



di Francesco Benincasa

Introduzione

Nel realizzare il progetto **Files-On-Cloud**, si è cercato di essere il più aderente possibile alle specifiche di progetto. Si sono apportati i seguenti aggiunte/cambiamenti rispetto alle specifiche iniziali:

- Tutti gli utenti sono caratterizzati da uno username in formato email. Questo cambio di requisito si è reso necessario al fine di utilizzare Firebase come sistema di autenticazione.
- Per gli utenti Consumer esiste un campo Codice Fiscale. In realtà, pur avendo un vincolo di unicità per tutti gli user di tipo *consumer*, questo campo deve essere visto come un semplice codice univoco. Non vengono eseguiti controlli di validità di tale campo come codice fiscale.
- Al fine semplificare la dimostrazione del funzionamento dell'applicazione, al momento della creazione dei vari tipi di utenti, alcuni campi sono prevalorizzati. Ovviamente l'utente è libero di cambiarli a proprio piacimento prima di salvare. Ad esempio, ogni utente in fase di creazione ha come password: *password* e come
- e indirizzo di posta elettronica uxcesco@gmail.com (per rendere la verifica del funzionamento del sistema più semplice).
- Sono stati inseriti dei limiti in termini di dimensioni dei file da caricare di 1MB.
- Il report per gli amministratori di default prende come data iniziale quella del primo giorno del mese precedente e come data finale la data odierna.

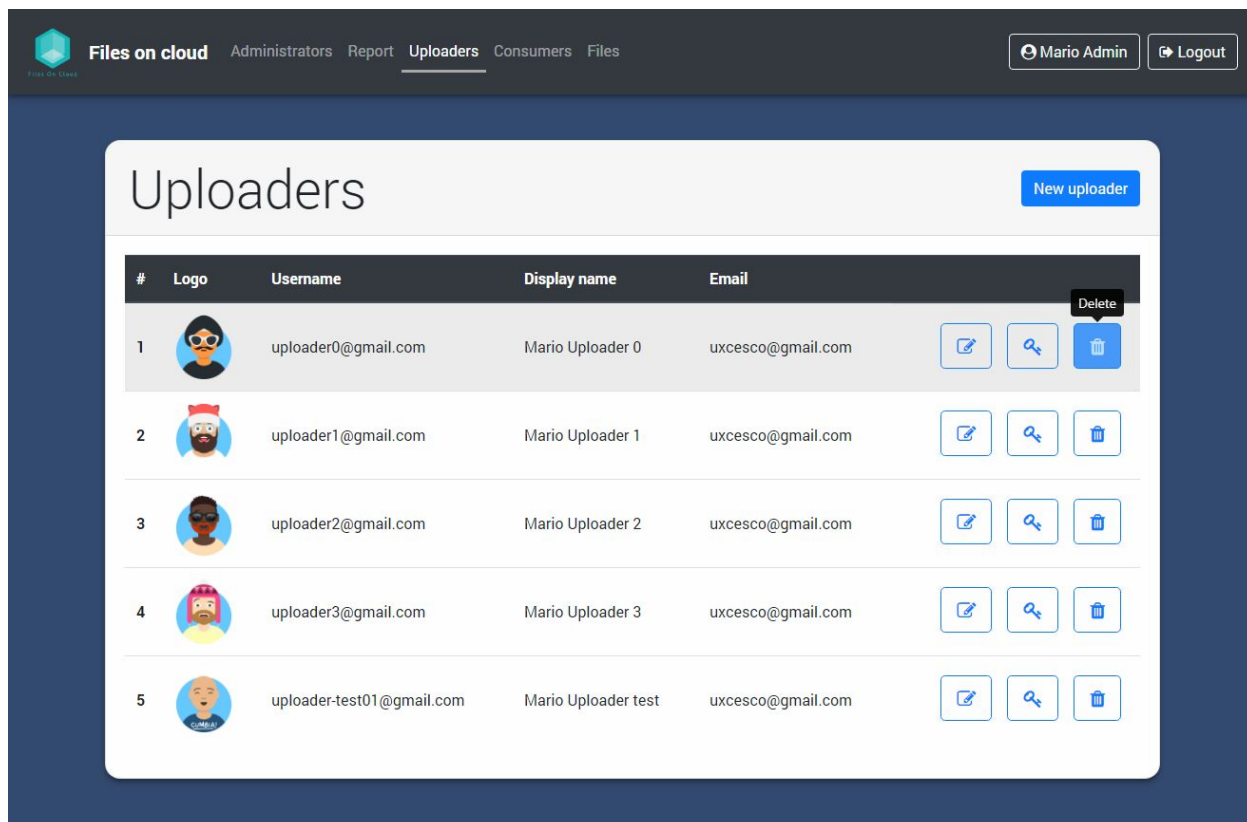
L'applicazione è visionabile all'indirizzo:

<https://programmazione-web-238419.appspot.com/>

I sorgenti sono disponibili su github:

<https://github.com/xcresco/files-on-cloud>

Di seguito è riportato uno screenshot dell'applicativo



Piattaforma tecnologica utilizzata

La soluzione software realizzata per essere eseguita sulla piattaforma cloud di Google. In particolare si e' utilizzato:

- Google App Engine: per ospitare l'applicazione web.
- Google SQL: per ospitare l'istanza di database MYSQL utilizzata da Files-on-cloud.
- Firebase Authenticator: per la gestione degli utenti e relativi login.
- Google storage per memorizzare i file da condividere.

Per l'invio di email ci si è basati sul sistema di mail offerto dalla piattaforma (che in locale non funziona).

Tecnologie utilizzate

Si sono utilizzate diverse tecnologie per realizzare il progetto. Di seguito sono riportate quelle principali suddivise per "area di competenza":

Linguaggi utilizzati

- Java per il backend
- Typescript per il frontend
- CSS
- HTML

Gestione del progetto e dei sorgenti

- Maven
- Git

Backend

- Spring Boot
- Spring
- Spring security
- Spring web
- JPA/Hibernate
- [FasterXML/Jackson](#): libreria per la conversione oggetti Java in JSON
- [Swagger](#): per la documentazione ed il testing dei servizi web realizzati con l'applicazione.
- Firebase Admin SDK

Frontend

- Angular 8
- Bootstrap
- [NG-Bootstrap](#)
- Font-awesome
- Firebase Client
- Node.JS (npm)

Tools ed ambienti di sviluppo utilizzati

- Eclipse (per il backend e la gestione del progetto in maven)
 - Plugin per Integrazione con Google Cloud SDK
- Webstorm (per la gestione del progetto Angular)
- DataGrip
- Google Cloud SDK
- Maven
- Atom

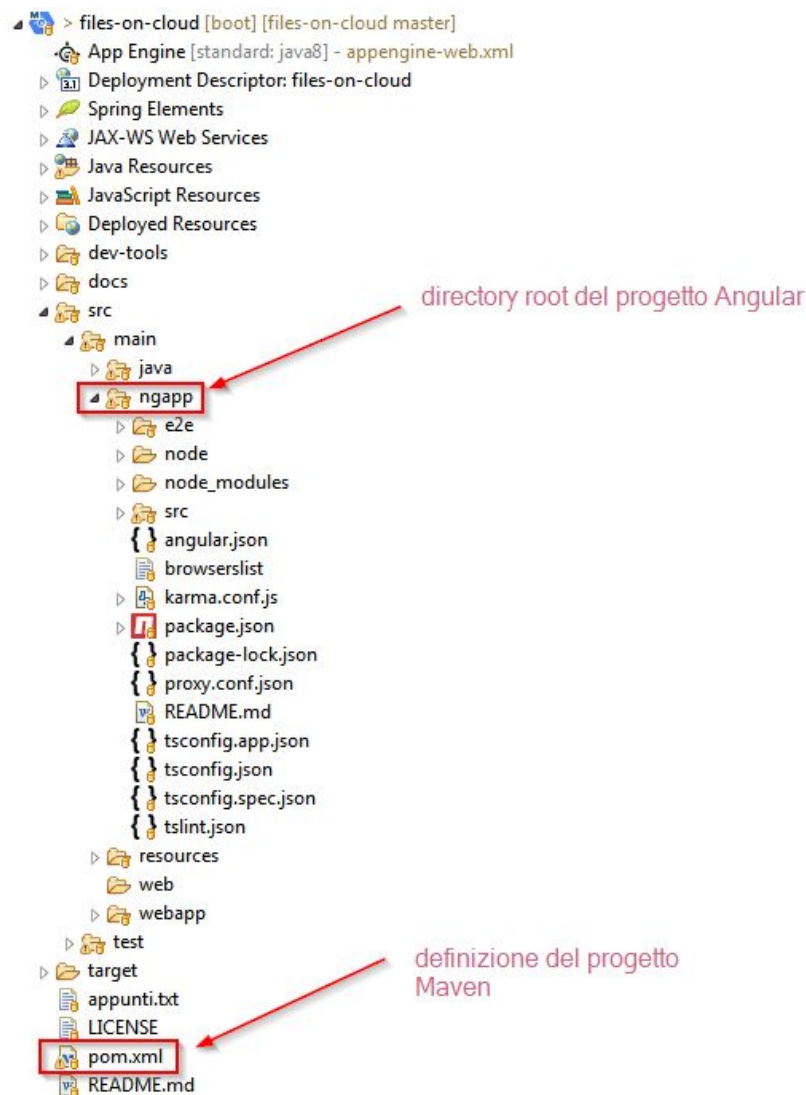
Gli strumenti di sviluppo sono stati scelti in quanto già utilizzati in altri progetti. Le scelte tecnologie sono state guidate dal fatto che:

- Sono tra quelle più usate attualmente nel mercato
- In buona parte già note

Configurazione cloud

Per la realizzazione del progetto è stato creato un apposito account gmail con il quale si sono creati due progetti, uno su <https://console.cloud.google.com> ed uno su <https://console.firebase.google.com/> .

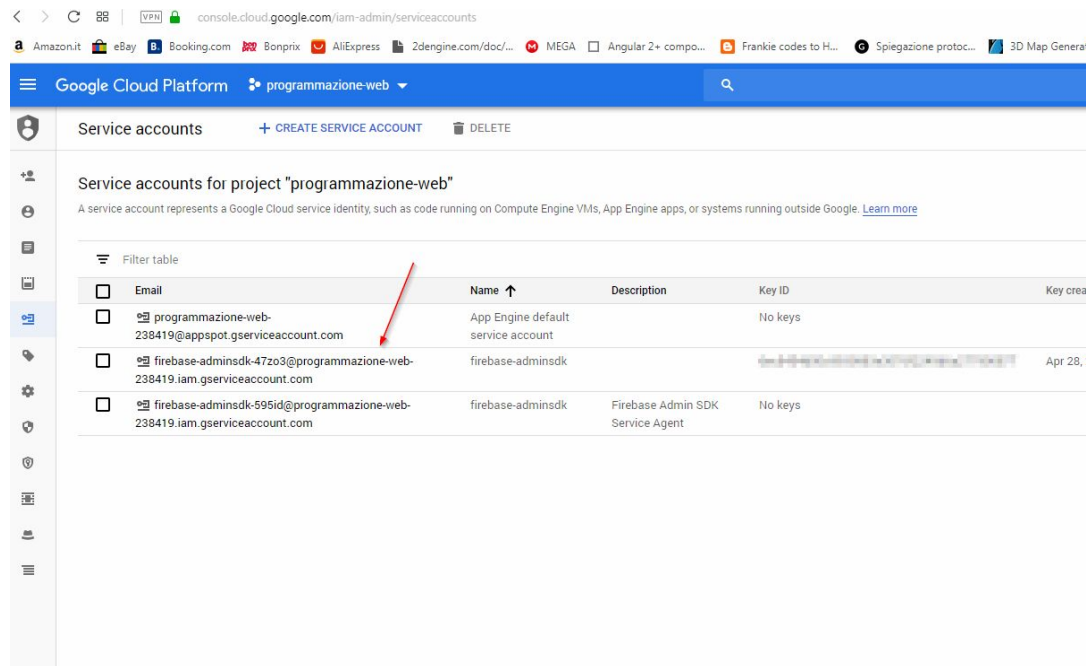
Note sulla configurazione dell'ambiente di sviluppo



Il progetto è suddiviso in due moduli:

- Il modulo “**Maven**”, che si occupa del progetto Java e della build complessiva del progetto. Al suo interno vengono utilizzate le credenziali di servizio generate mediante la console del progetto su cloud.

<https://cloud.google.com/iam/docs/creating-managing-service-account-keys>



Il file JSON che ospita la service key è reso disponibile all'applicazione web sotto forma di variabile d'ambiente `GOOGLE_APPLICATION_CREDENTIALS` (contiene il path del file).

```

1 serviceAccountKey.json
2 {
3   "type": "service_account",
4   "project_id": "programmazione-web-238419",
5   "private_key_id": "firebase-adminsdk-47zo3@programmazione-web-238419.iam.gserviceaccount.com",
6   "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoGBAL\n-----END PRIVATE KEY-----",
7   "client_email": "firebase-adminsdk-47zo3@programmazione-web-238419.iam.gserviceaccount.com",
8   "client_id": "firebase-adminsdk-47zo3@programmazione-web-238419.iam.gserviceaccount.com",
9   "auth_uri": "https://accounts.google.com/o/oauth2/auth",
10  "token_uri": "https://oauth2.googleapis.com/token",
11  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
12  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-47zo3%40programmazione-web-238419.iam.gserviceaccount.com"
13 }

