



This guide provides the necessary steps to implement the YouTube to MP4 conversion function directly into your WordPress theme.

1. The Conversion Function (manus-youtube-converter.php)

The provided PHP code implements a shortcode that displays a form. When a user submits a YouTube URL, the function calls the **Cobalt API** (a free, open-source API for media conversion) to process the video and returns a direct download link for the MP4 file.

Key Features:

- **API Used:** Cobalt API (using a public instance: <https://co.wuk.sh/>).
- **Method:** Uses WordPress's built-in `wp_remote_post` function for secure HTTP requests.
- **Output:** The shortcode renders a simple form and, upon successful submission, redirects the user to the MP4 download link.

2. Implementation Steps

You have two primary options for adding this code to your WordPress site:

Option A: Adding to the Theme's `functions.php` (Recommended by User)

This is the fastest way to implement the function, as requested.

1. **Access Your Theme Files:** Connect to your WordPress site via **FTP** or use your hosting provider's **File Manager**.
2. **Navigate to Theme Directory:** Go to `wp-content/themes/your-theme-name/`.
3. **Edit `functions.php`:** Open the `functions.php` file for editing.
4. **Add the Code:** Paste the entire content of the provided `manus-youtube-converter.php` file at the **very end** of your `functions.php` file, just before the closing `?>` tag (if one exists).



Option B: Creating a Simple Plugin (Recommended for Best Practice)

Creating a small plugin is generally safer and ensures the function remains active even if you change your theme.

1. **Create a New Directory:** In `wp-content/plugins/`, create a new folder named `manus-youtube-converter`.
2. **Create the Plugin File:** Inside the new folder, create a file named `manus-youtube-converter.php`.
3. **Add Plugin Header:** Paste the following header at the top of the file: `<?php
/** * Plugin Name: Manus YouTube to MP4 Converter * Description:
Adds a shortcode to convert YouTube videos to MP4 using the Cobalt
API. * Version: 1.0 * Author: Manus AI */ // [PASTE THE REST OF
THE CODE HERE]`
4. **Add the Code:** Paste the entire content of the provided `manus-youtube-converter.php` file immediately after the plugin header.
5. **Activate Plugin:** Log in to your WordPress admin dashboard, go to **Plugins**, and activate the **Manus YouTube to MP4 Converter** plugin.

3. Usage

Once the code is implemented, you can use the shortcode on any page or post:

Action	Shortcode
Insert Converter Form	<code>[youtube_to_mp4_converter]</code>
1. Create a New Page/Post: Go to your WordPress editor. 2. Add Shortcode Block: Insert a Shortcode block (or a Custom HTML block) into the content. 3. Paste the Shortcode: Enter <code>[youtube_to_mp4_converter]</code>. 4. Publish: Save and publish the page. The form will now be visible on your site.	

4. Important Considerations



Aspect	Detail
API Stability	The Cobalt API is a community-driven, open-source project. While generally reliable, public instances can be unstable, rate-limited, or shut down without notice. For high-volume or mission-critical use, consider hosting your own Cobalt instance or using a paid, commercial API [1].
Legal/ToS	Downloading content from YouTube may violate their Terms of Service. Ensure your use case complies with all applicable laws and YouTube's policies.
Server Requirements	The code uses <code>wp_remote_post</code> , which requires the cURL library to be enabled on your PHP installation (standard on most hosts).

References

[1] [Cobalt API Documentation: Methods](https://github.com/imputnet/cobalt/blob/main/docs/api.md), acceptable values, headers, responses and everything else related to making and parsing requests from a cobalt api instance. [https://github.com/imputnet/cobalt/blob/main/docs/api.md]

```
<?php
/**
 * YouTube to MP4 Converter Functionality
 *
 * This file contains the necessary PHP functions to create a
shortcode
 * for converting YouTube videos to MP4 format using the Cobalt API.
 *
 * Usage: [youtube_to_mp4_converter]
 *
 * IMPORTANT: This code should be added to your theme's functions.php
file
 * or included as a custom plugin.
 */

// 1. Core function to call the Cobalt API
function manus_cobalt_convert_video($youtube_url) {
    // The Cobalt API endpoint (using a known public instance)
    // NOTE: Public instances can be unstable or rate-limited. For
```



WordPress YouTube to MP4 Converter Implementation Guide

```
production,
    // it is recommended to host your own instance or use a paid API.
    $api_url = 'https://co.wuk.sh/';

    // Prepare the request body
    $body = array(
        'url' => $youtube_url,
        'videoQuality' => 'max', // Request the highest quality
available
        'youtubeVideoContainer' => 'mp4' // Explicitly request MP4
container
    );

    // Prepare the arguments for wp_remote_post
    $args = array(
        'body'      => json_encode($body),
        'headers' => array(
            'Content-Type' => 'application/json',
            'Accept'       => 'application/json',
        ),
        'timeout' => 30, // Set a reasonable timeout for the
conversion process
        'sslverify' => false // Set to true in a production
environment with proper SSL
    );

    // Make the POST request using WordPress's HTTP API
    if (!function_exists('wp_remote_post')) {
        // Fallback for environments where WordPress functions are not
loaded (should not happen in functions.php)
        return array('error' => 'WordPress HTTP functions not
available.');
```

```
    }
    $response = wp_remote_post($api_url, $args);

    // Check for WordPress HTTP errors
    if (is_wp_error($response)) {
        return array('error' => 'API Request Error: ' .
```



