# Get di tutti gli ordini

Il metodo deve accettare in input l’**ID** dell’utente che intende ricercare tutti i suoi ordini.

Successivamente sarà necessario effettuare i seguenti passaggi:

* **query sul database:** recuperare tutti gli ordini relativi all’utente con l’**ID** di pertinenza.

Nello specifico, i dati da restituire sono i seguenti:

* DATA\_REGISTRAZIONE, tabella ORDINE
* DATA\_AGGIORNAMENTO, tabella ORDINE
* DESCRIZIONE (stato ordine), tabella STATO\_ORDINE

e la lista di tutti i prodotti collegati all’ordine in esame, ogni elemento della lista dovrà contenere :

* ID PRODOTTO, tabella DETTAGLIO\_ORDINE
* DESCRIZIONE, tabella PRODOTTO
* NOME, tabella PRODOTTO
* QUANTITA, tabella DETTAGLIO\_ORDINE
  + **Ritorno dati e gestione errori**: è necessario ritornare al consumer un JSON contenente i dati sopracitati, e status code 200. Nel caso in cui si verifichi un errore nell’esecuzione della query, occorre restituire status code 500.

## FLOW DIAGRAM RECUPERO DI TUTTI GLI ORDINI

Immagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamente

## Architettura applicativa funzionalità

Richiesta Consumer

OrdineController

GetAllOrdiniByUserId

ManageOrdineBusiness

GetAllOrdiniByUserId

ManageOrdineData

GetAllOrdiniByUserId

IRepositoryOrdine

GetAllOrdiniByUserId

RepositoryOrdine

GetAllOrdiniByUserId

AcademyShopDB

-Ordine; -Prodotto;

-Stato\_Ordine; -Dettagli\_Ordine;

## Esplicazione metodo GetAllOriniByUserId

### Api Layer

Questo metodo è implementato all’interno della classe OrdineController e rappresenta pertanto l’implementazione dell’Operazione GetAllOrdiniByUserId. Riceve in input Id Utente.Invoca il metodo GetOrdiniByUserId della classe ManageOrdineBusiness , grazie al campo oOBL , che si occuperà di effettuare diversi controlli

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

Immagine che contiene testo, software, Software multimediale, Software per la grafica

Descrizione generata automaticamente

L'attributo **[HttpGet("GetAllOrdiniByUserId/{userId}")]** definisce la rotta per questa azione. Il segnaposto **{userId}** indica che **userId** sarà un parametro nella rotta. Il metodo restituisce un **Task<IActionResult>**, il che significa che è un metodo asincrono che restituisce un risultato che può essere un qualsiasi tipo di risultato di azione. Il metodo include un blocco **try-catch** per catturare eventuali eccezioni e restituire un errore 500 con un messaggio dettagliato.

### Business Layer

Nel business layer (ManageOrdineBusiness).Questo metodo in un blocco try/catch invoca il metodo presente nel ManageOrdineData,tramite il campo oUDL ,che restituisce gli ordini per l’utente eventuamente recuperato sul DataBase. Nel presente metodo dopo la restituzione del dato da parte ManageOrdineData vengono effettuati i seguenti controlli :

* Verifica se l’utente esiste se non è presente ritorna un errore
* Verifica se ci sono ordini per l’utente se non ci sono ritorna un StatusCode errore 400

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente

Il metodo **GetOrdiniByUserId** esegue alcune operazioni importanti come la verifica dell'esistenza dell'utente, verifica dell’esistenza dell’ordine e il recupero degli ordini.

### Data Layer o DAO

In questa classe ManageOrdineData faccio ritornare la firma del metodo presente nella classe IRepositoryOrdine, tramite il campo \_repositoryOrdine utile per la dipendenza esterna necessaria per effettuare operazioni di accesso e manipolazione dei dati degli ordini.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

### IRepository

L’interfaccia repository è progettata per definire operazioni relative alla gestione degli ordini.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente



### RepositoryOrdine

Questa classe **RepositoryOrdine** è un'implementazione concreta dell'interfaccia **IRepositoryOrdine**.Va a comunicare nel DataBase

Immagine che contiene testo, software, schermata

Descrizione generata automaticamente

Qui è presente ache il metodo per verificare se l‘utente è presente nel DB

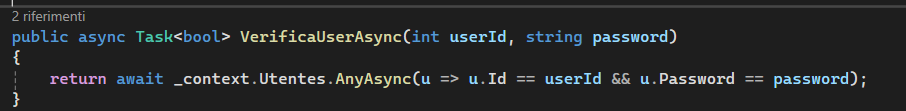


Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

}

}

Questo metodo implementa il metodo definito nell'interfaccia **IRepositoryOrdine**. Riceve come argomento l'ID dell'utente e restituisce una lista di oggetti **OrdineDettaglioDTOperGET** che rappresentano gli ordini relativi all'utente specificato.

esegue una query sul contesto del database **\_context** per recuperare gli ordini relativi a un determinato utente, identificato dall'ID **userId**. Dopo aver selezionato gli ordini, viene creata una lista di oggetti **OrdineDettaglioDTOperGET** utilizzando il metodo **Select**.

Utilizza l'operatore **Select** per mappare i risultati della query su oggetti **OrdineDettaglioDTOperGET**, che contiene i dettagli degli ordini per la visualizzazione.

Questa è la classe DTO (Creando una nuova Class Library(DtoLayer.Dto) )

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

### Database

Immagine che contiene testo, elettronica, schermata, software

Descrizione generata automaticamente

### **PROVE SWAGGER**

Apro il mio metodo GET

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamente

Come richiesto dal metodo devo inserire in input l’ID utente

Immagine che contiene testo, software, schermata, Carattere

Descrizione generata automaticamente

Inserisco l’Id dell’utente in questo esempio inserisco L’ID 1 che nel mio database ha 3 ordini pesenti:

Immagine che contiene testo, software, Software multimediale, computer

Descrizione generata automaticamente

Mi restituisce la lista degli ordini dell’’utente con l’Id 1 e uno status code 200.

Se provo a inserire un Id utente presente nel database ma che non ha nessun ordine mi dovrà restituire un Status code errore 500 con la scritta Ordini non presenti per questo utente. Esempio ID 11

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

Se inserisco un ID utente non presente nel database mi restituira Utente non trovato. ES. ID 111

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

.