PROJECT API

Overview

With the OODs principles, we designed our back-end focusing on database and providing APIs for others to perform CRUD (create, read, update and delete) operations on our database. We encapsulated all the related functions for different tables into different classes as shown below.

Besides, we implemented the creation of stored functions and triggers in other corresponding class. Then we use the InOrder, InOrderFunction, InOrderTriggers classes as a driver to connect and initialize tables, stored functions and triggers used in the database. In this way, we can make a clear modular structure for our program.

Since Order and OrderRecord are highly combined, we only allow to add Order records immediately when the user try to insert an Order. In this way, Order and Order Record become immutable. For the same reason, we also make InventoryRecord immutable.

You can find details about our APIs in the following pages

Supported operations

CustomerService

Function	Description	Parameters	Return
insert()	Inserts one row into the Customer table.	conn: the Connection object; customer: the Customer object.	true if inserted successfully, false other wise.
update()	Update a customer in the Customer table.	conn: the Connection object; customer: the Customer object.	true if updated successfully, false otherwise.
deleteById()	Delete a customer by it's ID.	conn: the connection object used in the driver; id: The id of the customer.	true if deleted successfully, otherwise false.
getById()	Gets a customer by its ID.	conn: the connection object used in the driver; id: The id of the customer.	the customer object, null if not found.

InventoryRecordService

Function	Description	Parameters	Return
insert()	Inserts one row into the InventoryRecord table.	conn: the connection object used in the driver; inventoryRecord: the InventoryRecord object.	true if inserted successfully, otherwise false.
getById()	Gets an inventory record by the product's SKU.	conn: the Connection object; sku: the product SKU.	the InventoryRecord instance, null if not found.
getAll()	Gets all inventory records ordered by the peoduct SKU.	conn: the Connection object; customer: the Customer object.	list of all inventory records

OrderService

Function	Description	Parameters	Return
shipOrder()	Ships the order by updating the ShipmentDate in the OrderTable table.	conn: the Connection object; orderId: the orderID. shipDate: the ship date.	true if updated successfully, otherwise false.
getById()	Gets an order by its ID.	conn: the Connection object; orderId: the order Id.	the order, null if not found.
getAll()	Gets all records.	conn: the Connection object	all records as a List.

ProductService

Function	Description	Parameters	Return
insert()	Inserts one row into the Product table.	conn: the Connection object; product: one product.	true if inserted successfully, false other wise.
deleteById()	Deleteone row by its ID.	conn: the connection object used in the driver; sku: The SKU of the product.	true if deleted successfully, otherwise false.
update()	Updates one product.	conn: the Connection object; product: The Production object.	true if updated successfully, false otherwise.
getProductBySku()	Get the product by SKU.	conn: the Connection object; sku: the product SKU.	the Product, null if cannot get the product.
getAllProducts()	Get all products in the Product table, the result list is ordered by product name.	conn: the Connection object	all products.