



Portus

Advanced File Hosting Script

Documentation

Portus



CONTENTS

1. Server Requirements.....	3
1.1 Additional Requirements.....	3
2. Installation.....	4
2.1 With Installer.....	4
2.2 Manual Installation.....	7
3. Plugins & Libraries.....	9
4. API Keys.....	11
4.1 Stripe.....	11
4.2 Google reCaptcha.....	11
4.3 Google Auth.....	13
4.4 Amazon S3.....	16
4.5 Cloudflare R2.....	19
4.6 Wasabi.....	21
4.7 FTP.....	25
4.8 Google Drive.....	26
4.9 Razorpay.....	33
5. Upload Settings.....	34
5.1 Uploadable Types.....	34
5.2 File Size Limit.....	34
5.3 Server Quota.....	34
5.4 Server Storage.....	35
6. Configuring Server Cron Job.....	35
7. Language.....	35
8. Affiliate.....	36
9. Security.....	36
10. Support.....	37

1. Server Requirements

Portus was coded with **Laravel 11**. The server requirements required for Laravel 11 are also required for Portus.

- PHP >= 8.2
- ctype PHP Extension
- cURL PHP Extension
- DOM PHP Extension
- Fileinfo PHP Extension
- Filter PHP Extension
- Hash PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PCRE PHP Extension
- PDO PHP Extension
- Session PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

1.1 Additional Requirements

Notice: Since high-size files take longer to upload, you need to make some adjustments on your server. If you do not make these adjustments, you may receive a 504 error when uploading a file.

Recommended php.ini values;

```
max_execution_time = 600  
max_input_time = 600
```

Recommended nginx.conf (If you are using nginx) values;

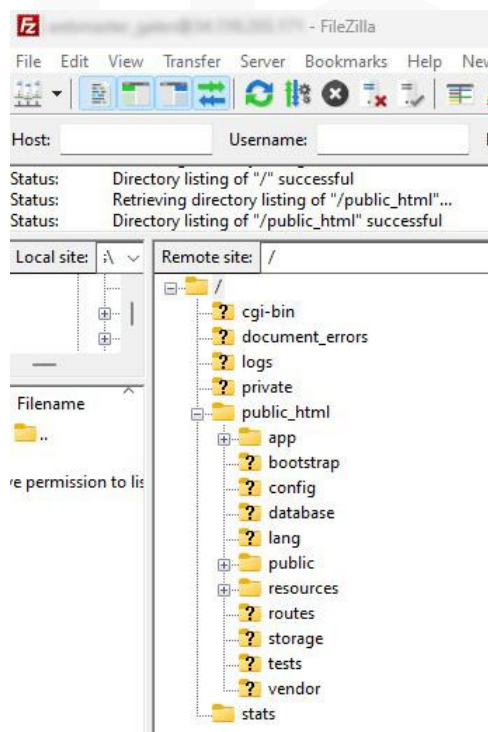
```
send_timeout 600s;  
proxy_connect_timeout 600s;  
proxy_read_timeout 600s;  
proxy_send_timeout 600s;
```

2. Installation

2.1 With Installer

You can easily install Portus with the installer. Installation can be done quickly and safely with the installer.

There is a **files.zip** in the installation file you downloaded. Upload all the files in the **files.zip file to the public_html** directory on your server. After uploading, the files should be like this;



After uploading the files, go to;

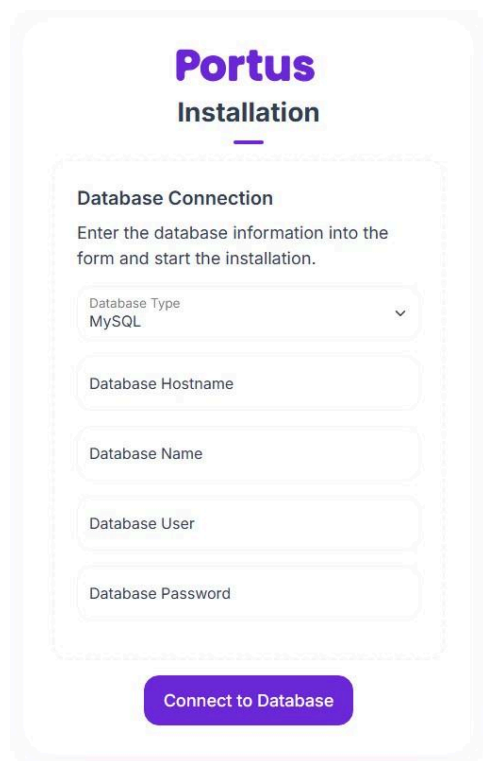
yourwebsiteaddress.com/install

If you installed it on localhost, go to this address;

localhost/install

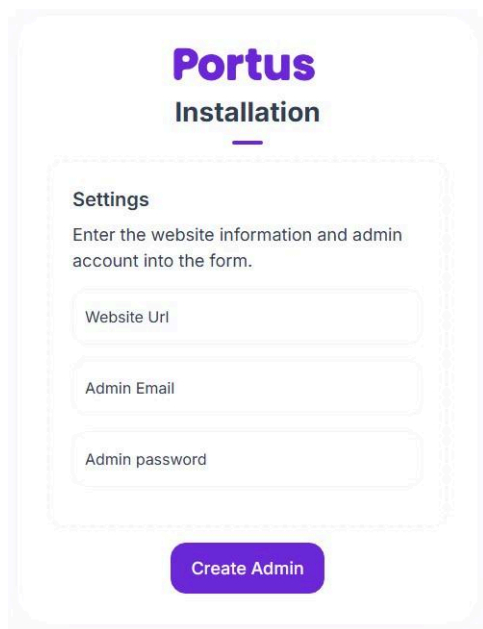
Notice: If you are going to install in a **subfolder** instead of **public_html**; You must add **public** to the end of the site address in the URL. For example; if you will install to the **test folder**, the address of the install page will be **localhost/test/public/install** or **yourwebsite.com/test/public/install**.

Start the installation by pressing “**Start Installation**”.

The image shows a screenshot of the 'Portus Installation' web interface. At the top, the title 'Portus Installation' is displayed in purple. Below it, a section titled 'Database Connection' contains the instruction: 'Enter the database information into the form and start the installation.' The form includes a dropdown menu for 'Database Type' (currently set to 'MySQL'), and five text input fields for 'Database Hostname', 'Database Name', 'Database User', and 'Database Password'. A purple button labeled 'Connect to Database' is positioned at the bottom of the form. A large, faint 'Portus' watermark is visible in the background of the entire page.

Enter your database information and establish a database connection.

If the connection is established successfully, click the “**Continue**” button.