WordPress YouTube to MP4 Converter Implementation Guide

```
$response->get_error_message());
}

$response_code = wp_remote_retrieve_response_code($response);
$response_body = wp_remote_retrieve_body($response);
$data = json_decode($response_body, true);

// Check for API errors or unexpected response
if ($response_code !== 200 || !isset($data['status'])) {
    $error_message = isset($data['error']) ? $data['error'] :
'Unknown API error.';
    return array('error' => "Conversion failed. Status Code:
{$response_code}. Message: {$error_message}");
}

// Handle successful response
if ($data['status'] === 'tunnel' || $data['status'] ===
'redirect') {
    return array(
        'success' => true,
        'download_url' => $data['url'],
        'filename' => $data['filename']
    );
} elseif ($data['status'] === 'error') {
    $error_code = isset($data['error']['code']) ?
$data['error']['code'] : 'Unknown';
    return array('error' => "Cobalt API Error: {$error_code}.
Please check the URL and try again.");
} else {
    // Handle other statuses like 'picker' or 'local-processing'
    if necessary,
    // but for a simple downloader, we'll treat them as an error
    for now.
    return array('error' => "Conversion status '{$data['status']}'
not supported for direct download.");
}
}
```



WordPress YouTube to MP4 Converter Implementation Guide

```
// 2. Shortcode function to display the form and handle submission
function manus_youtube_downloader_shortcode() {
    // Check if the form has been submitted
    if (isset($_POST['manus_convert_submit']) &&
isset($_POST['youtube_url'])) {
        // Sanitize the input URL
        $youtube_url = esc_url_raw(trim($_POST['youtube_url']));

        if (!empty($youtube_url)) {
            // Call the conversion function
            $result = manus_cobalt_convert_video($youtube_url);

            if (isset($result['success']) && $result['success']) {
                // Redirect the user to the download URL
                // This is the most reliable way to initiate a
download from an external service.
                wp_redirect($result['download_url']);
                exit;
            } else {
                // Display error message
                $error_message = $result['error'];
                $output = '<div style="color: red; border: 1px solid
red; padding: 10px; margin-bottom: 15px;">Error: ' .
esc_html($error_message) . '</div>';
            }
        } else {
            $output = '<div style="color: orange; border: 1px solid
orange; padding: 10px; margin-bottom: 15px;">Please enter a valid
YouTube URL.</div>';
        }
    }

    // Output the form
    $output .= '
<div class="manus-youtube-converter-wrapper" style="max-width:
600px; margin: 20px auto; padding: 20px; border: 1px solid #ccc;
border-radius: 5px;">
    <h3 style="margin-top: 0;">YouTube to MP4 Converter</h3>
```



WordPress YouTube to MP4 Converter Implementation Guide

```
        <form method="post" action="' .  
esc_url($_SERVER['REQUEST_URI']) . '">  
            <p>  
                <label for="youtube_url" style="display: block;  
margin-bottom: 5px; font-weight: bold;">Enter YouTube Video  
URL:</label>  
                <input type="url" id="youtube_url"  
name="youtube_url" placeholder="e.g.,  
https://www.youtube.com/watch?v=dQw4w9WgXcQ" required  
                style="width: 100%; padding: 10px; box-  
sizing: border-box; border: 1px solid #ddd; border-radius: 3px;">  
            </p>  
            <p>  
                <input type="submit" name="manus_convert_submit"  
value="Convert & Download MP4"  
                style="background-color: #ff0000; color:  
white; padding: 10px 15px; border: none; border-radius: 3px; cursor:  
pointer; font-weight: bold;">  
            </p>  
            <p style="font-size: 0.8em; color: #666;">  
                * This service uses a third-party API (Cobalt) for  
conversion.  
            </p>  
        </form>  
    </div>  
    ';  
  
    return $output;  
}  
  
// 3. Register the shortcode  
add_shortcode('youtube_to_mp4_converter',  
'manus_youtube_downloader_shortcode');  
  
// Optional: Ensure wp_remote_post is available if this file is loaded  
early  
if (!function_exists('wp_remote_post')) {  
    require_once(ABSPATH . 'wp-includes/pluggable.php');
```



WordPress YouTube to MP4 Converter Implementation Guide

```
require_once(ABSPATH . 'wp-includes/http.php');  
}
```