



Cours Angular

1

| TD MyBooks

| Firebase – Création de l'application – Configurations

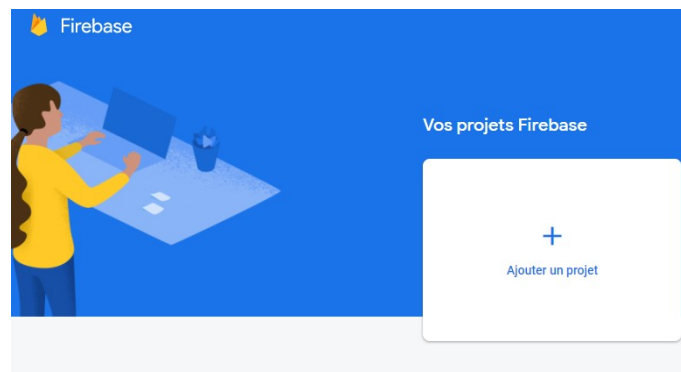
1] Principes de la nouvelle application

L'application va porter sur la gestion personnelle de livres avec mise en place d'un stockage sur le cloud avec une base noSQL Firebase (services d'hébergement pour les développeurs) et cette fois-ci le framework CSS Bootstrap.

- **Création du compte FireBase et initialisations**

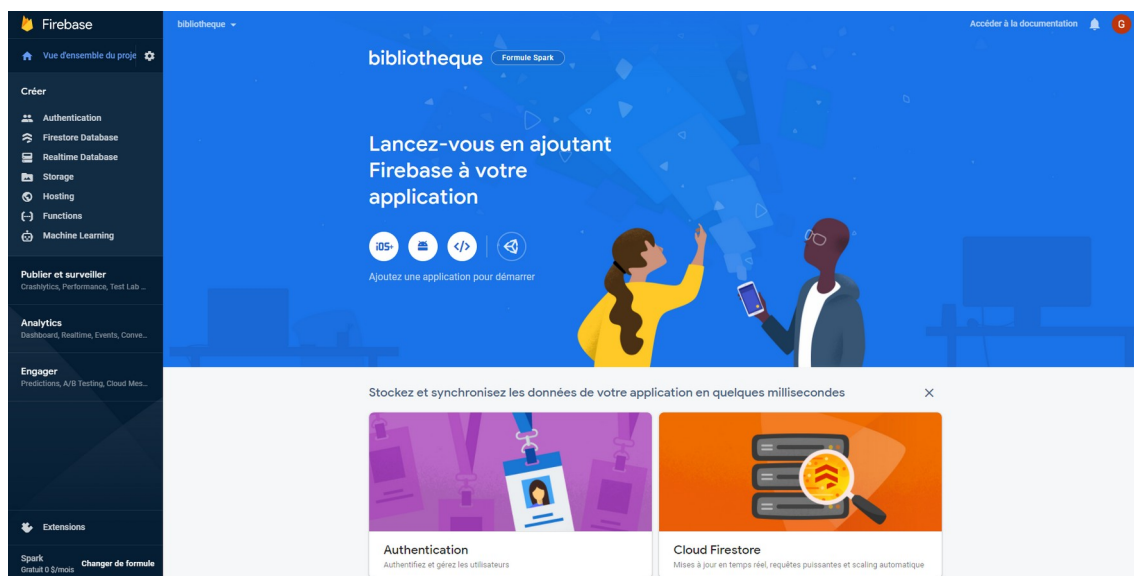
T1.1 Allez sur le site officiel <https://firebase.google.com/> et enregistrez-vous, classiquement avec votre compte Gmail.

T1.2 Une fois sur la console Firebase, créez un nouveau projet par l'icône « Ajouter un projet » :

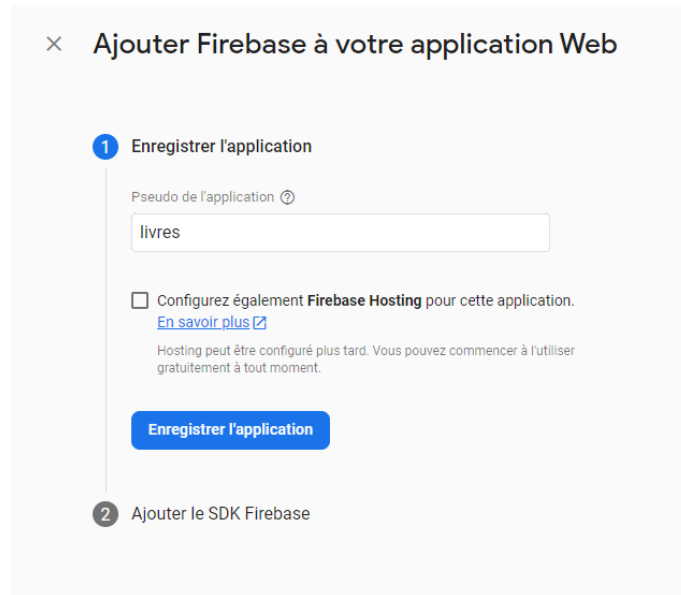


T1.3 Indiquez `bibliotheque` (sans l'accent) comme nom du projet, puis continuez ; à l'écran suivant **décochez l'activation pour Google Analytics**, non nécessaire pour cette application de démonstration.

