Gridfs example in php

Step 1: Download the Composer Installer

Go to the Composer download page:

Open your browser and visit the Composer official website:

https://getcomposer.org/

Download the Windows Installer:

On the Composer homepage, click on the "Getting Started" section and then click "Download Composer". Alternatively, you can directly download the Windows installer by clicking on the link:

Composer-Setup.exe

This will download a file named Composer-Setup.exe to your computer.

Step 2: Run the Composer Installer

Run the installer:

Once the download is complete, double-click the Composer-Setup.exe file to run the installer.

Step 3: Verify Composer Installation

1. **Open a Command Prompt:**

   Press `Win + R`, type `cmd`, and press Enter to open the Command Prompt.

2. **Check Composer version:**

   In the Command Prompt, type the following command to verify that Composer is installed correctly:

   ```
   composer --version
   ```

Step 4: **Add PHP to PATH manually**:

If PHP is not added to the PATH, you can manually add it:

- Open **System Properties → Advanced system settings → Environment Variables**.
- Under **System Variables**, find the `Path` variable and click **Edit**.
- Add the path to your PHP folder (e.g., `C:\php` or `C:\Program Files\php`).
- Click **OK** to save, then restart the Command Prompt and check Composer again with `composer --version`.

Step 5: Using Composer

- **Create a new directory for your project:**

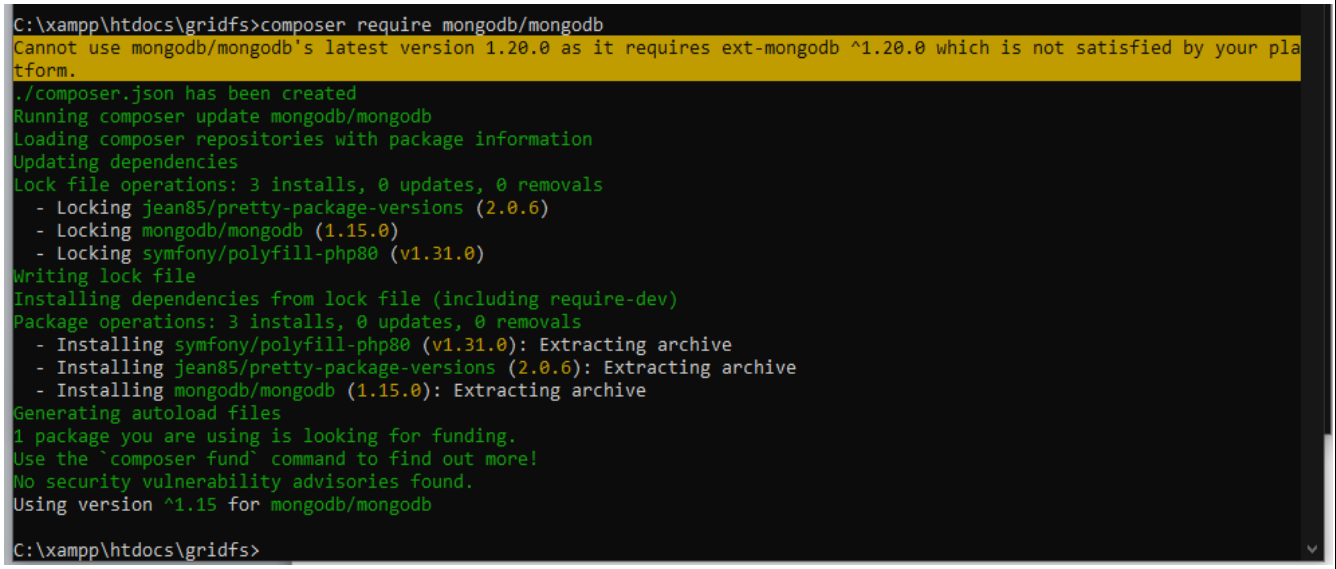
Navigate to the folder where you want your project to reside. For example:

```
mkdir my-php-project
cd my-php-project
```

- **Install a PHP library (e.g., MongoDB PHP driver):**

To install the MongoDB PHP driver via Composer, run:

```
composer require mongodb/mongodb
```



