

# Contacts Readme

L'applicazione Contacts è stata sviluppata da Alessio Dinatale per il progetto richiesto da Turing s.r.l in fase di colloquio.

## - Sviluppo tecnico

Vengono elencati alcuni componenti e punti tecnici delle applicazioni:

### Webapp Angular:

- Utilizzo del pattern **MVC**.
- Utilizzo del pattern **DTO**.
- Utilizzo dei **Router Guards**.
- Utilizzo della nuova metodologia **Standalone**.
- Utilizzo degli **Observer** per ottimizzazione.
- Utilizzo di **Interceptor** per la gestione del **JSON Web Token** (Gestione della sessione)
- Utilizzo degli **Angular Reactive Forms**.
- Utilizzo della libreria grafica **PrimeNG**.
- Utilizzo della libreria grafica **Lottie**.
- **Altro**

### Springboot App:

- Utilizzo del pattern **Controller-Service-Repository**.
- Utilizzo del pattern **Object-Relational-Mapping**
- Utilizzo delle librerie **Hibernate/JPA** per **ORM**.
- Utilizzo del pattern **DTO**.
- Utilizzo di validazione tramite **JAVA Validation**.
- Utilizzo di **Bean (uso IoC)**
- Utilizzo di classi **Configuration**
- Utilizzo del pattern **Spring Security**
- Utilizzo del pattern **JWT**
- Utilizzo di token **JWT** per autorizzazioni sulle API.
- Utilizzo di token **JWT** per mantenimento della sessione.
- Utilizzo di controllo della rotta basato su **RBAC**.
- **Altro**

Sono state utilizzate alcune buone norme di programmazione (**Split Methods, Singola Responsabilità, Dependency Injection, Singleton per Service ove necessario**) per quanto consentito dal tempo massimo disponibile, difatti tante altre buone norme considerate, non sono state implementate per velocizzare lo sviluppo (divisione in sotto componenti grafici in **Angular**, logica d'ottimizzazione per **risparmiare chiamate al backend** quando non necessario, parziale gestione di **Responsive** da parte dalla UI).

## - Altre tecnologie utilizzate

Si segnalano le seguenti tecnologie utilizzate:

**Postman** per il testing degli endpoint backend

**Docker** per containerizzare le diverse applicazioni e gestire la loro comunicazione.

## - Come installare l'applicazione

L'applicazione è composta da 3 entità principali:

- Webapp Angular (Frontend)
- Spring app (Backend)
- MySql DB

Per mandare in esecuzione l'intero sistema ed effettuare i dovuti test è stata implementata una soluzione che prevede l'utilizzo della tecnologia **Docker**

### Run tramite Docker

Per mandare in esecuzione l'intero sistema attraverso docker eseguire il comando nella root della cartella scaricata in cui sono presenti le applicazioni:

```
docker compose up
```

nella cartella che contiene le applicazioni Angular e Spring.

questo avvierà le 3 entità in container separati generando:

1 container - Database MySQL (con il nostro database Contacts al suo interno) **(ESPOSTO SULLA PORTA - 15900)**

2 container - Springboot App **(ESPOSTO SULLA PORTA - 16900)**

3 container - Webapp Angular **(ESPOSTO SULLA PORTA - 16990)**

entrambi i container sono configurati lavorando in una network privata.

per accedere all'app sarà sufficiente attendere l'avvio dei container ed entrare nell'applicazione tramite il seguente url: **http://localhost:16990**

## **- Come utilizzare l'applicazione**

L'architettura proposta per il progetto prevede un'applicazione full stack. Questo identifica alcune scelte necessarie (non previste direttamente nell'analisi dei requisiti) durante lo sviluppo, per garantire un utilizzo fluido e pratico.

Per utilizzare l'applicazione è necessario creare un nuovo account nella pagina di registrazione, questo ci consentirà di effettuare un Login successivo che ci permetterà di aver accesso al nostro Token personale per effettuare le operazioni richieste.

Il backend validerà e proteggerà difatti i suoi endpoints, analizzando il token, verificandolo e comprendendo se l'utente cerca di operare sulle proprie risorse o meno.

Il token inoltre servirà a mantenere attiva la sessione dell'utente, che potrà dopo aver effettuato il login, utilizzare l'app senza dover ripetere l'autenticazione (Silent Login)

Si consiglia per questioni di compatibilità tipiche dello sviluppo web, l'utilizzo di browser moderni per garantire la massima resa (es. Google Chrome).

Grazie dell'attenzione, Alessio Dinatale.