**public** List<CustomerGetallRes> getallcustomer(CustomerDetailsGetAllReq req) {

List<CustomerGetallRes> resList = **new** ArrayList<CustomerGetallRes>();

DozerBeanMapper mapper = **new** DozerBeanMapper();

**try** {

CriteriaBuilder cb = em.getCriteriaBuilder();

CriteriaQuery<CustomerGetallRes> query = cb.createQuery(CustomerGetallRes.**class**);

List<CustomerGetallRes> list = **new** ArrayList<CustomerGetallRes>();

Root<CustomerDetails> c =query.from(CustomerDetails.**class**);

query.multiselect(c.get("customerId").alias("customerId"),c.get("gstNo").alias("gstNo"),

c.get("companyId").alias("companyId"),c.get("branchCode").alias("branchCode"),

c.get("clientName").alias("clientName"),c.get("dateOfBirth").alias("dateOfBirth"),

c.get("createdBy").alias("createdBy"),c.get("mobileNo1").alias("mobileNo1")

);

//query.multiselect(c);

// Order By

List<Order> orderList = **new** ArrayList<Order>();

orderList.add(cb.asc(c.get("customerId")));

TypedQuery<CustomerGetallRes> result = em.createQuery(query);

list= result.getResultList();

**for**(CustomerGetallRes data : list) {

CustomerGetallRes res = **new** CustomerGetallRes();

res = mapper.map(data, CustomerGetallRes.**class**);

resList.add(res);

}

}

**catch**(Exception e) {

e.printStackTrace();

log.info("Log Details"+e.getMessage());

**return** **null**;

}

**return** resList;

}