Step 6: create a .php file in htdocs/gridfs directory

Gridfs_example.php

<?php

require 'vendor/autoload.php'; // Ensure you are using Composer's autoloader

// Create a MongoDB client

$client = new MongoDB\Client("mongodb://localhost:27017");

// Select a database and GridFS bucket

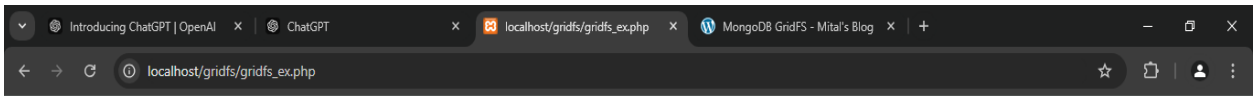
$database = $client->selectDatabase('test');

$bucket = $database->selectGridFSBucket();

```
// Example: Upload a file to GridFS
function uploadFile($filePath) {
    global $bucket;

    // Open the file to upload
    $file = fopen($filePath, 'rb'); // Open file in binary read mode

    // Upload the file to GridFS
    $fileId = $bucket->uploadFromStream(basename($filePath), $file);

    // Close the file
    fclose($file);

    return $fileId;
}

// Example: Download a file from GridFS
function downloadFile($fileId, $outputPath) {
    global $bucket;

    // Open a file for writing the downloaded content
    $file = fopen($outputPath, 'wb'); // Open file in binary write mode

    // Download the file from GridFS using its file ID
    $bucket->downloadToStream($fileId, $file);

    // Close the file
```

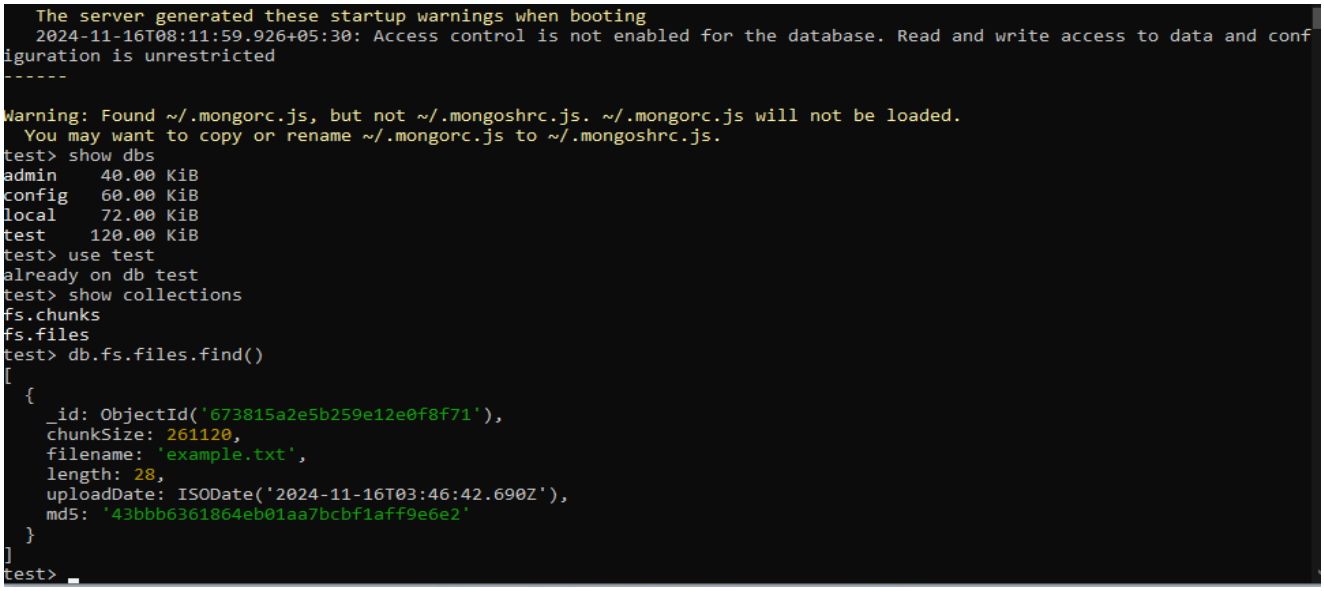
```
    fclose($file);
}

// Example usage

// Upload a file (e.g., 'example.txt')
$fileId = uploadFile('example.txt');
echo "File uploaded with ID: $fileId\n";

// Now download the file to a different location
downloadFile($fileId, 'downloaded_example.txt');
echo "File downloaded to 'downloaded_example.txt'.\n";
?>
```

