**The idea**

* Collect all the **Queries** and **Commands** related to one domain business to a **Unit of Work**
* Take **Save** and **Get** data out of **Queries** and **Commands**, bring them on **Unit of Work**

**Apply**

In the Shipping project, I try to apply for the Basket domain

* I created **BasketUnitOfWork** class, that collect all the commands and query related to the basket. Create a **Basket** property, store all current changes of a user. Implement **SaveChanges** function to save all those changes.

namespace Marketplace.Interview.Business

{

    public class BasketUnitOfWork : BasketOperationBase, IBasketUnitOfWork

    {

        public IAddToBasketCommand AddToBasketCommand { get; set; }

        public IRemoveFromBasketCommand RemoveFromBasketCommand { get; set; }

        public IGetBasketQuery GetBasketQuery { get; set; }

        public Basket.Basket Basket { get; set; }

        public BasketUnitOfWork()

        {

            Basket = GetBasket();

            AddToBasketCommand = new AddToBasketCommand { Basket = Basket };

            RemoveFromBasketCommand = new RemoveFromBasketCommand { Basket = Basket };

            GetBasketQuery = new GetBasketQuery { Basket = Basket };

        }

        public BasketUnitOfWork(IAddToBasketCommand addToBasketCommand,

            IRemoveFromBasketCommand removeFromBasketCommand,

            IGetBasketQuery getBasketQuery)

        {

            Basket = GetBasket();

            AddToBasketCommand = addToBasketCommand;

            RemoveFromBasketCommand = removeFromBasketCommand;

            GetBasketQuery = getBasketQuery;

            AddToBasketCommand.Basket = Basket;

            RemoveFromBasketCommand.Basket = Basket;

            GetBasketQuery.Basket = Basket;

        }

        public void SaveChanges()

        {

            SaveBasket(Basket);

        }

        public void Dispose()

        {

            GC.SuppressFinalize(this);

        }

    }

}

* Then update the implement of commands and queries, remove all Get and Save data directly, add Basket property. Here is some change in AddToBasketCommand:

namespace Marketplace.Interview.Business.Basket

{

    public class AddToBasketCommand : ~~BasketOperationBase,~~ IAddToBasketCommand

    {

        public AddToBasketResponse Invoke(AddToBasketRequest request)

        {

~~//var basket = GetBasket();~~

            var basket = Basket;

            request.LineItem.Id = basket.LineItems.MaxOrDefault(li => li.Id) + 1;

            basket.LineItems.Add(request.LineItem);

~~//SaveBasket(basket);~~

            return new AddToBasketResponse() { LineItemCount = basket.LineItems.Count };

        }

        public Basket Basket { get; set; }

    }

    public class AddToBasketRequest

    {

        public LineItem LineItem { get; set; }

    }

    public class AddToBasketResponse

    {

        public int LineItemCount { get; set; }

    }

}

* Here is how the BasketUnitOfWork using in the controller

namespace Marketplace.Interview.Web.Controllers

{

    [HandleError]

    public class HomeController : Controller

    {

        private readonly IGetShippingOptionsQuery \_getShippingOptions;

        private readonly IBasketUnitOfWork \_basketUnitOfWork;

        public HomeController(IBasketUnitOfWork basketUnitOfWork,

            IGetShippingOptionsQuery getShippingOptions)

        {

            \_basketUnitOfWork = basketUnitOfWork;

            \_getShippingOptions = getShippingOptions;

        }

        public ActionResult Index()

        {

            var basket = \_basketUnitOfWork.GetBasketQuery.Invoke(new BasketRequest());

            var shippingOptions = \_getShippingOptions.Invoke(new GetShippingOptionsRequest()).ShippingOptions;

            var viewModel = new InterviewViewModel { Basket = basket, ShippingOptions = shippingOptions };

            return View(viewModel);

        }

        public ActionResult RemoveItem(int id)

        {

            \_basketUnitOfWork.RemoveFromBasketCommand.Invoke(id);

            \_basketUnitOfWork.SaveChanges();

            return RedirectToAction("Index");

        }

        public ActionResult AddItem(LineItemViewModel lineItemViewModel)

        {

            var shippingOptions = \_getShippingOptions.Invoke(new GetShippingOptionsRequest()).ShippingOptions;

            var lineItem = new LineItem()

            {

                Amount = lineItemViewModel.Amount,

                ProductId = lineItemViewModel.ProductId,

                Shipping = shippingOptions[lineItemViewModel.ShippingOption],

                SupplierId = lineItemViewModel.SupplierId,

                DeliveryRegion = lineItemViewModel.DeliveryRegion,

            };

            \_basketUnitOfWork.AddToBasketCommand.Invoke(new AddToBasketRequest() { LineItem = lineItem });

            \_basketUnitOfWork.SaveChanges();

            return RedirectToAction("Index");

        }

        public ActionResult About()

        {

            return View();

        }

    }

}

* Currently the changed code is working as expectation.

**For continue develop**

* For other domain, we will create new UnitOfWork and make sure to separate Save data directly from the Commands.