```

Questo file verrà recupero dall'app in fase di avvio mediante il codice contenuto nella classe *WebStartup*.

```

@WebListener
public class WebStartup implements ServletContextListener {

    public void contextInitialized(ServletContextEvent sce) {
        try {

            FirebaseOptions options;

            options = new FirebaseOptions.Builder()
                // prende le credenziali da GOOGLE_APPLICATION_CREDENTIALS
                .setCredentials(GoogleCredentials.getApplicationDefault())
                // .setCredentials(GoogleCredentials.fromStream(serviceAccount))
                .setDatabaseUrl("https://programmazione-web-238419.firebaseio.com").build();

            FirebaseApp.initializeApp(options);

        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void contextDestroyed(ServletContextEvent sce) {

    }

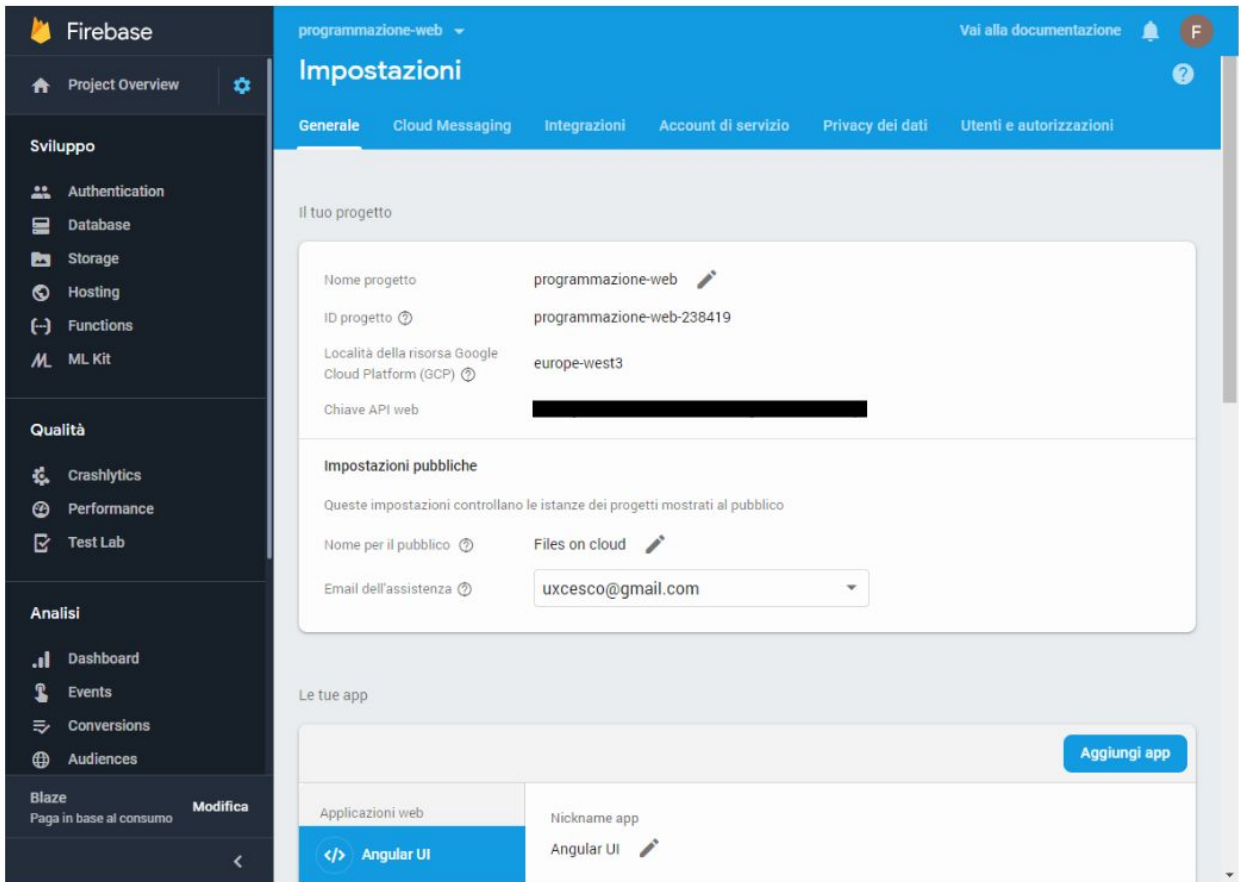
}

```

- Il modulo “**Angular**” ospita il codice del front-end realizzato in Angular. Questo modulo richiede la configurazione del progetto su firebase.

Le istruzioni su come ottenere le credenziali da usare con Firebase sono presenti al seguente url:

<https://firebase.google.com/docs/web/setup>



Le credenziali ottenute devono essere presenti nei file *environment.prod.ts* ed *environment.ts*.

