

# Гайд по 4 лабе WEB

## 1. PHP

Шаг 1. Скачиваем zip архив PHP 8.3 по [ссылке](#):

**PHP 8.3 (8.3.6)**

---

[Download source code](#) [27.01MB]  
[Download tests package \(phpt\)](#) [16.26MB]

**VS16 x64 Non Thread Safe (2024-Apr-10 15:14:02)**

- [Zip](#) [30.66MB]   
sha256: d0967a6ff800567ba60bfb9883086241dc264334ca95107cad42d8a39ff409b
- [Debug Pack](#) [25.69MB]  
sha256: 5617afafe4535f58a1079abd01454cc185746f1182d33c5d20e09df725da8d04
- [Development package \(SDK to develop PHP extensions\)](#) [1.26MB]  
sha256: 1c202c2c138fa408ee792c420fae6655af0e0506d3a09e327c748b54b4b11c9b

Шаг 2. Распаковываем скачанный архив в любое удобное место на диске C

Шаг 3. Проверяем корректность установки PHP

- 1) Зажимаем **Win+R** на клавиатуре, вводим в появившемся окне **cmd**, жмем **Enter**.
- 2) В появившейся командной строке вводим **php -v**
- 3) Должно выдать что-то типо:

```
C:\Windows\system32\cmd  x  +  v
Microsoft Windows [Version 10.0.22631.3447]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\User>php -v
PHP 8.3.6 (cli) (built: Apr 10 2024 14:53:44) (NTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
```

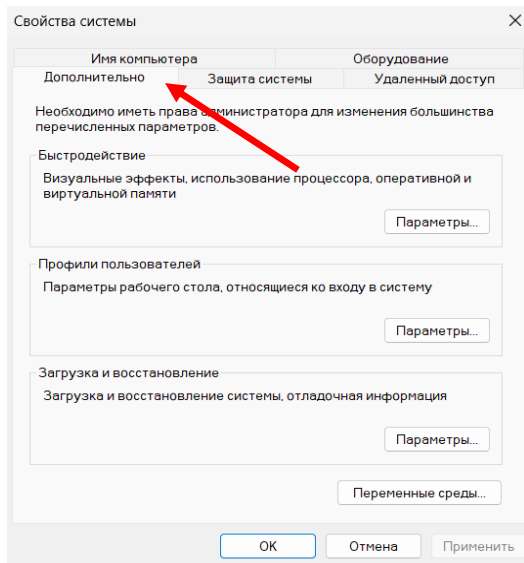
Если все так, пропускаем следующие шаги до раздела **2. Composer**. Если же командная строка ругается:

```
"php-v" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
```

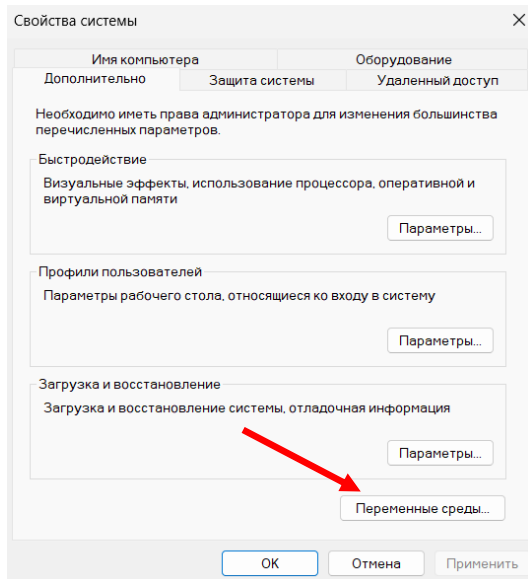
**Шаг 4.** Ну что, бедолага, давай чинить PHP.

1) Зажимай клавиши **Win+R**, вводи **sysdm.cpl** и жми **Enter**.

