

JANUARY 28 2015

Unit of Work + Repository Pattern in MVC5 and Web API 2 with Fluent NHibernate and Ninject



User Click & Replay - Business Process Monitoring

Ad User, Process & Technology performance monitoring, insights and auto

germainapm.com


Learn more

After a lot of digging around to figure out the best way to implement the Unit of Work + Repository pattern in an ASP.NET MVC 5 or Web API 2 application, I came up with this solution.

It was used for a custom CMS application that was part of a web development project in Sydney for a law firm marketing website.

My main goal was to keep the code simple and practical without too much over-engineering. To show my approach I've created a sample MVC5 application that displays a list of products, the result isn't very exciting but it allows you to see everything working end to end.

The sample solution is a typical 3 tiered architecture with a separate project for the Domain, Data and Web tiers and is configured to use SQL Server. To keep things (relatively) short I'm only going to talk about the pieces that relate to the Unit of Work & Repository patterns, I'm assuming you're already familiar with MVC5, Fluent NHibernate and Ninject.

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)
campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

Code available at <https://github.com/cornflourblue/mvc5-unit-of-work-example> (<https://github.com/cornflourblue/mvc5-unit-of-work-example>)

Update 21 May 2016:

- Added Web API 2 example project that implements Unit of Work and Repository pattern
- Added Rollback method to UoW class to enable rolling back transactions for caught exceptions.
- Replaced Session.Close() with Session.Dispose() based on best practice recommendations found online.
- Wrapped transaction commit and rollback calls in if statements to ensure transaction is active.

Unit of Work Pattern

The Unit of Work pattern is used to group one or more operations (usually database operations) into a single transaction or “unit of work”, so that all operations either pass or fail as one.

Repository Pattern


The Repository pattern is used to manage CRUD operations through an abstract interface that exposes domain entities and hides the implementation details of database access code.

Implementation

Domain Layer

The domain/business logic layer contains the interfaces for a unit of work and a generic repository.

IUnitOfWork Interface

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?source=share&utm_medium=social))
A lightweight Unit of Work interface that defines methods for beginning, committing and rolling back a transaction.

```
public interface IUnitOfWork
{
    void BeginTransaction();
    void Commit();
    void Rollback();
}
```

IRepository<T> Interface

A generic repository interface that exposes a standard set of methods for performing CRUD operations on entities within the system.

```
public interface IRepository<T> where T : IEntity
{
    IQueryable<T> GetAll();
    T GetById(int id);
    void Create(T entity);
    void Update(T entity);
    void Delete(int id);
}
```

Data Layer

The data layer contains the implementations of the above unit of work and repository interfaces using Fluent NHibernate as the ORM.

UnitOfWork Class

The UnitOfWork class contains methods for beginning, committing and rolling back a transaction, it also exposes a Session property that returns the current NHibernate Session associated with the unit of work. Each UnitOfWork instance contains a single session.

~~The static constructor is used to implement the Singleton pattern for the NHibernate session factory, in C# static constructors are~~

~~executed only once per application domain and are thread-safe which makes them ideal for implementing singletons.~~

```
public class UnitOfWork : IUnitOfWork
{
    private static readonly ISessionFactory _sessionFactory;
    private ITransaction _transaction;


    public ISession Session { get; private set; }

    static UnitOfWork()
    {
        // Initialise singleton instance of ISessionFactory, static constructors are only executed once during the
        // application lifetime - the first time the UnitOfWork class is used
        _sessionFactory = Fluently.Configure()
            .Database(MsSqlConfiguration.MsSql2008.ConnectionString(x => x.FromConnectionStringWithKey("UnitOfWorkExample")))
            .Mappings(x => x.AutoMappings.Add(
                AutoMap.AssemblyOf<Product>(new AutomappingConfiguration()).UseOverridesFromAssemblyOf<ProductOverrides>()))
            .ExposeConfiguration(config => new SchemaUpdate(config).Execute(false, true))
            .BuildSessionFactory();
    }

    public UnitOfWork()
    {
        Session = _sessionFactory.OpenSession();
    }

    public void BeginTransaction()
    {
        _transaction = Session.BeginTransaction();
    }

    public void Commit()
    {
        try
        {
            // commit transaction if there is one active
            if (_transaction != null && _transaction.IsActive)
                _transaction.Commit();
        }
    }
}
```