Une fois la création lancée, continuer pour arriver à l'écran ci-dessous :



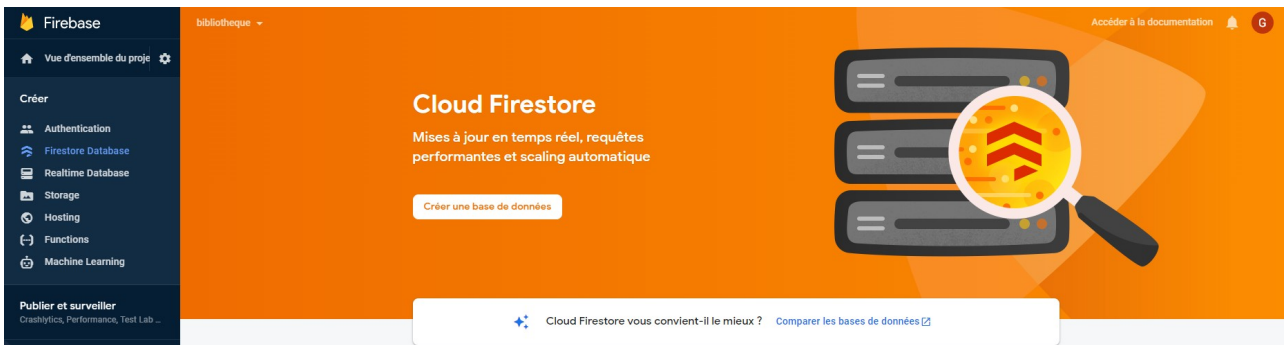
T1.4 L'application sera de type Web, aussi cliquez sur l'icône (</>) correspondante et indiquez le pseudo de l'application ; enregistrez ensuite :



Vous obtenez votre configuration, c'est à dire un ensemble de valeurs dans une constante `firebaseConfig` (**à sauvegarder pour l'instant dans un fichier texte simple**).

T1.5 Revenez sur l'écran du projet et cliquez cette fois sur la petite roue dentée à droite du lien « Vue d'ensemble du projet » en haut à gauche ; si vous sélectionnez « Paramètres du projet » vous retrouverez votre configuration et accessoirement c'est à cet endroit que vous pouvez supprimer l'application.

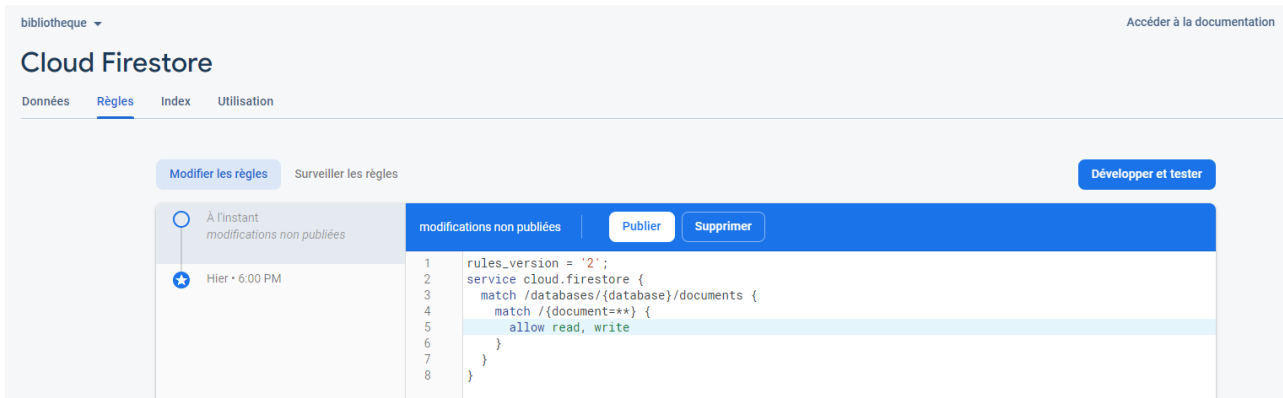
T1.5 Revenez sur l'écran du projet et cliquez cette fois sur **Firestore Database** pour cette fois créer l'espace de stockage :



T1.6 Cliquez sur le bouton « Créer une base de données », indiquez de « Démarrer en mode test » ; choisissez plutôt un niveau de Zone Cloud Firestore à `europa-west` (choisissez le numéro que vous voulez) et activez la base.