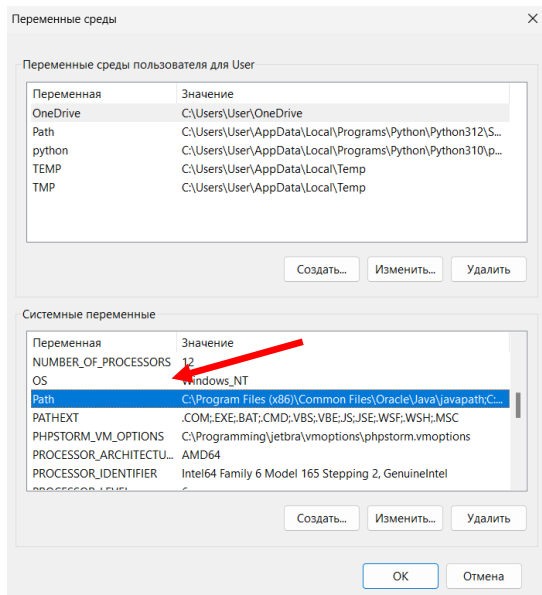
2) -> **Дополнительно**



3) -> **Переменные среды**

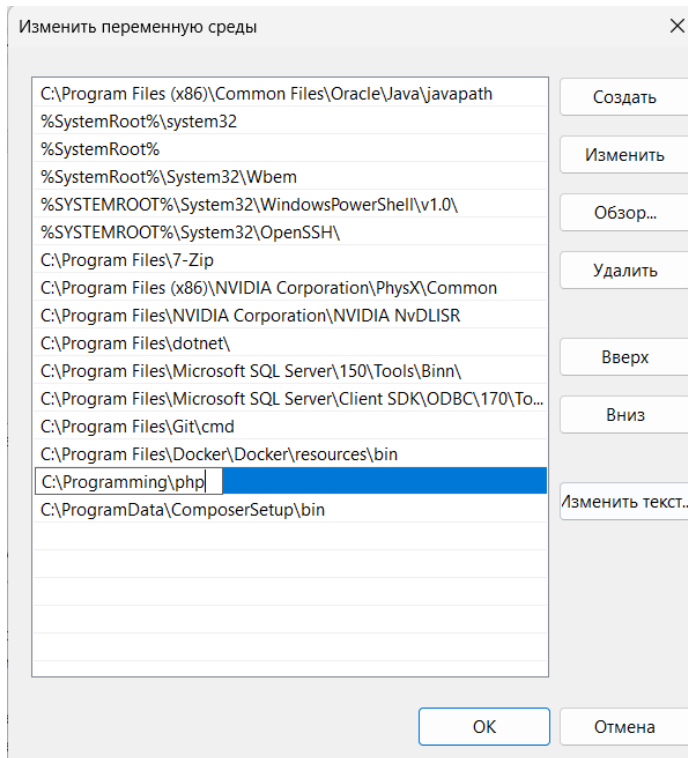


4) В появившемся окне в системных переменных ищем и выбираем **Path**



5) -> **Изменить**

6) Создаем путь к нашей **папке**, в которой лежит *php.exe*



7) Жмем **OK**

8) И опять **OK**

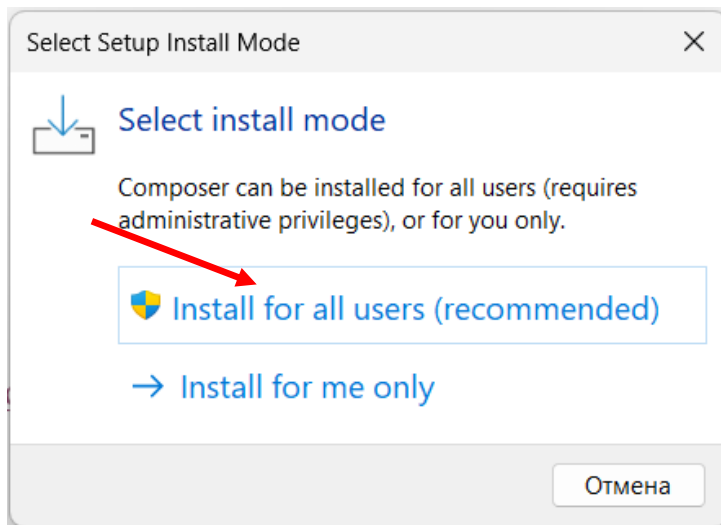
9) И ошесть **OK**

10) Повторяем **Шаг 3.**

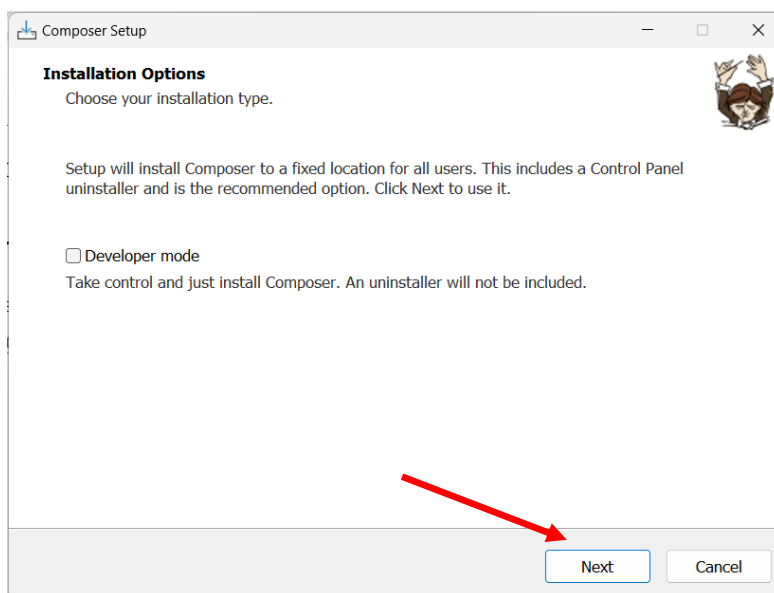
## 2. Ставим Composer на ПК

Шаг 1. Качаем [Composer для Windows](#)

Шаг 2. Next

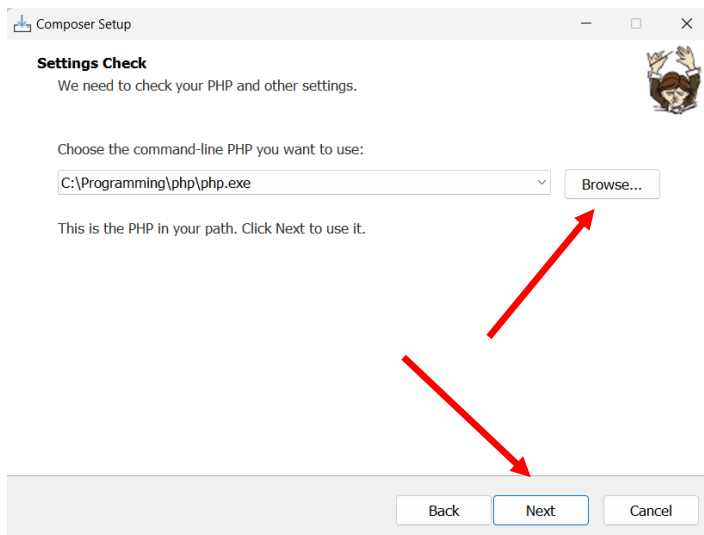


Шаг 3. Next

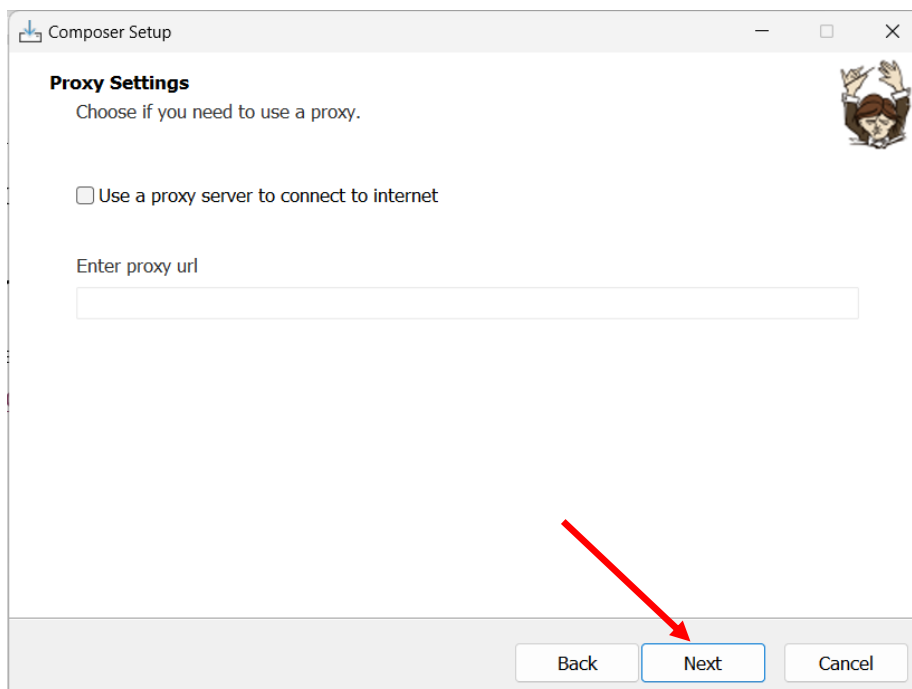


Шаг 4. Если в разделе **1. PHP** все прошло успешно, установщик должен сам определить путь до *php.exe*. Но если он этого не сделал, укажите путь вручную.

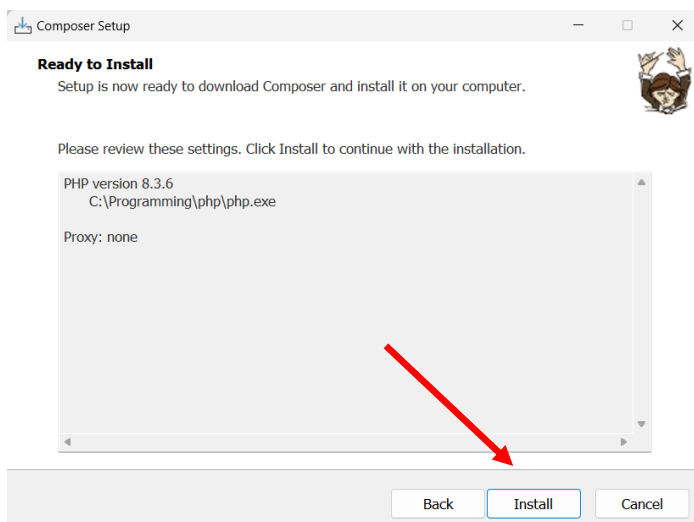
Жмем Next



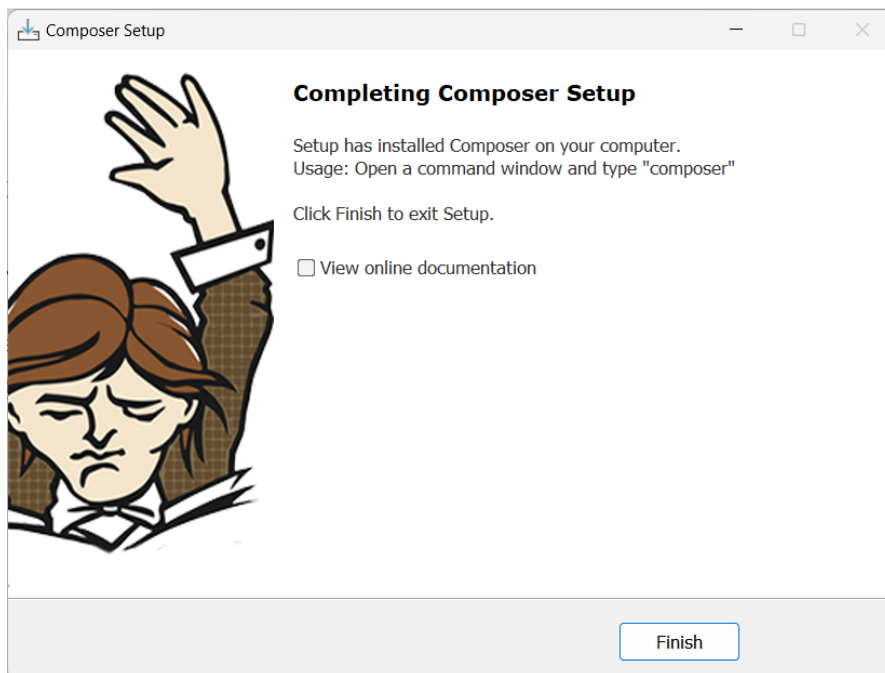
## Шаг 5. Нам этот ваш прокси и ... ненужен! Next