Portus
Installation

Settings
Enter the website information and admin account into the form.

Website Url

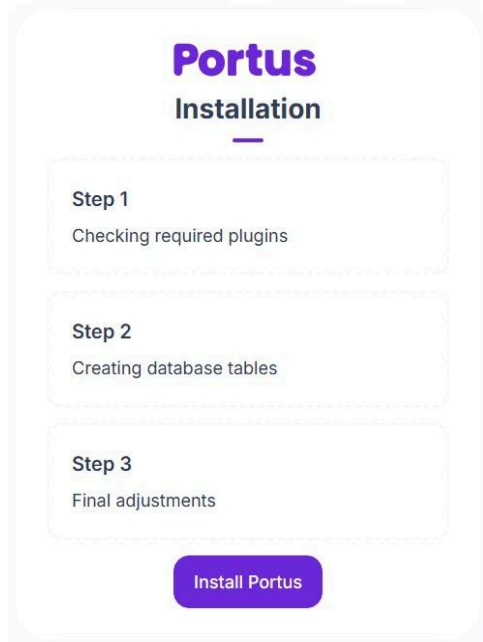
Admin Email

Admin password

Create Admin

In the next step, the website url, admin email and password must be entered. The **admin email and password are important**, you will log in to the website with this information.

After filling out the form, click on the “Continue” button.



Portus
Installation

Step 1
Checking required plugins

Step 2
Creating database tables

Step 3
Final adjustments

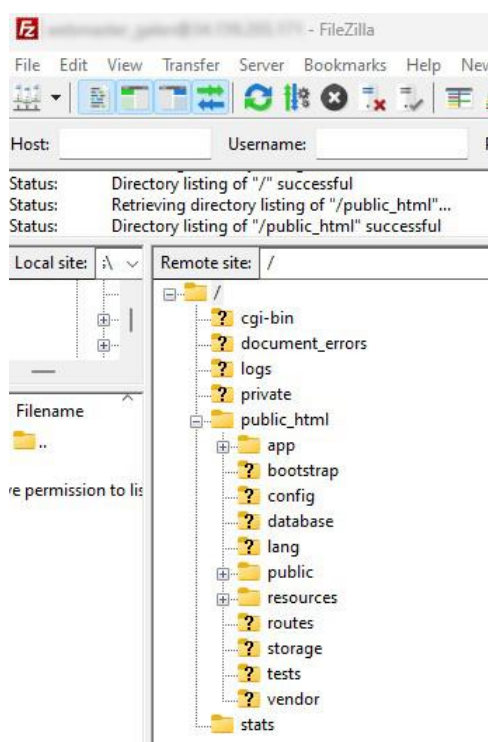
Install Portus

Finally, complete the installation by clicking the **“Install Portus”** button. It may take up to 1 minute for the installation to finish.

2.2 Manual Installation

You can also install Portus manually.

There is a **files.zip** in the installation file you downloaded. Upload all the files in the **files.zip** file to the **public_html** directory on your server. After uploading, the files should be like this;



Find **database.sql** in the installation file you downloaded. Import this file into the database.

Finally, it is necessary to make a few changes in the env file. Open the file named **.env** in the **public_html** folder in the files you transfer to your server.

Arrange the places below according to your needs;

```
APP_URL=YourWebsiteUrl
```

```
DB_CONNECTION=DatabaseType  
DB_HOST=DatabaseHost  
DB_PORT=DatabasePort  
DB_DATABASE=DatabaseName  
DB_USERNAME=DatabaseUser  
DB_PASSWORD=DatabasePassword
```

Write **%installed%** as the INSTALLED value.

```
INSTALLED=%installed%
```

Finally you need to change the APP_KEY value. You can do this with the **php artisan key:generate** command. Or you can generate a new key by clicking [here](#).

Default admin login for manual installation;

Email: admin@portus.com

Password: 123123123

3. Plugins & Libraries

Portus; made with Laravel 11, Bootstrap 5 and Vanilla JS. The following libraries were also used.

Name	Source	License
Animate.css	View	View
ApexCharts	View	View
Bootstrap 5	View	View
Ckeditor 5	View	View
Counter-Up 2	View	View
Dropzone	View	View
Flag Icons	View	View
Font Awesome	View	View
Gridjs	View	View
Vanilla Lazyload	View	View
OverlayScrollbars	View	View
Smooth Scroll	View	View
Toastify	View	View
WOW	View	View

Name	Source	License
jenssegers/agent	View	View
league/flysystem-aws-s3-v3	View	View
league/flysystem-ftp	View	View
whitecube/laravel-cookie-consent	View	View
mcamara/laravel-localization	View	View
stripe/stripe-php	View	View
josiasmontag/laravel-recaptchav3	View	View
yaza-putu/laravel-google-drive-storage	View	View
stechstudio/laravel-zipstream	View	View

Portus

Label ⓘ
e.g. example.com 0 / 50

reCAPTCHA type ⓘ

☒ reCAPTCHA v3 Verify requests with a score

☐ reCAPTCHA v2 Verify requests with a challenge

Domains ⓘ

+ Add a domain, e.g. example.com

Owners

(You)

+ Enter email addresses

☐ **Accept the reCAPTCHA Terms of Service**

You agree to explicitly inform visitors to your site that you have implemented reCAPTCHA v3 on your site and that their use of reCAPTCHA v3 is subject to the Google [Privacy Policy](#) and [Terms of Use](#). reCAPTCHA may only be used to fight spam and abuse on your site. reCAPTCHA must not be used for any other purposes such as determining credit worthiness, employment eligibility, financial status, or insurability of a user.

By accessing or using the reCAPTCHA APIs, you agree to the Google APIs [Terms of Use](#), Google [Terms of Use](#), and to the Additional Terms below. Please read and understand all applicable terms and policies before accessing the APIs.

reCAPTCHA Terms of Service ▾

☒ **Send alerts to owners** ⓘ

CANCEL **SUBMIT**

Copy **“SITE KEY”** and **“SECRET KEY”**. Open the **“Settings > Google reCAPTCHA”** page on your website. Paste **“SITE KEY”** to **“Google reCAPTCHA API”** and **“SECRET KEY”** to **“Google reCAPTCHA Secret API”**.