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)
catch <https://codefund.app/invite/oSKfmPLO69o>) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)


```
{
    // rollback if there was an exception
    if (_transaction != null && _transaction.IsActive)
        _transaction.Rollback();

    throw;
}
finally
{
    Session.Dispose();
}
}

public void Rollback()
{
    try
    {
        if (_transaction != null && _transaction.IsActive)
            _transaction.Rollback();
    }
    finally
    {
        Session.Dispose();
    }
}
}
```

Repository<T> Class

A generic repository class that implements methods for performing CRUD operations on domain entities. You may notice that there aren't any transactions in this class, that's because transactions need to be implemented at a higher level because a transaction may contain several operations across different repositories.

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

```
public class Repository<T> : IRepository<T> where T : IEntity
{
    private UnitOfWork _unitOfWork;
    public Repository(IUnitOfWork unitOfWork){
        _unitOfWork = (UnitOfWork)unitOfWork;
    }

    protected ISession Session { get { return _unitOfWork.Session; } }


    public IQueryable<T> GetAll()
    {
        return Session.Query<T>();
    }

    public T GetById(int id)
    {
        return Session.Get<T>(id);
    }

    public void Create(T entity)
    {
        Session.Save(entity);
    }

    public void Update(T entity)
    {
        Session.Update(entity);
    }

    public void Delete(int id)
    {
        Session.Delete(Session.Load<T>(id));
    }
}
```

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

Unit of Work in ASP.NET MVC 5

When implementing the Unit of Work and Repository pattern in MVC 5, the web layer is the responsible for the configuration of dependency injection and transaction management.

NinjectWebCommon Class


This class is automatically added when you install the Ninject.MVC5 package from NuGet and contains the configuration for dependency injection. I've left out the code that I didn't touch.

The unit of work binding sets the scope to "InRequestScope()" which ensures that the same IUnitOfWork instance will be used everywhere within a single request. Having a single Unit or Work per request is necessary for the pattern to function correctly.

```
public static class NinjectWebCommon
{
    ...

    /// <summary>
    /// Load your modules or register your services here!
    /// </summary>
    /// <param name="kernel">The kernel.</param>
    private static void RegisterServices(IKernel kernel)
    {
        // unit of work per request
        kernel.Bind<IUnitOfWork>().To<UnitOfWork>().InRequestScope();

        // default binding for everything except unit of work
        kernel.Bind(x => x.FromAssembliesMatching("*").SelectAllClasses().Excluding<UnitOfWork>().BindDefaultInterface());
    }
}
```

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

MVC 5 BaseController Class

The base controller is used for beginning and committing transactions, this implementation uses a transaction per action so everything within an action is treated as a single Unit of Work. There is also a check to ensure that transactions are not created for child actions since any child action will already be running within the transaction of it's parent action.

I'm using public property injection for the IUnitOfWork property rather than constructor injection so controllers that inherit from BaseController won't need to call the base constructor passing the dependency.

```
public class BaseController : Controller
{
    [Inject]
    public IUnitOfWork UnitOfWork { get; set; }

    protected override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        if (!filterContext.IsChildAction)
            UnitOfWork.BeginTransaction();
    }

    protected override void OnResultExecuted(ResultExecutedContext filterContext)
    {
        if (!filterContext.IsChildAction)
            UnitOfWork.Commit();
    }
}
```


Unit of Work in Web API 2

When implementing the Unit of Work and Repository pattern in Web API 2, the web api layer is the responsible for the configuration of dependency injection and transaction management.

NinjectWebCommon Class

Unlike the MVC 5 implementation, the NinjectWebCommon class must be added manually when used with the Ninject.Web.WebApi NuGet package, and it depends on a custom Ninject based dependency resolver.