## Шаг 6. Install



## Шаг 7. Если во время установки не было ошибок – это победа! **Finish**



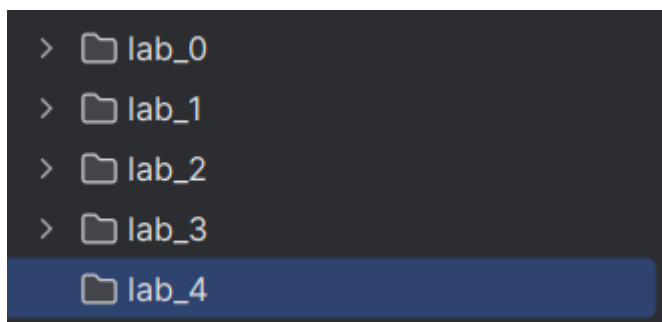
Так же можно проверить корректность установки командой  
***composer --version***

```
Командная строка
Microsoft Windows [Version 10.0.22631.3447]
(с) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

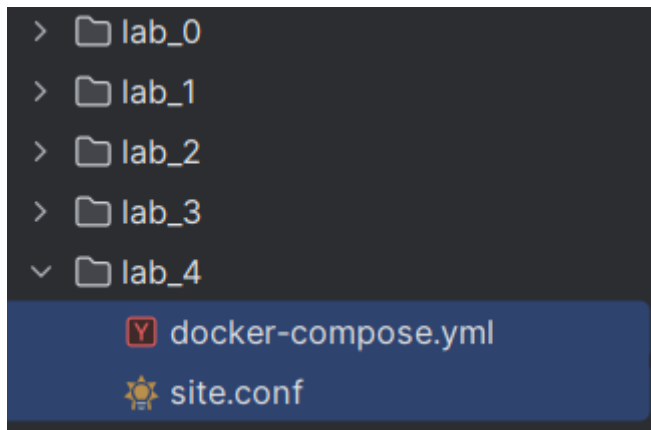
C:\Users\User>composer --version
Composer version 2.7.3 2024-04-19 21:40:57
PHP version 8.3.6 (C:\Programming\php\php.exe)
Run the "diagnose" command to get more detailed diagnostics output.
```

## 3. Ставим Composer на проект

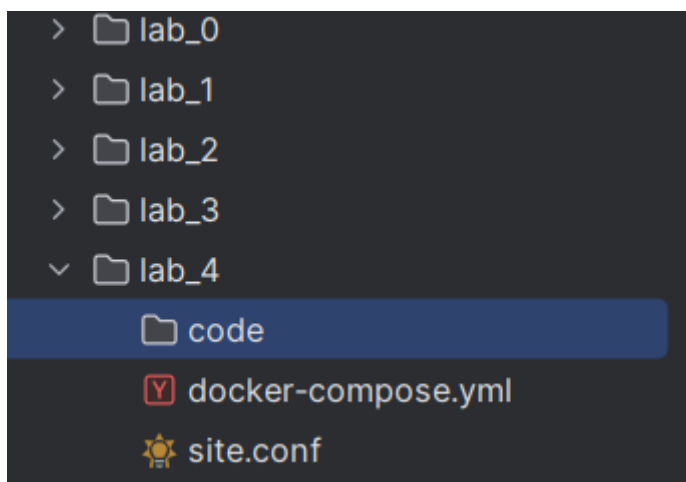
### Шаг 1. Создаем проект



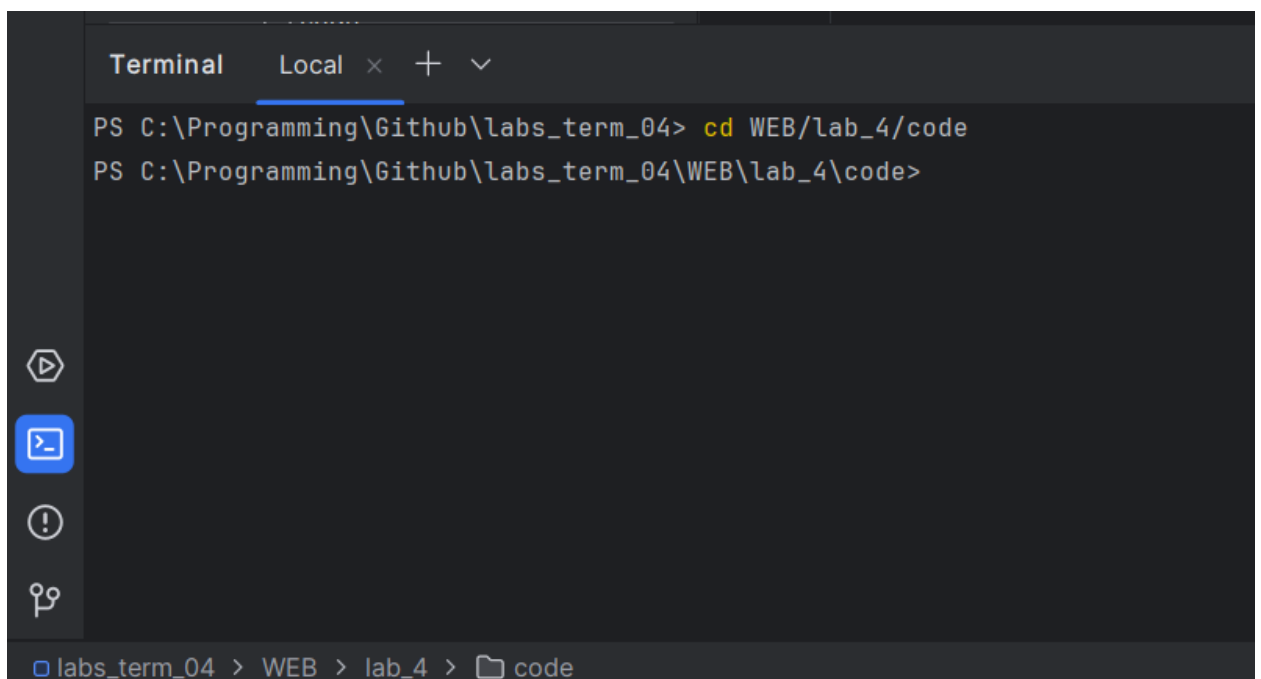
**Шаг 2.** Создаем в проекте *docker-compose.yml* и *site.conf*



