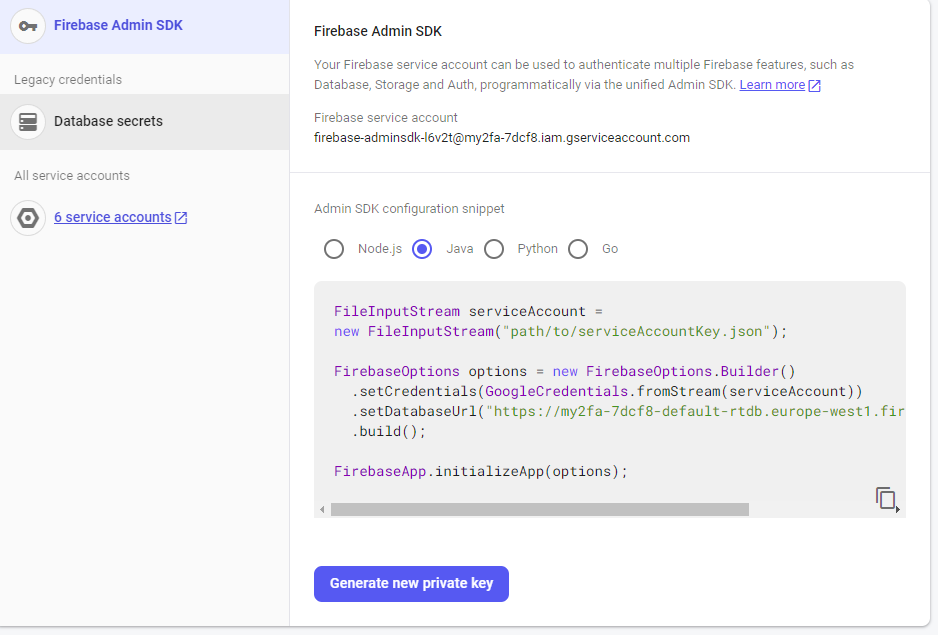
Conectare la baza de date firebase:

1. Mi-am facut un proiect: My2FAapp in care am folosit Firestore
2. De la Project Overview => Project Settings => Serivce Accounts => am luat codul de conectare la baza de date Firestore:



1. Am generat o cheie privata => JSON pe care l-am inclus in proiectul meu => config/ConnectionString.json

**Firestore & FirestoreTemplate:**

**Firestore** este clasa principală utilizată pentru a stabili o conexiune cu serviciul Cloud Firestore și a efectua operații de bază în această bază de date. Aceasta permite comunicarea cu Firestore și trimiterea de cereri către acest serviciu. Este partea esențială pentru a interacționa cu Firestore în aplicația dvs.

**FirestoreTemplate** este o clasă sau componentă specifică din cadrul framework-ului Spring Data Firestore. Aceasta oferă un nivel mai înalt de abstractizare și ușurință în realizarea operațiunilor CRUD (Create, Read, Update, Delete) în Firestore. FirestoreTemplate se bazează pe clasa Firestore pentru a efectua aceste operațiuni, dar furnizează o interfață mai ușor de utilizat și se concentrează pe **maparea datelor din Firestore în obiecte Java și invers.**

În esență, Firestore este clasa care gestionează comunicarea directă cu Firestore, în timp ce FirestoreTemplate este o componentă de nivel superior care oferă un set de operațiuni ușor de utilizat pentru efectuarea de operațiuni CRUD în Firestore.

În ceea ce privește operațiile CRUD în Firestore, acestea sunt efectuate utilizând atât Firestore (pentru operații de bază), cât și FirestoreTemplate (pentru operațiuni mai ușoare și mai abstractizate). Cu toate acestea, FirestoreTemplate oferă funcționalități mai extinse pentru a simplifica operațiunile de mapare a datelor și pentru a gestiona colecțiile de documente.

**Firestore vs FirestoreStub**

**În general, într-o aplicație de producție, ar trebui să utilizați o instanță Firestore pentru a comunica cu serviciul Cloud Firestore, nu un FirestoreStub. FirestoreStub este folosit în principal în scopuri de testare și nu este necesar într-o aplicație de producție.**

**Firestore este clasa principală pentru comunicarea cu Cloud Firestore și asigură toate funcționalitățile necesare pentru a efectua operațiuni în bază de date Firestore. FirestoreStub este o interfață sau o clasă de stub (imitare) care poate fi folosită în teste pentru a simula interacțiunile cu Firestore, dar nu ar trebui să fie utilizată în producție.**