Use this site key in the HTML code your site serves to users. [See client side integration](#)

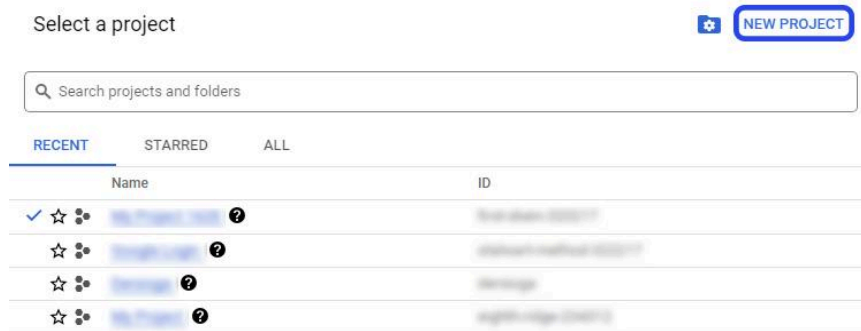
COPY SITE KEY 6Ld8fQAIAAAAHkfYs

Use this secret key for communication between your site and reCAPTCHA. [See server side integration](#)

COPY SECRET KEY 6Ld8fQAIAAAALWgrCpCb

4.3 Google Auth

Click [here](#) and go to Google Developer Console. Click the **“Select a project”** button in the top bar. Click **“NEW PROJECT”** in the opened field.



Select a project

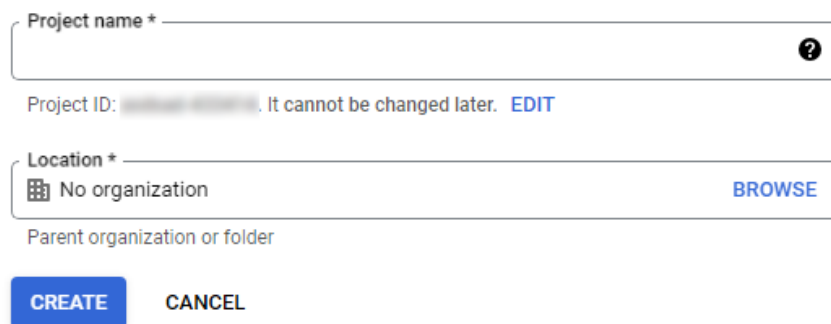
NEW PROJECT

Search projects and folders

RECENT STARRED ALL

	Name	ID
✓ ☆	My Project 12345 ?	12345678901234567890
☆	My Project 12345 ?	12345678901234567890
☆	My Project 12345 ?	12345678901234567890
☆	My Project 12345 ?	12345678901234567890

Fill the **“Project Name”** input. It doesn't matter what name you write here.



Project name *

Project ID: 12345678901234567890. It cannot be changed later. [EDIT](#)

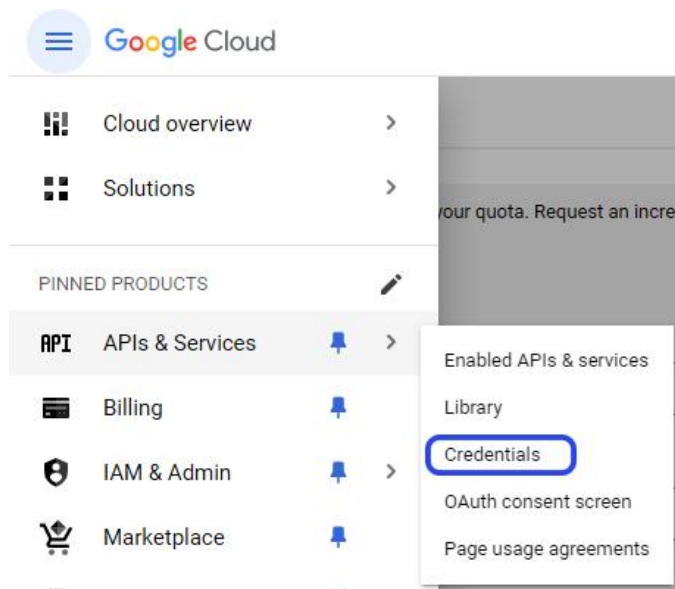
Location *

No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

Click on **“APIs & Services > Credentials”**.



Click on the **“Configure consent screen”** button. Select **“External”** and click on the **“Create”** button.

Fill the form and click on **“SAVE AND CONTINUE”**. You don't need to do anything in step 2 (Scopes) and step 3 (Test users), proceed through these steps by clicking the **“SAVE AND CONTINUE”** button. Click on **“BACK TO DASHBOARD”** on the Summary step.

 The image shows the 'App information' form in the Google Cloud console. The form is divided into several sections: 'App information' (with fields for App name, User support email, and App logo), 'App domain' (with fields for Application home page, Application privacy policy link, and Application terms of service link), 'Authorized domains' (with an 'ADD DOMAIN' button), and 'Developer contact information' (with an 'Email addresses' field). At the bottom right, there are 'SAVE AND CONTINUE' and 'CANCEL' buttons.

Click **“Credentials”** on the left. Click on **“CREATE CREDENTIALS > OAuth client ID”**.

Credentials [+ CREATE CREDENTIALS](#) [DELETE](#)

Create credentials to access Google APIs

☒ Remember this

API Keys

☐ Name

No API keys to display

API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

Select "Web application" from the **"Application type"** menu. Add your website url to **"Authorized JavaScript origins"** and add your **website url + /login/google/callback** to Authorized redirect URIs. Click on "CREATE".

Authorized JavaScript origins ⓘ

For use with requests from a browser

URIs 1 *

[+ ADD URI](#)

Authorized redirect URIs ⓘ

For use with requests from a web server

URIs 1 *

[+ ADD URI](#)

Note: It may take 5 minutes to a few hours for settings to take effect

[SAVE](#) [CANCEL](#)

You can see **"Your Client ID"** and **"Your Client Secret"** codes in the opened window.

4.4 Amazon S3

Step 1: Create an Amazon S3 Bucket

[Log in](#) to your AWS account; if you don't have one already, sign up for one.

Proceed to the [S3 section](#).

Select **Create Bucket**.

Enter a unique name for your bucket.

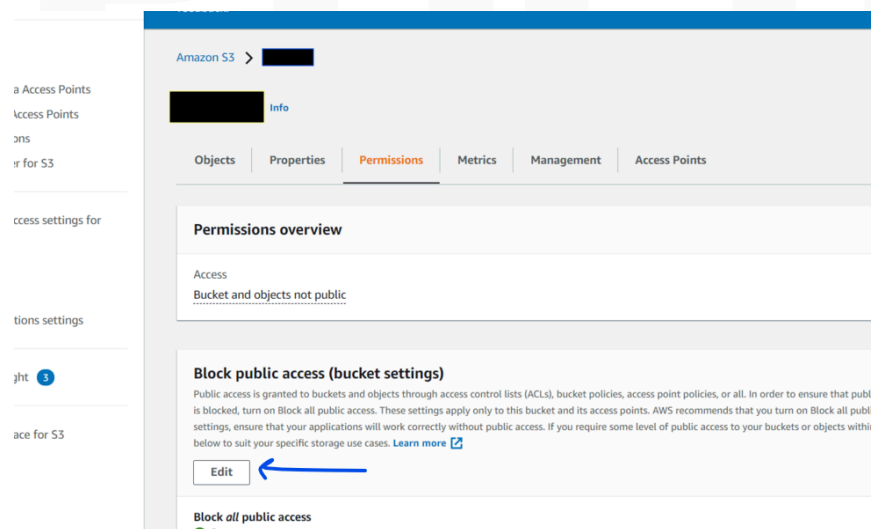
Select a region.