**Шаг 3.** Создаем в проекте папку, в которой будет наш код (у меня это *code*)



**Шаг 4.** В терминале переходим в нашу папку *code* с помощью команды *cd <путь к папке>*



**Шаг 4.** Вводим *composer init* и отвечаем на вопросы для создания вашего *composer.json*.

- *n* = *No*
- **Enter** = вариант, который указывается в *[]*
- Или вписываем свой вариант

```
PS C:\Programming\Github\labs_term_04\WEB\lab_4\code> composer init

Welcome to the Composer config generator

This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [user/code]:
Description []:
Author [Арсений <115050782+ArPaSokolov@users.noreply.github.com>, n to skip]: n
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []: project
License []:

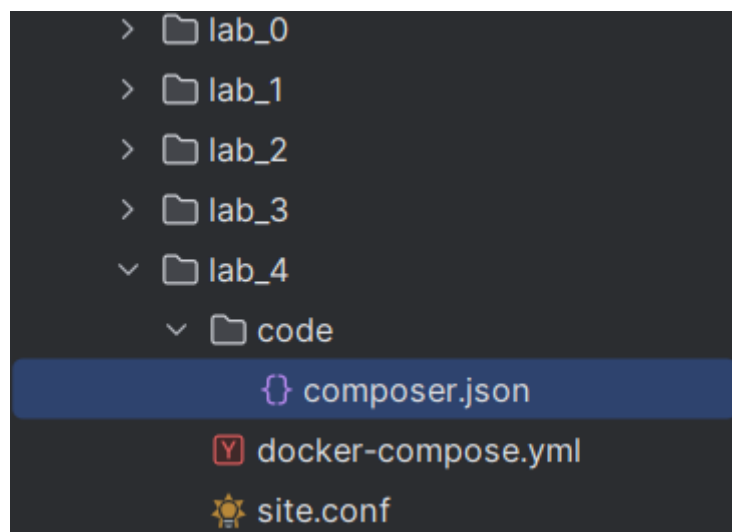
Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? n
Would you like to define your dependencies (require) interactively [yes]? n
Would you like to define your dev dependencies (require-dev) interactively [yes]? n
Add PSR-4 autoload mapping? Maps namespace "User\Code" to the entered relative path. [src/, n to skip]: n

{
    "name": "user/code",
    "type": "project",
    "require": {}
}

Do you confirm generation [yes]?
PS C:\Programming\Github\labs_term_04\WEB\lab_4\code>
```

У вас должен появиться файл *composer.json*

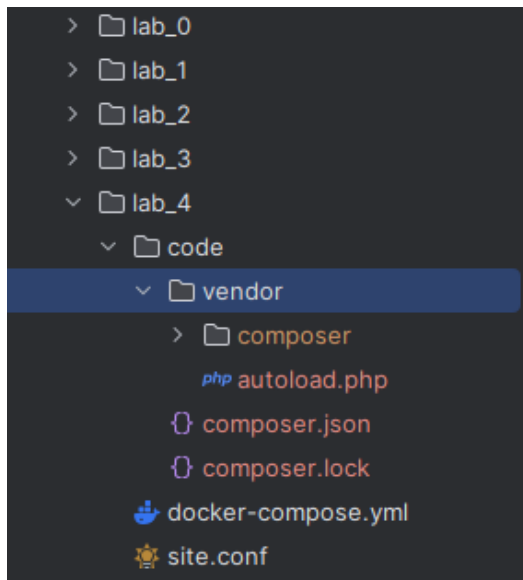




## Шаг 5. ВВОДИМ *composer install*

```
PS C:\Programming\Github\labs_term_04\WEB\lab_4\code> composer install
No composer.lock file present. Updating dependencies to latest instead of installing from lock file. See https://getcomposer.org/install for more information.
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
```

Если все прошло успешно у вас должны появиться *composer.lock* и папка *vendor* со следующим содержимым:



## Шаг 5. Подключаем Google API. Вводим *composer require google/apiclient*

У вас начнется установка, на 94% она задержится какое-то время – это нормально, без паники)

```
- Locking google/apiclient (v2.15.3)
- Locking google/apiclient-services (v0.345.0)
- Locking google/auth (v1.37.1)
- Locking guzzlehttp/guzzle (7.8.1)
- Locking guzzlehttp/promises (2.0.2)
- Locking guzzlehttp/psr7 (2.6.2)
- Locking monolog/monolog (3.0.0)
- Locking paragonie/constant_time_encoding (v2.6.3)
- Locking paragonie/random_compat (v9.99.100)
- Locking phpseclib/phpseclib (3.0.37)
- Locking psr/cache (3.0.0)
- Locking psr/http-client (1.0.3)
- Locking psr/http-factory (1.0.2)
- Locking psr/http-message (2.0)
- Locking psr/log (3.0.0)
- Locking ralouphie/getallheaders (3.0.3)
- Locking symfony/deprecation-contracts (v3.4.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 18 installs, 0 updates, 0 removals
- Downloading google/apiclient-services (v0.345.0)
- Installing paragonie/random_compat (v9.99.100): Extracting archive
- Installing paragonie/constant_time_encoding (v2.6.3): Extracting archive
- Installing phpseclib/phpseclib (3.0.37): Extracting archive
- Installing psr/log (3.0.0): Extracting archive
- Installing monolog/monolog (3.0.0): Extracting archive
- Installing ralouphie/getallheaders (3.0.3): Extracting archive
- Installing psr/http-message (2.0): Extracting archive
- Installing psr/http-factory (1.0.2): Extracting archive
- Installing guzzlehttp/psr7 (2.6.2): Extracting archive
- Installing symfony/deprecation-contracts (v3.4.0): Extracting archive
- Installing psr/http-client (1.0.3): Extracting archive
- Installing guzzlehttp/promises (2.0.2): Extracting archive
- Installing guzzlehttp/guzzle (7.8.1): Extracting archive
- Installing psr/cache (3.0.0): Extracting archive
- Installing firebase/php-jwt (v6.10.0): Extracting archive
- Installing google/auth (v1.37.1): Extracting archive
- Installing google/apiclient-services (v0.345.0): Extracting archive
- Installing google/apiclient (v2.15.3): Extracting archive
17/18 [=====] 94%
```

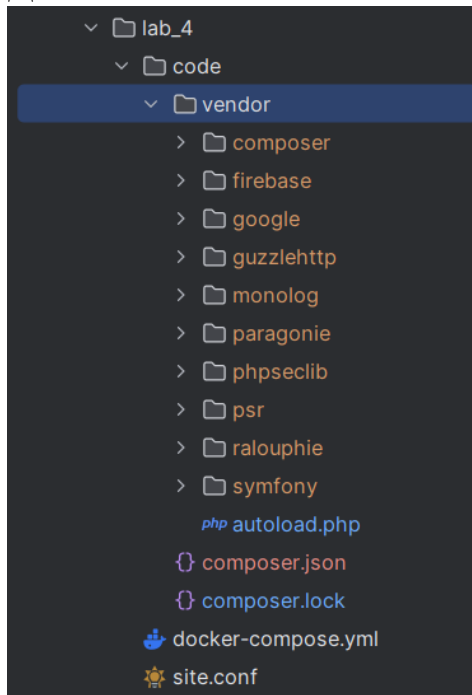
Если же у вас подобная ошибка:

```
Failed to download google/auth from dist: The zip extension and unzip/7z commands are both missing, skipping.  
The php.ini used by your command-line PHP is: C:\php\php.ini  
Now trying to download from source
```

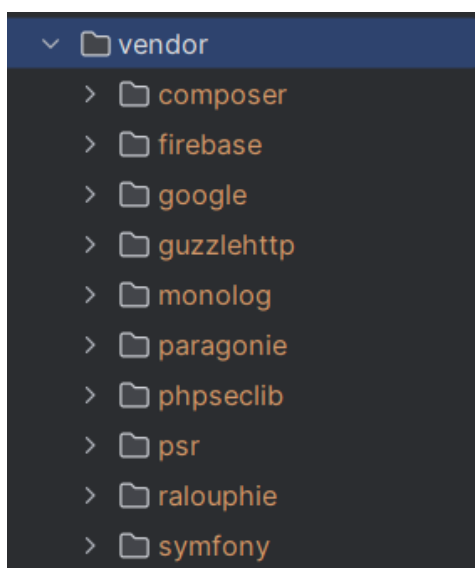
Попробуйте это [решение](#) (уже двоим спасло жизнь):

Нужно найти “;extension = zip” в php.ini и удалить точку с запятой, чтоб получилось “extension = zip”.

