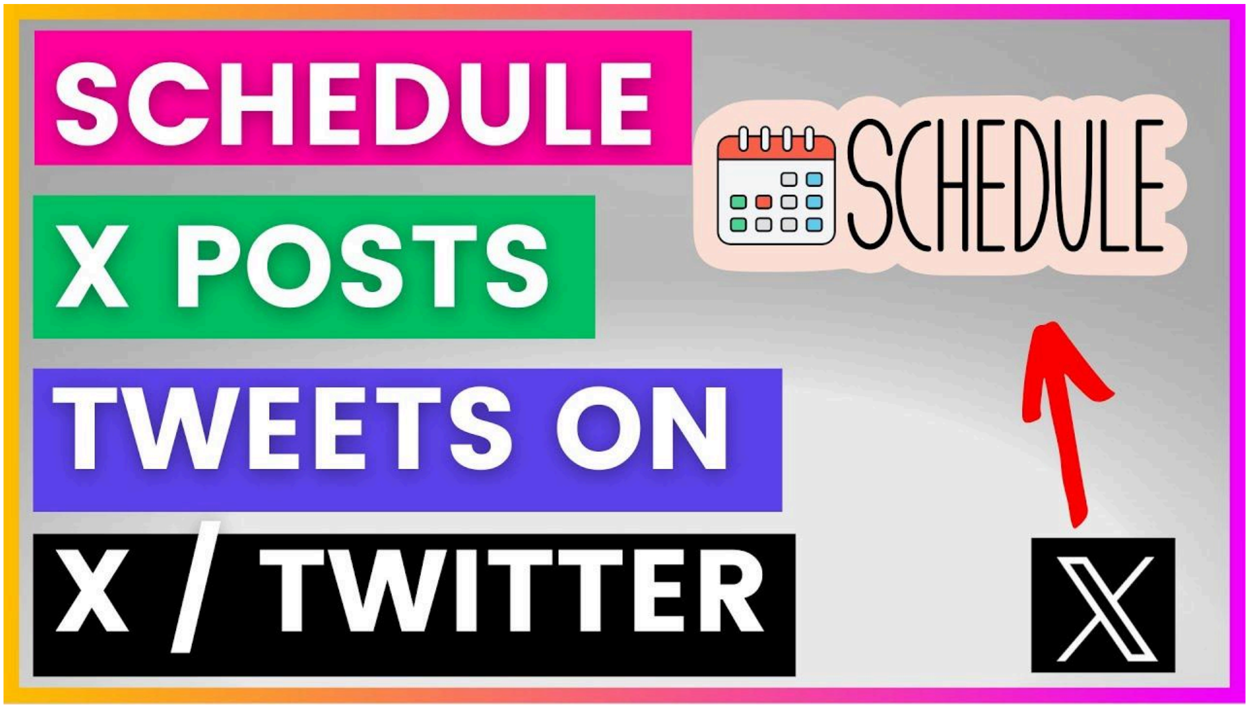

BY - Ambrish

Download Code - https://github.com/ambristech

🎯 Want to see how it's done?

👉 Download the **X Scheduler code** from my GitHub: github.com/ambristech/x-scheduler

👉 Explore my GitHub for more projects: github.com/ambristech

👉 Connect with me if you're into AI, cybersecurity, web dev, or digital marketing collaborations!

Create Twitter (X) Post Scheduler on XAMPP Using PHP

This tutorial guides you through building an **X Post Scheduler** using **PHP** on **XAMPP**. The app allows scheduling posts for X (Twitter) and automatically publishing them using the X API. We'll use PHP, MySQL, and the `abraham/twitteroauth` library, with an HTML form for scheduling. Screenshots are recommended at key steps.

Prerequisites

- **XAMPP**: Installed with Apache and MySQL (https://www.apachefriends.org/)
- **Composer**: Installed globally (https://getcomposer.org/)
- **X Developer Account**: Free tier with OAuth 1.0a credentials
- **Text Editor**: VS Code or Notepad++
- **Browser**: Chrome or Firefox

Step 1: Set Up XAMPP Environment

1. **Download and Install XAMPP**:
    - Go to: https://www.apachefriends.org/
    - Install to `C:\xampp`
2. **Start XAMPP**:
    - Open `C:\xampp\xampp-control.exe`
    - Start **Apache** and **MySQL**
    - **Screenshot**: XAMPP Control Panel with Apache and MySQL running
3. **Test Localhost**:
    - Open `http://localhost`

Check: Apache and MySQL running, dashboard visible

Step 2: Create Project Folder

1. **Create Folder**:
    - Path: `C:\xampp\htdocs\x-scheduler`
2. **Verify Access**:
    - Open `http://localhost/x-scheduler`

Check: Folder created, accessible

Step 3: Set Up X Developer Portal

1. **Sign Up**:
   - Go to: https://developer.x.com/
   - Apply for free developer account
2. **Create Project and App**:
   - Project: `XSchedulerApp`
   - App: `XScheduler`
3. **Get Credentials**:
   - Enable “Read and write” permissions
   - Save API Key, API Secret, Access Token, Access Token Secret
   - **Screenshot**: Developer Portal “Keys and tokens” (blur credentials)
4. **Save Credentials**:
   - Store securely

Check: App created, credentials saved

Step 4: Install Composer Dependencies

Navigate:

```
cd C:\xampp\htdocs\x-scheduler
```

1.

