[https://www.codeproject.com/Articles/741207/Repository-with-Unit-of-Work-IoC-and-Unit-Test](https://urldefense.com/v3/__https:/www.codeproject.com/Articles/741207/Repository-with-Unit-of-Work-IoC-and-Unit-Test__;!!PRtDf9A!9aW3M8Yj-gibWvvXCJ_tfhKc6DTxKsewfNQ_n0Ulsk4M7nwKQj1K9YPknjZJtkBEniY$)

[https://www.codeproject.com/Articles/579035/Creating-Unit-Testable-Applications-in-ASP-NET-MVC](https://urldefense.com/v3/__https:/www.codeproject.com/Articles/579035/Creating-Unit-Testable-Applications-in-ASP-NET-MVC__;!!PRtDf9A!40qOTKfJTq88MZghkaZQaEjvLRHNz8zxSXu5REDWF54BZ6MvYChqGRtk6SVoumh17HY$)

public void IndexControllerShowsAllLocationRecords()

{

Mock<IUnitOfWork> mock = new Mock<IUnitOfWork>();

mock.Setup(m => m.Repository<Location>().GetAll()).Returns(new List<Location> { new Location { Id = 1, LocationName = "Location 1" }, new Location { Id = 2, LocationName = "Location 2" }, new Location { Id = 3, LocationName = "Location 3" }, });

LocationController controller = new LocationController(mock.Object);

var model = (controller.Index() as ViewResult)?.ViewData.Model as List<LocationListViewModel>;

Assert.Equal("Location 1", model[0].LocationName);

Assert.Equal(3, model.Count);

Assert.Equal(2, model[1].LocationTableID);

}

[Fact]

public void DetailsShowsCorrectRecord()

{

Mock<IUnitOfWork> mock = new Mock<IUnitOfWork>();

mock.Setup(m => m.Repository<Location>().Get(It.IsAny<Func<Location, bool>>())).Returns(new Location { Id = 1, LocationName = "Location 1" });

LocationController controller = new LocationController(mock.Object);

var model = (controller.Details(1) as ViewResult)?.ViewData.Model as LocationListViewModel;

Assert.Equal("Location 1", model.LocationName);

Assert.Equal(1, model.LocationTableID);

}

private ApplicationContext GetContextWithData()

{

var options = new DbContextOptionsBuilder<ApplicationContext>()

.UseInMemoryDatabase(Guid.NewGuid().ToString())

.Options;

var context = new ApplicationContext(options);

context.Location.Add(new Location { Id = 1, LocationName = "Location 1" });

context.Location.Add(new Location { Id = 2, LocationName = "Location 2" });

//context.Activities.Add(new Activity { Id = 1, Title = "Title 1", Description = "Description 1", BeginDate = DateTime.Now, EndDate = DateTime.Now, ActivityStatusId = 1, Completed = true, Requester = "Requester 1", Sponsor = "Sponsor 1", RequestNumber = "1" });

//context.Activities.Add(new Activity { Id = 2, Title = "Title 2", Description = "Description 2", BeginDate = DateTime.Now, EndDate = DateTime.Now, ActivityStatusId = 1, Completed = true, Requester = "Requester 2", Sponsor = "Sponsor 2", RequestNumber = "2" });

context.SaveChanges();

return context;

}

private Repository<Location> GetRepository()

{

var options = new DbContextOptionsBuilder<ApplicationContext>()

.UseInMemoryDatabase(Guid.NewGuid().ToString())

.Options;

var context = new ApplicationContext(options);

context.Location.Add(new Location { Id = 1, LocationName = "Location 1" });

context.Location.Add(new Location { Id = 2, LocationName = "Location 2" });

//context.Activities.Add(new Activity { Id = 1, Title = "Title 1", Description = "Description 1", BeginDate = DateTime.Now, EndDate = DateTime.Now, ActivityStatusId = 1, Completed = true, Requester = "Requester 1", Sponsor = "Sponsor 1", RequestNumber = "1" });

//context.Activities.Add(new Activity { Id = 2, Title = "Title 2", Description = "Description 2", BeginDate = DateTime.Now, EndDate = DateTime.Now, ActivityStatusId = 1, Completed = true, Requester = "Requester 2", Sponsor = "Sponsor 2", RequestNumber = "2" });

context.SaveChanges();

Repository<Location> repoLocation = new Repository<Location>(context);

return repoLocation;

}