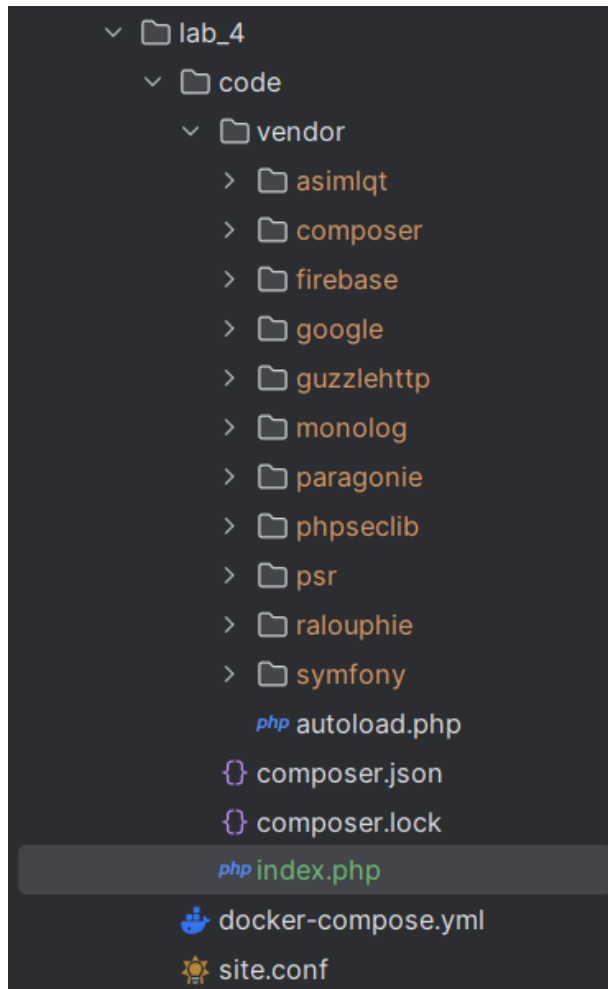
Далее имеем:



**Шаг 6.** Возможно, вам еще понадобится *composer require asimlqt/php-google-spreadsheet-client*



**Шаг 7.** Добавляем *index.php* (файл, указанный в *site.conf*) в папку *code*

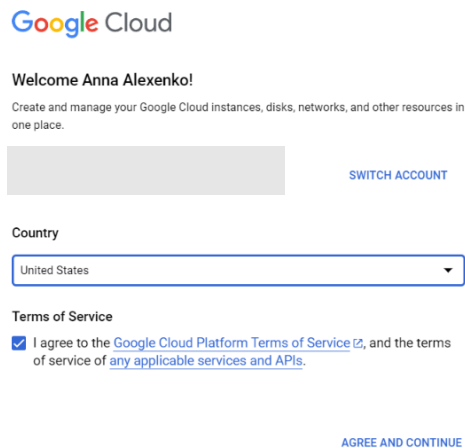


**Важно!** Папка *vendor* должна обязательно находиться в одной папке с вашим *index.php* (файлом, указанным в *site.conf*)

## 4. Создание Google API

Ох пойдет щас возня....

**Шаг 1.** Переходим по [ссылке](#), если попросит указать страну, выбираем любую страну (России в списке нет)



Google Cloud

Welcome Anna Alexenko!

Create and manage your Google Cloud instances, disks, networks, and other resources in one place.

Country

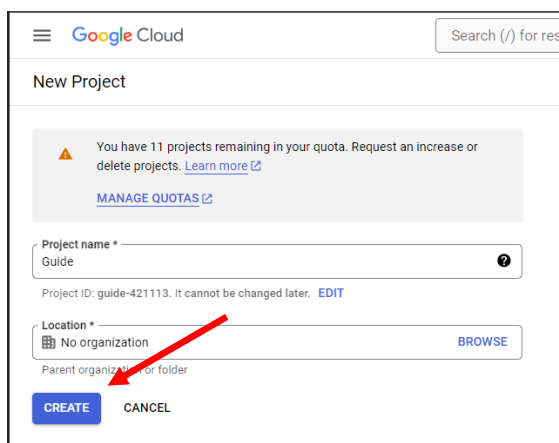
United States

Terms of Service

☒ I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#).

AGREE AND CONTINUE

**Шаг 2.** Указываем имя проекта, организацию не трогаем, жмем **CREATE**



Google Cloud

Search (/) for res

New Project

You have 11 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*

Guide

Project ID: guide-421113. It cannot be changed later. [EDIT](#)

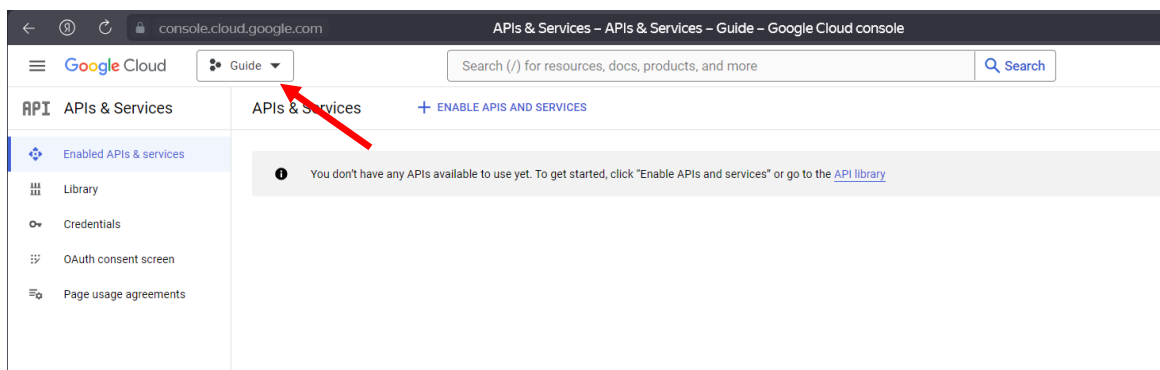
Location \*

No organization [BROWSE](#)

Parent organization or folder

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

**Шаг 3.** Нас перебрасывает на следующую страницу



console.cloud.google.com

APIs & Services - APIs & Services - Guide - Google Cloud console

Google Cloud

Guide

Search (/) for resources, docs, products, and more

Search

APIs & Services

APIs & Services

[+ ENABLE APIS AND SERVICES](#)

