



# **GDG DevFest Porto-Novo 2018**

# Firebase Codelabs



Marnel Gnacadja  
@marnel\_fr

<https://github.com/marnelfr/firebase-php/archive/master.zip>

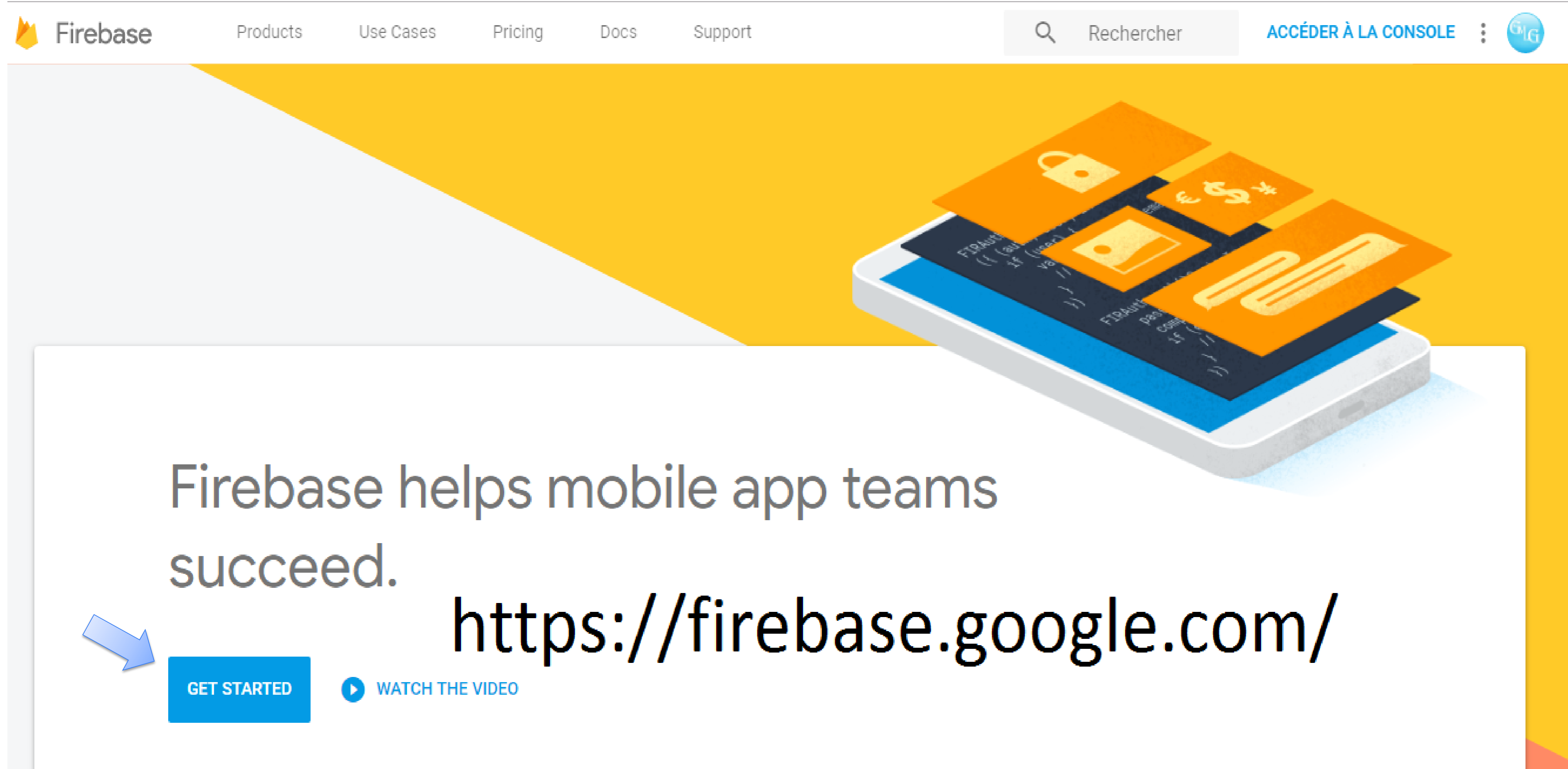
# Configurer Firebase



<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Home page - Firebase

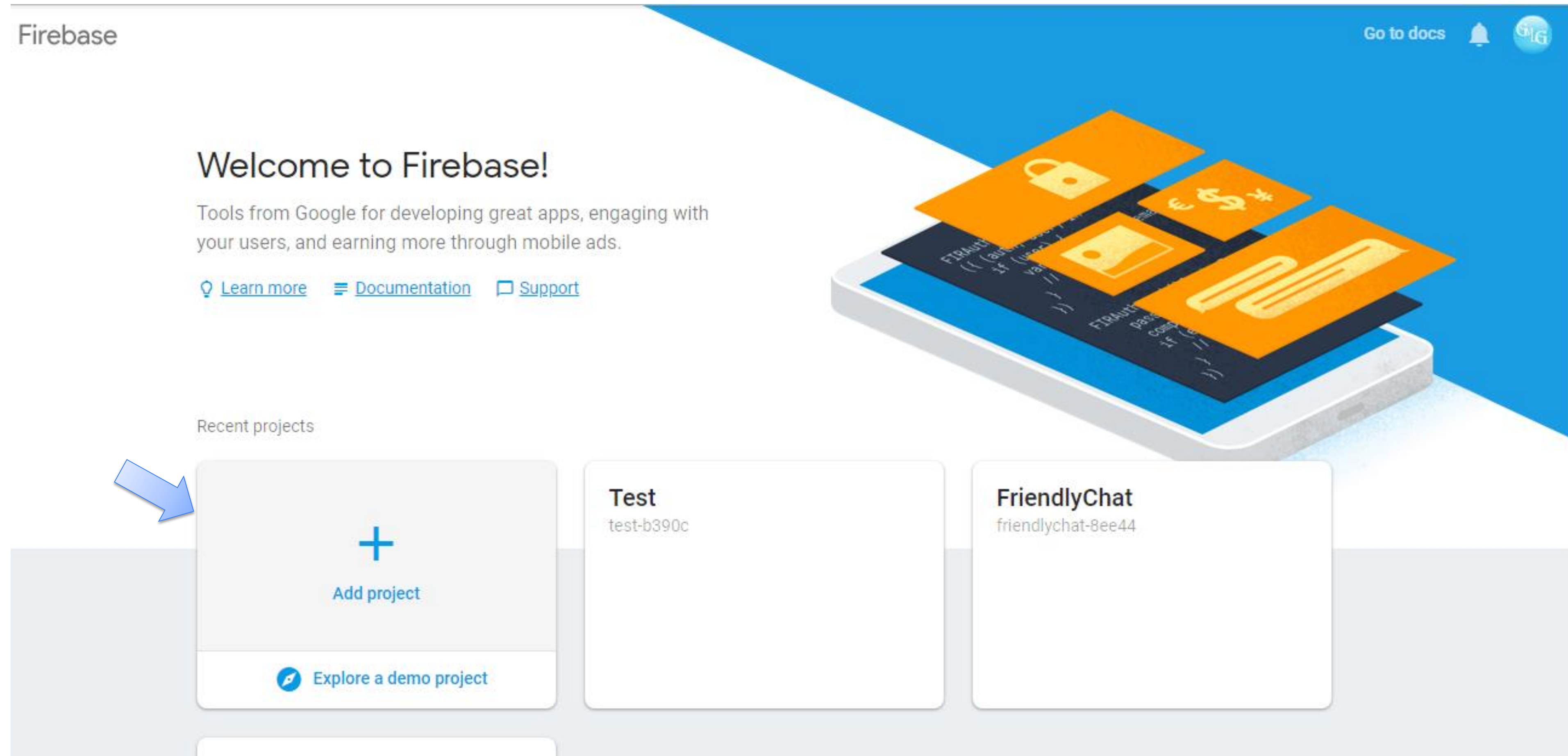


Cliquer sur "**GET STARTED**"

