**查询类型**

Axon框架区分了三种类型的查询，即：（1）点对点查询，（2）分散收集查询，和（3）订阅查询。

（1）点对点查询 queryGateway.**query**()

1. 分散收集查询 queryGateway.**scatterGather**();

（3）订阅查询。queryGateway.**subscriptionQuery**();

QueryBus是将查询分派到查询处理程序的机制。使用查询请求名称和查询响应类型的组合注册查询。可以为同一请求-响应组合注册多个处理程序，这些处理程序可用于实现分散聚集模式。在分派查询时，客户机必须指出它是希望从**单个处理程序**还是从**所有处理程序**获得响应。

*queryGateway.****query****(queryName, queryCommand, ResponseTypes.multipleInstancesOf(VehicleDTO.class))*

1. *② ③ ④*

*① 客户机指出它是希望从单个处理程序获得响应*

*② 查询请求名称 可以省略，默认情况下，查询的名称是**查询负载的完全限定类名*

*③ 查询负载*

*④ 查询响应类型有三种： optionalInstanceOf、instanceOf、.multipleInstancesOf*

**6.4.2.1 AxonServer查询总线**

Axon提供了一个现成的查询总线，即AxonServerQueryBus。它连接到axoniqaxonserver服务器来发送和接收查询。

AxonServerQueryBus是一种“分布式查询总线”。默认情况下，它使用SimpleQueryBus处理不同JVM上的传入查询。

**Springboot部分：**

还是只需要引入axon-spring-boot-starter依赖，springboot就能自动进行配置。

SimpleQueryBus在分派查询的线程中直接处理查询。要配置SimpleQueryBus（而不是AxonServerQueryBus）：

**Springboot部分：**

@Bean

public SimpleQueryBus queryBus(AxonConfiguration axonConfiguration, TransactionManager transactionManager) {

return SimpleQueryBus.builder()

.messageMonitor(axonConfiguration.messageMonitor(QueryBus.class, "queryBus"))

.transactionManager(transactionManager)

.errorHandler(axonConfiguration.getComponent(

QueryInvocationErrorHandler.class,

() -> LoggingQueryInvocationErrorHandler.builder().build()

))

.queryUpdateEmitter(axonConfiguration.getComponent(QueryUpdateEmitter.class))

.build();

}

@QueryHandler

public List<VehicleDTO> find(final FindAllVehicleUseCase.FindAllQuery query) {

return vehicleRepository.findAll().stream()

.map(VehicleMapper.INSTANCE::toDto)

.collect(Collectors.toList());

}

@QueryHandler

public Optional<VehicleDTO> findById(final FindByIdVehicleUseCase.FindByIdQuery query) {

String id = query.getVehicleId().getIdentifier();

return findById(id).map(VehicleMapper.INSTANCE::toDto);

}