**Atentie aici ca imi trebuie dependenta in maven:**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

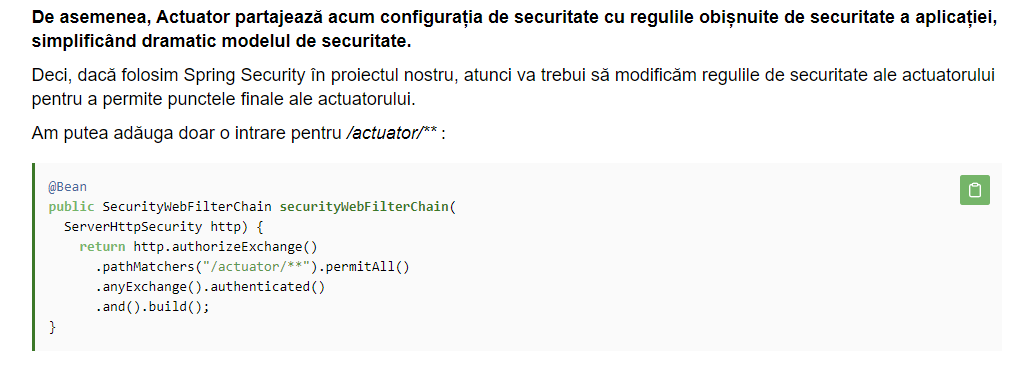
</dependency>

**Si in application.proprieties = management.endpoints.web.exposure.include=beans**

**Altfel nu va merge programul ca nu imi va gasi toate bean-uri corespunzatore pentru Firestore (mai exact FirestoreTemplate).**

**Explicatie:**

* ***Actuator :***
* Aduce aplicatiei noastra caracteristici pregatite pentru productie
* Monitorizarea aplicatiei noastra, stranegrea de valori si intelegerea traficului sau a starii bazei de date devine triviala cu aceasta dependenta.
* Beneficul acestei biblioteci este ca putem obtine instrumente de nivel de productie fara a fi necesar sa implementam noi aceste caracteristici.
* Actuatorul expune în principal **informații operaționale despre aplicația care rulează** - sănătate, valori, informații, dump, env etc. Utilizează puncte terminale HTTP sau bean-uri JMX pentru a ne permite să interacționăm cu el.
* Noile versiune accepta si modulul webflux si nu doar MVC
* **Spre deosebire de versiunile anterioare, Actuator vine cu majoritatea punctelor finale dezactivate.**
* Astfel, singurele două disponibile implicit sunt /health și /info .
* Dacă vrem să le activăm pe toate, am putea seta management.endpoints.web.exposure.include=\* . Alternativ, putem enumera punctele finale care ar trebui să fie activate.



Dependența Spring Boot Actuator și configurarea management.endpoints.web.exposure.include oferă funcționalități de monitorizare, administrare și expunere a informațiilor despre starea aplicației. Iată mai multe detalii despre fiecare aspect:

Dependența Spring Boot Actuator: Actuator este un modul al framework-ului Spring Boot care furnizează o serie de endpoint-uri predefinite și funcționalități pentru a monitoriza și administra aplicația dvs. Aceste funcționalități sunt utile în special în medii de producție pentru a verifica sănătatea, metricile și alte aspecte ale aplicației.

* management.endpoints.web.exposure.include=beans: Această configurare este utilizată pentru a specifica care endpoint-uri Actuator ar trebui să fie expuse și accesibile prin interfața web. În acest caz, management.endpoints.web.exposure.include=beans indică că endpoint-ul /actuator/beans va fi expus și poate fi accesat printr-o cerere HTTP.
* management.endpoints.web: Acesta este un subsistem de configurare care se ocupă de expunerea endpoint-urilor Actuator prin HTTP.
* exposure: Acesta specifică endpoint-urile care ar trebui să fie incluse în expunere (adică accesibile prin HTTP).
* include: beans: Această parte a configurării specifică endpoint-ul /actuator/beans pentru a fi inclus și expus prin HTTP.

Endpoint-ul /actuator/beans furnizează o listă de toate bean-urile gestionate (componentele) din aplicația dvs. Spring. Un "bean" în Spring este o componentă gestionată care poate fi injectată și utilizată în aplicație. Endpoint-ul /actuator/beans oferă o imagine de ansamblu a componentelor gestionate, inclusiv numele și tipul lor.

Această funcționalitate poate fi utilă pentru depurare, depanare și înțelegerea structurii aplicației dvs. Spring. Prin activarea acestui endpoint și accesarea lui prin intermediul unei cereri HTTP, puteți obține o imagine de ansamblu asupra componentelor gestionate din aplicația dvs.

***Este important să menționați că utilizarea Actuator în mediul de producție poate necesita o atenție specială pentru securitate, deoarece expunerea unor anumite informații sau funcționalități prin Actuator poate avea impact asupra securității aplicației. De aceea, este important să configurați Actuator cu atenție și să restricționați accesul la endpoint-uri în funcție de cerințele dvs. de securitate.***