

Firestore, Précis et concis

Table des matières

Connexion avec un compte Google et accès à la console	2
Création de projet Firebase	2
Authentification (users)	3
Firestore (base de données).....	6
Création d'une database	6
Création de collection.....	7
Rules.....	8
Playground.....	9
Cloud Storage (upload de fichiers).....	9
Upload de fichiers.....	9
Rendre les fichiers accessibles publiquement via une url	10
Ajout de Firebase à une application	13
Avec une application Web	13
Firestore CLI	15
Projet client (React)	16
Version 8	16
Configuration	16
Auth.....	17
Firestore.....	20
Storage, upload de fichier	22
Hosting.....	23
Tester son projet avant de le deployer.....	24
Rollback.....	25
Ajouter un nom de domaine personnalisé	25
Rewrites.....	27

Connexion avec un compte Google et accès à la console

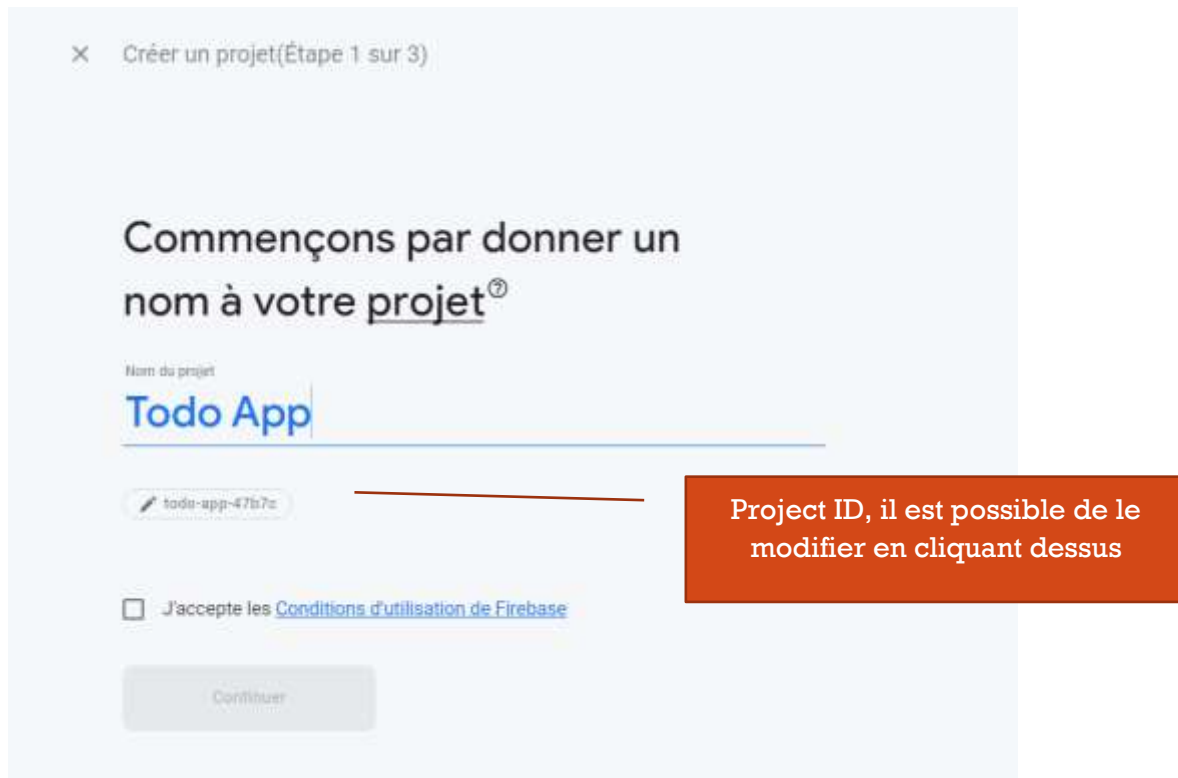
Se rendre à la page <https://firebase.google.com/>. Cliquer sur le lien « connexion » et se connecter avec un compte Google.

Cliquer sur « accéder à la console » <https://console.firebase.google.com/>

Création de projet Firebase

Cliquer sur le bouton « Créer un projet »

Etape 1 : donner un nom au projet



× Créer un projet(Étape 1 sur 3)

Commençons par donner un nom à votre projet[?]

Nom du projet

Todo App

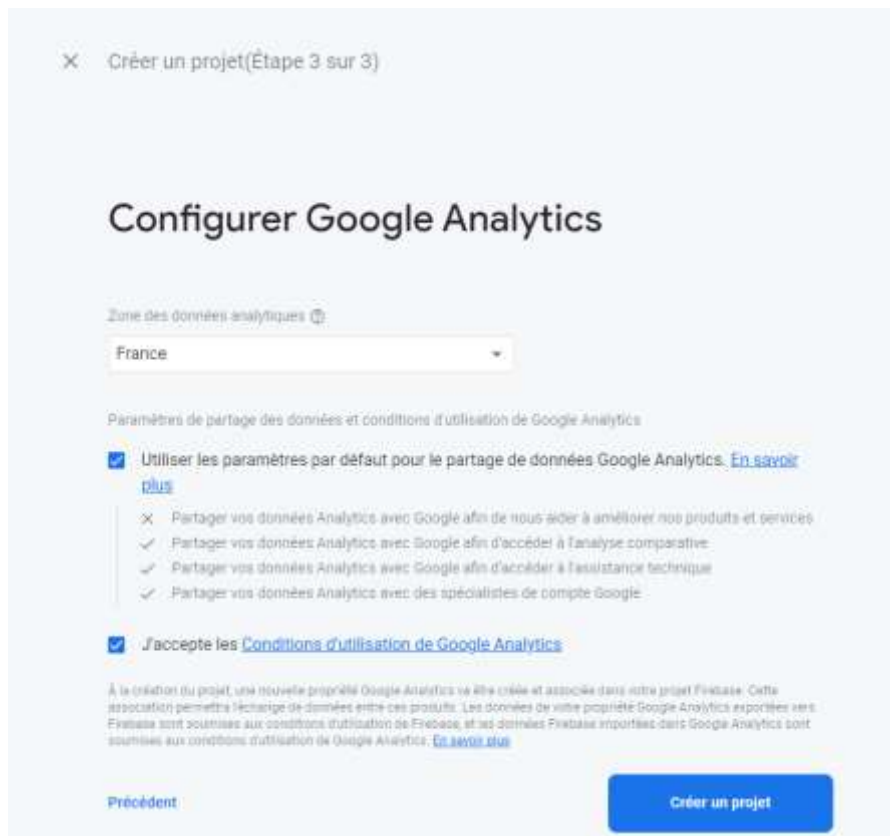
todo-app-47b7c

J'accepte les [Conditions d'utilisation de Firebase](#)

Continuer

Project ID, il est possible de le modifier en cliquant dessus

... Etapes suivantes : Google Analytics

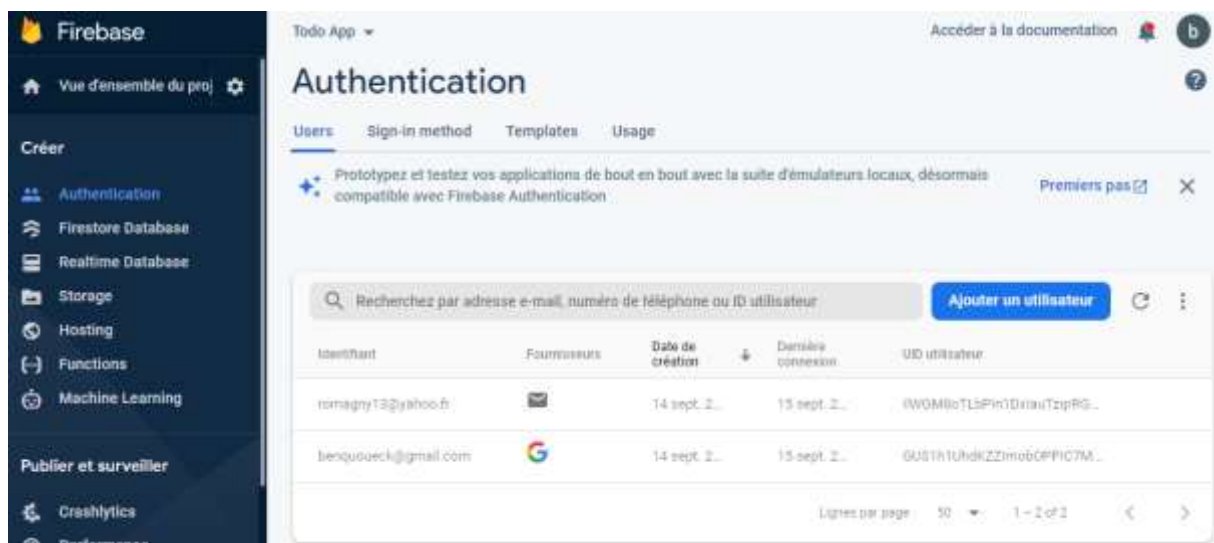


.. Puis cliquer sur « créer un projet »

Authentification (users)

Firebase permet de gérer facilement les utilisateurs : signin, signup, forgot password, etc.