Uncheck the box to block public access.

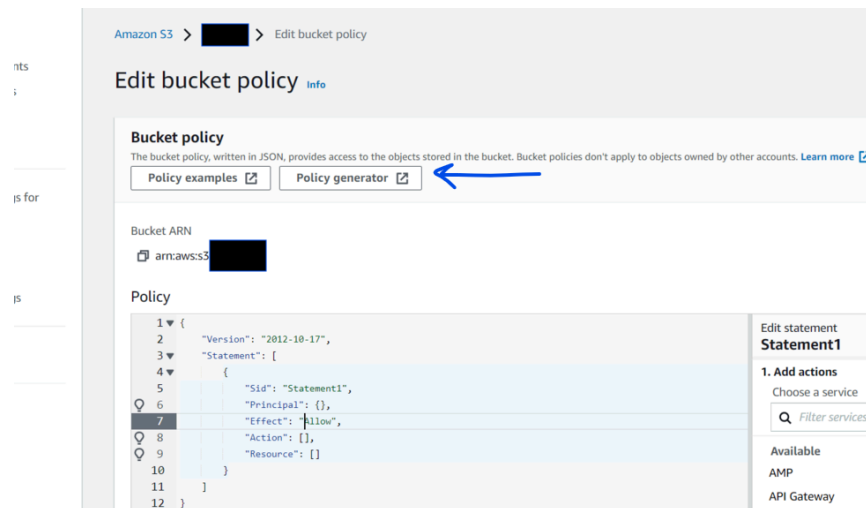
Leave all other default settings unchanged and select **Create**.

Step 2: Create an Amazon S3 Bucket Policy

Begin by selecting the bucket. Select the **Permissions** tab and then select “**Edit**” in the “Bucket Policy” session.



Copy Bucket ARN and proceed to the **Policy Generator**.



Edit the statement. Use the ARN you copied earlier.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal
Use a comma to separate multiple values.

AWS Service Amazon S3 ☐ All Services (*)
Use multiple statements to add permissions for more than one service.

Actions -- Select Actions -- ☒ All Actions (*)

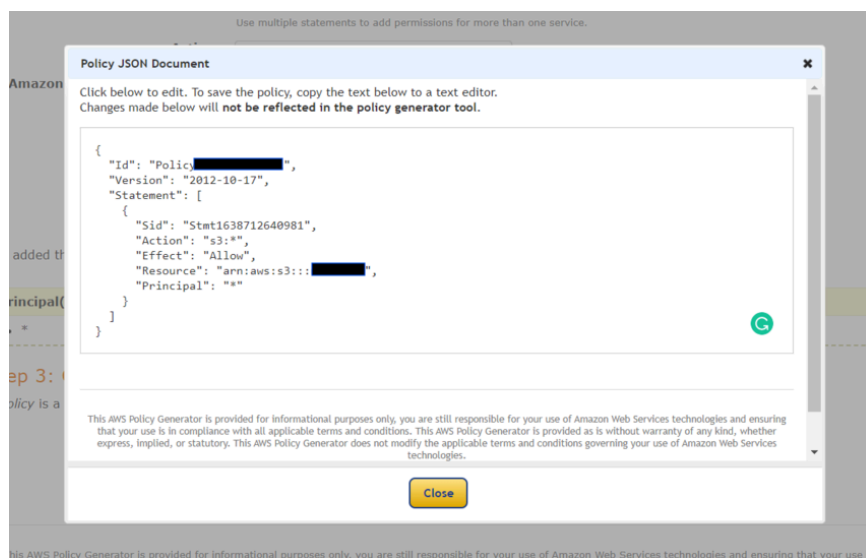
Amazon Resource Name (ARN)
ARN should follow the following format: arn:aws:s3:::\${BucketName}/\${KeyName}.
Use a comma to separate multiple values.

[Add Conditions \(Optional\)](#)

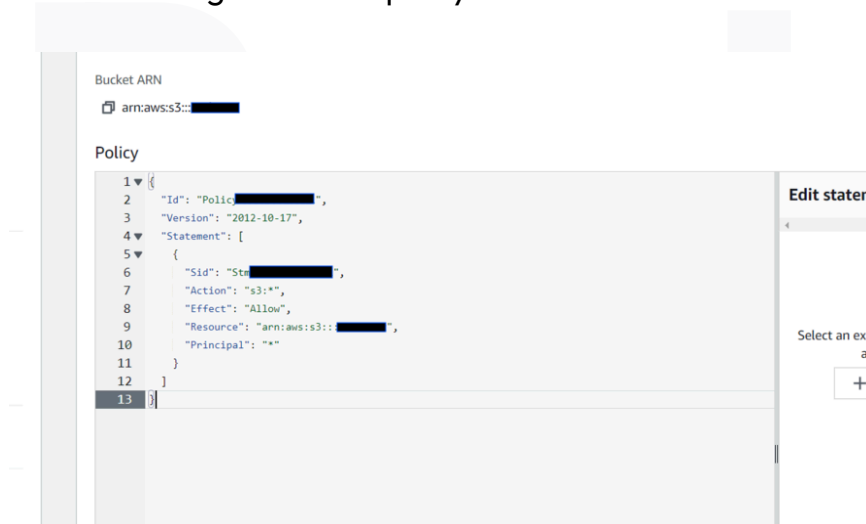
[Add Statement](#)

Step 3: Generate Policy

Generate the policy document.



The Policy is generated in JSON format. Copy and paste it into the **Bucket Policy** tab and save. Ensure that you have unchecked **Block all public access** in the permissions tab before creating the bucket policy.



Step 3: Create an IAM User

If you don't have an existing user, you can [create a new one](#) and add a bucket policy that allows the [IAM user to upload to an S3 bucket](#).

Select Add User.

Add the user name and select the AWS access type.

Set the password and then proceed to permissions.

Leave the other default settings unchanged and create the user.

Save the user's access key ID and secret access key since the secret access key can only be viewed once.