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Ajout de projet - Firebase



Cliquer sur **"Add projet"**

<https://github.com/marnelfr/firebase-php/archive/master.zip>





# Ajout de projet - Firebase

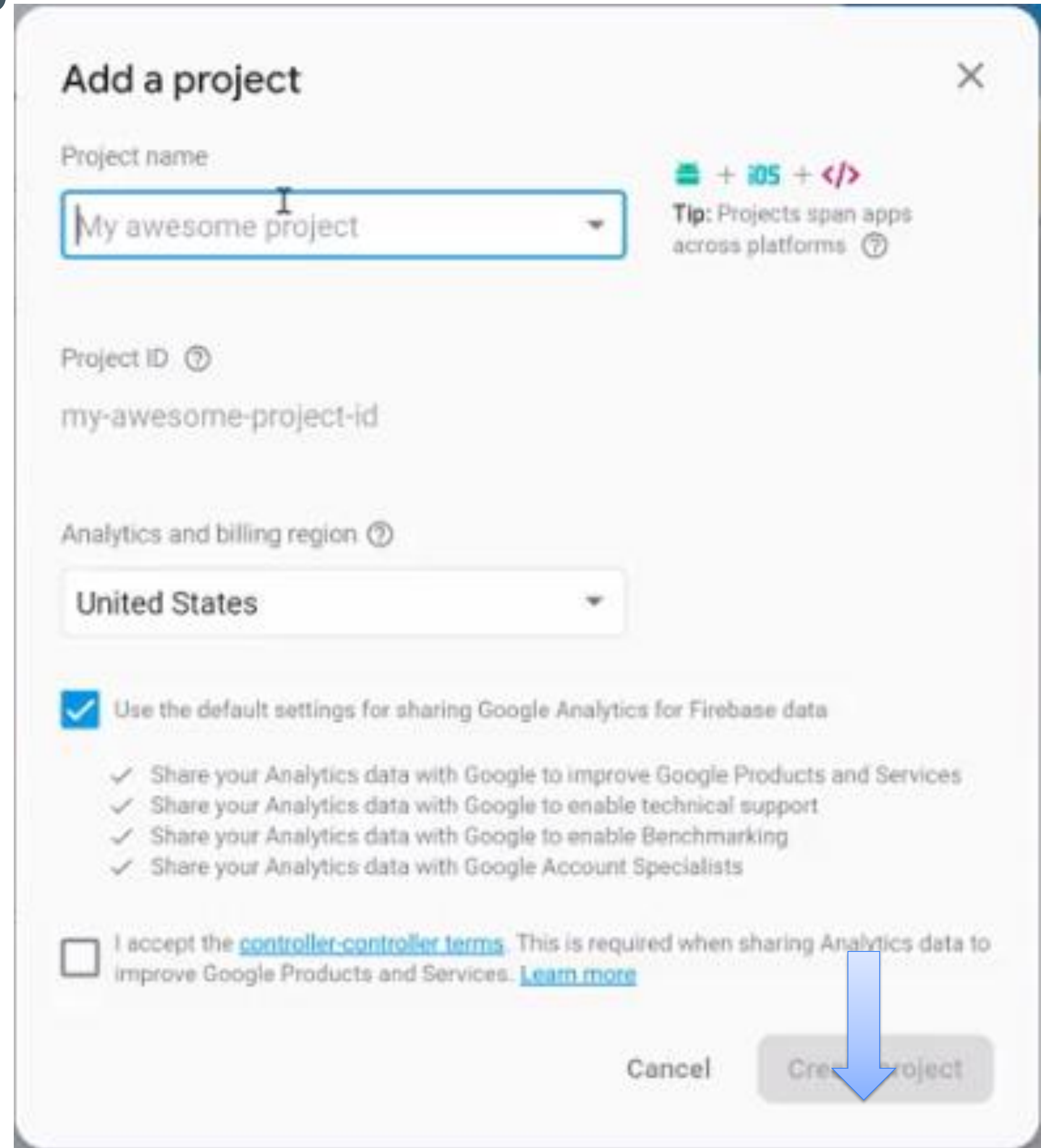
Donner un nom au projet => "**PHP-Firebase**"

Project-ID => "**php-firebase-1234**"

Choisir une région => "**Benin**"

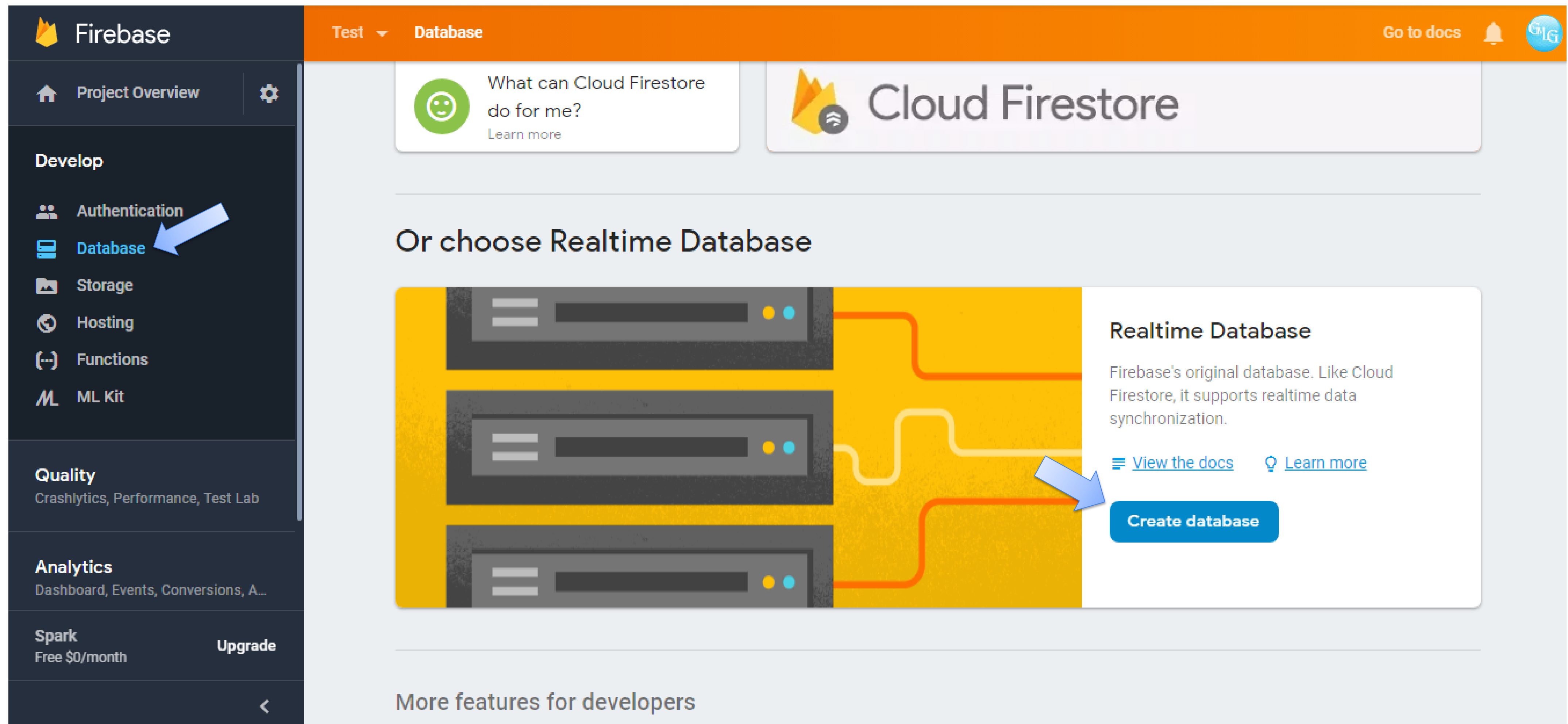
Accepter les termes

Cliquer sur "**Create project**"



The screenshot shows the 'Add a project' dialog box in the Firebase console. It includes a close button (X) in the top right corner. The 'Project name' field contains 'My awesome project' and is highlighted with a blue border. To its right, there are icons for Android, iOS, and a code editor icon, along with a tip: 'Tip: Projects span apps across platforms'. Below the project name is the 'Project ID' field, which contains 'my-awesome-project-id'. Underneath that is the 'Analytics and billing region' dropdown menu, currently set to 'United States'. A checkbox labeled 'Use the default settings for sharing Google Analytics for Firebase data' is checked. Below this, there are four checkmarks indicating the sharing of analytics data for various purposes: improving Google products, enabling technical support, enabling benchmarking, and sharing with account specialists. At the bottom, there is an unchecked checkbox for accepting the 'controller-controller terms', with a link to 'Learn more'. The bottom right corner features a 'Cancel' button and a 'Create project' button, with a large blue arrow pointing down to the 'Create project' button.