The unit of work binding sets the scope to "InRequestScope()" which ensures that the same IUnitOfWork instance will be used everywhere within a single request. Having a single Unit or Work per request is necessary for the pattern to function correctly.

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

```
public static class NinjectWebCommon
{
    private static readonly Bootstrapper bootstrapper = new Bootstrapper();


    /// <summary>
    /// Starts the application
    /// </summary>
    public static void Start()
    {
        DynamicModuleUtility.RegisterModule(typeof(OnePerRequestHttpModule));
        DynamicModuleUtility.RegisterModule(typeof(NinjectHttpModule));
        bootstrapper.Initialize(CreateKernel);
    }

    /// <summary>
    /// Stops the application.
    /// </summary>
    public static void Stop()
    {
        bootstrapper.ShutDown();
    }

    /// <summary>
    /// Creates the kernel that will manage your application.
    /// </summary>
    /// <returns>The created kernel.</returns>
    private static IKernel CreateKernel()
    {
        var kernel = new StandardKernel();
        kernel.Bind<Func<IKernel>>().ToMethod(ctx => () => new Bootstrapper().Kernel);
        kernel.Bind<IHttpModule>().To<HttpApplicationInitializationHttpModule>();

        RegisterServices(kernel);

        // Install our Ninject-based IDependencyResolver into the Web API config
        GlobalConfiguration.Configuration.DependencyResolver = new NinjectResolver(kernel);
    }
}
```

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)
return kernel; campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

```
}

/// <summary>
/// Load your modules or register your services here!
/// </summary>
/// <param name="kernel">The kernel.</param>
private static void RegisterServices(IKernel kernel)
{
    // unit of work per request
    kernel.Bind<IUnitOfWork>().To<UnitOfWork>().InRequestScope();

    // default binding for everything except unit of work
    kernel.Bind(x => x.FromAssembliesMatching("*").SelectAllClasses().Excluding<UnitOfWork>().BindDefaultInterface());
}
}
```

Web API UnitOfWorkActionFilter Class

The unit of work action filter is used for beginning and committing transactions, this implementation uses a transaction per action so everything within an action is treated as a single Unit of Work.

```
public class UnitOfWorkActionFilter : ActionFilterAttribute
{
    public IUnitOfWork UnitOfWork { get; set; }

    public override void OnActionExecuting(HttpContext actionContext)
    {
        UnitOfWork = actionContext.Request.GetDependencyScope().GetService(typeof(IUnitOfWork)) as IUnitOfWork;
        UnitOfWork.BeginTransaction();
    }

    public override void OnActionExecuted(HttpContext actionExecutedContext)
    {
        UnitOfWork = actionExecutedContext.Request.GetDependencyScope().GetService(typeof(IUnitOfWork)) as IUnitOfWork;
        if (actionExecutedContext.Exception == null)
        {
            // commit if no exceptions
            UnitOfWork.Commit();
        }
        else
        {
            // rollback if exception
            UnitOfWork.Rollback();
        }
    }
}
```

Add UoW Action Filter to Global Filters

The UnitOfWorkActionFilter is added globally in the WebApiConfig class.

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        ...


