pkg

Entities

Users

<<table>>
UserEntity
- id : int
- role : UserRoleEnum
- name : String
- email : String(unique)
- password : String

<<enum>>
UserRoleEnum
- user : String
- receptionist : String
- inventory_manager : String
- cooker : String
- administrator : String

AuthEntity
- user : UserEntity
- token : String

Customers

<<table>>
CustomerEntity
- id : int
- name : String
- email : String
- phone : String
- address : String
- observation : String

1

0..*

Orders

<<enum>>
OrderStatusEnum
- Registered : int
- Finished : int
- Canceled : int

<<table>>
OrderItemEntity
- id : int
- order_id : int
- product_id : int
- packing_id : int(nullable)
- amount : double
- unit_price : double

FullOrderItemEntity
- product : ProductEntity
- packing : PackingEntity

<<table>>
OrderEntity
- id : int
- customer_id : int
- order_date : Date
- delivery_date : Date
- status : OrderStatusEnum
- discount : double
- shipping : double
- observation : String

0..*

1

FullOrderEntity
- items : FullOrderItemEntity[]
- customer : CustomerEntity

0..* 0..*

Products

<<table>>
ProductPackingsEntity
- id : int
- product_id : int
- packing_id : int
- price : double
- quantity : int

1

<<table>>
ProductEntity
- id : int
- name : String
- description : String
- is_packed : boolean
- unit_cost : double
- unit_price : double(nullable)
- category_id : int
- is_activated : boolean

<<table>>
CategoryEntity
- id : int
- name : String
- image_url : String
- category_id : int(nullable)

0..*

0..*   1

0..*

1

0..1

<<table>>
FeatureEntity
- id : int
- name : String

0..*

1

1

<<table>>
PackingEntity
- id : int
- name : String
- unit : String
- unit_abbreviation : String
- size : double
- cost : double

0..*

1

FullPackingEntity
- productPacking : ProductPackingsEntity

0..*

<<table>>
ProductFeaturesEntity
- id : int
- product_id : int
- feature_id : int

0..*

FullProductEntity
- features : FeatureEntity[]
- packings : FullPackingEntity[]
- category : CategoryEntity

ProductPackingsWithSuggestionEntity
- suggestedPriceMin : double
- suggestedPriceMax : double

FullPackingWithSuggestionEntity
- productPacking : ProductPackingsWithSuggestionEntity

FullProductWithSuggestionEntity
- suggestedPriceMin : double
- suggestedPriceMax : double
- features : FeatureEntity[]
- packings : FullPackingWithSuggestionEntity[]
- category : CategoryEntity

pkg

Repositories

Entity

<<interface>>
ICustomerRepository

<<interface>>
IRepository

<<bind>>
<Entity->CustomerEntity>

+ Create(entity : Entity) : void
+ Update(entity : Entity) : void
+ Show(entityId : int) : Entity
+ List(filters : Object, sort : Object[]) : Entity[]
+ Delete(entityId : int) : void

<<bind>>
<Entity->UserEntity>

<<interface>>
IUserRepository

+ *ShowByEmail(email : String) : UserEntity*

<<bind>>
<Entity->FullProductEntity>

<<bind>>
<Entity->FullOrderEntity>

<<interface>>
IOrderRepository

+ *ListItems(orderId : int) : OrderItemEntity[]*

<<interface>>
IProductRepository

+ CreatePacking(packing : PackingEntity) : void
+ UpdatePacking(packing : PackingEntity) : void
+ ListPackings() : PackingEntity[]
+ DeletePacking(packingId : int) : void
+ CreateFeature(feature : FeatureEntity) : void
+ UpdateFeature(feature : FeatureEntity) : void
+ ListFeatures() : FeatureEntity[]
+ DeleteFeature(featureId : int) : void
+ CreateCategory(category : CategoryEntity) : void
+ UpdateCategory(category : CategoryEntity) : void
+ ListCategories() : CategoryEntity[]
+ *DeleteCategory(categoryId : int) : void*
+ *ShowProductPacking(productPackingId : int) : ProductPackingsEntity*
+ *UpdateProductPacking(p : ProductPackingsEntity) : void*

pkg

Controllers

AuthController

+ Login(email : String, password : String) : AuthEntity
+ Authenticate(token : String) : boolean

UserController

+ Create(user : UserEntity) : void
+ Update(userId : int, user : UserEntity) : void
+ Show(userId : int) : UserEntity
+ List(filters : Object) : UserEntity[]
+ Delete(userId : int) : void
+ ListRoles() : UserRoleEnum[]

CustomerController

+ Create(customer : CustomerEntity) : void
+ Update(customerId : int, customer : CustomerEntity) : void
+ Show(customerId : int) : CustomerEntity
+ List(filters : Object) : CustomerEntity[]
+ Delete(customerId : int) : void

OrderController

+ Create(order : FullOrderEntity) : void
+ Update(orderId : int, order : FullOrderEntity) : void
+ Show(orderId : int) : FullOrderEntity
+ List(filters : Object) : FullOrderEntity[]
+ Delete(orderId : int) : void
+ ListOrderStatus() : OrderStatusEnum[]
+ FinishOrder(orderId : int) : void
+ CancelOrder(orderId : int) : int
+ GenerateReceipt(orderId : int) : StringBuffer

ProductController

+ Create(product : FullProductEntity) : void
+ Update(productId : int, product : FullProductEntity) : void
+ Show(productId : int) : FullProductEntity
+ List(filters : Object) : FullProductEntity[]
+ Delete(ProductId : int) : void
+ CreatePacking(packing : PackingEntity) : void
+ UpdatePacking(packingId : int, packing : PackingEntity) : void
+ ListPackings() : PackingEntity[]
+ DeletePacking(packingId : int) : void
+ CreateFeature(feature : FeatureEntity) : void
+ UpdateFeature(featureId : int, feature : FeatureEntity) : void
+ ListFeatures() : FeatureEntity[]
+ DeleteFeature(featureId : int) : void
+ CreateCategory(category : CategoryEntity) : void
+ UpdateCategory(categoryId : int, category : CategoryEntity) : void
+ ListCategories() : CategoryEntity[]
+ DeleteCategory(categoryId : int) : void
+ RegisterManufacturing(productPackingId : int, quantity : int) : void
+ GetPriceSuggestion(product : FullProductEntity) : FullProductWithSuggestionEntity

Todos os métodos podem receber como parâmetro a requisição e resposta HTTP, extrair os parâmetros representados a partir da requisição e retornar o valor via resposta

○ **IUserRepository**

○ **ICustomerRepository**

○ **IOrderRepository**

○ **IProductRepository**

pkg

Views

Auth

Login

User

Users   UsersForm

Product

Products   ProductsForm   Categories   Features   Packings

Order

Orders   OrdersData

Customer

Customers   CustomersForm