4.5 Cloudflare R2

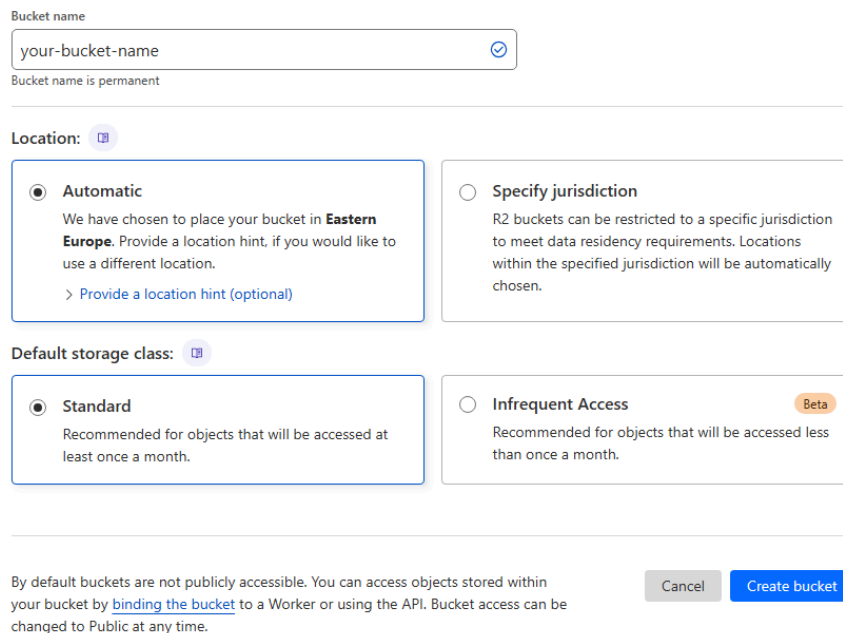
[Log in](#) to your Cloudflare account; if you don't have one already, sign up for one.

Proceed to the [R2 section](#).

Select **Create Bucket**.

Create a bucket

Get started by creating a new empty bucket. You'll be able to add data to your bucket using the dashboard or [Wrangler CLI](#).



Bucket name

your-bucket-name

Bucket name is permanent

Location:

☒ Automatic

We have chosen to place your bucket in **Eastern Europe**. Provide a location hint, if you would like to use a different location.

[Provide a location hint \(optional\)](#)

☐ Specify jurisdiction

R2 buckets can be restricted to a specific jurisdiction to meet data residency requirements. Locations within the specified jurisdiction will be automatically chosen.

Default storage class:

☒ Standard

Recommended for objects that will be accessed at least once a month.

☐ Infrequent Access Beta

Recommended for objects that will be accessed less than once a month.

By default buckets are not publicly accessible. You can access objects stored within your bucket by [binding the bucket](#) to a Worker or using the API. Bucket access can be changed to Public at any time.

Cancel Create bucket

Enter a unique name for your bucket and click the **"Create bucket"**.

Click **"Manage API tokens"** on the overview page.



Click **"Create API token"**.

Create API Token

Token name

your-token-name

Permissions

Specify the R2 Storage permission type for this token.

☐ **Admin Read & Write:** Allows the ability to create, list and delete buckets, edit bucket configurations, as well as list, write and read objects

☐ **Admin Read only:** Allows the ability to list buckets and view bucket configuration, as well as list and read objects.

☒ **Object Read & Write:** Allows the ability to read, write, and list objects in specific buckets.

☐ **Object Read only:** Allows the ability to read and list objects in specific buckets.

Permissions can be further customized or created for multiple accounts from your [Profile](#).

Specify bucket(s)

☐ Apply to all buckets in this account (including newly created buckets)

☒ Apply to specific buckets only

your-bucket-name

After filling out the form, click **"Create API Token"**.

Use the following credentials for S3 clients:

Access Key ID

9abb3f9099123a Your Cloudflare Key

Click to copy

Secret Access Key

f710ffb3aff09c46c2abfcd9e2 Your Cloudflare Secret

Click to copy

Use jurisdiction-specific endpoints for S3 clients:

Default

https://5c638ce5d9e6 Your Cloudflare Endpoint

Click to copy

Click on your bucket on the overview page and go to the **"Settings"** tab.

Click the **"Connect Domain"** button in the **"Custom domains"** area. Add a subdomain for your own domain here. For example;

Your main domain;
[yourdomain.com](#)

Your custom domain for Cloudflare R2;
[assets.yourdomain.com](#)

or

[cdn.yourdomain.com](#)

or

[files.yourdomain.com](#)

Don't forget to create a record for the subdomain. This subdomain is your Cloudflare url.

<p>Custom domains</p> <p>When a custom domain is connected to your bucket, the contents of your bucket will be made publicly accessible through that domain. Websites connected can also benefit from Cloudflare features such as bot management, Access, and Cache. Learn more</p>		
Domain	Status	Access to Bucket
assets.yourdomain.com	Active	Allowed

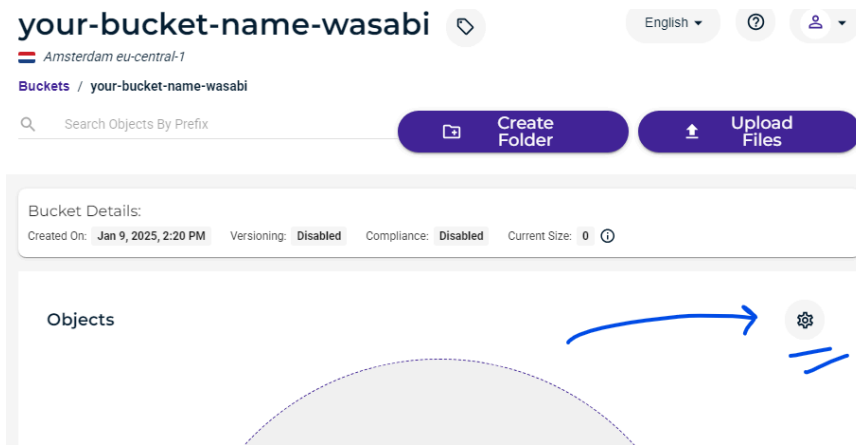
4.6 Wasabi

[Log in](#) to your Wasabi account; if you don't have one already, sign up for one.

Proceed to the [buckets](#) page.

Fill in the **Bucket Name** and **Region** fields, you do not need to change the other fields.
Create your bucket.

Click your bucket name on the buckets page. And click the settings icon.



Go to the "**Permissions**" tab and edit "**Bucket Policy**".