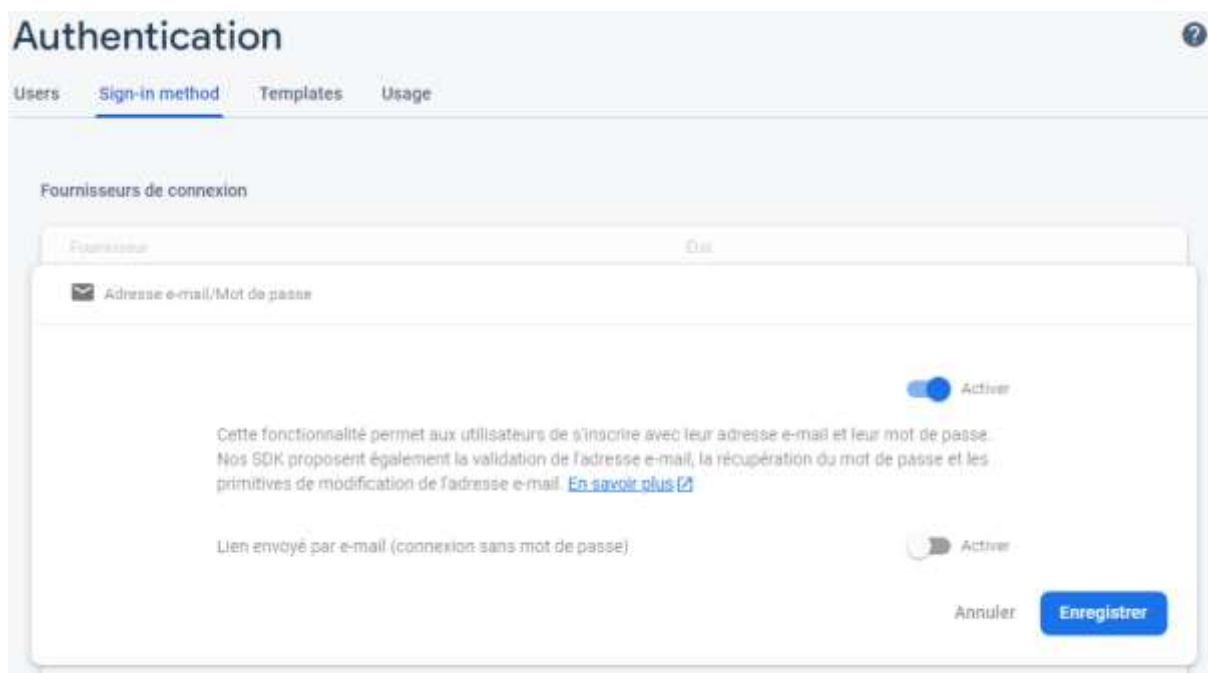
Les utilisateurs créés sont listés. On peut également voir leur « **uid** » qui peut être utilisé pour les collections.



Plusieurs possibilités :

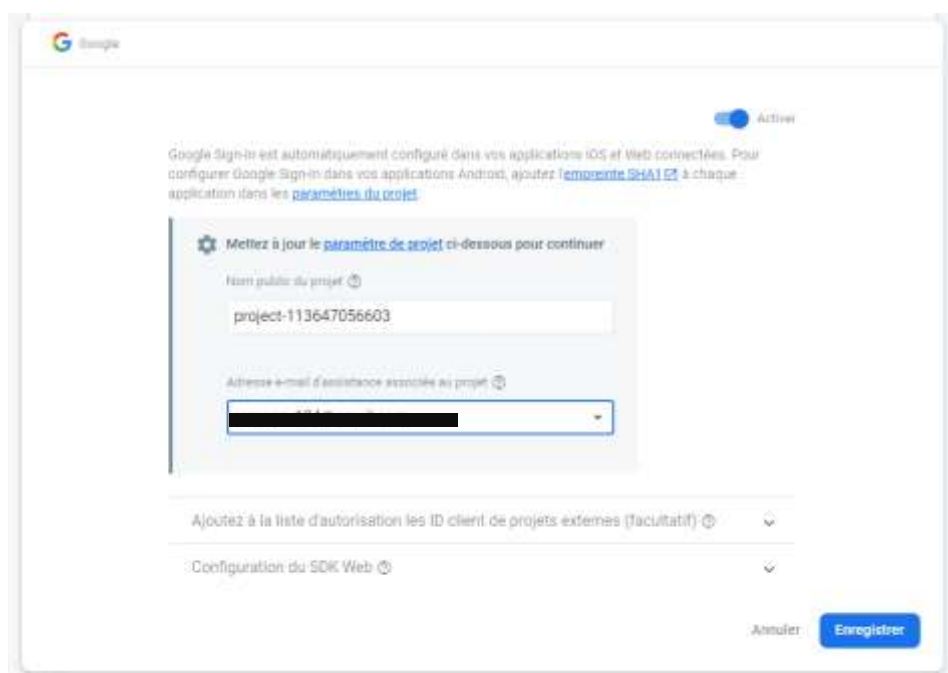
- Par email et password (provider : « password »)
- Par téléphone
- **OAuth providers** (provider: « Google », « Facebook », etc.)

Par email et password : Onglet « Authentication »... « sign-in method » ... « Adresse e-mail/Mot de passe »



The screenshot shows the 'Authentication' settings page with the 'Sign-in method' tab selected. Under the heading 'Fournisseurs de connexion', there is a table with two columns: 'Fournisseur' and 'État'. The first row is 'Adresse e-mail/Mot de passe', which is currently active (toggle switch is blue). Below this row, there is a description: 'Cette fonctionnalité permet aux utilisateurs de s'inscrire avec leur adresse e-mail et leur mot de passe. Nos SDK proposent également la validation de l'adresse e-mail, la récupération du mot de passe et les primitives de modification de l'adresse e-mail. [En savoir plus](#)'. Below this, there is another option 'Lien envoyé par e-mail (connexion sans mot de passe)' which is currently inactive (toggle switch is grey). At the bottom right, there are 'Annuler' and 'Enregistrer' buttons.

OAuth providers : Exemple Google



The screenshot shows the Google Sign-in configuration page. At the top, there is a 'Google' logo and a toggle switch labeled 'Activer' which is turned on. Below this, there is a paragraph of text: 'Google Sign-in est automatiquement configuré dans vos applications iOS et Web connectées. Pour configurer Google Sign-in dans vos applications Android, ajoutez le [paramètre SHA1](#) à chaque application dans les [paramètres du projet](#)'. Below this, there is a section titled 'Mettez à jour le paramètre de projet ci-dessous pour continuer'. This section contains two input fields: 'Nom public du projet' with the value 'project-113647056603' and 'Adresse e-mail d'assistance associée au projet' with a dropdown menu showing a partially obscured email address. Below this section, there are two expandable sections: 'Ajoutez à la liste d'autorisation les ID client de projets externes (facultatif)' and 'Configuration du SDK Web', both with expand/collapse icons. At the bottom right, there are 'Annuler' and 'Enregistrer' buttons.

On peut éditer le modèle des mails envoyés

Authentication

Users Sign-in method Templates Usage

Modèles

- Email
- Validation de l'adresse e-mail**
- Réinitialisation du mot de passe
- Modification de l'adresse e-mail
- Paramètres SMTP
- SMS
- Validation par SMS

Langue du modèle: anglais

Validation de l'adresse e-mail

Lorsqu'un utilisateur se connecte avec une adresse e-mail et un mot de passe, vous pouvez lui envoyer un e-mail de confirmation pour qu'il valide son adresse. [En savoir plus](#)

Non de l'expéditeur: non indiqué

De: noreply@todo-app-33cca.firebaseio.com

Adresse de réponse: noreply

Objet: Verify your email for %APP_NAME%

Message: Hello %DISPLAY_NAME%,
Follow this link to verify your email address.
https://todo-app-33cca.firebaseio.com/_/auth/action?mode=action&obCode=code
If you didn't ask to verify this address, you can ignore this email.
Thanks,
Your %APP_NAME% team

Firestore (base de données)

C'est la première version de base de données (NoSQL) de Firebase. Permet de définir des collections ainsi que des règles d'accès aux différentes collections/ documents

Création d'une database

Onglet « Firestore Database » ... « Créer une base de données »

Créer une base de données

1 Règles de sécurité pour Cloud Firestore

2 Définir l'emplacement Cloud Firestore

Après avoir défini votre structure de données, vous devez spécifier des règles pour sécuriser vos données.