# Ajout de base de données- Firebase



The screenshot shows the Firebase console interface. On the left, a dark sidebar contains the 'Database' option, which is highlighted with a blue arrow. The main content area has an orange header with 'Test' and 'Database' tabs. Below the header, there's a section for 'Cloud Firestore' and a section titled 'Or choose Realtime Database'. This section features a graphic of three server racks connected by lines, with a blue arrow pointing to the 'Create database' button. The button is blue with white text. Above the button, there's a description of 'Realtime Database' and links to 'View the docs' and 'Learn more'.

Firestore

What can Cloud Firestore do for me? [Learn more](#)

Cloud Firestore

Or choose Realtime Database

**Realtime Database**

Firestore's original database. Like Cloud Firestore, it supports realtime data synchronization.

[View the docs](#) [Learn more](#)

**Create database**

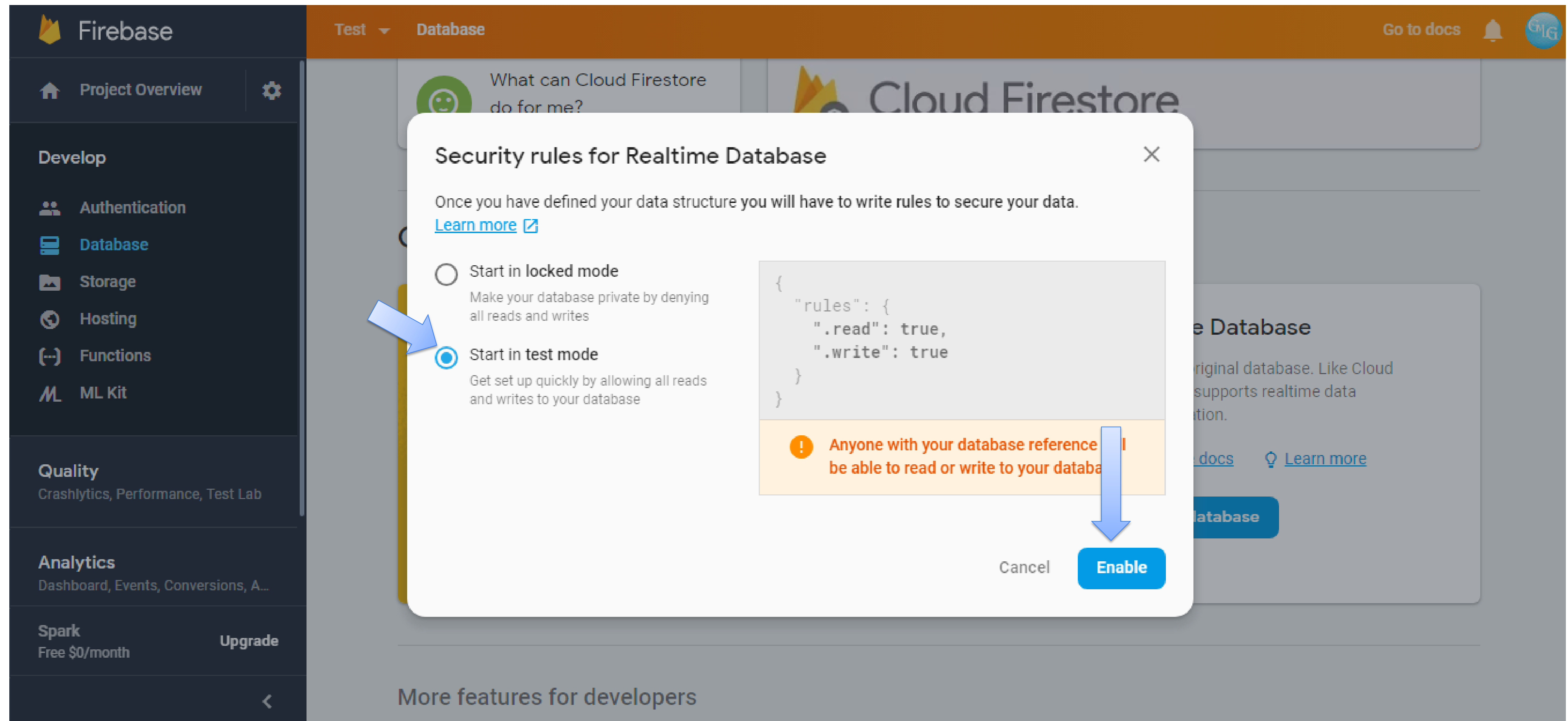
More features for developers

Cliquer sur **Database** => **Create database**

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Ajout de base de données- Firebase



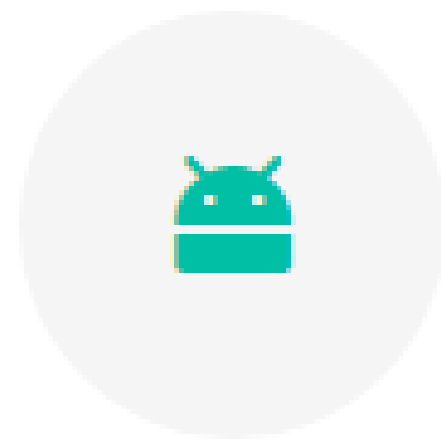
Mettons nous en **test mode** => **Enable**