Paste the following codes into the bucket policy field and save;

```
{
  "Id": "Policy1736089665775",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1736089664410",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::your-bucket-name-wasabi"
    }
  ]
}
```

Don't forget to change your-bucket-name-wasabi to your own.

Click "**Users**" in the sidebar. Click the "**Create User**" button.

Create a Username

UserName

Type of Access

☒ Programmatic (create API keys) ☐ Console

Multi-Factor Authentication (MFA)

☐ Require MFA ⓘ

Select **"AmazonS3FullAccess"** in the policies step.

Start typing to find policies for the user.

Attach Policy To User

The summary should be like this;

User Details

Username:

your-username

Console Access:

No

API Access:

Yes


Multi-Factor Authentication (MFA):

Not required

Permissions

Groups

Policies

AmazonS3FullAccess 

After creating the user;

Create Access Key ×

Note: Download your key file now, which contains your new access key and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.

Access Key:

K91OD9

Your Wasabi Key

Secret Key: [Hide](#)



zGEgRyE

Your Wasabi Secret

[Download CSV](#) [Copy Keys To Clipboard](#)

Bucket List

Your Wasabi Region

Bucket Name	Owner	Region
 your-bucket-name-wasabi		 eu-central-1

In our example;

Wasabi Region = eu-central-1

Wasabi Bucket = your-bucket-name-wasabi

Wasabi Url = <https://s3.eu-central-2.wasabisys.com/>

The wasabi url is as follows;

https://s3.YOUR_REGION.wasabisys.com/

4.7 FTP

You can store files on your backup server via ftp.

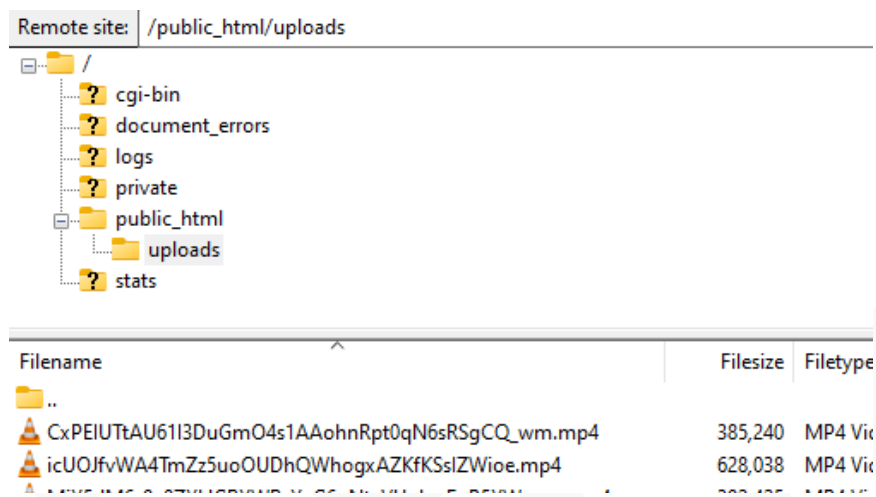
FTP Path is the path of the folder where the files will be stored.

FTP URL is the public url where uploaded files will be displayed.

For example;

FTP Path = /public_html/uploads

FTP URL = <http://192.168.10.10/uploads/>



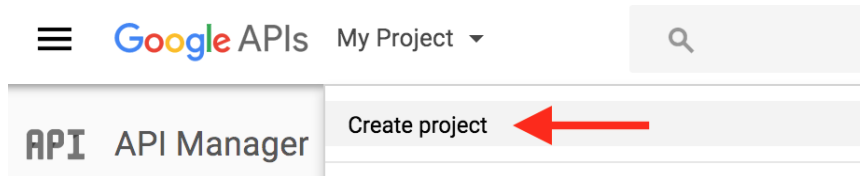
When you go to

http://192.168.10.10/uploads/CxPEIUTtAU61I3DuGmO4s1AAohnRpt0qN6sRSgCQ_wm.mp4 you should be able to watch the video in the browser without any problems.

4.8 Google Drive

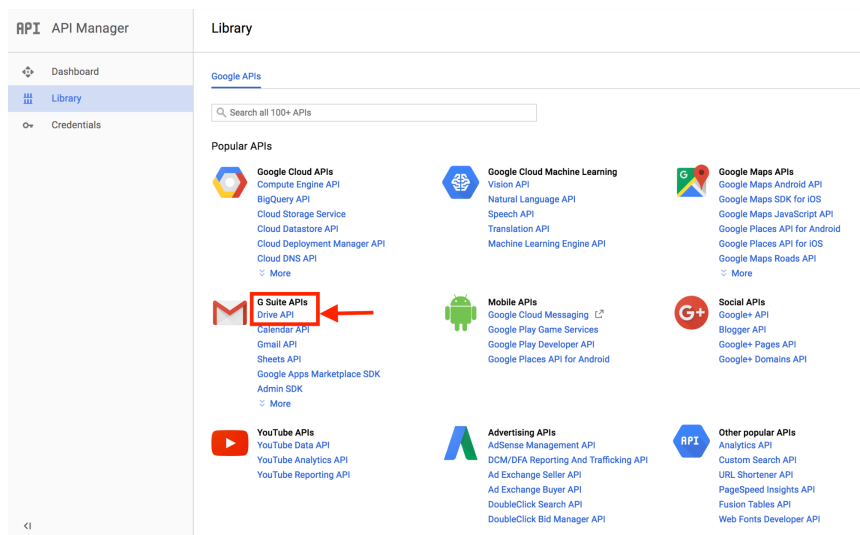
Step 1: Getting your Client ID and Secret

[Log in](#) to your Google Account. Create a new project using the dropdown at the top.



After you enter a name, it takes a few seconds before the project is successfully created on the server.

Make sure you have the project selected at the top. Then go to **Library** and click on **"Drive API"** under **"G Suite APIs"**.



And then Enable it.



Google Drive API




ENABLE

Go to "**Credentials**" and click on the tab "**OAuth Consent Screen**". Fill in a "**Product name shown to users**" and Save it. Don't worry about the other fields.

API Manager


Credentials


Credentials OAuth consent screen Domain verification

Email address 

Product name shown to users



Homepage URL (Optional)

Product logo URL (Optional) 

 This is how your logo will look to end users
Max size: 120x120 px

Privacy policy URL
Optional until you deploy your app


Terms of service URL (Optional)

The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

Then go back to Credentials, click the button that says "**Create Credentials**" and select "**OAuth Client ID**".

Create credentials 

API key
Identifies your project using a simple API key to check quota and access.
For APIs like Google Translate

OAuth client ID
Requests user consent so your app can access the user's data.
For APIs like Google Calendar.

