

Computer Grafica 2023-2024

Esercitazione 5

React: ReactRouter

Utilizzare il linguaggio typescript e i componenti React di Material UI per realizzare le seguenti funzionalità.

Cominciare installando le librerie di Material UI per TS e quelle di React Router Dom

```
npm install @mui/material @mui/icons-material @emotion/react @emotion/styled
```

```
npm install --save @types/material-ui
```

```
npm install react-router-dom
```

Obiettivo dell'esercizio è realizzare un'applicazione interattiva che consenta di ottenere le informazioni per confezionare un cocktail partendo dalla scelta dell'ingrediente di base che lo costituisce. Interagendo con l'applicazione, l'utente potrà selezionare uno specifico cocktail e accedere alle informazioni di dettaglio sugli ingredienti che lo compongono e sulla sua preparazione, garantendo una navigazione fluida tra le diverse pagine presentate.

Tutte le informazioni mostrate saranno ricavate dal sito **thecocktaildb.com** che offre le API per accedere ai contenuti descritte alla pagina <https://www.thecocktaildb.com/api.php>

PASSO 1

Realizzare una pagina che mostra un elenco di superalcolici e permette di selezionarli.

Per ottenere l'elenco dei superalcolici eseguire una fetch come quella che segue all'indirizzo url evidenziato (dove setResult e setError sono due funzioni da implementare per aggiornare lo stato)

```
fetch('https://www.thecocktaildb.com/api/json/v1/1/list.php?i=list', {
  mode: 'cors',
})
  .then(v => v.json())
  .then(j => setResult(j.drinks))
  .catch(e => setError(e))
```

I dati restituiti elencano i prodotti in questa forma

```
{"drinks":
  [{"strIngredient1": "Light rum"},
  {"strIngredient1": "Applejack"},
  {"strIngredient1": "Gin"},
  {"strIngredient1": "Dark rum"},
  ... ]}
```

filtrare le chiavi "strIngredient" di ogni oggetto dell'array di dati restituito dalla chiave "drink" nel modo seguente `Object.keys(val).filter(...)` per selezionare 5 ingredienti che si preferiscono (o che iniziano con una determinata lettera).

Presentare gli ingredienti di base dei cocktail sotto forma di card con immagine e titolo descrittivo: i dati non restituiscono le immagini necessarie, ma si possono ottenere le thumbnails di dimensioni diverse (aggiungendo -Small o -Medium al fondo della url seguente)

<https://www.thecocktaildb.com/images/ingredients/gin-Small.png> (dove occorre sostituire la parola 'gin' con il nome dell'ingrediente cercato).



PASSO 2

L'applicazione deve permettere di navigare tra più pagine: quindi occorre impostare il routing.

Utilizzare React Router per impostare la navigazione tra le pagine. Nella versione 6.4 di React Router <https://reactrouter.com/en/main/routers/create-browser-router> è stata introdotta la funzione `createBrowserRouter` che consente di definire un oggetto router con un insieme di rotte definite (**vedi codice in rosso sottostante**).

Le route sono definite da `path`, `element`, `loader`, e `children`.

- Il campo `path` specifica il percorso dell'url associato a una determinata route.
- L'attributo `element` definisce il componente React che deve essere renderizzato quando la route corrisponde all'url.
- L'attributo `children` è un array che contiene le descrizioni di route figlie di quella in cui `children` è definito. I componenti associati a queste route sono renderizzati all'interno del componente associato alla route "contenitore".

Realizzare un componente esterno `<Template/>` che faccia da contenitore: esso deve presentare una barra di navigazione in alto e il componente `<Outlet/>`.

Come funziona il componente `<Outlet/>`?

Prendendo a esempio il codice che segue, quando l'URL corrisponde a `/`, sia `ParentComponent` che `ChildComponent` vengono renderizzati e `<Outlet/>` è il punto del componente genitore in cui vengono renderizzati gli elementi descritti nel componente figlio.

```
const ParentComponent = () => {
  return (
    <div>
      <h1>Title</h1>
      <Outlet />
    </div>
  );
};

const ChildComponent = () => {
  return (
    <div>
      <h2>Subtitle</h2>
    </div>
  );
};

const App = () => {

  const router = createBrowserRouter([
```

```

{
  path: '/',
  element: <ParentComponent />,
  children: [
    {
      index: true,
      element: <ChildComponent />,
    },
  ],
},
]);

return (
  <RouterProvider router={router} />
);
};

```

Oltre a path, element e children un altro attributo opzionale per definire il router è **loader**: specifica una funzione asincrona che carica i dati necessari per la route, è utile quando serve recuperare dati prima di renderizzare il componente.

```

const router = createBrowserRouter([
  {
    path: "/",
    element: <Ingredients />,
    loader: async function () {
      return fetch("https://www.thecocktaildb.com/api/json/v1/1/list.php?i=list ");
    }
  },
]);

```

Questa versione aggiornata di React Router consente quindi di specificare la chiamata asincrona che ritorna il risultato di una fetch all'interno della definizione stessa del router.