[En savoir plus](#)

Démarrer en mode de production
Par défaut, vos données sont privées. L'accès client en lecture/écriture ne sera autorisé qu'en fonction de vos règles de sécurité.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document}/** {
      allow read, write: if false;
    }
  }
}
```

Tous les accès tiers en lecture et en écriture seront refusés

Si vous activez Cloud Firestore, vous ne pourrez pas utiliser Cloud Datastore avec ce projet, notamment depuis l'application App Engine associée.

Annuler **Suivant**

En mode test, la base de données sera supprimée après un certain temps (à éviter sauf pour des tests)

... Zone du cloud

Créer une base de données

1 Règles de sécurité pour Cloud Firestore

2 Définir l'emplacement Cloud Firestore

Le paramètre d'emplacement définit l'endroit où sont stockées vos données Cloud Firestore.

⚠ Une fois défini, cet emplacement ne peut plus être modifié. Ce paramètre s'appliquera également à votre bucket Cloud Storage par défaut.

[En savoir plus](#)

Zone Cloud Firestore

Sélectionner la région la plus proche

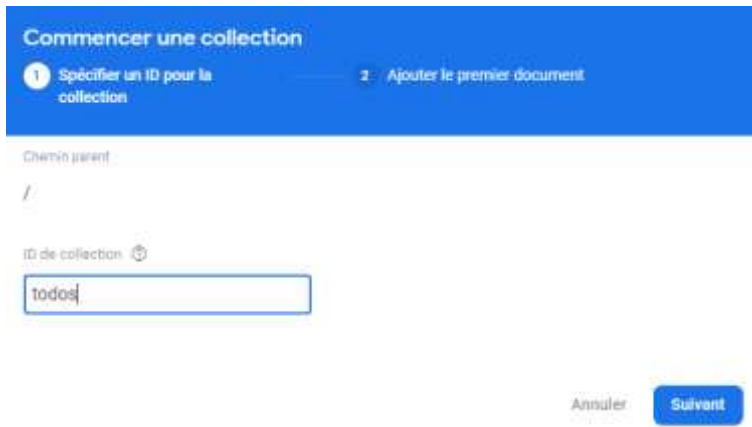
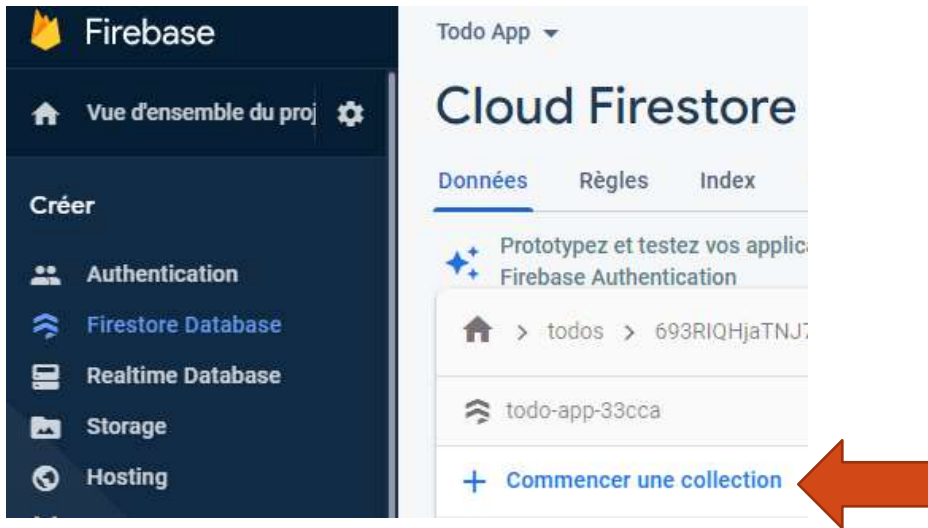
Si vous activez Cloud Firestore, vous ne pourrez pas utiliser Cloud Datastore avec ce projet, notamment depuis l'application App Engine associée.

Annuler

Activer

Création de collection

Cliquer sur « commencer une collection »

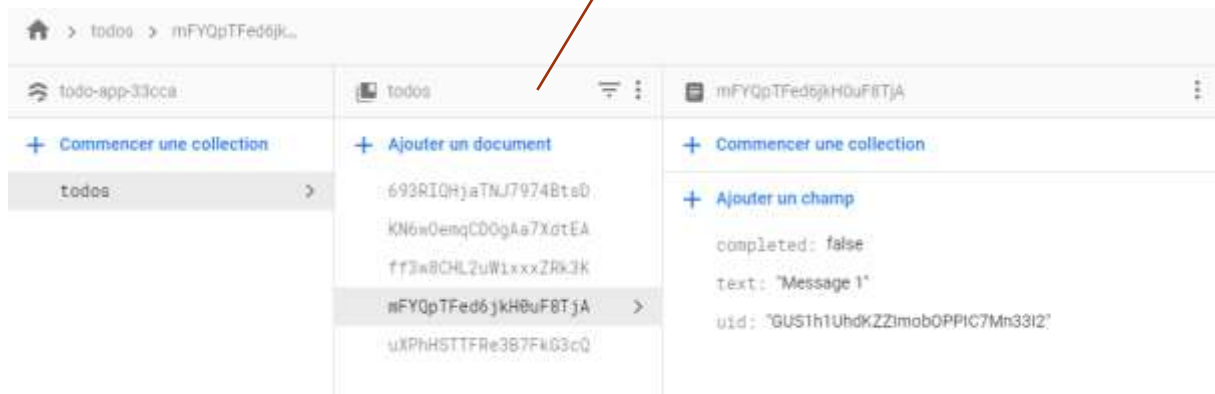


On peut ensuite définir des documents



Exemple de documents

On peut voir l'id des documents



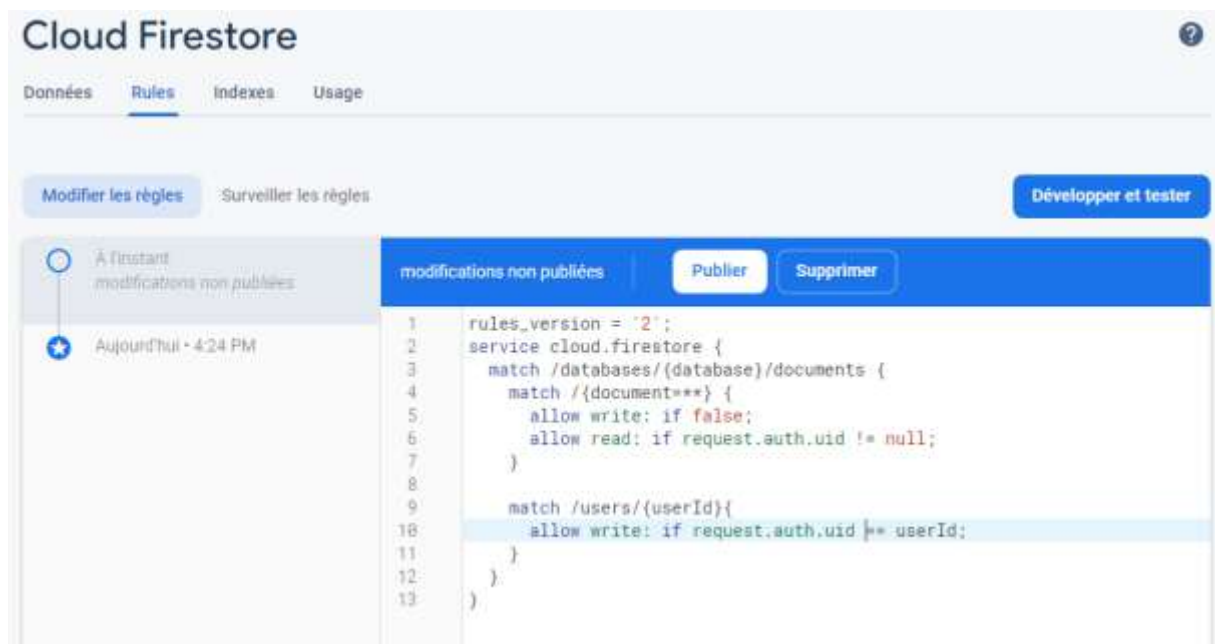
Rules

Exemple sur la collection « users »:

- *Non autorisé en écriture sur tout*
- *Autorisation de lecture sur tout si authentifié.*
- *Autorisation d'écriture sur la collection « users » seulement si le userId correspond*

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow write: if false;
      allow read: if request.auth.uid != null;
    }

    match /users/{userId}{
      allow write: if request.auth.uid == userId;
    }
  }
}
```