<https://github.com/marnelfr/firebase-php/archive/master.zip>





# DocOfficiel - Langues supportés



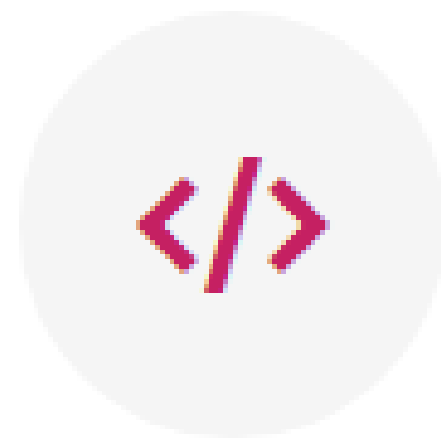
Android guides

Get started now



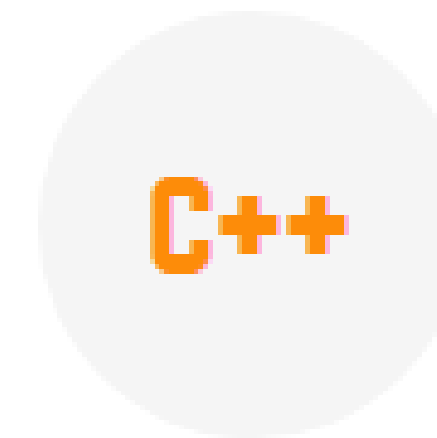
iOS guides

Get started for Swift & Obj-C



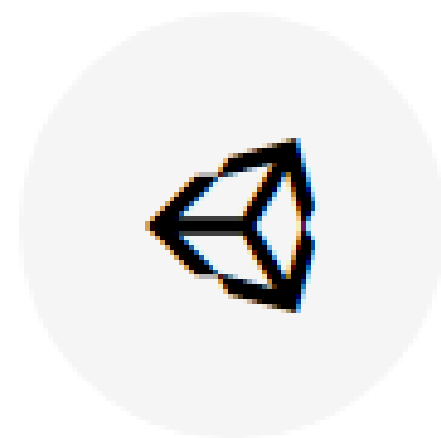
Web guides

Get started for JS Client



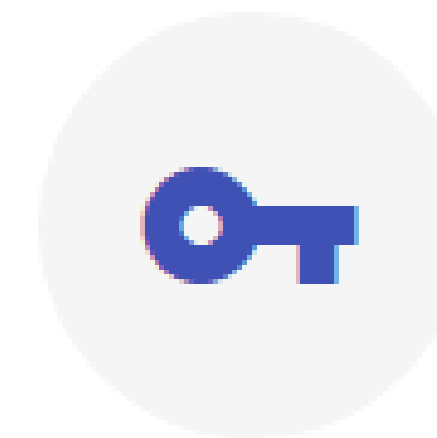
C++ guides

Get started now



Unity guides

Get started now



Admin guides

Get started now

Il n'existe pas de documentation officielle pour PHP

<https://github.com/marnelfr/firebase-php/archive/master.zip>




# Et PHP alors ?



<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Firestore Admin SDK

 **Firestore Admin SDK for PHP**  
latest

Search docs

Overview

Setup

Realtime Database

Authentication

User management

Storage


Remote Config

Cloud Messaging

Troubleshooting

**HIRE DEVELOPERS**  
With Read the Docs

Recruiting developers?  
[Read the Docs can help!](#)

 Read the Docs

v: latest ▾

Docs » Firestore Admin SDK for PHP

 [Edit on GitHub](#)

## Firestore Admin SDK for PHP

Interact with [Google Firestore](#) from your PHP application.

license **MIT** stars **634** downloads **373k** downloads **361/day**

### Note

If you are interested in using the PHP Admin SDK as a client for end-user access (for example, in a web application), as opposed to admin access from a privileged environment (like a server), you should instead follow the [instructions for setting up the client JavaScript SDK](#).

The source code can be found at <https://github.com/kreait/firebase-php/>.

A ready-to-go repository with usage examples can be found at <https://github.com/jeromegamez/firebase-php-examples>.

## User Guide

- [Overview](#)
  - [Requirements](#)

Un package, mon préféré ;) permettant d'utiliser  
firebase avec PHP

<https://github.com/marnelfr/firebase-php/archive/master.zip>



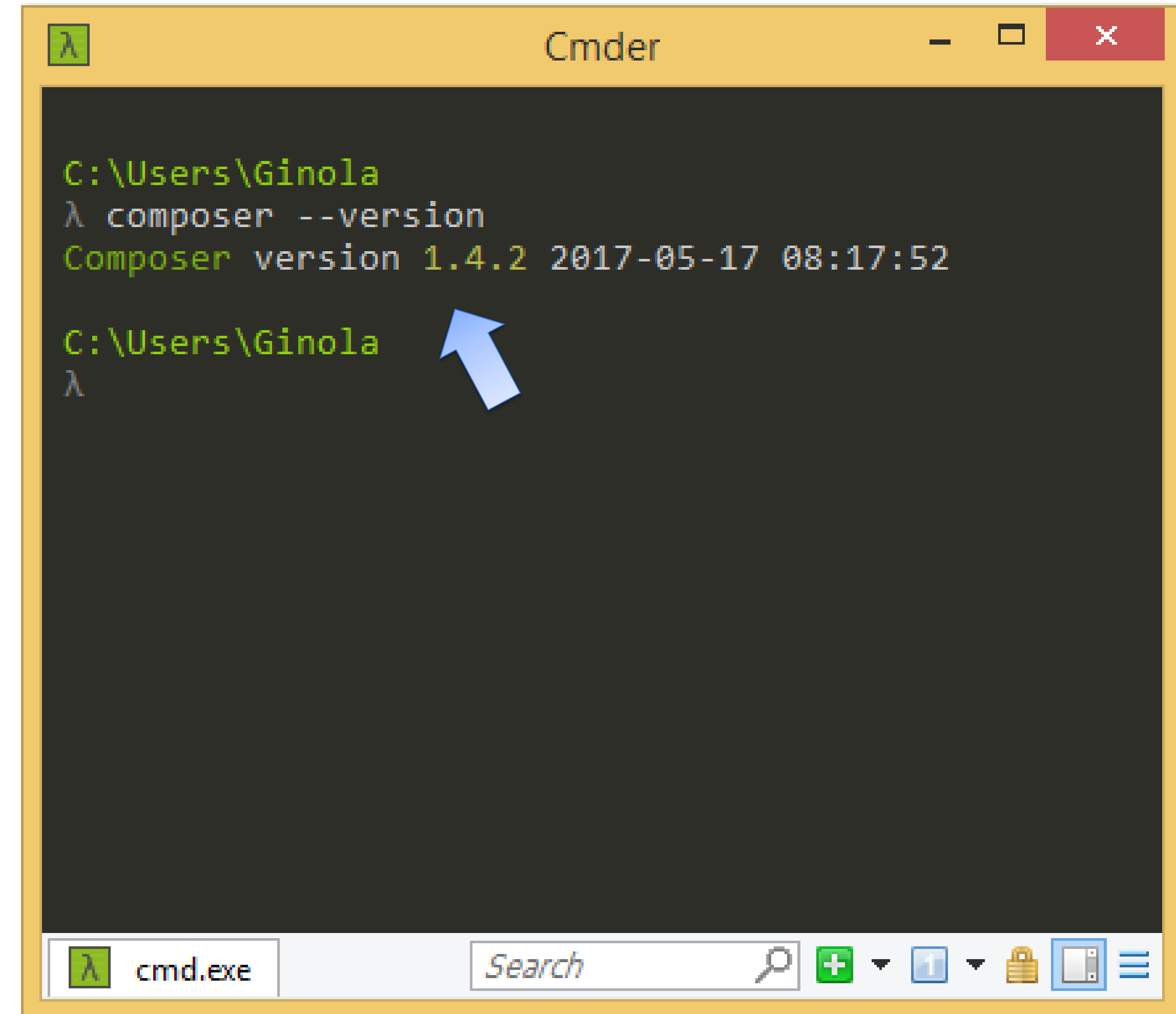
# Outil indispensable

(Enfin, pas vraiment 😊)