Step 7: run file in browser



File uploaded with ID: 673815a2e5b259e12e0f8f71 File downloaded to 'downloaded_example.txt'.

Step 7: check file in mongodb



Gridfs Example manually add file

How to Add files in gridFS

step -1 download database tools in mongodb official website

https://www.mongodb.com/try/download/database-tools

Tool name:MongoDB Command Line Database Tools Download

step -2 extract files and copy to all files in bin folder.

step -3 use this command

C:\MongoDb\bin>mongofiles -d mydatabase put unit4.pdf

step -4 connect mongodb server

test> use mydatabase

mydatabase> db.fs.files.find()

[
{
_id: ObjectId(‘672c42cf8f5961ed7650ca6a’),
length: Long(‘427620’),
chunkSize: 261120,
uploadDate: ISODate(‘2024-11-07T04:32:15.120Z’),

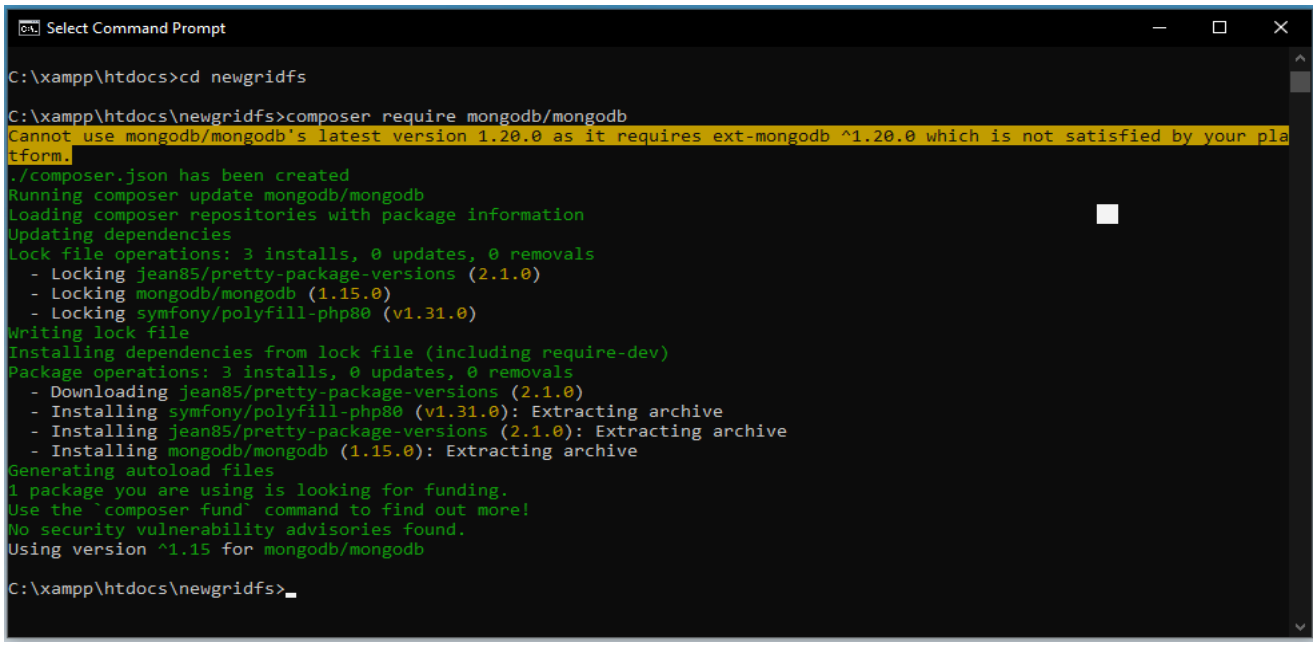
filename: ‘unit4.pdf’,
metadata: {}
}
]

db.fs.chunks.find({files_id:ObjectId(‘672c42cf8f5961ed7650ca6a’)})