... cliquer sur « **publier** »

Playground

Permet de tester les différentes actions sur une collection. Exemple



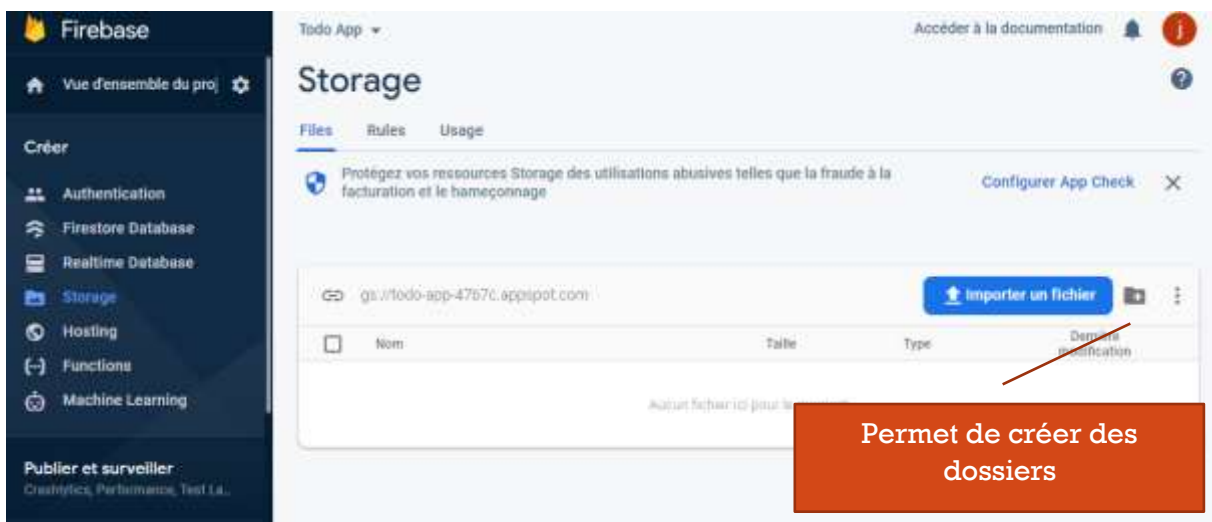
Cloud Storage (upload de fichiers)

Permet d'héberger des fichiers tels que des images sur firebase.

Dans les collections pour une image de profil par exemple on indiquera l'url vers cette image.

Upload de fichiers

Onglet « Storage ». On peut créer des dossiers et uploader des fichiers

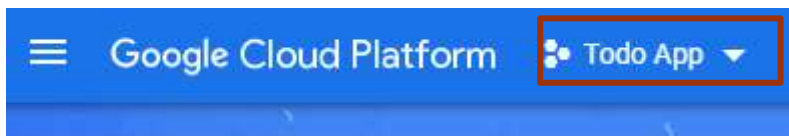


Exemple création d'un dossier « uploads » et imports d'images

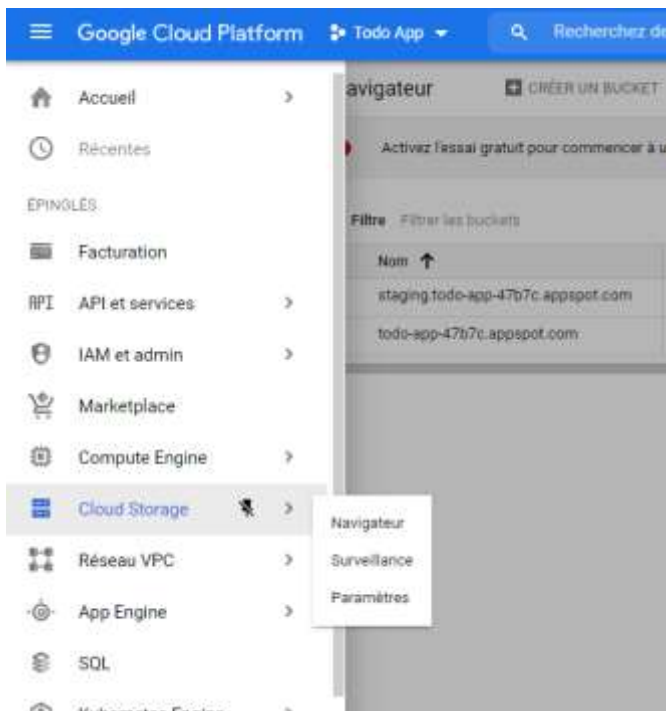


Rendre les fichiers accessibles publiquement via une url [Documentation](#)

Aller sur la console Google Cloud <https://console.cloud.google.com/> et sélectionner le bon projet



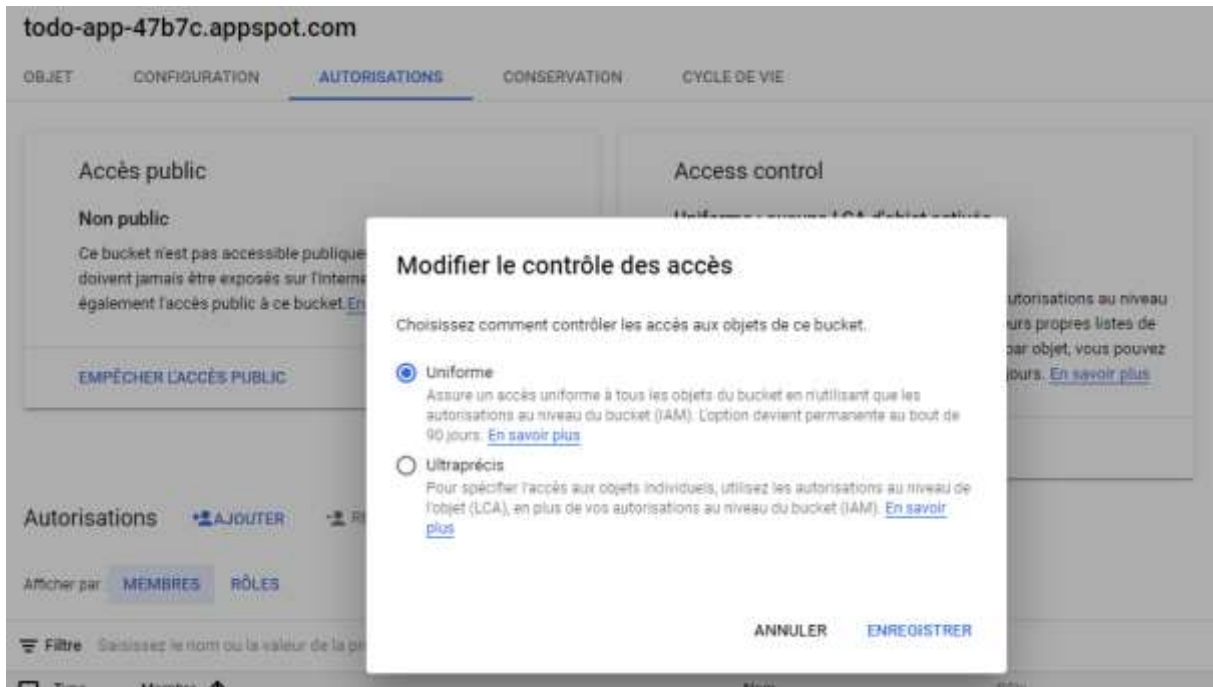
... puis dans la sidebar sélectionner « cloud storage »