Initialize Composer:

```
composer init
```

2.
   - Package: `yourname/x-scheduler`
   - Description: “X Post Scheduler”
   - License: `MIT`

Here are the answers to each question prompted after composer init:

1. **Package name (<vendor>/<name>) [name/x-scheduler]:**
   name/x-scheduler (press Enter to accept the default).
2. **Description []:**
   X post scheduler
3. **Author [author@gmail.com, n to skip]:**
   n (to skip).

4. **`Minimum Stability []:`**
   `stable`
5. **`Package Type (e.g. library, project, metapackage, composer-plugin) []:`**
   `project`
6. **`License []:`**
   `MIT`
7. **`Would you like to define your dependencies (require) interactively [yes]?`**
   `no (or specify packages if needed, e.g., laravel/framework).`
8. **`Search for a package:`**
   `(Skip if you answered no above; otherwise, enter package names or press Enter to finish).`
9. **`Would you like to define your dev dependencies (require-dev) interactively [yes]?`**
   `no`
10. **`Add PSR-4 autoload mapping? Maps namespace "name\XScheduler" to the entered relative path. [src/, n to skip]:`**
    `src/ (or n to skip).`

Add Dependencies:

```
 composer require abraham/twitteroauth
composer require vlucas/phpdotenv
```

3.
4. **Verify**:
   - Check `composer.json`, `composer.lock`, `vendor/`

Check: Dependencies installed

Step 5: Create .env for Credentials

1. **Create .env**:

File: `C:\xampp\htdocs\x-scheduler\.env`

```
 CONSUMER_KEY=your_api_key
```

```
CONSUMER_SECRET=your_api_secret
ACCESS_TOKEN=your_access_token
ACCESS_TOKEN_SECRET=your_access_token_secret
```

Check: `.env` created

Step 6: Create HTML Form for Scheduling

1. **Create index.html**:

File: `C:\xampp\htdocs\x-scheduler\index.html`

```
 <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>X Post Scheduler</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 20px; }
    .post { margin-bottom: 20px; }
    label { display: block; margin: 5px 0; }
    input, button { padding: 5px; margin: 5px 0; }
  </style>
</head>
<body>
  <h1>X Post Scheduler</h1>
  <form action="schedule.php" method="POST">
    <div class="post">
      <label>Post Text (280 chars max):</label>
      <input type="text" name="posts[0][text]" maxlength="280" required>
      <label>Schedule Time (YYYY-MM-DD HH:MM:SS):</label>
      <input type="datetime-local" name="posts[0][schedule_time]" required>
    </div>
    <button type="submit">Schedule Post</button>
  </form>
</body>
</html>
```

   -

2. **Test**:
   - Open `http://localhost/x-scheduler/index.html`

Check: Form displays

Step 7: Create Database and Table

1. **Access phpMyAdmin**:
   - Open `http://localhost/phpmyadmin`
2. **Create Database**:
   - Name: `x_scheduler`
3. **Create Table**:

SQL:

```
 CREATE TABLE scheduled_posts (
   id INT AUTO_INCREMENT PRIMARY KEY,
   text VARCHAR(280) NOT NULL,
   schedule_time DATETIME NOT NULL,
   status ENUM('pending', 'posted', 'failed') DEFAULT 'pending',
   retry_count INT DEFAULT 0
);
```

Check: Database and table created

Step 8: Create PHP Script for Scheduling and Posting

1. **Create schedule.php**:

File: `C:\xampp\htdocs\x-scheduler\schedule.php`

```
 <?php
use Abraham\TwitterOAuth\TwitterOAuth;
require 'vendor/autoload.php';
use Dotenv\Dotenv;

$dotenv = Dotenv::createImmutable(__DIR__);
$dotenv->load();

$consumerKey = $_ENV['CONSUMER_KEY'];
$consumerSecret = $_ENV['CONSUMER_SECRET'];
$accessToken = $_ENV['ACCESS_TOKEN'];
$accessTokenSecret = $_ENV['ACCESS_TOKEN_SECRET'];
```

```
$connection = new TwitterOAuth($consumerKey, $consumerSecret, $accessToken,
$accessTokenSecret);
$connection->setApiVersion('1.1');

$db = new PDO('mysql:host=localhost;dbname=x_scheduler', 'root', '');

function postScheduledTweets() {
    global $connection, $db;
    $stmt = $db->query("SELECT * FROM scheduled_posts WHERE schedule_time <= NOW()
AND status = 'pending'");
    $posts = $stmt->fetchAll(PDO::FETCH_ASSOC);
    foreach ($posts as $post) {
        try {
            $result = $connection->post('tweets', ['text' => $post['text']]);
            if (isset($result->id_str)) {
                $db->prepare("UPDATE scheduled_posts SET status = 'posted' WHERE id =
?")->execute([$post['id']]);
                echo "Posted: {$post['text']}<br>";
            } else {
                $retryCount = $post['retry_count'] ?? 0;
                if ($retryCount < 3) {
                    $db->prepare("UPDATE scheduled_posts SET retry_count = retry_count + 1
WHERE id = ?")->execute([$post['id']]);
                    echo "Retrying: {$post['text']}<br>";
                } else {
                    $db->prepare("UPDATE scheduled_posts SET status = 'failed' WHERE id =
?")->execute([$post['id']]);
                    echo "Failed to post: {$post['text']}<br>";
                }
                error_log("HTTP Code: " . $connection->getLastHttpCode());
                error_log("Response: " . print_r($connection->getLastBody(), true));
            }
        } catch (Exception $e) {
            $db->prepare("UPDATE scheduled_posts SET status = 'failed' WHERE id =
?")->execute([$post['id']]);
            echo "Error posting: {$e->getMessage()}<br>";
            error_log("Exception: " . $e->getMessage());
        }
    }
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $posts = $_POST['posts'];
    foreach ($posts as $post) {
        $stmt = $db->prepare("INSERT INTO scheduled_posts (text, schedule_time) VALUES (?,
?)");
        $stmt->execute([$post['text'], $post['schedule_time']]);
    }
    echo "Posts scheduled successfully!";
} else {
    postScheduledTweets();
}
?>
```

○

2. **Create cron.php**:

File: `C:\xampp\htdocs\x-scheduler\cron.php`

```
 <?php
require 'schedule.php';
postScheduledTweets();
?>
```

○

3. **Verify**:

Check: Files created

Step 9: Test the App Locally

1. **Schedule Post**:
    - Open `http://localhost/x-scheduler/index.html`
    - Text: “My first scheduled post! #XAPI”
    - Time: 1-2 min ahead
    - **Screenshot**: Browser with post confirmation
2. **Check Database**:
    - `http://localhost/phpmyadmin`
    - Verify `scheduled_posts`
    - **Screenshot**: phpMyAdmin with new entry
3. **Run cron.php**:
    - Open `http://localhost/x-scheduler/cron.php`
4. **Check X**:
    - Verify post (@yourusername)
    - **Screenshot**: X profile with tweet
5. **Check Database**:

   - `status`: `posted`
6. **Check Errors**:
   - Check `error_log`

Check: Post scheduled, published, database updated

Step 10: Simulate Cron Job Locally

1. **Schedule Post**:
   - Text: “Auto post from XAMPP! #XAPI”
2. **Run cron.php**:
   - Open `http://localhost/x-scheduler/cron.php` every minute
   - **Screenshot**: Browser with `cron.php` output
3. **Check**:
   - Post on X, `status`: `posted`

Check: Cron simulation works

Step 11: Secure and Optimize

1. **Protect .env**:

`.htaccess`:

```
 <FilesMatch "\.(env|lock|json)$">
   Order Allow,Deny
   Deny from all
</FilesMatch>
```

   -
2. **Error Logging**:
   - Included in `schedule.php`
3. **Rate Limits**:
   - Monitor 1500 posts/month
4. **Retry Logic**:
   - Included in `schedule.php`
   - **Screenshot**: Code editor with retry logic

Check: Security implemented

Step 12: Next Steps

1. **Choose Hosting**:
    - Any Hosting With (PHP 8.1+)
2. **Deploy**:
    - Upload to `public_html`
    - Set up database, cron job
3. **Test Live**

Summary

1. XAMPP Control Panel
2. XAMPP dashboard
3. Empty `http://localhost/x-scheduler`
4. Developer Portal credentials
5. File Explorer with `vendor/`
6. `.env` file
7. `index.html` form
8. phpMyAdmin `scheduled_posts`
9. Project files
10. Post confirmation
11. `scheduled_posts` entry
12. X profile tweet
13. `cron.php` output
14. `schedule.php` retry logic

Troubleshooting

- **Composer**: `composer clear-cache`, `composer update`
- **API**: Check credentials, permissions
- **Database**: Verify `x_scheduler`, `root`
- **Post Failure**: Check `error_log` (401, 403, 429)

IF you face any problem you can contact me on my Social media Accounts 🙂

Download Code - https://github.com/ambristech