->insert($document);

$manager->executeBulkWrite($database, $bulk);
echo '<pre>';
var_dump($manager);
echo '</pre>';
}
}
catch (MongoDB\Driver\Exception\Exception $e) {
    echo "Failed to Create Document ", $e->getMessage();
}
?>
```

```
<?php
```

```
$host="localhost:27017";
$userdb="Bookshop";
$database =$userdb.".DatabaseBooks";

try{

$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");
if ($manager) {
$bulk = new MongoDB\Driver\BulkWrite;
$bulk->insert(['title' => "MongoDB Tutorials",
                'author name' => "Mikey",
                'email' => "Mikey@yahoo.co",
                'ISBN' => "978-0-470-40030-2",'year' => "2005"]);
$bulk->insert(['title' => "PHP and MongoDB",
                'author name' => "Davey Bungooligan",
                'email' => "Davey@hotmail.com",
                'ISBN' => "978-0-13-801014-0",'year' => "2009"]);
$bulk->insert(['title' => "MySQL Databases",
                'author name' => "Suzy B",
                'email' => "Suzy@aol.com",
                'ISBN' => "0-13-80-801014-5",'year' => "2011"]);
$bulk->insert(['title' => "Database System with Java",
                'author name' => "Mikey",
                'email' => "Mikey@yahoo.com",
                'ISBN' => "865-0-740-61020-1",'year' => "2004"]);
$bulk->insert(['title' => "High Performance MySQL",
                'author name' => "Terrence",
                'email' => "Terrence@gmail.com",
                'ISBN' => "845-0-540-71020-2",'year' => "2014"]);
$bulk->insert(['title' => "Fundamentals of Database Systems",
                'author name' => "Philomena Jones",
                'email' => "Philomena@bradford.gov",
                'ISBN' => "765-0-840-81010-1", 'year' => "2005"]);

$manager->executeBulkWrite($database, $bulk);
}

}catch(Exception $e){
echo "<center><h1>Doesn't work</h1></center>";
print_r($e);
exit;
}
if ($host){ // select a database
echo "Database". $userdb. "selected";
echo "<center><h1>Good connection</h1></center>";
}

?>
```

- Insert the following six documents using PHP.
- If you copy the code shown on the left side into a PHP file and run it from localhost, it will do the job for you.

```
{title: "MongoDB Tutorials", author name: "Mikey", email: "Mikey@yahoo.co", ISBN: "978-0-470-40030-2", year: "2005"}
```

```
{title: "PHP and MongoDB", author name: "Davey Bungooligan", email: "Davey@hotmail.com", ISBN: "978-0-13-801014-0", year: "2009"}
```

```
{title: "MySQL Databases", author name: "Suzy B", email: "Suzy@aol.com", ISBN: "0-13-80-801014-5", year: "2011"}
```

```
{title: "Database System with Java", author name: "Mikey", email: "Mikey@yahoo.com", ISBN: "865-0-740-61020-1", year: "2004"}
```

```
{title: "High Performance MySQL", author name: "Terrence", email: "Terrence@gmail.com", ISBN: "845-0-540-71020-2", year: "2014"}
```

```
{title: "Fundamentals of Database Systems", author name: "Philomena Jones", email: "Philomena@bradford.gov", ISBN: "765-0-840-81010-1", year: "2005"}
```

## Finding a document

You can find a document or documents matching a particular pattern using the **find** function. If you want to find all the database books in the DatabaseBooks collection, you can do this easily using the following PHP code.

```
<?php

$host="localhost:27017";
$userdb="Bookshop";
$database=$userdb.".DatabaseBooks"; /* this is the database with collection */
$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");
if ($manager) {
    $query = new MongoDB\Driver\Query([]);
    $cursor = $manager->executeQuery($database, $query);

    foreach ( $cursor as $r )
    {
        echo '<pre>';
        var_dump( $r );
        echo '</pre>';
    }
}

?>
```

## Finding a document using MongoDB\Driver\Manager with PHP

We can use **MongoDB\Driver\Query()** with a variable **\$filter** to specify the query criteria.

**\$filter** variable is created. Filter can contain any condition to be met to select a records. If you are familiar with relational databases then you can consider it as a where clause.

Example find all the books by author name called **Derick Rethans**

```
<?php

$host="localhost:27017";
$userdb="Bookshop";
$database=$userdb.".DatabaseBooks"; /* this is the database with collection */
$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");
$filter = ['name' => 'Derick Rethans'];
if ($manager) {
    $query = new MongoDB\Driver\Query($filter);
    $cursor = $manager->executeQuery($database, $query);

        foreach ( $cursor as $r )
    {
echo '<pre>';
        var_dump( $r );
echo '<pre>';
    }
}