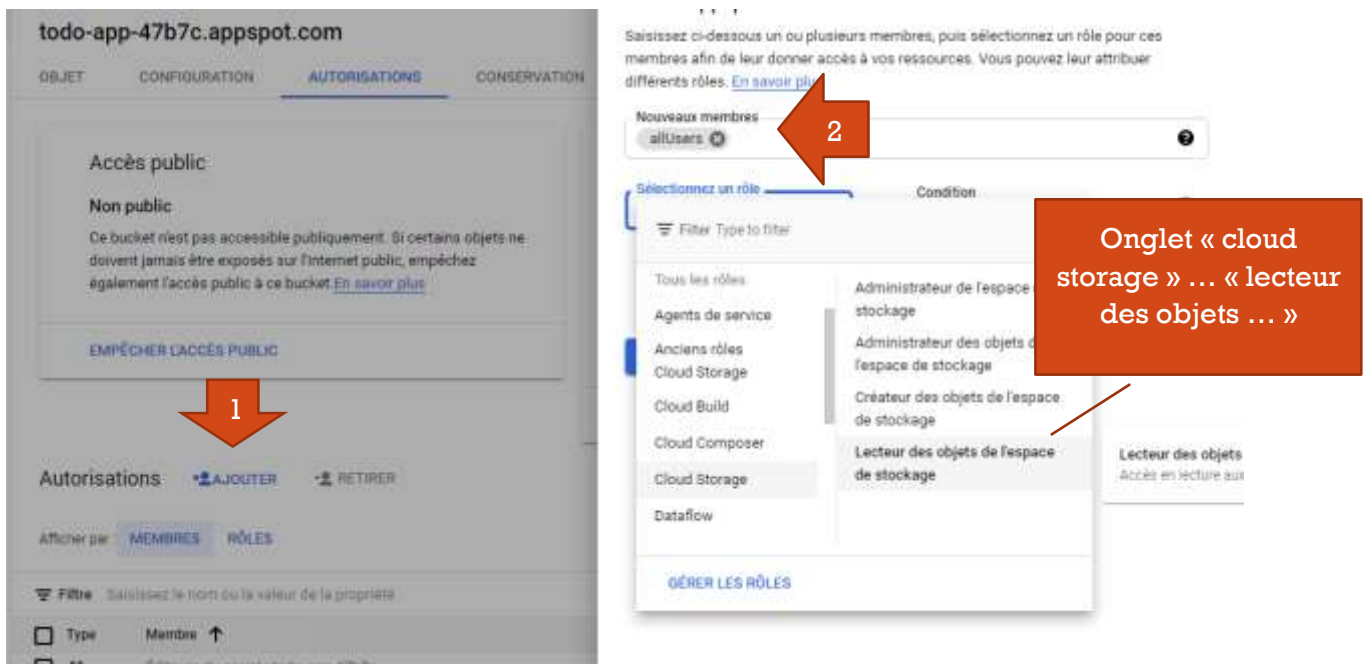
Cliquer sur le lien correspond

Filtre Filtrer les buckets		
<input type="checkbox"/> Nom ↑	Création	Type d'emplacement
<input type="checkbox"/> staging.todo-app-47b7c.appspot.com	13 août 2021 à 16:24:23	Multi-region
<input type="checkbox"/> todo-app-47b7c.appspot.com	13 août 2021 à 16:24:23	Multi-region

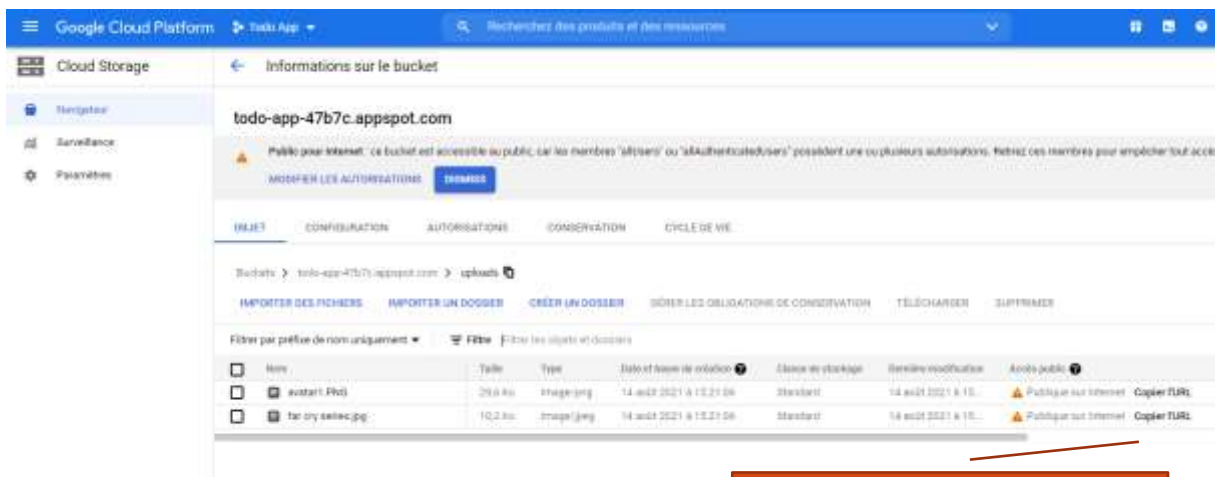
Passer les autorisations en « uniforme » pour appliquer les règles à tous les objets



Ajout des autorisations de lecture à tous les utilisateurs « allUsers »



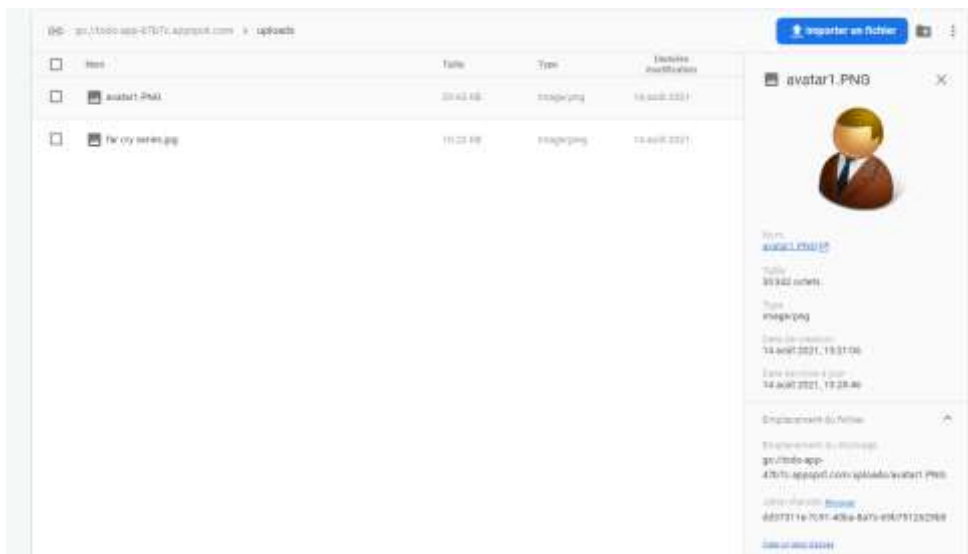
... « Enregistrer »



URL publique

Exemple d'url publique : « <https://storage.googleapis.com/todo-app-47b7c.appspot.com/uploads/avatar1.PNG> »

Depuis la console firebase on peut avoir les détails de chaque fichier




En emplacement du stockage on semble avoir l'url du fichier (exemple « <gs://todo-app-47b7c.appspot.com/uploads/avatar1.PNG> »). Il suffit de remplacer « gs : » par « <https://storage.googleapis.com/> »

Ajout de Firebase à une application



Avec une application Web

Cliquer sur le bouton  et donner un pseudo à l'application



... cliquer sur « Enregistrer l'application ». On obtient alors un exemple de code avec les informations pour se connecter depuis une application web

Installation et configuration du SDK

npm CDN Configuration

Si vous utilisez déjà [npm](#) et un bundler de module comme [webpack](#) ou [Rollup](#), vous pouvez exécuter la commande suivante pour installer la dernière version du SDK :

```
$ npm install firebase
```

Ensuite, initialisez Firebase et commencez à exploiter les SDK pour les produits que vous souhaitez utiliser.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyCbnQ5NqNN3iJ2KEnpwNWXawN2YS_0mFL8",
  authDomain: "todo-app-33cca.firebaseio.com",
  projectId: "todo-app-33cca",
  storageBucket: "todo-app-33cca.appspot.com",
  messagingSenderId: "24423858774",
  appId: "1:24423858774:web:4f7b6b832fe59da49b2637",
  measurementId: "G-WF6CDQK73K"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Remarque : Cette option utilise le [SDK JavaScript modulaire](#), qui réduit la taille du SDK.

... cliquer sur « Accéder à la console ». L'application est ajoutée.

The screenshot shows the Firebase console interface for a project named 'Todo App'. On the left is a dark sidebar with navigation options: 'Vue d'ensemble du projet', 'Créer' (with sub-items for Authentication, Firestore Database, etc.), 'Publier et surveiller' (with sub-items for Crashlytics, Performance, Test Lab, etc.), 'Analytics', 'Engager', and 'Extensions'. The main content area has a blue header with 'Todo App' and 'Accéder à la documentation'. Below the header is a notification banner about email updates. The main content area features a large illustration of a woman and a man, with the text '1 application' and '+ Ajouter une application'. A red callout box points to the '+ Ajouter une application' button with the text 'Formule Spark : gratuit'. Another red callout box points to the main content area with the text 'On peut ajouter d'autres applications ou consulter/ modifier celle(s) existante(s)'. At the bottom, there is a banner for 'Stockez et synchronisez les données de votre application en quelques millisecondes'.

Firestore CLI [Documentation](#)

Installation en **global**

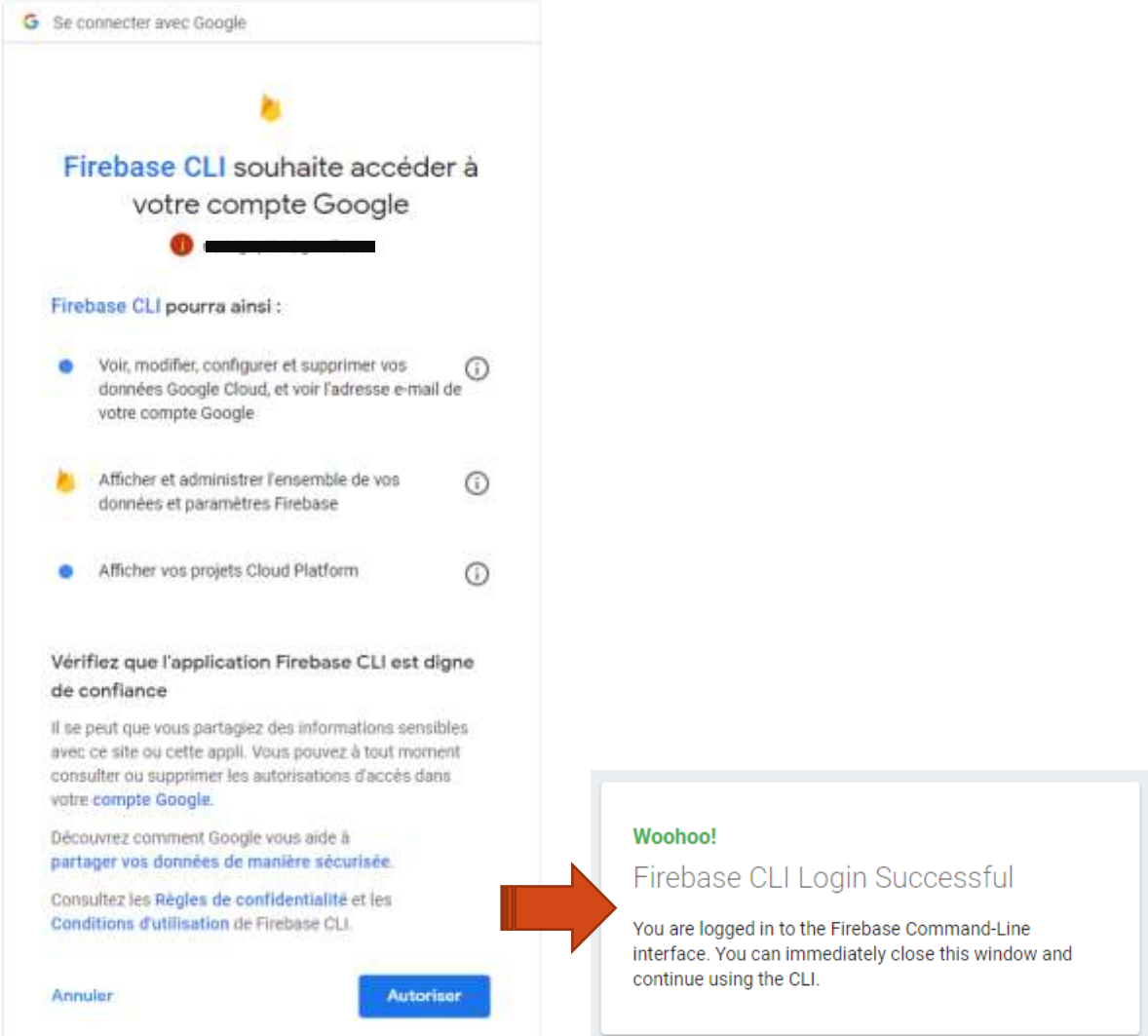
```
npm i -g firebase-tools
```

Login

Depuis une invite de commande

```
firebase login
```

... Une page web s'ouvre demandant de s'authentifier avec un compte Google puis on est dirigé vers un écran de consentement



Se connecter avec Google

Firestore CLI souhaite accéder à votre compte Google

Firestore CLI pourra ainsi :

- Voir, modifier, configurer et supprimer vos données Google Cloud, et voir l'adresse e-mail de votre compte Google
- Afficher et administrer l'ensemble de vos données et paramètres Firestore
- Afficher vos projets Cloud Platform

Vérifiez que l'application **Firestore CLI** est digne de confiance

Il se peut que vous partagiez des informations sensibles avec ce site ou cette appli. Vous pouvez à tout moment consulter ou supprimer les autorisations d'accès dans votre [compte Google](#).

Découvrez comment Google vous aide à [partager vos données de manière sécurisée](#).

Consultez les [Règles de confidentialité](#) et les [Conditions d'utilisation](#) de Firestore CLI.

[Annuler](#) [Autoriser](#)

Woohoo!
Firestore CLI Login Successful

You are logged in to the Firestore Command-Line interface. You can immediately close this window and continue using the CLI.

Pour se déconnecter

```
firebase logout
```

Une fois connecté on peut utiliser les commandes du cli et lister les projets par exemple

```
firebase projects:list
```

```
C:\Users\romag>firebase projects:list  
✓ Preparing the list of your Firebase projects
```

Project Display Name	Project ID	Project Number	Resource Location ID
Todo App	todo-app-47b7c	113647056603	[Not specified]

```
1 project(s) total.
```

Projet client (React)

Version 8

[Github de firebase js sdk](#)

Puis installation de firebase en dépendance

```
npm i firebase@8.10.0
```

Voir si c'est la dernière version

Configuration

Créer un fichier contenant les informations de connexion et l'importer en haut de « App » (ou ajouter directement le code en haut de « App »)

```
import firebase from "firebase/app";  
import "firebase/auth";  
import "firebase/firestore";  
  
const firebaseConfig = {  
  apiKey: "AIzaSyCbhQ5NqMN31J2KEmpwNWXawN2YS_OmFL8",  
  authDomain: "todo-app-33cca.firebaseio.com",  
  projectId: "todo-app-33cca",  
  storageBucket: "todo-app-33cca.appspot.com",  
  messagingSenderId: "24423850774",  
  appId: "1:24423850774:web:4f7b6b832fe59da49b2637",  
  measurementId: "G-WF6CDQK73K"  
};  
  
const firebaseApp = firebase.initializeApp(firebaseConfig);  
  
export default firebaseApp;
```

Import des features utilisées par le projet client

Auth

Documentation : <https://firebase.google.com/docs/auth>

On peut créer un fichier util « auth » avec les différentes méthodes de gestion des users. Ce sont les méthodes qui seront utilisées à la soumission de formulaire signin, signup, forgot password, etc.

```
import firebase from "firebase/app";

const addAuthListener = cb =>
  firebase.auth().onAuthStateChanged(user => cb(user));

const getCurrentUser = () => firebase.auth().currentUser;

const signIn = async (email, password) =>
  await firebase.auth().signInWithEmailAndPassword(email, password);

const signInWithGoogle = async () => {
  const provider = new firebase.auth.GoogleAuthProvider();
  return await firebase.auth().signInWithPopup(provider);
};

const signUp = async (email, password) =>
  await firebase.auth().createUserWithEmailAndPassword(email, password);

const signOut = async () => await firebase.auth().signOut();

const sendPasswordResetEmail = async email =>
  await firebase.auth().sendPasswordResetEmail(email);

export {
  addAuthListener,
  getCurrentUser,
  signIn,
  signInWithGoogle,
  signUp,
  signOut,
  sendPasswordResetEmail
};
```

Création d'un hook « **auth-hook** » qui permettra de récupérer les informations d'utilisateur au chargement de l'application React.

```
import { useState, useEffect } from "react";
import { addAuthListener, getCurrentUser } from "./auth";

const useAuth = () => {
  const [loading, setLoading] = useState(true);
  const [user, setUser] = useState(getCurrentUser());
```

```

useEffect(() => {
  const unsubscribe = addAuthListener(user => {
    setUser(user);
    setLoading(false);
  });

  return () => {
    unsubscribe();
  };
}, []);

return { loading, user };
};

export default useAuth;

```

On peut créer un contexte « auth-context » pour partager les informations de l'utilisateur

```

import React, { useContext } from "react";

export const AuthContext = React.createContext({});

export function useAuthContext() {
  const context = useContext(AuthContext);
  if (!context) {
    throw new Error(
      "useAuth must be used within a AuthProvider. Wrap a parent component in <AuthProvider> to fix this error."
    );
  }
  return context;
}

```

« App »

Utilisation du hook

```

export default function App() {
  const { loading, user } = useAuth();

```

... et définition de la value du provider

```

if (loading)
  return (
    <div className="m-auto align-middle">
      <CircleToBlockLoading />
    </div>
  );
return (

```

```
<AuthContext.Provider value={{ user, authenticated: user !== null }}>
```

Création d'un composant "**ProtectedRoute**" qui permet de rediriger les utilisateurs vers la page de connexion s'ils essaient d'accéder à une page protégée

```
import React from "react";
import { Route, Redirect } from "react-router-dom";

export const ProtectedRoute = ({ authenticated, ...props }) => {
  if (!authenticated) return <Redirect to="/auth" />;
  return <Route {...props} />;
};
```

Exemple d'utilisation (« App »)

```
<Switch>
  <ProtectedRoute path="/" authenticated={user !== null} exact>
    <Todos />
  </ProtectedRoute>
  <Route path="/auth" exact>
    <AuthPage />
  </Route>
  <Route path="/forgot-password" exact>
    <ForgotPassword />
  </Route>
  <Route path="*">
    <Redirect to="/" />
  </Route>
</Switch>
```

Exemple de méthodes de formulaire pour signin

```
const handleSubmit = async event => {
  event.preventDefault();
  try {
    // validation

    const result = await signIn(values.email, values.password);
    history.push("/");
  } catch (err) {
    setError(err.message);
  }
};
```

Utilisation du hook useHistory de react router pour rediriger

```
const handleSigninWithGoogle = async event => {
  event.preventDefault();
  try {
    const result = await signInWithGoogle();
    history.push("/");
  } catch (err) {
    setError(err.message);
  }
};
```

Firestore

Documentation : <https://firebase.google.com/docs/firestore/quickstart#web-version-8>

On peut se créer un fichier util « firestore »

```
import firebase from "firebase/app";

const getItems = snapshot => {
  let items = [];
  snapshot.forEach(doc => {
    const item = doc.data();
    items.push({
      id: doc.id,
      ...item
    });
  });
  return items;
};

const getDocs = async collectionPath => {
  const snapshot = await firebase
    .firestore()
    .collection(collectionPath)
    .get();
  return getItems(snapshot);
};

const getDocsByUid = async (collectionPath, uid) => {
  const snapshot = await firebase
    .firestore()
    .collection(collectionPath)
    .where("uid", "=", uid)
    .get();
  return getItems(snapshot);
};

const addDoc = async (collectionPath, data) => {
  const doc = await firebase.firestore().collection(collectionPath).add(data);
  return {
    id: doc.id,
    ...data
  };
};

const setDoc = async (collectionPath, documentPath, data) =>
  await firebase
    .firestore()
    .collection(collectionPath)
    .doc(documentPath)
    .set(data);

const deleteDoc = async (collectionPath, documentPath) =>
  await firebase
    .firestore()
    .collection(collectionPath)
    .doc(documentPath)
    .delete();

const onSnapshot = (collectionPath, cb) => {
  const unsubscribe = firebase
    .firestore()
    .collection(collectionPath)
    .onSnapshot(snapshot => cb(getItems(snapshot)));
  return unsubscribe;
};

const onSnapshotByUid = (collectionPath, uid, cb) => {
  const unsubscribe = firebase
    .firestore()
    .collection(collectionPath)
    .where("uid", "=", uid)
    .onSnapshot(snapshot => cb(getItems(snapshot)));
  return unsubscribe;
};

export {
  getDocs,
  getDocsByUid,
  addDoc,
  setDoc,
  deleteDoc,
  onSnapshot,
  onSnapshotByUid
};
```

Filtre par uid (champ ajouté
au document)

Utilisation

Exemple d'**ajout** (méthode appelée à la soumission d'un formulaire)

```
const COLLECTION_PATH = "todos";
```

```
const handleSubmit = async event => {
  event.preventDefault();

  const text = inputRef.current.value.trim();
  if (text === "") return;

  try {
    const newTodo = await addDoc(COLLECTION_PATH, {
      uid: user.uid,
      text,
      completed: false
    });
    inputRef.current.value = "";
  } catch (error) {
    console.error("Error", error);
  }
};
```

Update

```
const handleToggle = async () => {
  try {
    await setDoc(COLLECTION_PATH, item.id, {
      uid: user.uid,
      text: item.text,
      completed: !item.completed
    });
  } catch (error) {
    console.error("Error", error);
  }
};
```

Delete

```
const handleDelete = async () => {
  try {
    await deleteDoc(COLLECTION_PATH, item.id);
  } catch (error) {
    console.error("Error", error);
  }
};
```

On peut utiliser « onSnapshot » pour être notifié et recharger automatiquement les données

```

const TodoPage = () => {
  const { user } = useAuthContext();
  const [items, setItems] = useState([]);

  useEffect(() => {
    // automatic reloading on insert, update, delete
    const unsubscribe = onSnapshotByUid(COLLECTION_PATH, user.uid, items =>
      setItems(items)
    );
    return () => unsubscribe();
  }, []);
}

```

Storage, upload de fichier

Documentation : <https://firebase.google.com/docs/storage/web/start>

[Article sur Medium intéressant](#)

On pourrait créer un fichier util « storage » avec les méthodes utilisées

Exemple avec React

```

import React, { useState } from "react";
import firebase from "firebase/app";

export default function UploadFile() {
  const [file, setFile] = useState();
  const onFileChange = (event) => {
    if (event.target.files && event.target.files.length === 1) {
      const pickedFile = event.target.files[0];
      setFile(pickedFile);
      console.log(pickedFile.name);
    }
  };

  const submitHandler = async (event) => {
    event.preventDefault();

    try {
      const storage = firebase.storage().ref().child(`/uploads/${file.name}`);
      await storage.put(file);
    } catch (err) {
      console.log("An error encountered", err);
    }
  };

  return (
    <div>
      <form onSubmit={submitHandler}>
        <input type="file" accept=".jpg,.png,.jpeg" onChange={onFileChange} />
        <button type="submit">Save</button>
      </form>
    </div>
  );
}

```

Sauvegarde du fichier avec son nom et extension dans le dossier « uploads »

Ne pas oublier dans la configuration de firebase d'importer « firebase/storage »

```

import firebase from "firebase/app";
import "firebase/auth";
import "firebase/storage";
import "firebase/analytics";

const firebaseConfig = {

```

```
apiKey: "AIzaSyBUf1ro2LpKtZa0-brms30_YQLAlj0q45c",
authDomain: "todo-app-47b7c.firebaseio.com",
projectId: "todo-app-47b7c",
storageBucket: "todo-app-47b7c.appspot.com",
messagingSenderId: "113647056603",
appId: "1:113647056603:web:da46a4093ef55a4c78befe",
measurementId: "G-GZCLT1H8KT",
});

const app = firebase.initializeApp(firebaseConfig);
firebase.analytics();
```

Hosting

Documentation : <https://firebase.google.com/docs/hosting/quickstart>

Le CLI permet de déployer un nouveau projet firebase ou un projet existant

1. Installer le [Firebase CLI](#) si ce n'est pas fait
2. Création de firebase.json et « .firebaserc »

```
firebase init hosting
```

```
? Please select an option: Use an existing project
? Select a default Firebase project for this directory: todo-app-47b7c (Todo App)
```

Un fichier « firebase.json » est généré

```
{
  "hosting": {
    "public": "dist",
    "ignore": [
      "firebase.json",
      "**/.*",
      "**/node_modules/**"
    ],
    "rewrites": [
      {
        "source": "**",
        "destination": "/index.html"
      }
    ]
  }
}
```

Un fichier « .firebaserc » avec l'id du projet Fireabse utilisé

```
{
  "projects": {
    "default": "todo-app-47b7c"
  }
}
```

3. Déploiement

Ne pas oublier de build le projet (commande « npm run build ») auparavant pour générer les fichiers dans le dossier « dist » par exemple qui seront déployés

```
firebase deploy --only hosting
```

Il est possible d'ajouter un message

```
firebase deploy --only hosting -m "First commit"
```

```
PS C:\Projects\todo-react-firebase-v8> firebase deploy --only hosting

=== Deploying to 'todo-app-47b7c'...

i  deploying hosting
i  hosting[todo-app-47b7c]: beginning deploy...
i  hosting[todo-app-47b7c]: found 6 files in dist
+  hosting[todo-app-47b7c]: file upload complete
i  hosting[todo-app-47b7c]: finalizing version...
+  hosting[todo-app-47b7c]: version finalized
i  hosting[todo-app-47b7c]: releasing new version...
+  hosting[todo-app-47b7c]: release complete

+  Deploy complete!

Project Console: https://console.firebase.google.com/project/todo-app-47b7c/overview
Hosting URL: https://todo-app-47b7c.web.app

Update available 9.16.3 → 9.18.0
To update to the latest version using npm, run npm install -g firebase-tools
For other CLI management options, visit the CLI documentation
(https://firebase.google.com/docs/cli#update-cli)
```

url du projet déployé

Exemple : <https://todo-app-47b7c.web.app/>

4. Mise à jour ultérieures

```
firebase deploy --only hosting -m "Deploying the best new feature ever."
```

Tester son projet avant de le deployer

Il est possible de tester son projet avec la commande

```
firebase serve
```

```
PS C:\Projects\todo-react-firebase-v8> firebase serve

=== Serving from 'C:\Projects\todo-react-firebase-v8' ...

i  hosting: Serving hosting files from: dist
+  hosting: Local server: http://localhost:5000
```


Rollback

Il est possible d'annuler une update depuis l'onglet "Hosting"

Tableau de bord Utilisation

Surveillez les requêtes Web envoyées à votre site avec Cloud Logging Premiers pas

Domaines todo-app-33cca

Ajouter un domaine personnalisé

Domaine	État
todo-app-33cca.web.app Par défaut	
todo-app-33cca.firebaseio.com Par défaut	

Historique des versions de todo-app

État	Heure	Déploiement	Fichiers
★ Actuellement	16 sept. 2021 19:28	b benquoieck@gmail.com 55437a... current item updated on s...	...
🚀 Déploiement effectué	16 sept. 2021 18:58	b benquoieck@gmail.com 540e33... linkify and sidebar	Rollback Supprimer

« current » (ou « actuellement » en fr) indique la version active

Ajouter un nom de domaine personnalisé

Depuis l'onglet "Hosting"

Ajouter un domaine personnalisé

1 Saisissez un domaine 2 Valider la propriété 3 Mettre en service

Saisissez le nom exact du domaine qui doit s'afficher lorsque les internautes visitent votre site. Il peut s'agir d'un domaine (yourdomain.com) ou d'un sous-domaine (app.yourdomain.com).

Domaine

my-awesome-site.com

Rediriger my-awesome-site.com vers un site Web existant

★ Vous avez besoin d'un domaine ? Avec Google Domains, lancez rapidement votre site en ligne.

Trouver un domaine


Annuler Continuer

Une aide pour acquérir un nom de domaine est suggérée

Ajouter un domaine personnalisé

1 Saisissez un domaine — 2 Valider la propriété — 3 Mettre en service


Ajoutez les enregistrements TXT ci-dessous à votre fournisseur DNS pour confirmer que vous êtes le propriétaire de **my-awesome-site.com**. Ne supprimez pas vos enregistrements TXT.



Type	Hôte	Valeur
TXT	my-awesome-site.com	google-site-verification=qgbvZDe3800NQpL7eP4--N6SuyUWc-0LOzo4F5ylDT0 

Annuler **Valider**

Copier la valeur de TXT qu'il faudra ajouter


On peut rechercher des noms de domaines sur godaddy (qui seront moins chers qu'avec le lien propose par firebase) (<https://fr.godaddy.com/domainsearch/>)







my-awesome-site.fr  [voir mon panier](#)

 Sous réserve de certaines restrictions. 
 11,99€ **1,00 €** pour la première année

[Ajouter au panier](#)

my-awesome-site.eu Ajouter ceci : **0,01 €/an**
lorsque vous vous enregistrez pour 2 ans ou plus. Prix pour la 1ère année 0,01 € Années supplémentaires 13,19 €

Les domaines incluent la protection de confidentialité de base. 

myawesome.fr <small>Sous réserve de certaines restrictions. </small>	11,99€ 1,00 € pour la première année	 Ajouter
myawesomesites.com	24,17€ 1,37 € pour la première année, avec un enregistrement de 2 an(s)	 Ajouter
ANNONCE my-awesome-site.live	32,80€ 2,21 € pour la première année	 Ajouter
myawesomesites.fr <small>Sous réserve de certaines restrictions. </small>	11,99€ 1,00 € pour la première année	 Ajouter

Ajout de la valeur TXT après achat du domaine

Records

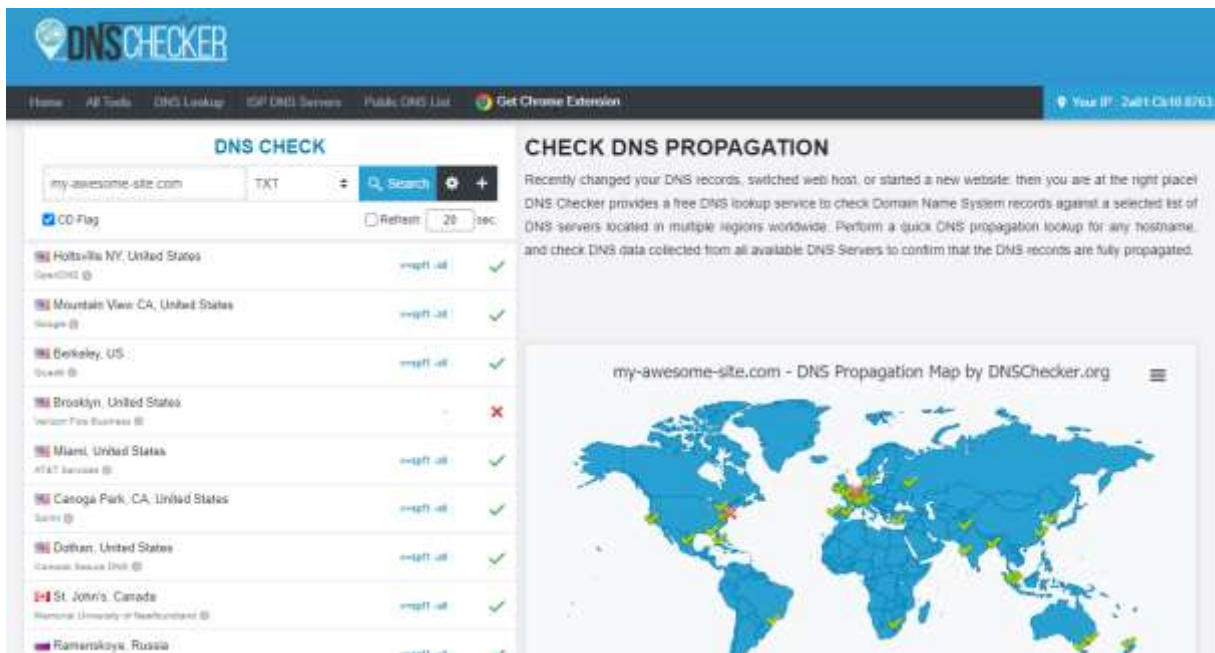
Last updated 20/07/2018 16:09

Type	Name	Value	TTL	
CNAME	www	@	1 Hour	
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com	1 Hour	
NS	@	ns17.domaincontrol.com	1 Hour	
NS	@	ns18.domaincontrol.com	1 Hour	
SOA	@	Primary nameserver: ns17.domaincontrol.com	1 Hour	

Type * Host * TXT Value *

TTL *

<https://dnschecker.org/> permet de vérifier la disponibilité



DNS CHECKER

Home All Tools DNS Lookup ISP DNS Servers Public DNS List Get Chrome Extension Your IP: 208.130.8763

DNS CHECK

my-awesome-site.com TXT Search +


CD-Flag Refresh 20 sec

Holtsville NY, United States LevelDNS @	→ipfl -all	✓
Mountain View CA, United States Nudge @	→ipfl -all	✓
Berkeley, US Glean @	→ipfl -all	✓
Brooklyn, United States Verizon Fios Business @	-	✗
Miami, United States AT&T Services @	→ipfl -all	✓
Canoga Park, CA, United States Sant @	→ipfl -all	✓
Dothan, United States Comcast Business Direct @	→ipfl -all	✓
St. John's, Canada Memorial University of Newfoundland @	→ipfl -all	✓
Ramenskoye, Russia	→ipfl -all	✓

CHECK DNS PROPAGATION

Recently changed your DNS records, switched web host, or started a new website: then you are at the right place! DNS Checker provides a free DNS lookup service to check Domain Name System records against a selected list of DNS servers located in multiple regions worldwide. Perform a quick DNS propagation lookup for any hostname, and check DNS data collected from all available DNS Servers to confirm that the DNS records are fully propagated.

my-awesome-site.com - DNS Propagation Map by DNSChecker.org



Rewrites

On peut définir une redirection vers index.html (firebase.json)

<https://firebase.google.com/docs/hosting/full-config#rewrites>

```
"hosting": {  
  // ...  
  
  // Serves index.html for requests to files or directories that do not exist  
  "rewrites": [{  
    "source": "**",  
    "destination": "/index.html"  
  }]  
}
```