T1.7 Toujours dans **Firestore Database**, vous obtenez un écran avec votre bibliothèque mais sans collection ; cliquez sur l'onglet Règles et modifiez le code comme ci-dessous pour ensuite le publier.

L'explication est que l'on supprime le délai de 30 jours indiqué car nous sommes en test mais on aimerait ne pas être limité dans le temps !



2] Création de l'application

- **Construction de l'application**

T2.1 Dans le répertoire des projets **Angular**, créez la nouvelle application cette fois avec **obligatoirement** le routage :

```
angular$ ng new mybooks --style=scss --routing=true
angular$ cd mybooks
```

T2.2 Installez les différents modules nécessaires, aussi bien pour **Bootstrap** que pour **Firebase** ; ouvrez ensuite votre projet par **Visual Studio Code** :

***Rappel :** l'option `save` indique que le paquetage s'installe uniquement dans le projet et non globalement.*

```
mybooks$ npm install --save bootstrap
mybooks$ npm install --save bootstrap-icons
mybooks$ npm install --save @ng-bootstrap/ng-bootstrap
mybooks$ npm install --save @angular/fire
```

Note : si l'installation de `ng-bootstrap` échoue par le client Angular (c'est pourtant celle recommandée : <https://ng-bootstrap.github.io/#/home>), tentez une installation classique : `npm install @ng-bootstrap/ng-bootstrap`.

T2.3 Au lieu d'intégrer Bootstrap dans le fichier `angular.json`, intégrez ces deux lignes directement dans `styles.scss` :

```
/* Importing Bootstrap SCSS file. */
@import '~bootstrap/dist/css/bootstrap.css';
@import '~bootstrap-icons/font/bootstrap-icons';
```

- **Configuration pour Firebase et autres modules**

T2.4 Indiquez dans le fichier `src/environments/environment.ts` votre configuration **Firebase** (avec vos clefs évidemment) :

```
export const environment = {
  production: false,
  firebase: {
    apiKey: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    authDomain: "bibliotheque-d288f.firebaseio.com",
    projectId: "bibliotheque-d288f",
    storageBucket: "bibliotheque-d288f.appspot.com",
    messagingSenderId: "565614153278",
    appId: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  }
}
```

```
}
};
```

T2.5 Ajoutez les lignes pour les importations nécessaires dans le fichier `src/app/app.module.ts` :

```
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
import { provideFirebaseApp, initializeApp } from '@angular/fire/app';
import { getFirestore, provideFirestore } from '@angular/fire/firestore';
import { environment } from 'src/environments/environment';
import { FormsModule } from '@angular/forms';
```

Avec les déclarations dans la section `imports` :

```
FormsModule,
NgbModule,
provideFirebaseApp(() => initializeApp(environment.firebase)),
provideFirestore(() => getFirestore()),
```

Ces importations concernent donc Firebase mais aussi le module pour les formulaires car nous en ferons un dans la saisie d'un livre, ainsi que des éléments pour Bootstrap.

3] Structure et éléments de l'application

- **« Squelette » de l'application**

La logique de présentation de l'application sera la suivante : dès le lancement de celle-ci la page d'accueil affichera les livres en stock à partir de FireBase avec un formulaire de saisie ; le menu Recherche permettra comme son nom la recherche d'un livre dans la base.

T3.1 Générez les composants (dans un sous-répertoire `components` pour améliorer la lisibilité de la structure) :

```
mybooks$ ng generate component components/home
mybooks$ ng generate component components/book
mybooks$ ng generate component components/books
mybooks$ ng generate component components/search
```

Le composant `home` est un composant template, les autres des composants métiers. `Book` sera pour l'ajout d'un livre, alors que `books` s'occupera de la liste des livres dans la base.

- **Composants spécifiques**

Nous aurons besoin d'un service utilisable par tous les composants et aussi un composant nommé `edit-book` pour l'implémentation de l'édition d'un enregistrement de livre.

```
mybooks$ ng generate service services/book
mybooks$ ng generate component modal/editBook
```

Le service s'appuiera sur un modèle (comprenez ici une interface) de livre. Il est possible aussi de créer une classe mais dans notre application, le principe de l'interface suffit car nous n'avons pas besoin de constructeur ni de méthodes.

T3.2 Créez un sous-répertoire `models` dans `src/app` et ensuite un fichier `book.model.ts` avec ceci :

```
export interface IBook {
  id?: string;
  titre : string;
  auteur : string;
```

```
    editeur : string;  
    genre : string;  
}
```

Notez le ? pour que TypeScript plus tard ne sorte pas d'erreur car cette variable est initialisée par Firebase et nous fournie à la création.