com.example.demo

com.example.demo::DemoApplication

+main(String[] args): void
+home(): String

com.example.demo.categories

com.example.demo.categories::Category

-id: Long
-title: String
-description: String

+Category(): ctor
+Category(Long id, String title, @Nullable String description): ctor
+getId(): Long
+setId(Long id): void
+getTitle(): String
+setTitle(String title): void
+getDescription(): String
+setDescription(@Nullable String description): void

com.example.demo.categories::CategoryService

-categoryRepository: CategoryRepository

+CategoryService(CategoryRepository categoryRepository): ctor
+checkIfExists(Long id): boolean
+getCategoryById(Long id): Category

«interface»
com.example.demo.categories::CategoryRepository

com.example.demo.clients

com.example.demo.clients::Client

-id: Long
-name: String
-last_name: String
-mobile: String
-creation_date: LocalDateTime

+Client(): ctor
+Client(String name, String last_name, String mobile): ctor
+setCreation_date(LocalDateTime dateTime): void
+setId(Long id): void
+setName(String name): void
+setLast_name(String last_name): void
+setMobile(String mobile): void
+getId(): Long
+getName(): String
+getLast_name(): String
+getMobile(): String
+toString(): String

com.example.demo.clients::ClientService

-clientRepository: ClientRepository

+ClientService(ClientRepository clientRepository): ctor
+getClients(int pageNo, int pageSize): List<Client>
+addClient(Client c): Client
+updateClient(Client c, Long id): Client
+checkIfExists(Long id): boolean

com.example.demo.clients::ClientController

-clientService: ClientService

+ClientController(ClientService clientService): ctor
+getClients(@PathVariable int pageNo, @PathVariable int pageSize): List<Client>
+addClient(@Valid @RequestBody Client c): Client
+editClient(@Valid @RequestBody Client c, @PathVariable("id") Long id): Client

«interface»
com.example.demo.clients::ClientRepository

com.example.demo.core

com.example.demo.core::CustomExceptionHandler

+handleException(BindException ex): Map<String, String>

com.example.demo.products

com.example.demo.products::Product

-id: Long
-name: String
-description: String
-category: Category
-creation_date: LocalDateTime

+Product(): ctor
+Product(String name, String description, Category category): ctor
+setId(Long id): void
+setName(String name): void
+setDescription(String description): void
+setCategory(Category category): void
+setCreation_date(): void
+getId(): Long
+getName(): String
+getDescription(): String
+getCategory(): Category
+getCreation_date(): LocalDateTime
+toString(): String

com.example.demo.products::ProductController

-productService: ProductService
-categoryService: CategoryService

+ProductController(ProductService productService, CategoryService categoryService): ctor
+getProducts(@PathVariable int pageNo, @PathVariable int pageSize): List<Product>
+addProduct(@Valid @RequestBody Product p): Product
+editProduct(@Valid @RequestBody Product p, @PathVariable("id") Long id): Product

com.example.demo.products::ProductService

-productRepository: ProductRepository

+ProductService(ProductRepository productRepository): ctor
+getProducts(int pageNo, int pageSize): List<Product>
+addNewProduct(Product p): Product
+updateProduct(Product p, Category cat, Long id): Product
+checkIfExists(Long id): boolean

«interface»
com.example.demo.products::ProductRepository

com.example.demo.sales

com.example.demo.sales::Sale

-id: Long
-total: double
-client: Client
-seller: Seller
-creation_date: LocalDateTime
~sale_transactions: List<Transaction>

+Sale(): ctor
+Sale(Client client, Seller seller): ctor
+getId(): Long
+setId(Long id): void
+getSale_transactions(): List<Transaction>
+getTotal(): double
+setTotal(): void
+getClient(): Client
+setClient(Client client): void
+getSeller(): Seller
+setSeller(Seller seller): void
+getCreation_date(): LocalDateTime
+setCreation_date(): void
+setSale_transactions(List<Transaction> sale_transactions): void

com.example.demo.sales::SaleController

-saleService: SaleService
-productService: ProductService
-sellerService: SellerService
-clientService: ClientService
-transactionService: TransactionService

+SaleController(SaleService saleService, ProductService productService, SellerService sellerService, ClientService clientService, TransactionService transactionService): ctor
+getSales(@PathVariable int pageNo, @PathVariable int pageSize): List<Sale>
+AddSale(@Valid @RequestBody Sale s): Sale
+editSaleTransactions(@Valid @RequestBody List<Transaction> transactions, @PathVariable("id") Long id): Sale

com.example.demo.sales::SaleService

-saleRepository: SaleRepository

+SaleService(SaleRepository saleRepository): ctor
+checkIfExists(Long id): boolean
+getAllSales(int pageNo, int pageSize): List<Sale>
+getSaleById(Long id): Sale
+addOrUpdateSale(Sale s): Sale

«interface»
com.example.demo.sales::SaleRepository

com.example.demo.sales.transactions

com.example.demo.sales.transactions::Transaction

~id: TransactionKey
-sale: Sale
-product: Product
-quantity: int
-unit_price: double

+Transaction(): ctor
+Transaction(TransactionKey id, Sale sale, Product product, int quantity, double unit_price): ctor
+setId(TransactionKey id): void
+getSale(): Sale
+setSale(Sale sale): void
+getProduct(): Product
+setProduct(Product product): void
+getQuantity(): int
+setQuantity(int quantity): void
+getUnit_price(): double
+setUnit_price(double unit_price): void
+equals(Object obj): boolean
+toString(): String

com.example.demo.sales.transactions::TransactionKey

~productId: Long
~saleId: Long

+TransactionKey(): ctor
+TransactionKey(Long productId, Long saleId): ctor
+getProductId(): Long
+setProductId(Long productId): void
+getSaleId(): Long
+setSaleId(Long saleId): void
+equals(Object obj): boolean
+hashCode(): int

com.example.demo.sales.transactions::TransactionService

~logger = LoggerFactory.getLogger(TransactionService.class): Logger
-transactionRepository: TransactionRepository

+TransactionService(TransactionRepository transactionRepository): ctor
+addMultipleTransactions(List<Transaction> transactions, Sale sale): List<Transaction>
+getTransactionBySale(Long sale_id): List<Transaction>
+updateTransactions(List<Transaction> old_trans, List<Transaction> new_trans): void

«interface»
com.example.demo.sales.transactions::TransactionRepository

~findBySaleId(Long sale_id): List<Transaction>

com.example.demo.sellers

com.example.demo.sellers::Seller

-id: Long
-name: String
-Agency: String
-details: String

+Seller(): ctor
+Seller(Long id, String name, @Nullable String agency, String details): ctor
+getId(): Long
+setId(Long id): void
+getName(): String
+setName(String name): void
+getAgency(): String
+setAgency(@Nullable String agency): void
+getDetails(): String
+setDetails(String details): void

com.example.demo.sellers::SellerService

-sellerRepository: SellerRepository

+SellerService(SellerRepository sellerRepository): ctor
+checkIfExists(Long id): boolean
+getSellerById(Long id): Category

«interface»
com.example.demo.sellers::SellerRepository