Dependency Manager for PHP

= =>



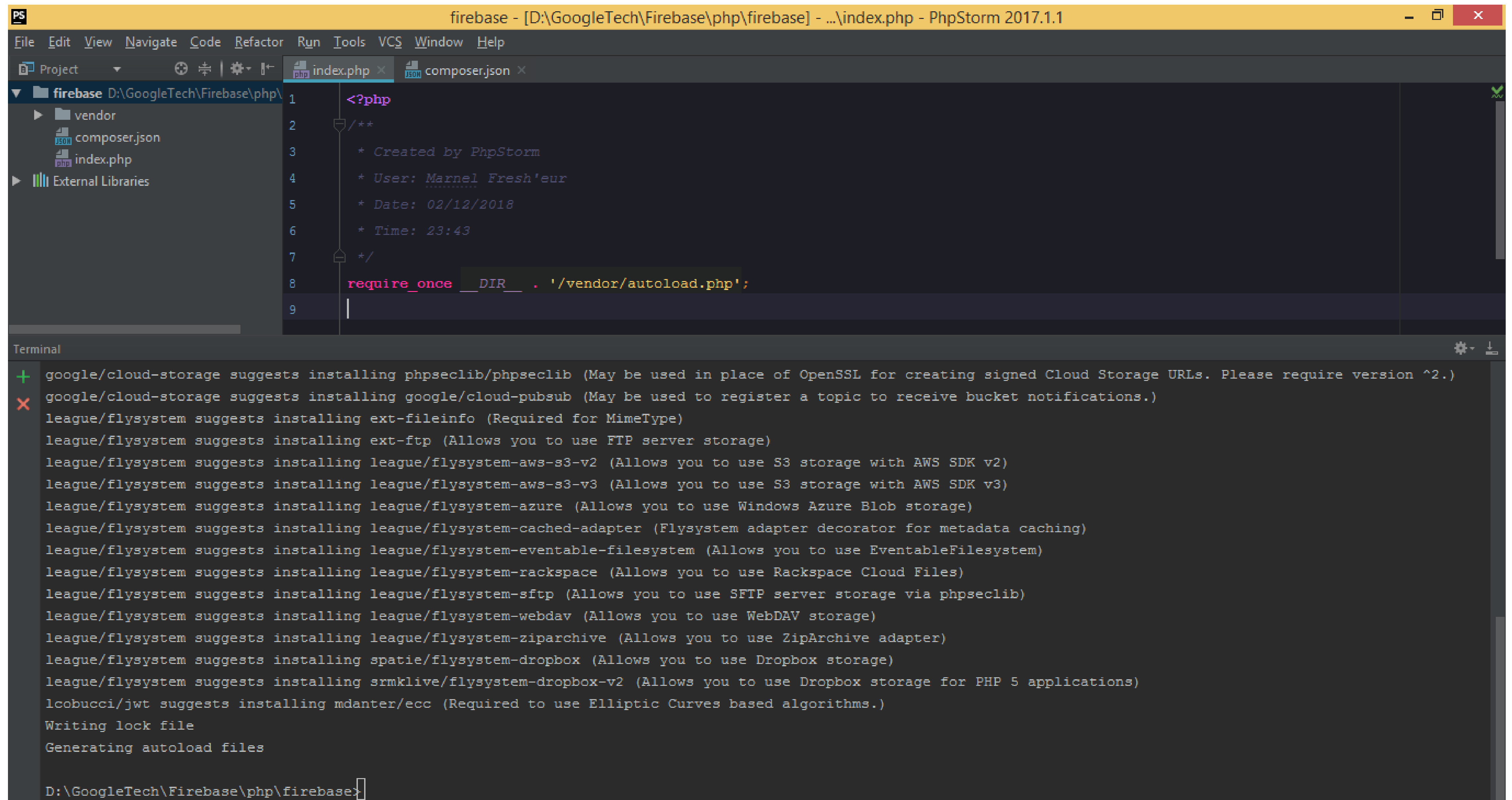
```
C:\Users\Ginola
λ composer --version
Composer version 1.4.2 2017-05-17 08:17:52

C:\Users\Ginola
λ
```

**composer require kreait/firebase-php ^4.18**

<https://github.com/marnelfr/firebase-php/archive/master.zip>





firebase - [D:\GoogleTech\Firebase\php\firebase] - ...\index.php - PhpStorm 2017.1.1

File Edit View Navigate Code Refactor Run Tools VCS Window Help

Project index.php x composer.json x

firebase D:\GoogleTech\Firebase\php\

- vendor
  - composer.json
  - index.php
- External Libraries

```
1 <?php
2 /**
3  * Created by PhpStorm
4  * User: Marnel Fresh'eur
5  * Date: 02/12/2018
6  * Time: 23:43
7  */
8 require_once __DIR__ . '/vendor/autoload.php';
9
```

Terminal

```
+ google/cloud-storage suggests installing phpseclib/phpseclib (May be used in place of OpenSSL for creating signed Cloud Storage URLs. Please require version ^2.)
X google/cloud-storage suggests installing google/cloud-pubsub (May be used to register a topic to receive bucket notifications.)
league/flysystem suggests installing ext-fileinfo (Required for MimeType)
league/flysystem suggests installing ext-ftp (Allows you to use FTP server storage)
league/flysystem suggests installing league/flysystem-aws-s3-v2 (Allows you to use S3 storage with AWS SDK v2)
league/flysystem suggests installing league/flysystem-aws-s3-v3 (Allows you to use S3 storage with AWS SDK v3)
league/flysystem suggests installing league/flysystem-azure (Allows you to use Windows Azure Blob storage)
league/flysystem suggests installing league/flysystem-cached-adapter (Flysystem adapter decorator for metadata caching)
league/flysystem suggests installing league/flysystem-eventable-filesystem (Allows you to use EventableFilesystem)
league/flysystem suggests installing league/flysystem-rackspace (Allows you to use Rackspace Cloud Files)
league/flysystem suggests installing league/flysystem-sftp (Allows you to use SFTP server storage via phpseclib)
league/flysystem suggests installing league/flysystem-webdav (Allows you to use WebDAV storage)
league/flysystem suggests installing league/flysystem-ziparchive (Allows you to use ZipArchive adapter)
league/flysystem suggests installing spatie/flysystem-dropbox (Allows you to use Dropbox storage)
league/flysystem suggests installing srmklive/flysystem-dropbox-v2 (Allows you to use Dropbox storage for PHP 5 applications)
lcobucci/jwt suggests installing mdanter/ecc (Required to use Elliptic Curves based algorithms.)
Writing lock file
Generating autoload files
D:\GoogleTech\Firebase\php\firebase>
```