Enabled APIs & services

Library

Credentials

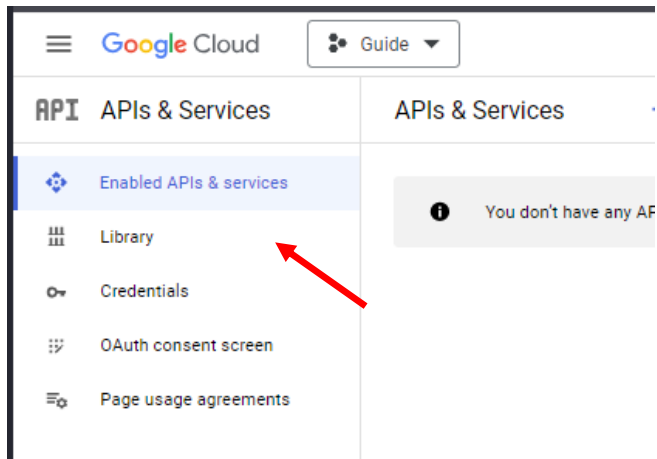
OAuth consent screen

Page usage agreements

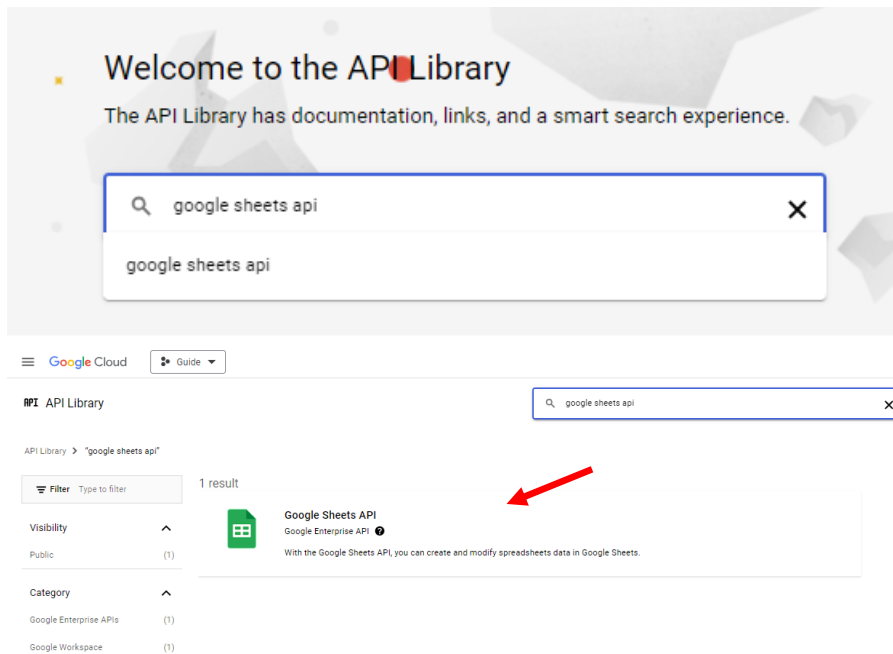
You don't have any APIs available to use yet. To get started, click "Enable APIs and services" or go to the [API library](#)

Сверху видим, что проект создан успешно

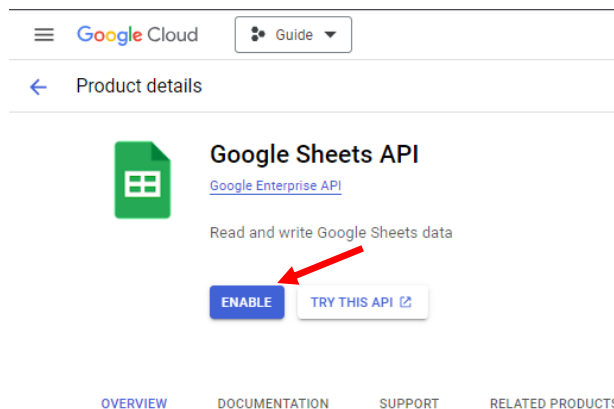
## Шаг 4. -> Library



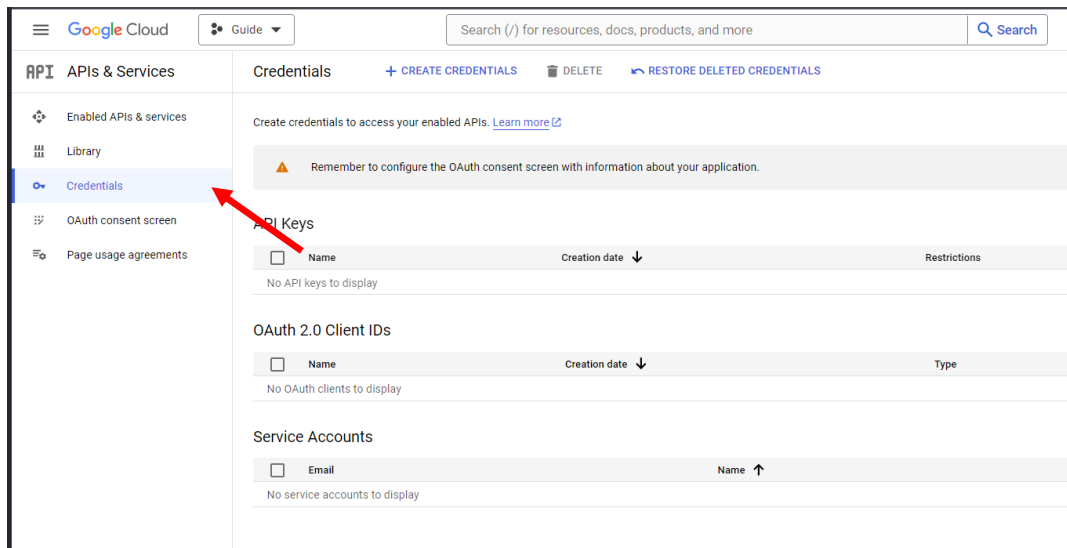
## Шаг 5. Вводим в поиске google sheets api, жмем Enter



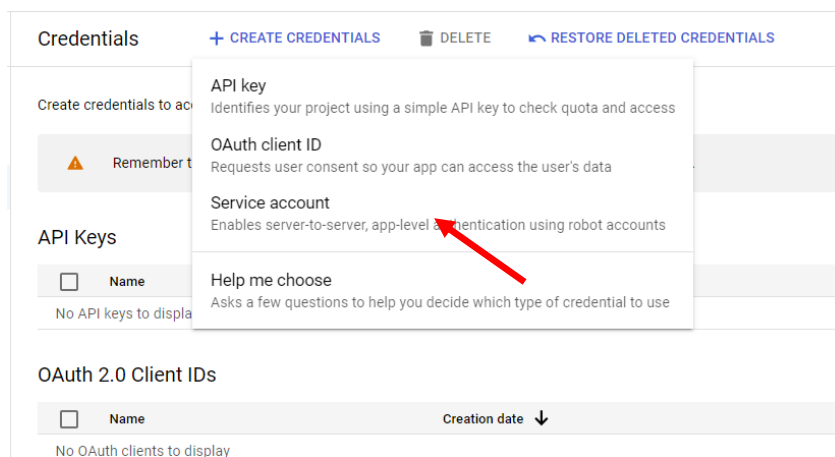
## Шаг 6. Жмем ENABLE



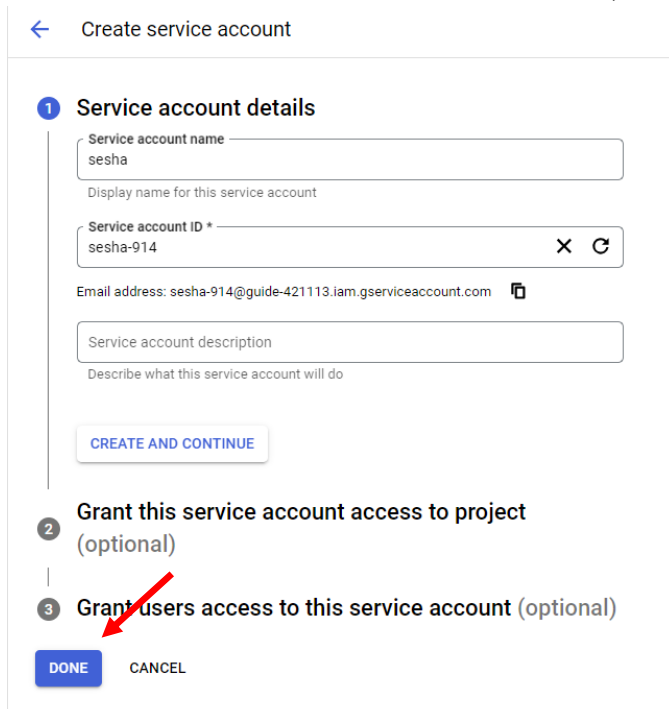
## Шаг 7. -> Credentials



## Шаг 8. -> Service account




## Шаг 9. Заполняем обязательные поля, жмем DONE



Теперь у нас есть сервисный аккаунт, запоминаем его

**Credentials** [+ CREATE CREDENTIALS](#) [DELETE](#) [RESTORE DELETED CREDENTIALS](#)

Create credentials to access your enabled APIs. [Learn more](#)

 Remember to configure the OAuth consent screen with information about your application.

