

Relazione Progetto Sistemi Distribuiti: Supermercato

Emilio Casella matr.204898

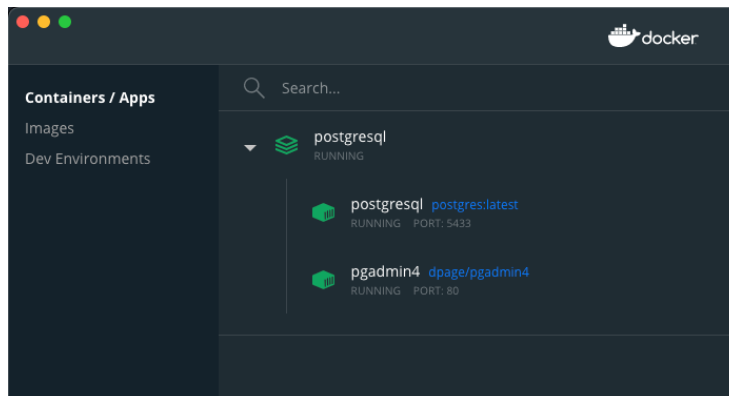
July 5, 2021

1 Introduzione

Ho deciso di implementare, ai fini della materia in esame, una web app relativa ad un Supermercato, utilizzando le tecnologie Spring Boot (lato back end) e Angular (lato front end).

Il database utilizzato è un dump di un programma inerente la gestione di un supermarket, che si trova facilmente in rete, dal quale sono state eliminate alcune tabelle non necessarie e apportate alcune modifiche per renderlo congeniale al mio scopo.

Sono stati implementati 2 contenitori in Docker, il primo contenente il DBMS PostgreSQL, mentre il secondo contiene PGAdmin 4, che consente di gestirlo in maniera appropriata.



2 Autenticazione

Si è scelto di utilizzare l'autenticazione di base fornita da Spring Boot, creando due tipologie di accesso:

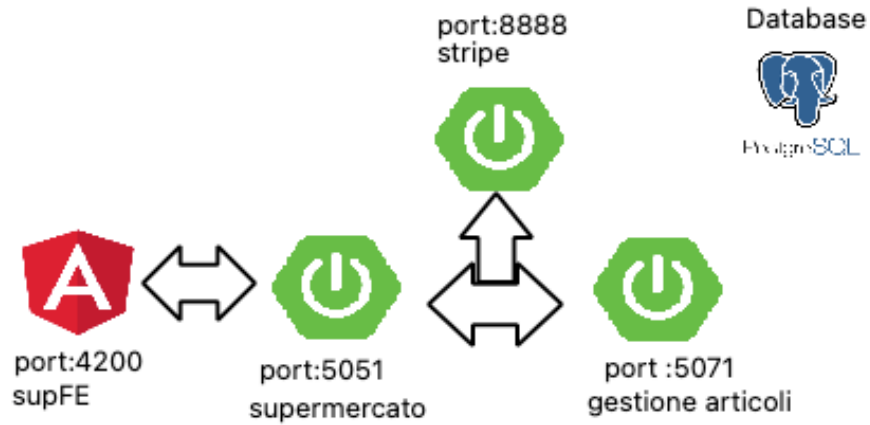
- **Admin:** Si occupa della gestione degli articoli e delle promozioni in essere, non avendo la possibilità di effettuare acquisti, non essendo necessario ai fini del ruolo assunto.
- **Utente:** Effettua esclusivamente procedere all'acquisto di prodotti presenti nel DB, non potendo accedere ad altre operazioni.

3 Architettura

Lato back end, sono stati implementati tre microservizi che si occuperanno di gestire il tutto:

- **Supermercato:** Lato utente, si occupa della possibilità di effettuare degli acquisti, tramite l'implementazione di un carrello.
Lato admin, si occupa della gestione degli articoli, garantendo la ricerca di un articolo per codice, codice a barre oppure per descrizione; garantendo, inoltre, le operazioni di inserimento-modifica e rimozione di un articolo.
- **Gestione supermercato:** Si occupa della gestione dei prezzi e delle promozioni relative agli articoli, garantendo le operazioni di inserimento-modifica di una promo legata ad un articolo presente nel db con relativa possibilità di rimozione, ad esclusivo uso dell'admin.
- **Stripe:** Si occupa di indirizzare all'utilizzo di un pagamento esterno, tramite il portale stripe, ai fini dell'acquisto dei prodotti presenti nel carrello di un utente.