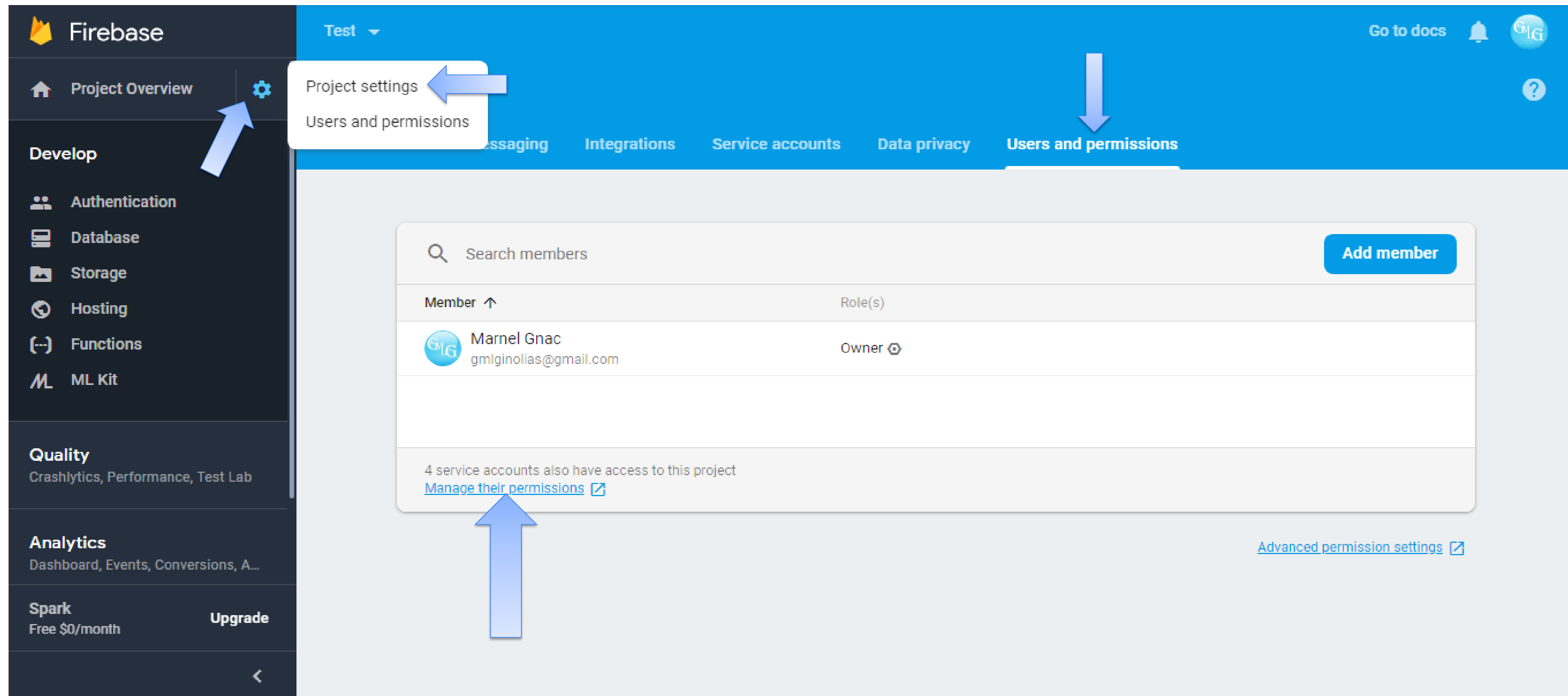
## Installation de krait/firebase-php

<https://github.com/marnelfr/firebase-php/archive/master.zip>





# Obtenir les identifiants à notre base de données

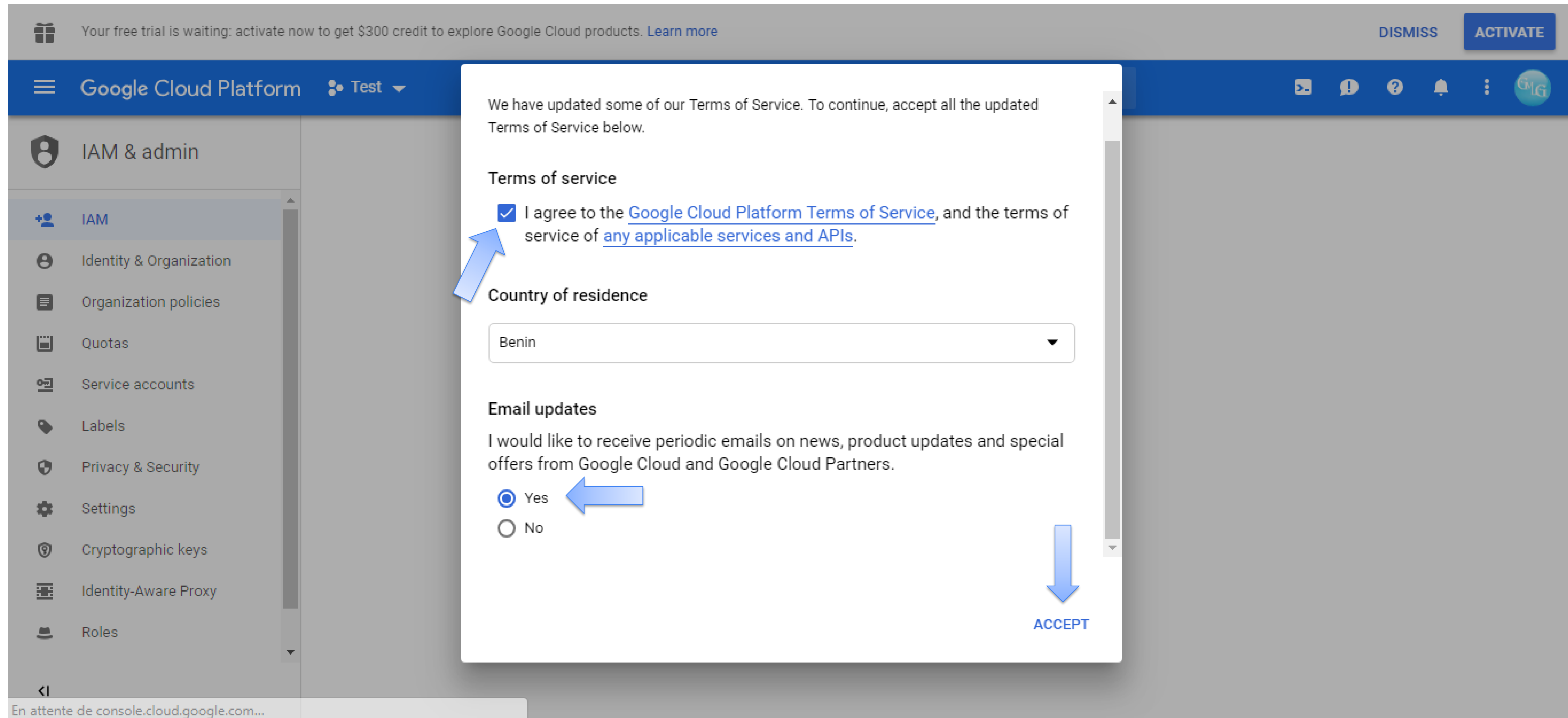


Cliquer sur  => "Project settings" => "Users and permissions" => "Manage their permissions"

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Un nouveau onglet ? 🤔



The screenshot shows the Google Cloud Platform console interface. At the top, there is a banner for a free trial. The left sidebar contains navigation links for IAM & admin, IAM, Identity & Organization, Organization policies, Quotas, Service accounts, Labels, Privacy & Security, Settings, Cryptographic keys, Identity-Aware Proxy, and Roles. The main content area displays a modal dialog for accepting the Terms of Service. The dialog includes a checkbox for agreeing to the terms, a dropdown menu for the country of residence (set to Benin), and radio buttons for email updates (set to Yes). A blue arrow points to the 'ACCEPT' button at the bottom right of the dialog.

Your free trial is waiting: activate now to get \$300 credit to explore Google Cloud products. [Learn more](#) DISMISS ACTIVATE

Google Cloud Platform Test

IAM & admin

- IAM
- Identity & Organization
- Organization policies
- Quotas
- Service accounts
- Labels
- Privacy & Security
- Settings
- Cryptographic keys
- Identity-Aware Proxy
- Roles

We have updated some of our Terms of Service. To continue, accept all the updated Terms of Service below.

**Terms of service**

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

**Country of residence**

Benin

**Email updates**

I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners.

☒ Yes ☐ No

[ACCEPT](#)

En attente de console.cloud.google.com...

Oui!! Ici, on va accepter **termes d'usage**, dire si on **veut recevoir des annonces** de nouveauté, puis cliquer sur => "**Accept**"

<https://github.com/marnelfr/firebase-php/archive/master.zip>




# Un nouveau onglet ?

Google Cloud Platform

Service accounts

Service accounts for project "Test"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more](#)

	Email	Name	Description	Key ID	Key creation date	Actions
<input type="checkbox"/>	test-b390c@appspot.gserviceaccount.com	App Engine default service account		No keys		
<input type="checkbox"/>	firebase-adminsdk-ha51u@test-b390c.iam.gserviceaccount.com	firebase-adminsdk	Firebase Admin SDK Service Agent	No keys		