**API Keys**

<input type="checkbox"/>	Name	Creation date	↓	Re
No API keys to display				

**OAuth 2.0 Client IDs**

<input type="checkbox"/>	Name	Creation date	↓
No OAuth clients to display			


**Service Accounts**

<input type="checkbox"/>	Email	
<input type="checkbox"/>	<a href="mailto:sesha-914@guid-421113.iam.gserviceaccount.com">sesha-914@guid-421113.iam.gserviceaccount.com</a>	

## Шаг 10. -> OAuth client ID

**Credentials** [+ CREATE CREDENTIALS](#) [DELETE](#) [RESTORE DELETED CREDENTIALS](#)

Create credentials to access your enabled APIs. [Learn more](#)

 Remember to configure the OAuth consent screen with information about your application.

**API Keys**

<input type="checkbox"/>	Name	Creation date	↓	Re
No API keys to display				

**OAuth 2.0 Client IDs**

<input type="checkbox"/>	Name	Creation date	↓
No OAuth clients 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

## Шаг 11. -> CONFIGURE CONSENT SCREEN

[←](#) Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

 To create an OAuth client ID, you must first configure your consent screen


[CONFIGURE CONSENT SCREEN](#)

## Шаг 12. Ничего не выбираем, жмем CREATE


OAuth consent screen

Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.


User Type

☐ Internal 

Only available to users within your organization. You will not need to submit your app for verification. [Learn more about user type](#)

☐ External 

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. [Learn more about user type](#)



CREATE

[Let us know what you think](#) about our OAuth experience

## Шаг 13. Заполняем обязательные поля

1 OAuth consent screen — 2 Scopes — 3 Test users — 4 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name \*  
Guide

The name of the app asking for consent

User support email \*

For users to contact you with questions about their consent. [Learn more](#)

App logo

This is your logo. It helps people recognize your app and is displayed on the OAuth consent screen.

After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of "Testing". [Learn more](#)

Logo file to upload 

BROWSE


Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

Developer contact information

Email addresses \*

These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE

 CANCEL

Жмем **SAVE AND CONTINUE**



## Шаг 14. Пропускаем, жмем **SAVE AND CONTINUE**

Edit app registration

✓ OAuth consent screen — 2 **Scopes** — 3 Test users — 4 Summary

Scopes express the permissions you request users to authorize for your app and allow your project to access specific types of private user data from their Google Account. [Learn more](#)

ADD OR REMOVE SCOPES

### Your non-sensitive scopes

API ↑	Scope	User-facing description
No rows to display		

### 🔒 Your sensitive scopes

Sensitive scopes are scopes that request access to private user data.

API ↑	Scope	User-facing description
No rows to display		

### 🔒 Your restricted scopes

Restricted scopes are scopes that request access to highly sensitive user data.

API ↑	Scope	User-facing description
No rows to display		

SAVE AND CONTINUE

CANCEL

## Шаг 15. Пропускаем, жмем **SAVE AND CONTINUE**

✓ OAuth consent screen — ✓ Scopes — 3 **Test users** — 4 Summary

### Test users

While publishing status is set to "Testing", only test users are able to access the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. [Learn more](#)

+ ADD USERS

Filter Enter property name or value ?

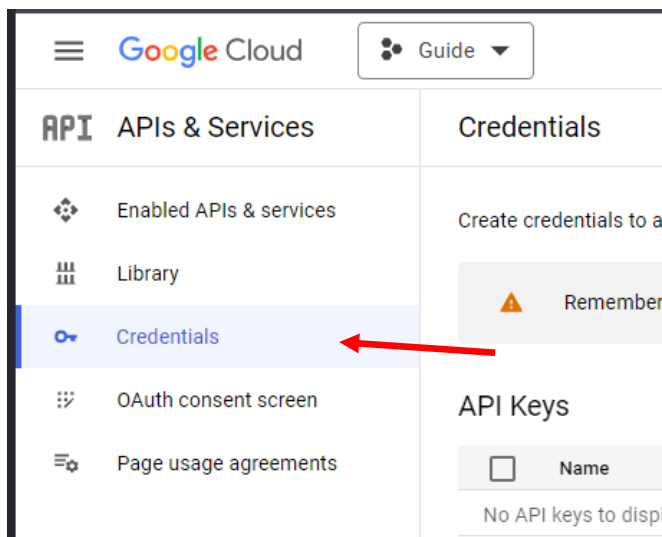
User information

No rows to display

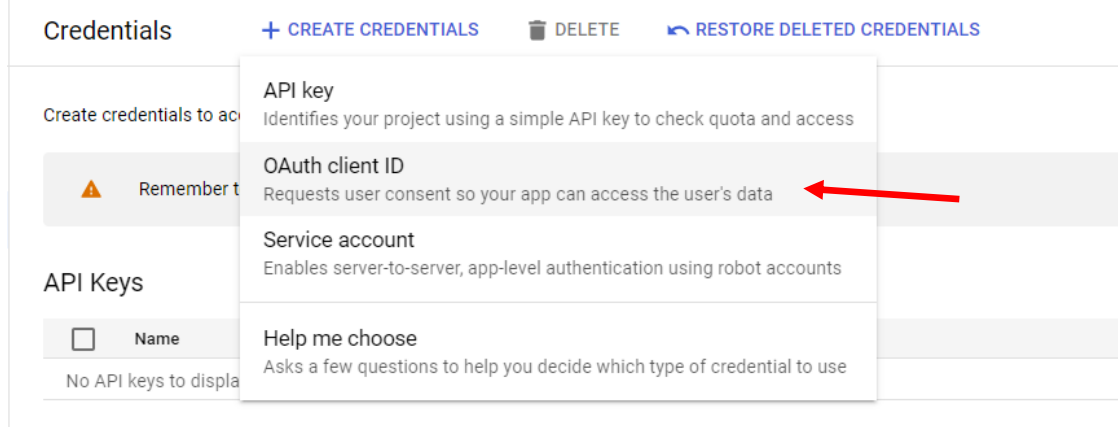
SAVE AND CONTINUE

CANCEL

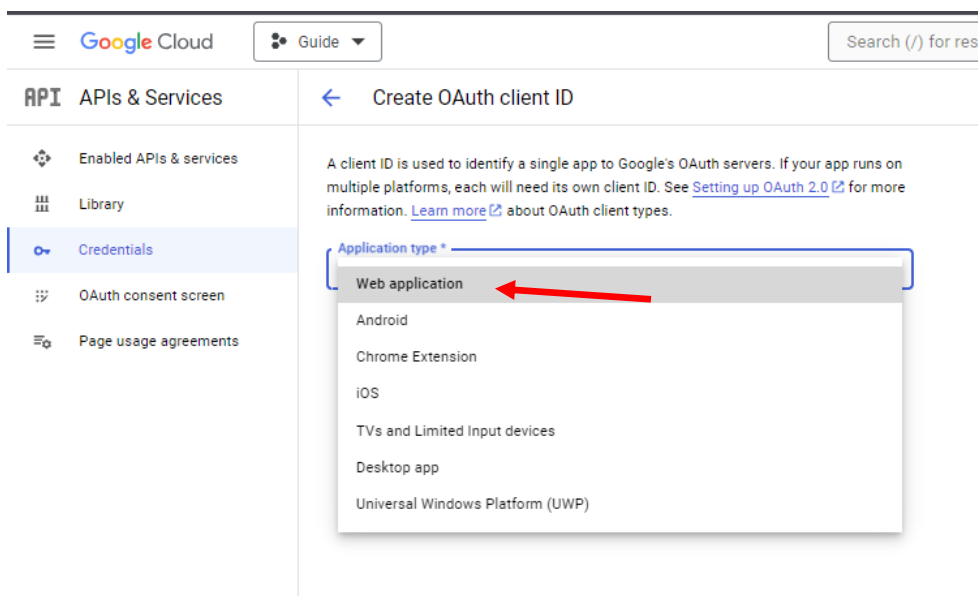
## Шаг 16. Возвращаемся в -> Credentials