Service account key
Enables server-to-server, app-level authentication using robot accounts.
For use with Google Cloud APIs.

Help me choose
Asks a few questions to help you decide which type of credential to use

Choose "**Web Application**" and give it a name.

Add **<https://developers.google.com/oauthplayground>** in "**Authorized redirect URIs**". You will need to use this in the next step to get your refresh token. Once you have the token, you can remove the URI.

API Manager

← Create client ID

Application type

- ☒ Web application
- ☐ Android [Learn more](#)
- ☐ Chrome App [Learn more](#)
- ☐ iOS [Learn more](#)
- ☐ PlayStation 4
- ☐ Other

Name

Test

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard ([http://*.example.com](#)) or a path ([http://example.com/subdir](#)). If you're using a nonstandard port, you must include it in the origin URI.

[http://www.example.com](#)

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

[https://developers.google.com/oauthplayground](#)

[http://www.example.com/oauth2callback](#)

Create Cancel

Click Create and take note of your **Client ID** and **Client Secret**.

Note: You can go through an approval process to use Google Drive. If you add yourself as a test, you can use Google Drive temporarily.

Step 2: Getting your Refresh Token

Go to **<https://developers.google.com/oauthplayground>**. Make sure you added this URL to your Authorized redirect URIs in the previous step.

In the top right corner, click the settings icon, check "**Use your own OAuth credentials**" and paste your Client ID and Client Secret.

?

↔

⚙

OAuth 2.0 configuration

OAuth flow: **Server-side**

OAuth endpoints: **Google**

Authorization endpoint:

Token endpoint:

Access token location: **Authorization header w/ Bearer prefix**

Access type: **Offline**

Force prompt: **Consent Screen**

☒ Use your own OAuth credentials

You will need to list the URL <https://developers.google.com/oauthplayground> as a valid redirect URI in your [Google APIs Console](#)'s project. Then enter the client ID and secret assigned to a web application on your project below:

OAuth Client ID:

OAuth Client secret:

Note: Your credentials will be sent to our server as we need to proxy the request. Your credentials will not be logged.

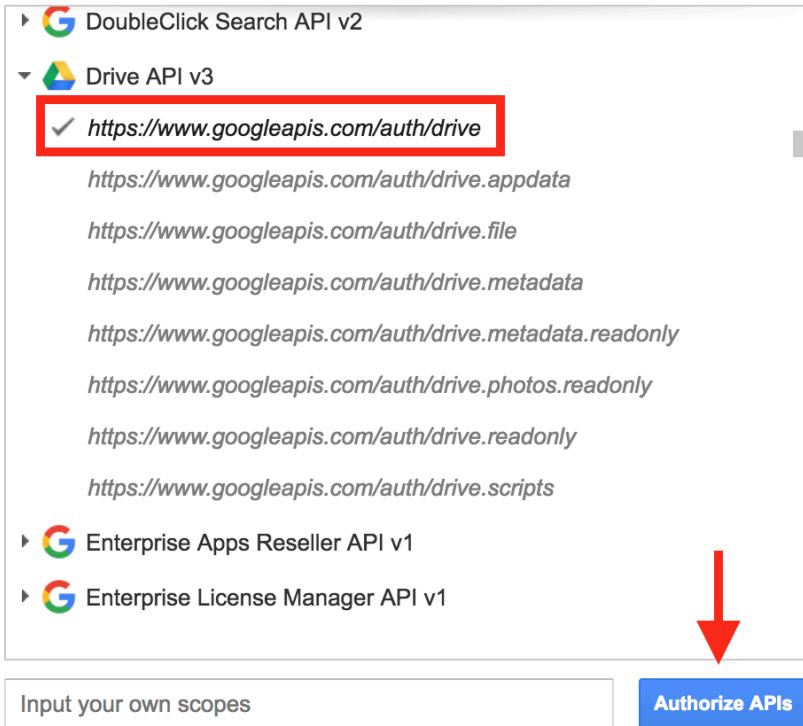
[Close](#)


US


In step 1 on the left, scroll to **"Drive API v3"**, expand it and check the first drive scope.

▼ Step 1 Select & authorize APIs


Select the **scope** for the APIs you would like to access or input your own OAuth scopes below. Then click the **"Authorize APIs"** button.




▶  DoubleClick Search API v2

▼  Drive API v3

- ✓ <https://www.googleapis.com/auth/drive>
- <https://www.googleapis.com/auth/drive.appdata>
- <https://www.googleapis.com/auth/drive.file>
- <https://www.googleapis.com/auth/drive.metadata>
- <https://www.googleapis.com/auth/drive.metadata.readonly>
- <https://www.googleapis.com/auth/drive.photos.readonly>
- <https://www.googleapis.com/auth/drive.readonly>
- <https://www.googleapis.com/auth/drive.scripts>

▶  Enterprise Apps Reseller API v1

▶  Enterprise License Manager API v1

Input your own scopes

Authorize APIs

Click **"Authorize APIs"** and allow access to your account when prompted. There will be a few warning prompts, just proceed.

When you get to step 2, check **"Auto-refresh the token before it expires"** and click **"Exchange authorization code for tokens"**.

OAuth 2.0 Playground



▸ Step 1 Select & authorize APIs

▾ Step 2 Exchange authorization code for tokens

Once you got the Authorization Code from Step 1 click the **Exchange authorization code for tokens** button, you will get a refresh and an access token which is required to access OAuth protected resources.

Authorization code:

Exchange authorization code for tokens

Refresh token:

Access token:

Refresh access token

☒ Auto-refresh the token before it expires.

Note: The OAuth Playground does not store refresh tokens, but as refresh tokens never expire, user should go to their Google Account [Authorized Access](#) page if they would like to manually revoke them.

When you get to step 3, click on step 2 again and you should see your **refresh token**.

OAuth 2.0 Playground



▸ Step 1 Select & authorize APIs

▾ Step 2 Exchange authorization code for tokens

Once you got the Authorization Code from Step 1 click the **Exchange authorization code for tokens** button, you will get a refresh and an access token which is required to access OAuth protected resources.

Authorization code: 4/Kbtgkos_8KeM5V8BLirblnOmb_WnthO

Exchange authorization code for tokens

Refresh token: 1/KAmvB-dvOrIAzuycy_w31cjMruc

Access token: ya29.GlvhAx1kSU9U66q7POw_WO

Refresh access token

☒ Auto-refresh the token before it expires.