Gridfs Example with html form

1. Install MongoDB PHP Driver

composer require mongodb/mongodb



2. **Example of GridFS in MongoDB with PHP**
    1. HTML Form for File Upload (`index.html`)

    index.php

    <!DOCTYPE html>
    <html lang="en">
    <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>File Upload to MongoDB GridFS</title>
    </head> <body> <h1>Upload File to MongoDB</h1>
    <form action="upload.php" method="post" enctype="multipart/form-data">
    <label for="file">Select file:</label>
    <input type="file" name="file" id="file" required>

 <button type="submit" name="submit">Upload File</button> </form>
    </body> </html>
    2. **PHP Script to Handle File Upload (`upload.php`)**

    <?php

```
require 'vendor/autoload.php'; // Composer autoloader

// Connect to MongoDB
$client = new MongoDB\Client("mongodb://localhost:27017");
$db = $client->test; // Replace with your database name
$gridFS = $db->selectGridFSBucket();

// Check if a file has been uploaded
if (isset($_FILES['file']) && $_FILES['file']['error'] == 0) {
   // Get the uploaded file details
   $fileTmpPath = $_FILES['file']['tmp_name'];
   $fileName = $_FILES['file']['name'];

   // Open the uploaded file
   $stream = fopen($fileTmpPath, 'rb');

   // Upload file to GridFS
   try {
      // Upload the file and store it in GridFS with the original file name
      $gridFS->uploadFromStream($fileName, $stream);
      fclose($stream);
      echo "File uploaded successfully!";
   } catch (Exception $e) {
      echo "Error uploading file: " . $e->getMessage();
   }
} else {
   echo "No file uploaded or file upload error.";
}
?>
```

Key Elements and Explanation:

1. **Composer Autoloader**:

   ```
   require 'vendor/autoload.php';
   ```

   - Ensures the MongoDB PHP library (installed via Composer) is loaded. Make sure you have run `composer require mongodb/mongodb` in your project to include this library.

   ```
   $client = new MongoDB\Client("mongodb://localhost:27017");
   $db = $client->test; // Replace with your database name
   $gridFS = $db->selectGridFSBucket();
   ```

   - Establishes a connection to the MongoDB server.
   - `selectGridFSBucket()` is used to interact with MongoDB's GridFS for file storage.
2. **File Upload Handling**:

   ```
   if (isset($_FILES['file']) && $_FILES['file']['error'] == 0) {
       $fileTmpPath = $_FILES['file']['tmp_name'];
   ```

```
$fileName = $_FILES['file']['name'];
```

- Checks if a file is uploaded and has no errors.
- Retrieves the temporary file path (`tmp_name`) and the original file name (`name`).

3. **Uploading the File to GridFS**:

```
$stream = fopen($fileTmpPath, 'rb');
$gridFS->uploadFromStream($fileName, $stream);
fclose($stream);
```

- Opens the uploaded file in binary read mode (`rb`).
- Uploads the file to GridFS with its original name using `uploadFromStream()`.
- Closes the file stream after uploading.

4. **Error Handling**:

```
} catch (Exception $e) {
    echo "Error uploading file: " . $e->getMessage();
```

- Captures and displays any exceptions that occur during the upload.

5. **Fallback for Missing/Invalid File**:

```
echo "No file uploaded or file upload error.";
```

- Notifies the user if no file was uploaded or if there was an error.

3. PHP Script to List Uploaded Files (`list_files.php`)

```
<?php
require 'vendor/autoload.php'; // Ensure you're loading the MongoDB library

$client = new MongoDB\Client("mongodb://localhost:27017");
$database = $client->selectDatabase('test');
$collection = $database->selectCollection('fs.files');

$files = $collection->find();

foreach ($files as $document) {

// Access the filename field

   $filename = $document['filename'];
   echo "Filename: " . htmlspecialchars($filename) . "<br>";
}
?>
```