?>
```

We can use MongoDB\Driver\Query() with \$options as a variable to specify additional query parameters.

\$option variable is created. \$options array contain additional options such as sorting, Projection, limit. mongodb driver's Query method is used to construct a query and accepts filter and options parameters.

Example find five books and sorts according to the book title in ascending order

```
<?php

$host="localhost:27017";
$userdb="Bookshop";
$database=$userdb.".DatabaseBooks"; /* this is the database with collection */
$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");
$filter = [];
$options = ['sort' => [ 'title' => 1], 'limit' => 5]; //projection can be used for inclusion and exclusion
if ($manager) {
    $query = new MongoDB\Driver\Query($filter,$options);
    $cursor = $manager->executeQuery($database, $query);

        foreach ( $cursor as $r )
    {
        echo '<pre>';
        var_dump( $r );
        echo '<pre>';
    }
}

?>
```

**Exercises:**

- Use find to find all the emails.
- Find all the books that were published after 2005.
- Find all the database books named **MySQL Databases**.

We can use **MongoDB\Driver\Query()** with **\$filter** and **\$options** as variables to define query criteria and customize query behaviour.

**Example find all the books published after 2005 and sorts according to the year in descending order**

```
<?php

$host="localhost:27017";
$userdb="Bookshop";
$database=$userdb.".DatabaseBooks"; /* this is the database with collection */
$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");

$filter = ['year' => ['$gt' => '2005']];
$options = ['sort' => ['year' => 1] ];
if ($manager) {
    $query = new MongoDB\Driver\Query($filter,$options);
    $cursor = $manager->executeQuery($database, $query);

    foreach ( $cursor as $r )
    {
        echo '<pre>';
        var_dump( $r );
        echo '</pre>';
    }
}

?>
```

## \$where

We can even filter using an arbitrary JavaScript expression using \$where. This will allow us to compare two fields in a single document.

### Lets use \$where with MongoDB\Driver\Manager and PHP

Example for \$Where

```
<?php

$host="localhost:27017";
$userdb="Bookshop";
$database=$userdb.".DatabaseBooks"; /* this is the database with collection */
$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");

$filter = ['title' => ['$exists' => 'true'], '$where' =>'this.title.length >20'];
$options = ['sort' => ['year' => -1]];
if ($manager) {
    $query = new MongoDB\Driver\Query($filter,$options);
    $cursor = $manager->executeQuery($database, $query);

    foreach ( $cursor as $r )
    {
        echo '<pre>';
        var_dump( $r );
        echo '</pre>';
    }
}

?>
```

## Exercises:

- Check out the PHP script on the right—do you understand what it's intended to do?
- If you're unsure, try running the script to see the results.

## Now

- Use `find` to get all the people who are exactly 99 years old
- Find all the people who are eligible for a bus pass (people older than 65)
- Find all the teenagers, greater than 12 and less than 20.

## Exercise using `$where`

- Use `$where` to find all the people who have a cat.
- Find all the people who are younger than their cats. Remember, not everyone has a cat, so you will need to use a Boolean `&&` to filter out the non-cat owners.
- Does anyone have the same name as their cat?

```
<?php

$host="localhost:27017";
$userdb="people";
$database = $userdb.".person";

$manager = new MongoDB\Driver\Manager("mongodb://{$host}/{$userdb}");

$names = [
    'Yolanda',
    'Iska',
    'Malone',
    'Frank',
    'Foxton',
    'Pirate',
    'Poppelhoffen',
    'Elbow',
    'Fluffy',
    'Paphat'
];

function randName($names) {
    $n = count($names);
    return $names[array_rand($names)] . ' ' . $names[array_rand($names)];
}

function randAge($max) {
    return rand(0, $max - 1);
}

$bulk = new MongoDB\Driver\BulkWrite;

for ($i = 0; $i < 100; ++$i) {
    $person = [
        'name' => randName($names),
        'age' => randAge(100)
    ];

    if (mt_rand() / mt_getrandmax() > 0.8) {
        $person['cat'] = [
            'name' => randName($names),
            'age' => randAge(18)
        ];
    }

    $bulk->insert($person);
}

$result = $manager->executeBulkWrite($database, $bulk);
echo "Inserted " . $result->getInsertedCount() . " documents.\n";
?>
```