Add-Ordine

dbo. Dettaglio\_Ordine

dbo. Ordine

dbo.Utente

dbo. Prodotto

addOrdineAsync()

AcademyShopDB

RepositoryOrdine

IRepositoryOrdine

addOrdine()

addOrdine()

addOrdine()

addOrdine()

OrdineController

ManageOrdineBusiness

ManageOrdineData

## SPECIFICHE: Inserimento nuovo ordine.

### OrdineController:

Il metodo addOrdineAsync() richiede in input **ID e Quantità** del prodotto da decurtare e **ID** dell’Utente relativo al nuovo ordine.

Le informazioni in output sono l’**ID** dell’ordine appena creato ed il codice di stato relativo.

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

*Test esempio tramite Swagger(Risposta in caso di operazione avvenuta con successo):*

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

### ManageOrdineBusiness:

Il metodo addOrdine() della classe ManageOrdineBusiness invoca il metodo prodottoExists(), il quale controlla l’esistenza del Prodotto dando come ritorno un booleano usato per il controllo dell’esistenza dello stesso.  
In caso il Prodotto non sia presente sul database ritorna lo status code 404, se il controllo risulta positivo recupera il prodotto da database tramite l’invocazione del metodo getProdottoAsync().  
In seguito viene controllata la correttezza dei dati che devono essere inseriti sul database.

Se presente almeno un errore viene riportato lo status code 400, altrimenti la quantità del prodotto viene aggiornata nell’oggetto Prodotto e viene invocato il metodo addORdine() della classe ManageOrdineData.

Sono previsti dei costrutti catch per raccogliere gli errori lanciati dal metodo addOrdine() della classe ManageOrdineData ed uno dedicato ad eccezioni generiche “Exception” per la gestione di casi imprevisti.

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

*Metodi getProdottoAsync() e prodottoExists() :*

*Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente*

*Test esempio tramite Swagger(Risposta in caso di prodotto non presente nel database ed in caso di quantità non valida):*

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

### ManageOrdineData:

Questo metodo invoca il metodo addOrdine del Repository Layer onde garantire l’applicazione del Pattern LOOSE COUPLING.

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

### AddOrdine (Interfaccia IRepositoryOrdine e Classe Repository):

Questo metodo viene richiamato all’interno dell’interfaccia IRepository e implementato all’interno della classe Repository.

La creazione del nuovo ordine viene risolta da due transazioni che conterranno le operazioni sul database.  
Nella prima transazione viene valorizzato l’oggetto Ordine che andrà ad essere inserito nel database.  
Se la transazione viene eseguita con successo viene chiusa ed è eseguita la seconda, in caso contrario esegue un rollback e viene propagata un eccezione TransactionAbortedException().

All’interno della seconda transazione viene valorizzato il DettaglioOrdine da inserire e viene salvato insieme al Prodotto la cui quantità è stata precedentemente aggiornata

Se la transazione viene eseguita con successo viene chiusa e viene passato come ritorno del metodo l’id dell’Ordine creato, in caso contrario esegue un rollback, viene cancellato l’Ordine creato nella prima transazione e viene propagata un eccezione TransactionException().

*Prima transazione del metodo addOrdine():*

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente

*Seconda transazione del metodo addOrdine():*

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente