# ContentOrdine

Il componente ContentOrdine gestisce la visualizzazione e la gestione degli ordini. Include funzioni per recuperare ordini da un'API. utilizza diverse librerie e componenti

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

## Il componente utilizza gli stati di React per gestire vari aspetti:

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

* orders: Array di ordini.
* currentPage: Pagina corrente per la paginazione.
* ordersPerPage: Numero di ordini per pagina.

## Questa funzione recupera gli ordini da un'API e aggiorna lo stato degli ordini.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* **API\_URL**: URL dell'API con il parametro userId.
* **fetch**: Esegue una richiesta HTTP GET all'URL specificato.
* **response.ok**: Controlla se la risposta è stata positiva.
* **response.json()**: Converte la risposta in formato JSON.
* **setOrders(data)**: Aggiorna lo stato orders con i dati ricevuti.

## Effetto useEffect

L'effetto viene eseguito al montaggio del componente per recuperare gli ordini.

Immagine che contiene Carattere, Elementi grafici, schermata, testo

Descrizione generata automaticamente

Esegue **fetchOrders** solo una volta quando il componente viene montato.

In sintesi :

* fetchOrders recupera gli ordini dall'API e aggiorna lo stato.
* useEffect esegue fetchOrders una volta al montaggio del componente.
* handleOnEndCreate recupera nuovamente gli ordini e aggiorna la pagina corrente quando un nuovo ordine è stato creato con successo.

return (

    <div className="header">

      <div className="title\_container\_ordine">

        <div className="title\_text">

          <h1>Ordini</h1>

          <h2>{nome} {cognome}</h2>

        </div>

      </div>

      <div className="welcome\_container\_ordine">

        <OrdineCreate onEndCreate= {handleOnEndCreate}/>

        <div className="banner\_container">

          <img src={banner} alt="Logo" />

        </div>

      </div>

      <div className="products\_container\_ordine">

        <div className="all\_products\_div">

          <h2>Tutti i prodotti</h2>

          <div className='pagination'>

                  <button onClick={handlePrevPage} disabled={currentPage === 1}>

                      <FontAwesomeIcon icon={faChevronLeft} />

                  </button>

                  <span className='pagination\_number'>{currentPage}/{totalPages}</span>

                  <button onClick={handleNextPage} disabled={currentPage === totalPages}>

                      <FontAwesomeIcon icon={faChevronRight} />

                  </button>

            </div>

        </div>

        <table>

          <thead>

            <tr>

              <th>Nome Prodotto</th>

              <th>Descrizione Prodotto</th>

              <th>Stato Ordine</th>

              <th>Quantità</th>

              <th>Data Registrazione</th>

              <th>Data Aggiornamento</th>

              <th>Azioni</th>

            </tr>

          </thead>

          <tbody>

            {order.map((order, index) => (

              <tr key={index}>

                <td>{order.prodottoNome}</td>

                <td>{order.prodottoDescrizione}</td>

                <td>{order.statoOrdineDescrizione}</td>

                <td>{order.quantita}</td>

                <td>

                  {new Date(order.dataRegistrazione).toLocaleDateString()}

                </td>

                <td>{order.dataAggiornamento == null ? (

                    <span>Non aggiornato</span>

                    ) : (

                     <span>  {new Date(order.dataAggiornamento).toLocaleDateString()}</span>

                     )}

                </td>

                <td>

                  <div className='icons-container'>

                  <OrdineDetails order={order}/>

                  <OrdineUpdate setOrders={setOrders} fetchOrders={fetchOrders} order={order} orders={orders}/>

                  <OrdineDelete ordine={order} onEndDelete= {handleOnEndDelete}/>

                  </div>

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

Il componente rende la struttura della pagina con ordini, paginazione e azioni per ogni ordine.

Utilizza gli stati e gli effetti di React per mantenere e aggiornare lo stato dell'applicazione.