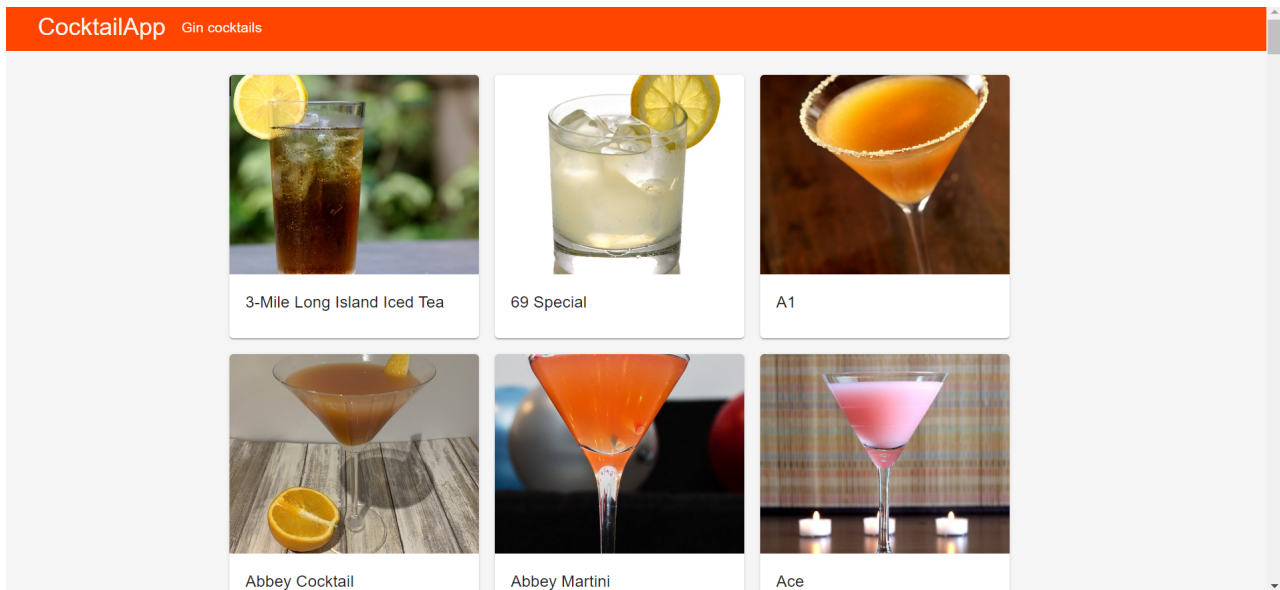
Impostare il router inserendo la route della prima pagina già realizzata.

PASSO 3

La navigazione deve permettere di raggiungere altre due pagine:

- La selezione di un superalcolico reindirizza alla pagina con l'elenco dei cocktail che lo usano come ingrediente.
- Ogni card dell'elenco dei cocktail è un link attivo che porta a una pagina di descrizione dello stesso.
- Dalla barra di navigazione in alto si può tornare alla homepage iniziale cliccando sul button "CocktailApp" oppure tornare all'elenco dei cocktail che hanno per ingrediente l'ultimo selezionato cliccando su "Gin cocktails", questo button non è attivo quando si è nella schermata iniziale.

Realizzare quindi la seconda pagina con le card dei cocktail e la barra di navigazione in alto come in figura:



Aggiornare il router:

- La url da cui ottenere i cocktail corrispondenti al superalcolico (e.g. Gin) è <https://www.thecocktaildb.com/api/json/v1/1/filter.php?i=Gin>
- La route della pagina dei cocktail deve variare a seconda del superalcolico selezionato e indicare il suo nome nel path (e.g. <http://localhost:3000/cocktail/Gin>).
- Rendere la chiamata API dinamica in base alla stringa inserita nel campo di input ``...i= ${req.params.cocktail}``, estrarre tale valore dal parametro di richiesta passato al loader.
- Quando si lancia la ricerca mostrare “loading” durante il caricamento, il testo dell’errore se la fetch restituisce errore o le card effettive se la ricerca va a buon fine.

PASSO 4

Mostrare in una terza pagina i dettagli (nome, descrizione, immagine, ingredienti) relativi al cocktail (e.g. Manhattan) selezionato ottenuti con una fetch all’indirizzo

<https://www.thecocktaildb.com/api/json/v1/1/search.php?s=Manhattan>

Le informazioni relative agli ingredienti sono ritornate come chiave valore (e.g. "strIngredient1":"Sweet Vermouth", "strIngredient2":"Bourbon"), in alcuni casi il nome dell’ingrediente non è indicato (e.g. "strIngredient3":"null"). Quindi per ottenere gli ingredienti filtrare le chiavi di ogni oggetto dell’array di dati `Object.keys(val).filter(...)` per verificare che includa `strIngredient` e che il valore relativo alla chiave sia diverso da `null`