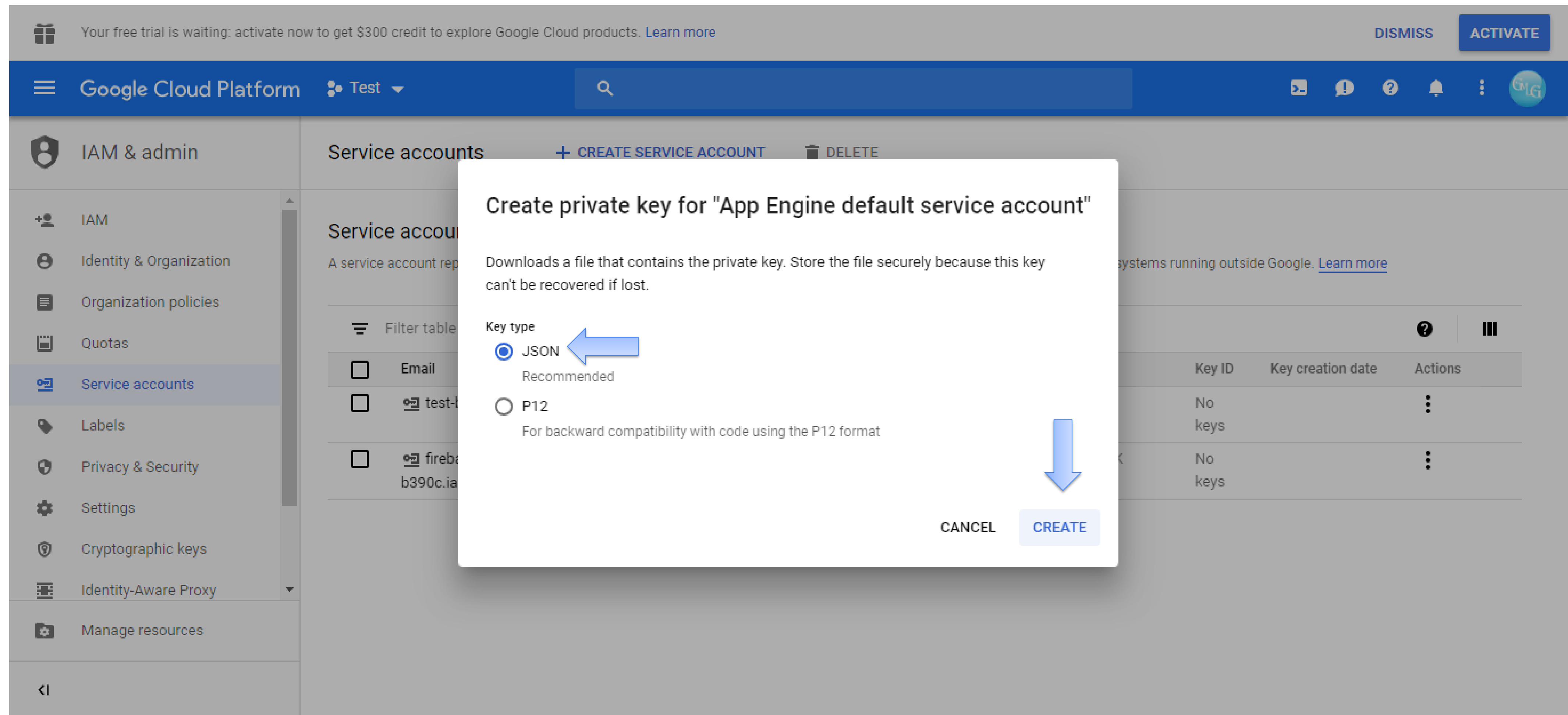
Edit  
Delete  
Create key

Dans l'onglet, cliquer sur "Service accounts" =>  => "Create key"

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Un nouveau onglet ? 🤔



Choisir le format "Json"  
Cliquer sur  
**"Create"**

==>

Génère et télécharge un fichier json  
de nom semblable à  
**test-b390c-a5cf2394a813.json**



# Du code, du code!!!!



<https://github.com/marnelfr/firebase-php/archive/master.zip>





# Connexion à la base de données

```
8
9  require_once __DIR__ . '/vendor/autoload.php';
10
11  use Krait\Firebase\Factory;
12  use Krait\Firebase\ServiceAccount;
13
14  $serviceAccount = ServiceAccount::fromJsonFile( filePath: __DIR__ . '/key/test-b390c-a5cf2394a813.json' );
15
16  $firebase = (new Factory)
17      ->withServiceAccount($serviceAccount)
18      ->create();
19
20  $database = $firebase->getDatabase();
21
22  var_dump($database);
```



Je m'en fou! J'ai déjà mon fichier **Json**

Grâce au fichier **json**, on peut simplement utiliser la méthode statique **fromJsonFile** de la classe **ServiceAccount**

Ceci va nous donner son instance dont à besoin la classe **Factory** pour établir la connexion à la base de données grâce à la méthode **getDatabase**

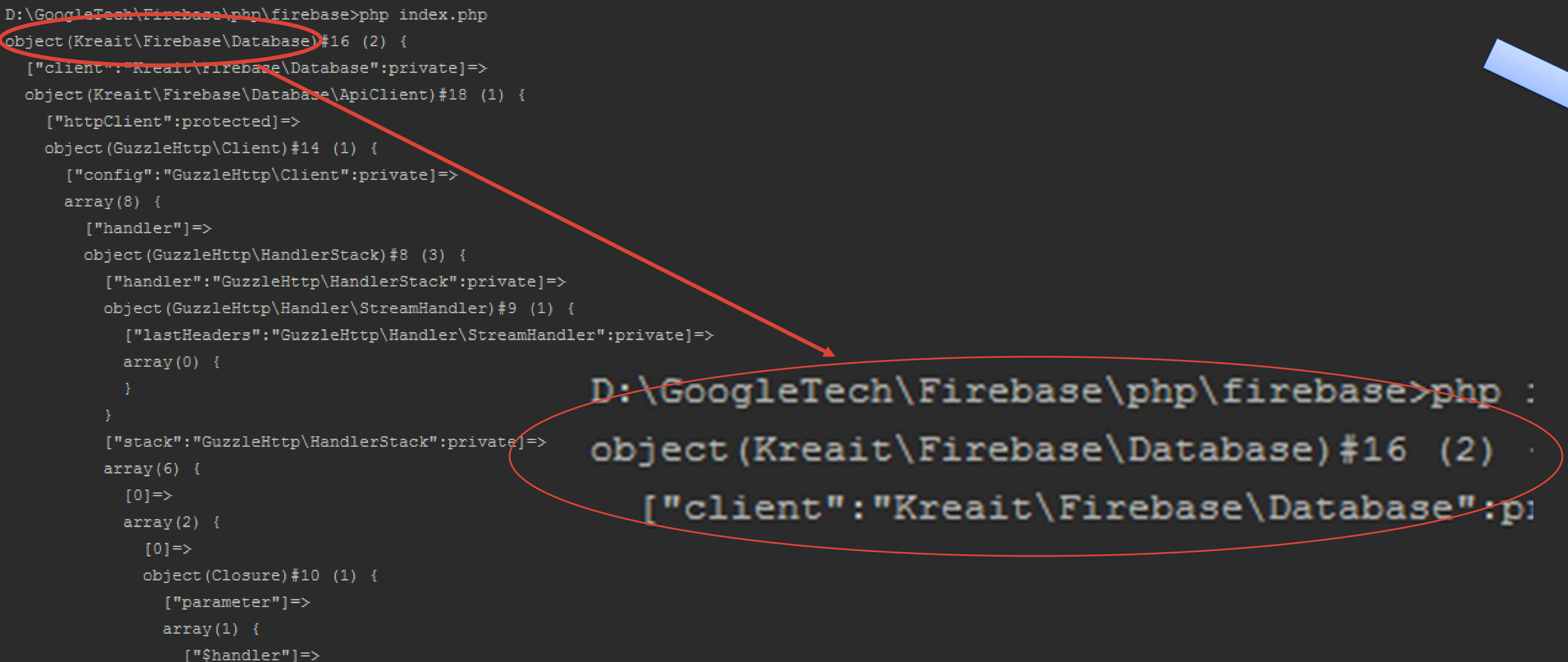

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Connexion à la base de données

```
D:\GoogleTech\Firebase\php\firebase>php index.php
object(Kreait\Firebase\Database)#16 (2) {
    ["client":Kreait\Firebase\Database:private]=>
    object(Kreait\Firebase\Database\ApiClient)#18 (1) {
        ["httpClient":protected]=>
        object(GuzzleHttp\Client)#14 (1) {
            ["config":GuzzleHttp\Client:private]=>
            array(8) {
                ["handler"]=>
                object(GuzzleHttp\HandlerStack)#8 (3) {
                    ["handler":GuzzleHttp\HandlerStack:private]=>
                    object(GuzzleHttp\Handler\StreamHandler)#9 (1) {
                        ["lastHeaders":GuzzleHttp\Handler\StreamHandler:private]=>
                        array(0) {
                        }
                    }
                }
            }
        }
    }
    ["stack":GuzzleHttp\HandlerStack:private]=>
    array(6) {
        [0]=>
        array(2) {
            [0]=>
            object(Closure)#10 (1) {
                ["parameter"]=>
                array(1) {
                    ["$handler"]=>

```



On se retrouve donc avec un objet de la classe  
**Kreait/Firebase/Database**  
Contenant notre connexion à la base de données



# Insertion en base de données

```
20 $database = $firebase->getDatabase();
21
22 $newPost = $database->getReference( path: 'blog/post')
23     ->push([
24         'title' => 'Mon premier post',
25         'content' => 'Ceci est mon premier post dans une base de données firebase ;)'
26     ]);
27
```

**getReference** permet de définir une référence, un chemin d'accès aux données dans la base de données.

Un peu comme les collections dans mongoDB



Facon c'est facile

**Push**, comme si on avait à faire à un simple tableau PHP, permet de rajouter des enregistrement à notre collection

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Affichage des données dans la console



Mon poste !!