## Шаг 17. Снова выбираем -> OAuth client ID



## Шаг 18. Выбираем Web application



## Шаг 19. Заполняем обязательные поля и жмем **CREATE**

Google Cloud

Guide

Search (/) for resources, docs

APIs & Services

Create OAuth client ID

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type \*  
Web application

Name \*  
Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins

For use with requests from a browser

+ ADD URI

Authorized redirect URIs

For use with requests from a web server

+ ADD URI

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

CREATE CANCEL

## Шаг 20. Сохраняем данные об аккаунте в виде *json* файла

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth access is restricted to the [test users](#) listed on your [OAuth consent screen](#)

Client ID

4...

6.app

Client secret

C...

Creation date

A...

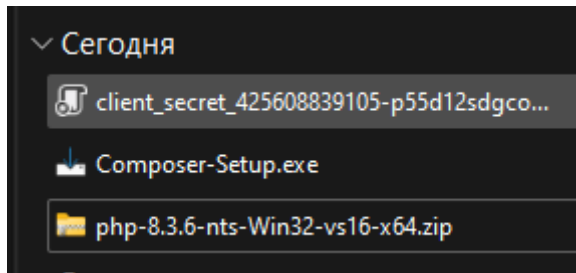
Status

Enabled

DOWNLOAD JSON

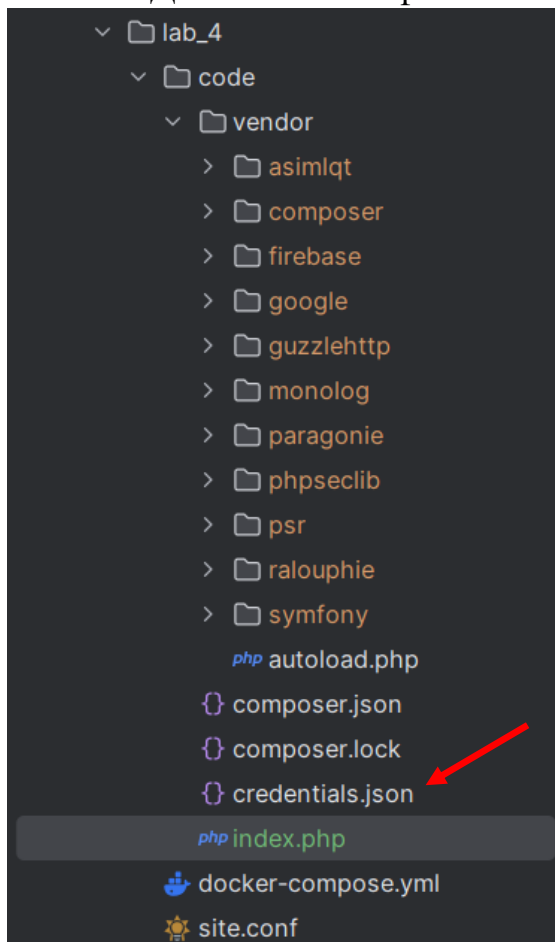
OK

**Шаг 21.** У вас в загрузках появится этот файл



Я переименовал его в *credentials.json*

**Шаг 22.** Добавляем это файл в наш проект:

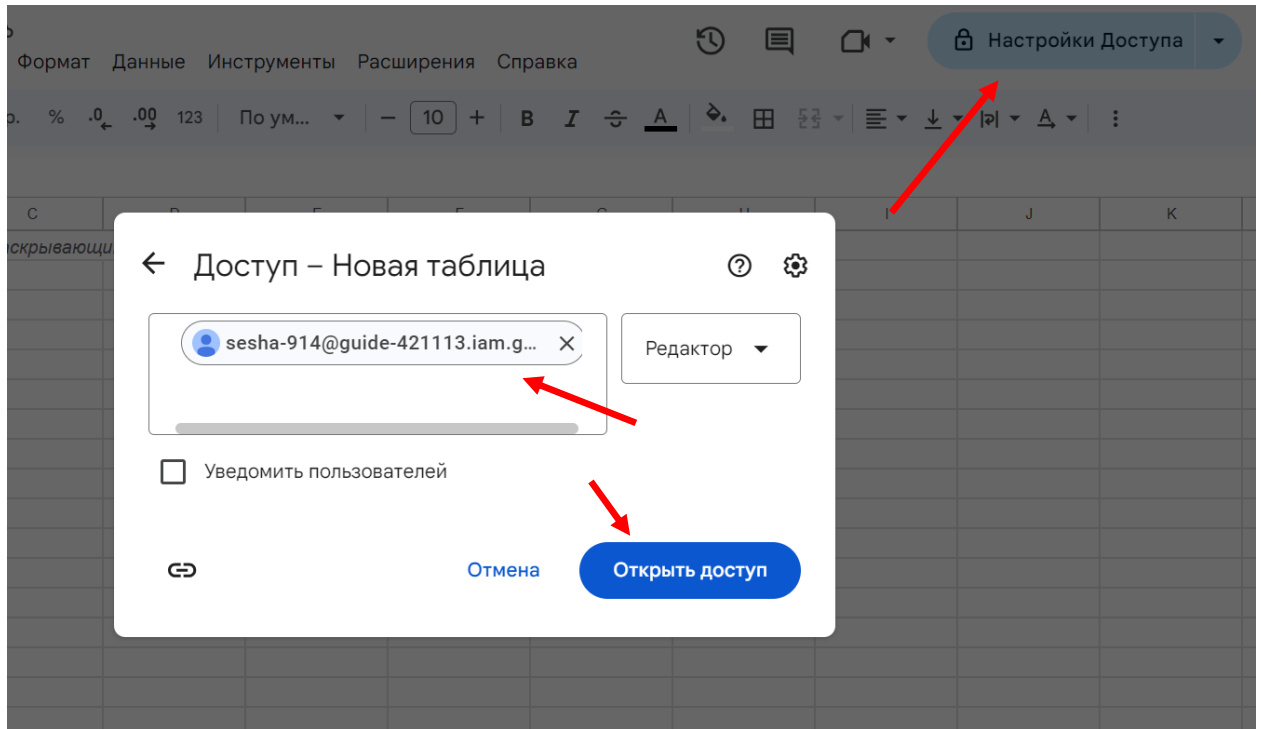


Фух...

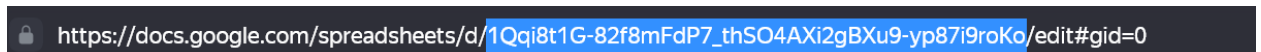
## 5. Google Sheets

**Шаг 1.** Создаем таблицу по [ссылке](#), тут вы уже сами разберетесь

**Шаг 2.** Помните сервисный аккаунт, который мы создали? Теперь нужно предоставить ему доступ к нашей таблице:



**Шаг 3.** А еще сохраните себе id таблицы, чтобы в будущем обращаться к ней



## 6. Работа с Google Sheets

Есть хорошая [статья](#)

Гайд составил <https://t.me/SeshaXD>

Очень рад, если помог вам)