============================= ManyToOne =================================

Post Man result OneToOne ki and OneToMany ki chesinatte vuntadi…….!!!

===================================================================================

ManyToOne Mapping main aim :

// Get Address details by User ID

// Get User details by Address ID

================ Unidirectional ManyToOne coding ===================

**// Get Customer details by Order ID -- only child ----- > Parent only Possible**

**Unidirectional\_Customer\_Controller**

**@GetMapping("/customerByOrder/{orderId}")**

**public Unidirectional\_Customer getCustomerByOrderId(@PathVariable Long orderId) {**

**Unidirectional\_Order order = unidirectionalOrderRepo.findById(orderId)**

**.orElseThrow(() -> new RuntimeException("Order not found"));**

**return order.getCustomer(); // Get customer from the Order entity**

**}**

Out put :

{

    "name": "chandu",

    "coustomerid": 1

}

================ Bidirectional ManyToOne coding ===================

**// Get Customer details by Order ID**

**Bidirectional\_Customer\_Controller**

**@GetMapping("/customerByOrder/{orderId}")**

**public Bidirectional\_Customer getCustomerByOrderId(@PathVariable Long orderId) {**

**Bidirectional\_Order order = bidirectionalOrderRepo.findById(orderId)**

**.orElseThrow(() -> new RuntimeException("Order not found"));**

**return order.getCustomer(); // Get customer from the Order entity**

**}**

**Out Put :** <http://localhost:8080/BICustomer/customerByOrder/111>

{

    "id": 1,

    "name": "chandu",

    "orders": [

        {

            "id": 111,

            "product": "AC",

            "price": 4556.00

        },

        {

            "id": 333,

            "product": "Car",

            "price": 4454.00

        }

    ]

}

**Bidirectional\_Order\_Controller**

**// Get Orders by Customer ID (User ID)**

**@GetMapping("/ordersByCustomer/{customerId}")**

**public List<Bidirectional\_Order> getOrdersByCustomerId(@PathVariable Long customerId) {**

**return BidirectionalOrdersRepo.findByCustomer\_Id(customerId); // Find orders by customer ID**

**}**

**Bidirectional\_Order\_Repo**

**public interface Bidirectional\_Order\_Repo extends JpaRepository<Bidirectional\_Order, Long> {  
 List<Bidirectional\_Order> findByCustomer\_Id(Long customerId);  
  
}**

**Out Put :** http://localhost:8080/BIOrders/ordersByCustomer/1

[

    {

        "id": 111,

        "product": "AC",

        "price": 4556.00

    },

    {

        "id": 333,

        "product": "Car",

        "price": 4454.00

    }

]

===================== \*\*\*\* \*\* \*\*\*\*\* ===============================