Notre poste est bien enregistrée en base de données et trouvable dans la **console firabase**

<https://github.com/marnelfr/firebase-php/archive/master.zip>





# Récupération de données de la base de données

```
32 $reference = $database->getReference( path: 'blog/post' );
33 $snapshot = $reference->getSnapshot();
34 $value = $snapshot->getValue();
35
36 var_dump($value);
```

**getSnapshot** récupère les données contenu dans une référence de la base de données au moment où elle est appelé

Un peut comme notre

**select \* from blog.post**

**getValue** retourne simplement son contenu

<https://github.com/marnelfr/firebase-php/archive/master.zip>





# Recuperation de données de la base de données

```
D:\GoogleTech\Firebase\php\firebase>php index.php
array(2) {
  ["-LSq0TES8bpiKpSkA-2F"]=>
  array(2) {
    ["content"]=>
    string(63) "Ceci est mon premier post dans une base de données firebase ;)"
    ["title"]=>
    string(16) "Mon premier post"
  }
  ["-LSq7Yirdb7f88D59q4c"]=>
  array(2) {
    ["content"]=>
    string(65) "Ceci est mon deuxième post dans une base de données firebase :D"
    ["title"]=>
    string(18) "Mon deuxième post"
  }
}

D:\GoogleTech\Firebase\php\firebase>
```

Affichage de contenu de notre  
variable **\$value**

<https://github.com/marnelfr/firebase-php/archive/master.zip>



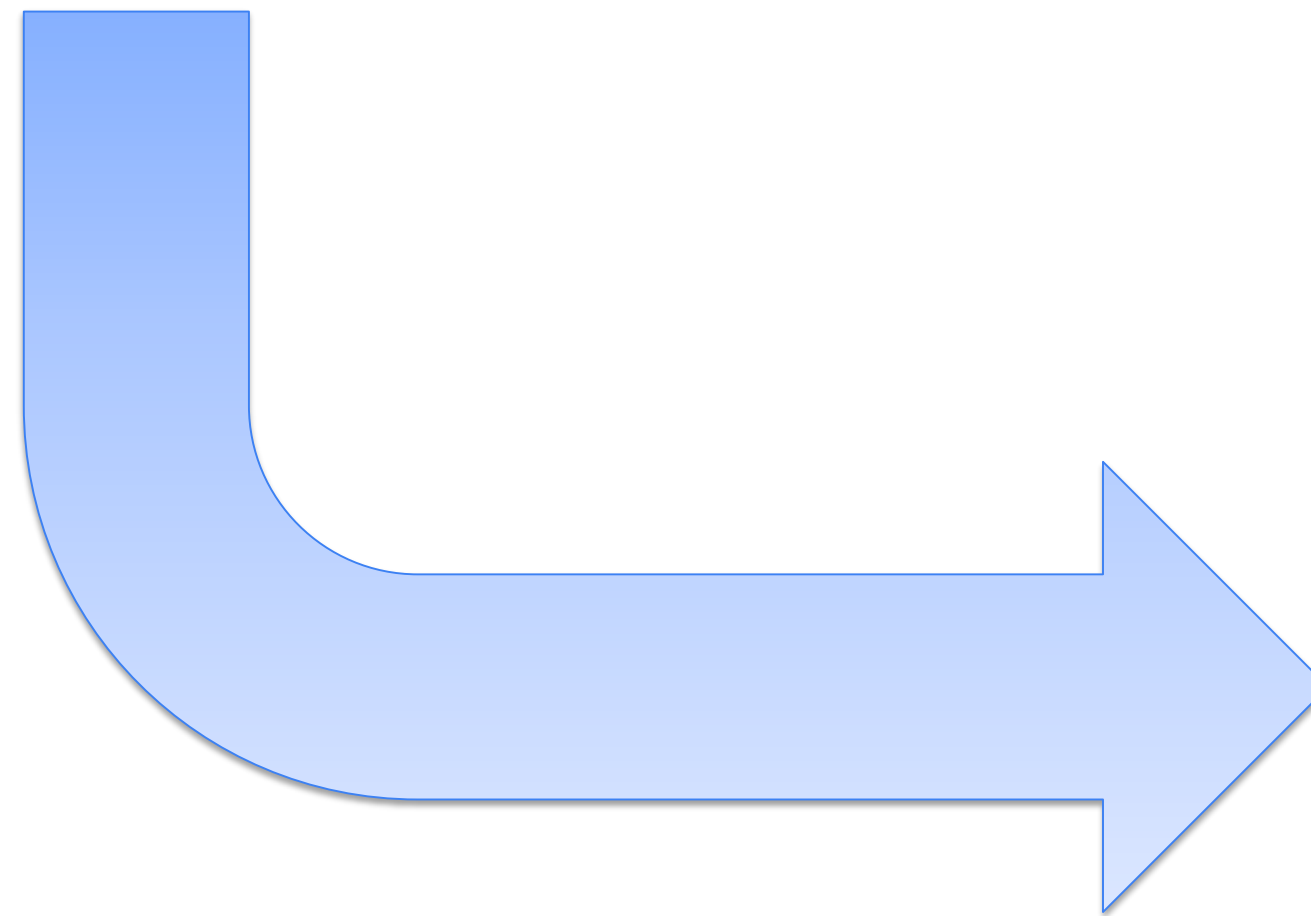
# Quelques methods de snapshot

- **exists()** renvoie **vrai** si le snapshot contient des données donc n'est pas vide.
- **getChild()** revoie le snapshot d'une sous-reference.
- **getReference()** renvoie à la reference mere de la reference sur laquelle on l'a appelé.
- **getValue()** renvoie la donnée contenu dans le snapshot.
- **hasChild()** renvoie **vrai** si la sous-reference contient des données.
- **hasChildren()** renvoie **vrai** si le snapshot à 'des' propriétés enfants.
- **numChildren()** renvoie le nombre de propriété enfant, un peu notre count() ici.



# Interroger la base de données

```
46 $data = $database->getReference( path: 'personne')
47     ->orderByChild( path: 'taille')
48     ->startAt( value: 1.50)
49     ->endAt( value: 3.10)
50     ->limitToFirst( limit: 25)
51     ->getSnapshot();
52
```



**Select \***  
**From** personne  
**Where** taille >= 1,50  
**And** taille <= 3,10  
**Order by** taille  
**Limit** 25



- Utilisant Git:

```
git clone https://github.com/marnelfr/firebase-php.git
```

- Téléchargement direct:

```
https://github.com/marnelfr/firebase-php/archive/master.zip
```

<https://github.com/marnelfr/firebase-php/archive/master.zip>



# Firestore sous tout force



- Utilisant Git:

```
git clone https://github.com/firebase/friendlychat-web
```

- Téléchargement direct:

```
https://github.com/firebase/friendlychat-web/archive/master.zip
```

- Documentation:

```
https://firebase.google.com/docs/web/setup
```





# Débrouillez-vous !!!



Documentation:

<https://console.firebase.google.com/>

<https://firebase-php.readthedocs.io/en/latest/>

<https://www.gograph.com/clipart/teasing-tongue-dumb-emoji-gg84007627.html>

[\*\*https://github.com/marnelfr/firebase-php/archive/master.zip\*\*](https://github.com/marnelfr/firebase-php/archive/master.zip)



*Merçi!*



Marnel Gnacadja  
@marnel\_fr