

Presentazione Codice Progetto LabTv di Giovanni Manichetti

Sommario

- **Home page**
- **Api Service**
- **Ritorno alla Home TS**
- **List components**
- **Scroll**
- **List Move Component/Output**
- **Passaggio dell'Id alla Home**
- **Detail Movie Component TS**
- **Video**
- **Film Simili**
- **Richiesta di acquisto di un film**
- **Pulsante di Ricerca**
- **Pulsante di Ricerca**
- **List Movie Component/Ricerca**
- **Login**
- **Area Riservata**
- **Navbar**

Home page

La pagina Home lato HTML gestisce:

Un campo di input per la ricerca dei film,

Dei messaggi di errore,

Due componenti.

Il primo componente è List-Movie, il secondo componente Detail-Movie.

I dati vengono scambiati tramite input output

Pagina Home lato TS

Qui ci sono le chiamate al server:

Lista Movie

Dettaglio Movie

Video Movie.

La lista movie ha una particolarità.

Prima di vedere la particolarità ci spostiamo nell' Api Service.

Api Service

Qui sono presenti i tre metodi di richiesta già elencati, in più una proprietà che immagazzina la lista dei film.

A scopo di test, per non effettuare tante chiamate al server, i dati sono stati salvati in tre proprietà della classe.

Due proprietà sono state commentate (in realtà anche la terza), una è rimasta attiva ma vuota.

Ritorno alla Home TS

All'avvio della Home, c'è il controllo se la proprietà nell'Api Service è piena, se non lo è viene effettuata la richiesta al server.

La Response, lavorata invia i dati nella proprietà dell'Api Service e nella proprietà della Home component, proprietà che verrà passata a list component.

Ovviamente se la proprietà nell'Api Service è piena, non viene effettuata la richiesta al server, ma il trasferimento dei dati dalla proprietà dell'api service alla proprietà della Home component.

List Move components

Riceve tutta la lista, prendi i primi 25 elementi e li inserisce in un Array di visualizzazione.

Un contatore tiene traccia per i prossimi inserimenti.

Scroll

Con la tecnica dello scroll infinito, si caricano altri 25 film fino a 100.

Installato il modulo che permette lo scroll, il modulo viene importato nel file app.module.ts.

Vengono implementate le funzioni di scroll sia lato HTML che lato TS

List Move Component/Output

List Movie component ha un solo output, l'ID del film cui si richiede il dettaglio.

Passaggio dell'Id alla Home

La Home riceve l'ID del film cui si richiedono le info in dettaglio, fa richiesta al server, e la Response, in caso positivo, viene inviata al Detail Movie Component.

Detail Movie Component TS

All'avvio, il file TS, prepara uno script che viene iniettato nel HTML per un probabile IFrame. Il Detail Movie Component riceve i dati di dettaglio del film che li mostra all'utente.

Qui gli output sono molti:

un output di "ritorno alla lista" (Show List collegato al pulsante omonimo),

un output che invia un ID per la richiesta del video (richieste effettuata sempre dalla Home),

un output per la richiesta di acquisto (ne parlerò in dettaglio più avanti),

L'ultimo output è collegato alla lista dei film simili (ne parlerò tra poco).

Video

Viene inviato l'ID del film in dettaglio alla home, la home richiede i dati al server, ricevuti i dati li restituisce al Detail Movie complement che tramite una modale mostra il video trailer.

Film Simili

Abbiamo parlato dei film simili al video cui abbiamo richiesto il dettaglio, ma se l'utente vorrebbe vedere più info su uno di questi film simili?

In questa versione un output trasmette il titolo alla Home, la Home trasmette il titolo al List Movie Component quale fa la ricerca tra i 100 film ricevuti, e se è presente mostra una card del film, in alternativa mostra un messaggio di film non trovato.

In una ipotetica versione futura/successiva, anziché passare il titolo, si può passare l'ID alla Home.

La Home farà la richiesta al server inviando il dettaglio del film allo stesso componente. Questo riduce il numero di passaggi, ma aumenta il numero di chiamate al server.

Richiesta di acquisto di un film

Uno output, senza parametri, invia la richiesta alla Home.

La Home ha già i dati del film in dettaglio.

La Home fa un controllo, se l'utente non è loggato mostra un alert che invita l'utente ad effettuare il Login o la registrazione.

Se l'utente è loggato, fa un controllo del film se già presente tra i preferiti e mostra un messaggio che informa l'utente che il film è già presente nella sua area riservata, in caso contrario, inserisce il film nell'array dei preferiti, e mostra all'utente il messaggio di inserimento del film nell'area riservata.

Pulsante di Ricerca

Nella Navbar è presente il pulsante di Ricerca, che permette all'utente di cercare un film tramite titolo.

Come funziona.

Aggiungi al Path la parola ricerca.

Questo nuovo Path, viene sempre riconosciuto dall' App-routing.module.ts e reindirizza sempre alla Home.

List Movie Component/Ricerca

La Home mostra il componente List Movie.

Il componente List Movie fa un controllo dell'indirizzo URL, se la parola "Ricerca" è presente, fa apparire il campo di input dove l'utente potrà inserire il titolo del film da ricercare.

La ricerca avviene tramite un Filter.

Dall'array che contiene tutta la lista dei film, viene filtrato nell'array di supporto ciò che l'utente sta cercando, e successivamente mostrato.

Un "ngModelChange" permette una ricerca e visualizzazione più rapida.

Login

Dopo aver parlato delle richieste fatte ad un server remoto, passiamo alle richieste di un server locale.

Cliccando su login appare la form di Login con la possibilità di accedere alla form di registrazione.

Qui è presente un componente padre (Login component) e due componenti figli (login form component - register form component).

Il componente Patria riceve e trasmette dati ai componenti figli.

Esegue anche la richiesta di accesso al server memorizza il token e il nome dell'utente nell'area storage, in caso tutto sia andato bene.

In alternativa, mostra dei messaggi di errore..

Le informazioni nello storage sono utili per il componente dell'area riservata e per la navbar.

Area Riservata

Verifica che l'utente è il login, in caso contrario reindirizza l'utente per il login.

Ad utente loggato, verifica se l'array favouriteMovie è pieno, mostrando il contenuto sotto forma di Cards, con la possibilità di rimuoverli dalla dallo stesso array.

In caso contrario mostra un messaggio che non ci sono film acquistati e invita l'utente ad acquistare un film.

Navbar

Contro la se l'utente è loggato cambiando alcune cose; tipo il pulsante login/registrati che diventa logout, e il nome dell'utente sotto l'icona di login.

Questo grazie al metodo "ngDoCheck"