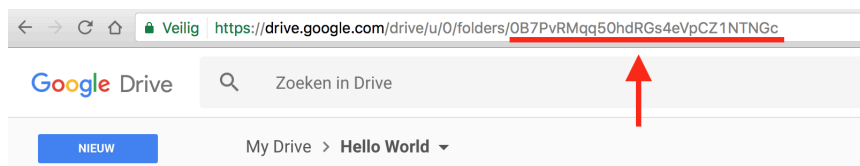
The access token will expire in **3566** seconds.

Note: The OAuth Playground does not store refresh tokens, but as refresh tokens never expire, user should go to their Google Account [Authorized Access](#) page if they would like to manually revoke them.

Step 3: Getting your Root Folder ID

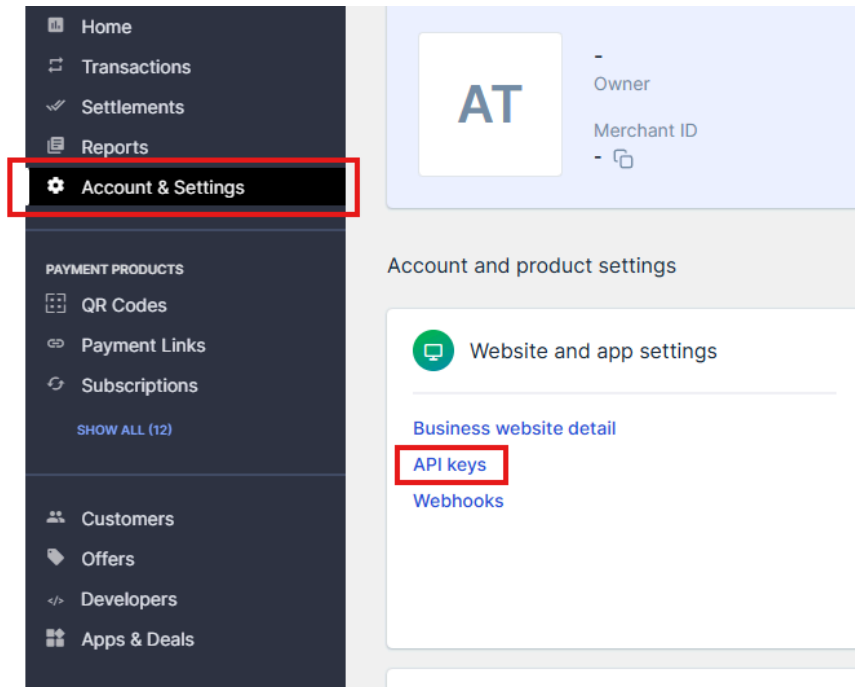
If you want to store files in your Google Drive root directory, then the folder ID can be **null**. Else go into your Drive and create a folder.

Because Google Drive allows duplicate names, it identifies each file and folder with a unique ID. If you open your folder, you will see the **Folder ID** in the URL.



4.9 Razorpay

To use the Razorpay payment method, you need to get API keys. Create an account [here](#). After creating an account, go to the API keys page.



Enter the keys on this page in the Payments > Settings > Gateway > Razorpay field on your website.

5. Upload Settings

5.1 Uploadable Types

By default, 15 different file formats can be uploaded. These formats are; ZIP, APNG, AVIF, GIF, ICO, JPE, JPEG, JPG, PNG, WEBP, HEIC, MP4, AVI, WMV, WEBM

To edit the uploadable formats, go to the **config/upload.php** file and change the **UPLOADABLE_TYPES** variable. The format you add must be supported by your server.

5.2 File Size Limit

The file upload limit for users is 50 MB for each file. If you want to change this value, go to the **config/upload.php** file and change the **MAX_FILE_SIZE** variable. This value must be written in KB. (1 MB = 1024 KB)

Upload limit may conflict with your server settings. To prevent this, you can edit your php.ini file.

Maximum allowed size for uploaded files;

```
upload_max_filesize = 50M
```

Must be greater than or equal to upload_max_filesize;

```
post_max_size = 50M
```

5.3 Server Quota

Specifying the server's storage limit is required for both the script and the statistical graphics. If you have allocated 40 GB for file storage on your server; go to the **config/upload.php** file and change the **SERVER_QUOTA** variable to 41943040. This value must be written in KB. (1 GB = 1024 MB and 1 MB = 1024 KB)

5.4 Server Storage

There are 6 different storages in the script; **local**, **Amazon S3**, **Cloudflare R2**, **Wasabi**, **FTP** and **Google Drive**. Storage settings can be adjusted on page;

Admin dashboard > Settings > Storage

6. Configuring Server Cron Job

Portus automatically deletes uploaded files under certain conditions. For this, you need to create a cron job on your server.

Command;

```
cd /your_website_full_path/public_html && php artisan schedule:run >> /dev/null 2>&1
```

Time value; * * * * *

The cron job you created should look like this; *

■ Command	Minute	Hour	Day	Month	Day of Week
■ <code>cd /home/web/portus.aneontech.com/public_html && php artisan schedule:run >> /dev/null 2>&1</code>	*	*	*	*	*

* You should write the path of your website according to your server, it may be different.

7. Language

To add a new language, you can duplicate an existing language in the lang folder. The script pulls the language information from the lang.json file. For example, if we want to add German language; duplicate the **en** folder in the lang folder and make the folder name as **de**. The content of the lang.json file in de folder will be as follows;

```
"langName": "Deutsch",  
"langCode": "de",  
"langDirection": "LTR",  
"langFlag": "fi fi-de fis"
```

After making this edit, you can edit the translations in the php files in the **de** folder.

Finally, you need to give permission to the new language you added. Go to;

public_html/config/laravellocalization.php

And find the language you added. (For example; Deutsch)

```
//'de' => ['name' => 'German', 'script' => 'Latn', 'native' => 'Deutsch', 'regional' => 'de_DE'],
```

Remove the slashes at the beginning of de. The final state should be like this;

```
'de' => ['name' => 'German', 'script' => 'Latn', 'native' => 'Deutsch', 'regional' => 'de_DE'],
```

That's all.

8. Affiliate

Portus uses ip2location.io to detect visitors' location information. ip2location is free within the limit of 50000 lookups per month. If you want to increase the limit, you can get a premium account [here](#).

9. Security

In laravel 11, the public directory is the entry point for all requests entering your application. Due to this situation, a public segment is added to the URL. Portus hides the public segment in the URL with the htaccess file. But this may create **security vulnerabilities**.

If you can set the **domain root as the public folder**, delete the .htaccess file in public_html. In this way, you increase the security level of your website.

Changing domain root may vary depending on the server type or hosting control panel you use. You can request **support** on this issue by writing to us.

10. Support

You can send us an email for all kinds of support before installation, during the installation and after installation. Our e-mail address; **aneontech.help@gmail.com**



Portus