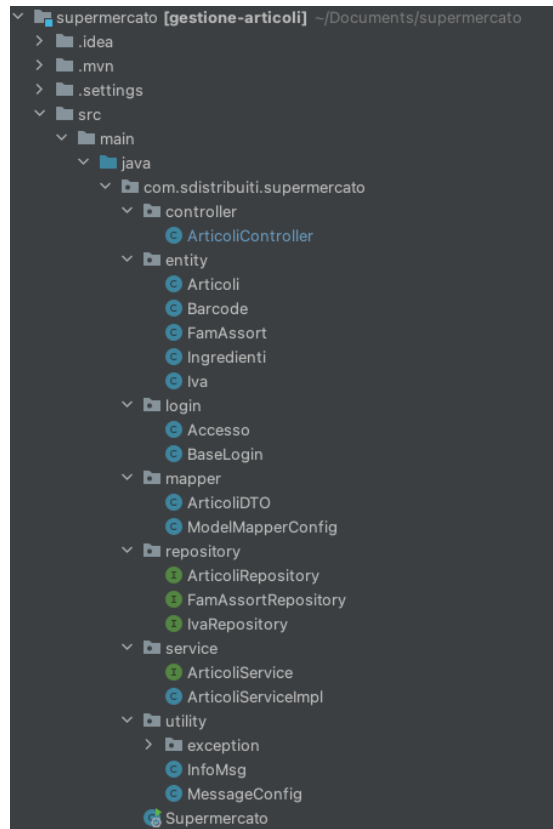
Tutti questi servizi, si interfacciano con il front end angular, che tramite l'utilizzo di librerie come angular material e bootstrap, permette di gestire il tutto.



4 Struttura dei microservizi

La struttura dei microservizi prevede una divisione tipica di questo ambito, implementando un package per gli entity contenente la classe Articoli,Barcode etc.,e vari package contenti controller,services etc.

Si specifica, che si è deciso di utilizzare il ModelMapper, che rende molto semplice la gestione delle varie entity utilizzate. Si noti la presenza di un package utility, contenente tutto il necessario per la gestione delle eccezioni, e del package Login, contenente il servizio di autenticazione.



Il microservizio di gestione articoli ha una divisione simile, mentre il microservizio stripe implementa esclusivamente le classi model e controller, necessarie al collegamento al servizio esterno di pagamento dedicato.

5 Testing

Ai fini del testing dei vari microservizi, sono stati utilizzati JUnit Test (per la ricerca di articoli e promozioni, con e senza paginazione) e dei test manuali, tramite Postman, delle varie chiamate di POST, GET e DEL di articoli o promozioni.

```
public class SupermercatoTest
{

    @Autowired
    private ArticoliRepository articoliRepo;

    @Test
    public void TestFindByCodArt() throws Exception
    {
        Assertions.assertThat(articoliRepo.findByCodArt("7999182"))
            .extracting(Articoli::getDescrizione)
            .isEqualTo("GHIACCIO 2LT IS MORI");
    }

    @Test
    public void TestCatchByDescrizioneLikeNoPag()
    {
        List<Articoli> items = articoliRepo.catchByDescrizioneLike("GHIACCIO%");
        assertEquals( expected: 1, items.size());
    }

    @Test
    public void TestCatchByDescrizioneLikePag()
    {
        List<Articoli> items = articoliRepo.findByDescrizioneLike( descrizione: "GHIACCIO%", PageRequest.of( page: 0, size: 1));
        assertEquals( expected: 1, items.size());
    }

    @Test
    public void TestCatchByBarcode() throws Exception
    {
        Assertions.assertThat(articoliRepo.catchByBarcode( ean: "9989988422990"))
    }

}
```

Home Workspaces Reports Explore Search Postman

My Workspace New Import SupermercatoTest / api/articoli/inserisci

POST http://localhost:5051/api/articoli/inserisci

Params Authorization Headers Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "codArt": "123456789",
3   "descrizione": "Pane Fresco F.lli Rossi",
4   "cat": "BGT",
5   "codStat": "null",
6   "pzCart": 0,
7   "pesoNetto": 0.1,
8   "idStatoArt": "1",
9   "dataCrea": "2018-02-17",
10  "prezzo": 0.8,
11  "barcode": {
12    "barcode": "9000099400994",
13    "idTipoArt": "CP"
14  },
15  "ingredienti": null,
16  },
17 }
```

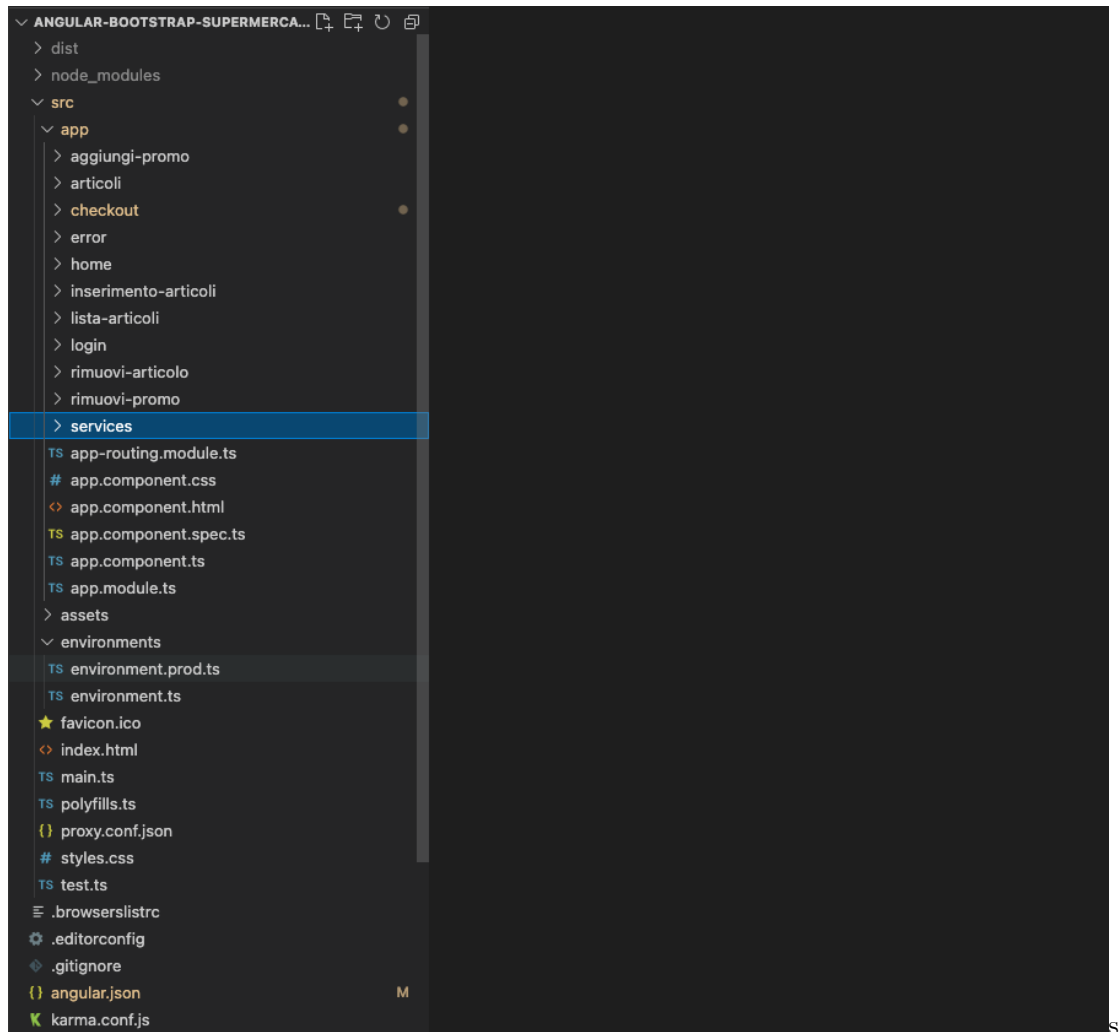
Response

Hit Send to get a response

Find and Replace Console Bootcamp Runner Trash

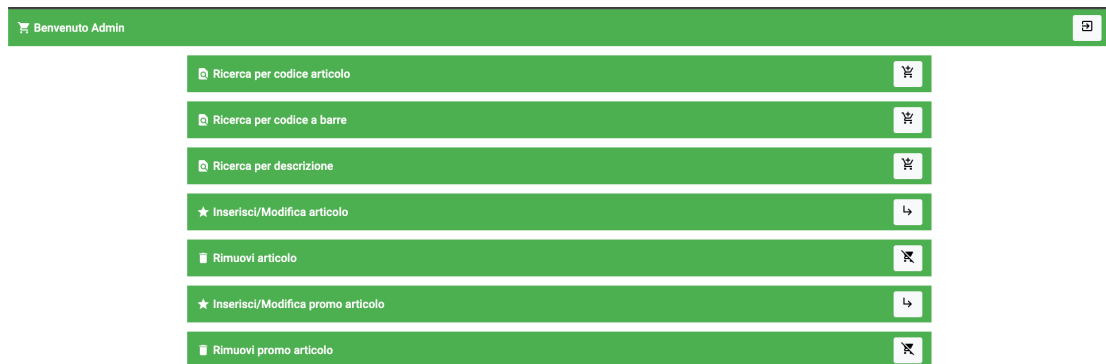
6 Struttura frontend

Il front end è stato sviluppato tramite l'utilizzo di Angular 11, Bootstrap 4 e Angular Material. Sono stati implementati vari pacchetti molto esplicativi in merito al componente offerto.

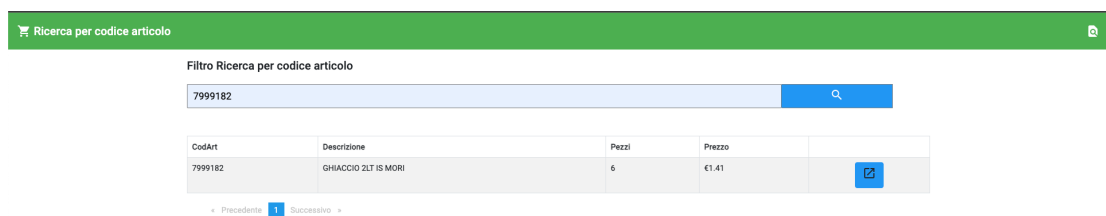


7 Risultato finale

Ecco come appare la schermata Admin.



Si può scegliere tra le varie tipologie di ricerca, che presentano tutte lo stesso layout



e vedere il dettaglio di un articolo.

Dettaglio Articolo

Codice	Descrizione	Unità di Misura	Pezzi Disponibili
7999182	GHIACCIO ZLT IS MORI	PZ	6
Peso Netto	Stato	Data Creazione	Prezzo
2	1	2015-07-03	1.41

Iva	Aliquota
IVA RIVENDITA 22%	22

Categoria
SURGELATI

Codice a barre
9989980422990

Si può inserire un articolo (la funzione vale anche come modifica),

Inserimento Articolo

Codice Articolo:
min 5 max 20 numeri

Descrizione Articolo
min 6 parole max 80

Unità di Misura

Iva

Descrizione Ingredienti:

Peso Netto:
0

Barcode:
min 8 max 13 numeri

Pezzi:
0

Famiglia Assortimento
-1

Procedi

oppure rimuoverlo tramite il codArt;

Rimozione Articoli

Filtro Rimozione Articoli

Inserisci codice Articolo da rimuovere...

inserire o rimuovere una promozione.

Inserimento Promo Articolo

Codice Articolo:
min 5 max 20 numeri

Prezzo :

0

Sconto :

0

Procedi

Rimozione Promo Articoli

Filtro Rimozione Promo Articoli

Inserisci codice Articolo da rimuovere...

Ecco come appare, invece, la schermata utente.
Si cercano i prodotti desiderati per descrizione,

Filtro Ricerca prodotti da acquistare

GHIACCIO

CodArt	Descrizione	Pezzi	Prezzo	
7999182	GHIACCIO 2LT IS MORI	6	€1.41	
006544101	I GENUINI 10 GHIACCIOLI ASSORT.650 GR	6	€2.28	
033748801	CUKI CUBETTI GHIACCIO X10	48	€1.03	
057146201	GIOSTYLE GHIACCIO ICEBERG 2X400	8	€1.70	

« Precedente 1 Successivo »



si riempie il carrello e si procede all'acquisto, approssimando l'importo all'intero superiore.

Carrello

Descrizione Articolo	Pezzi	Prezzo	
I GENUINI 10 GHIACCIOLI ASSORT.650 GR	1	€2.28	
PONTI INSALATA PER PASTA GR.370	1	€3.07	

Totale Carrello5.35

Close

Tramite Stripe, si inseriscono i dati di pagamento e di spedizione. Se il tutto va a buon fine, si riceve una mail riepilogativa dell'ordine effettuato e si effettua automaticamente il logout.

← Emilio Casella **TEST MODE**

Carrello Spesa
6,00 €

Powered by stripe | Termini Privacy

Dati di spedizione

Email

Indirizzo di spedizione

Nome

Italia

Inserisci indirizzo

[Inserisci l'indirizzo manualmente](#)

Dettagli di pagamento

Informazioni sulla carta

1234 1234 1234 1234   

MM / AA CVC

☒ L'indirizzo di fatturazione è uguale all'indirizzo di spedizione

Paga 6,00 €