

Reference:

http://www.tutorialspoint.com/mongodb/mongodb_php.htm

<http://www.sitepoint.com/introduction-to-mongodb/>

#1>. To use mongodb with php you need to use mongodb php driver.

#2>. Make a connection and Select a database

NOTE: To make a connection, you need to specify database name, if database doesn't exist then mongodb creates it automatically.

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
?>
```

#3>. Create a collection

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->createCollection("mycol");
echo "Collection created successfully";
?>
```

#4>. Insert a document

NOTE: To insert a document into mongodb, insert() method is used.

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->mycol;
echo "Collection selected successfully";
$document = array(
    "title" => "MongoDB",
    "description" => "database",
    "likes" => 100,
    "url" => "http://www.tutorialspoint.com/mongodb/",
    "by", "tutorials point"
);
$collection->insert($document);
echo "Document inserted successfully";
?>
```

NOTE: The MongoDB extension for PHP performs all operations asynchronously by default so you don't have to wait until the server responds that the data was successfully saved. This feature lets you move on to the next task without waiting for confirmation whether the data was saved or not and can be very convenient, but for important data you may want to be sure the document object was really saved and no errors happened. You can achieve this by passing an array as the second parameter of insert() with the key safe set to true.

Try it again, but this time pass the second parameter to insert().

```
<?php
$insertOpts = array("safe" => true);
$collection->insert($document, $insertOpts);
$collection->insert($document, $insertOpts);
?>
```

Now you'll get the exception I mentioned earlier since you are performing a safe insert and another object with the same ID already exists in the collection.

#4 b>. Insert multiple document in a collection:

```
<?php

// connects to localhost on port 27017 by default
$m = new Mongo();

// database selection for mongo db object

$db = $m->php_test;

$users = array();
for ($i = 0; $i < 10; $i++) {
    $users[] = array('username' => 'user' . $i, 'i' => $i);
}

// table section for database object

$collection = $db->user;

$collection->batchInsert($users);

// close the connection to MongoDB
$m->close();

?>
```

#5>. Find all documents

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->mycol;
echo "Collection selected successfully";

$cursor = $collection->find();
// iterate cursor to display title of documents
foreach ($cursor as $document) {
    echo $document["title"] . "\n";
}
?>
```

#5 b>. Count all records of a collection:

```
<?php

// connects to localhost on port 27017 by default
$m = new MongoClient();

// database selection for mongo db object

$db = $m->php_test;

// table section for database object
$collection = $db->user;

// fetch all product documents
$cursor = $collection->find();

// How many results found
$num_docs = $cursor->count();

echo '<b>Total number of documents in user collection are : </b>' . $num_docs;
```

```
// close the connection to MongoDB
$m->close();
?>
```

#6>. Save method for document:

If you want to modify an existing document object, you can use the save() method. If the passed array has no _id key, the document will be inserted and a primary key will be assigned. If the array already has an _id key then the document object will be updated. Like insert(), save() also accepts an optional second parameter.

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->mycol;
echo "Collection selected successfully";

// now update the document
$collection->save( array('$set'=>array("title"=>"MongoDB Tutorial")));
echo "Document updated successfully";
// now display the updated document
$cursor = $collection->find();
// iterate cursor to display title of documents
echo "Updated document";
foreach ($cursor as $document) {
    echo $document["title"] . "\n";
}
?>
```

NOTE: The save() method is not the only way to update an existing object; you can also use the update() method. **update() takes the criteria of the object you want to update as the first parameter**, and the updated object as the second parameter.

#7>. Update a document

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->mycol;
echo "Collection selected successfully";

// now update the document
$collection->update(array("title"=>"MongoDB"),
array('$set'=>array("title"=>"MongoDB Tutorial")));
echo "Document updated successfully";
// now display the updated document
$cursor = $collection->find();
// iterate cursor to display title of documents
echo "Updated document";
foreach ($cursor as $document) {
    echo $document["title"] . "\n";
}
?>
```

NOTE: if you don't specify \$set modifier then update will replace all values of document with current update operation

Example: In above example if we remove '\$set=>' text then updated record has only on attribute 'title'

#8>. Delete a document

```
<?php
// connect to mongodb
$m = new MongoClient();
echo "Connection to database successfully";
// select a database
$db = $m->mydb;
echo "Database mydb selected";
$collection = $db->mycol;
echo "Collection selected successfully";

// now remove the document
$collection->remove(array("title"=>"MongoDB Tutorial"),false);
echo "Documents deleted successfully";

// now display the available documents
$cursor = $collection->find();
// iterate cursor to display title of documents
echo "Updated document";
foreach ($cursor as $document) {
    echo $document["title"] . "\n";
}
?>
```

NOTE: In the above given example second parameter is boolean type and used for **justOne** field of remove() method.

#9>.Remaining mongodb methods findOne(), save(), limit(), skip(), sort() etc works same as explained in above