        // add UoW action filter globally
        config.Filters.Add(new UnitOfWorkActionFilter());
    }
}
```

Web Development Sydney

Feel free to drop me a line (</contact.aspx>) if you're looking for web development services in Sydney Australia, I also provide remote contracting services for clients outside Sydney.



Tags: [ASP.NET \(/posts/tag/aspnet\)](/posts/tag/aspnet), [Unit of Work \(/posts/tag/unit-of-work\)](/posts/tag/unit-of-work), [Repository \(/posts/tag/repository\)](/posts/tag/repository), [MVC5 \(/posts/tag/mvc5\)](/posts/tag/mvc5), [Fluent NHibernate \(/posts/tag/fluent-nhibernate\)](/posts/tag/fluent-nhibernate), [Ninject \(/posts/tag/ninject\)](/posts/tag/ninject), [ASP.NET Web API \(/posts/tag/aspnet-web-api\)](/posts/tag/aspnet-web-api), [Design Patterns \(/posts/tag/design-patterns\)](/posts/tag/design-patterns)

Sha. [Supporter](#)  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)
campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

More ASP.NET Posts

- [C# - Pure Pagination Logic in C# / ASP.NET \(/post/2018/10/17/c-pure-pagination-logic-in-c-aspnet\)](/post/2018/10/17/c-pure-pagination-logic-in-c-aspnet)
- [ASP.NET Core Razor Pages - Pagination Example \(/post/2018/10/15/aspnet-core-razor-pages-pagination-example\)](/post/2018/10/15/aspnet-core-razor-pages-pagination-example)
- [C# / ASP.NET - Domain Validation Logic in Domain Driven Design \(DDD\) \(/post/2016/12/12/c-aspnet-domain-validation-logic-in-domain-driven-design-ddd\)](/post/2016/12/12/c-aspnet-domain-validation-logic-in-domain-driven-design-ddd)
- [ASP.NET MVC - Pagination Example with Logic like Google \(/post/2015/10/30/aspnet-mvc-pagination-example-with-logic-like-google\)](/post/2015/10/30/aspnet-mvc-pagination-example-with-logic-like-google)
- [Fluent NHibernate - Unproxy Entity Objects with AutoMapper \(/post/2015/07/30/fluent-nhibernate-unproxy-entity-objects-with-automapper\)](/post/2015/07/30/fluent-nhibernate-unproxy-entity-objects-with-automapper)
- [C# - Incremental Delay to Prevent Brute Force or Dictionary Attack \(/post/2015/06/03/csharp-incremental-delay-to-prevent-brute-force-or-dictionary-attack\)](/post/2015/06/03/csharp-incremental-delay-to-prevent-brute-force-or-dictionary-attack)
- [ASP.NET - Fix for HTTPS/SSL Insecure/Mixed Content Warning \(/post/2014/09/25/aspnet-fix-for-https-ssl-insecure-mixed-content-warning\)](/post/2014/09/25/aspnet-fix-for-https-ssl-insecure-mixed-content-warning)
- [Dynamic LINQ - Using strings to sort by properties and child object properties \(/post/2014/07/16/dynamic-linq-using-strings-to-sort-by-properties-and-child-object-properties\)](/post/2014/07/16/dynamic-linq-using-strings-to-sort-by-properties-and-child-object-properties)
- [Web API 2 + ELMAH - Log ALL Unhandled Exceptions \(/post/2014/05/03/web-api-2-elmah-log-all-unhandled-exceptions\)](/post/2014/05/03/web-api-2-elmah-log-all-unhandled-exceptions)
- [Moq - Get Setup\(\) to return a dynamic collection from a fake repository \(/post/2014/04/28/moq-get-setup-to-return-a-dynamic-collection-from-a-fake-repository\)](/post/2014/04/28/moq-get-setup-to-return-a-dynamic-collection-from-a-fake-repository)
- [Post a simple string value from AngularJS to .NET Web API \(/post/2014/04/18/post-a-simple-string-value-from-angularjs-to-net-web-api\)](/post/2014/04/18/post-a-simple-string-value-from-angularjs-to-net-web-api)
- [ASP.NET Web API - elmah.axd 404 error - No HTTP resource was found that matches the request URI 'http://.../elmah.axd' \(/post/2014/03/14/aspnet-web-api-elmah-404-error-no-http-resource-was-found-that-matches-the-request-uri-http-elmah-axd\)](/post/2014/03/14/aspnet-web-api-elmah-404-error-no-http-resource-was-found-that-matches-the-request-uri-http-elmah-axd)
- [ASP.NET MVC - Required Checkbox with Data Annotations \(/post/2013/10/16/aspnet-mvc-required-checkbox-with-data-annotations\)](/post/2013/10/16/aspnet-mvc-required-checkbox-with-data-annotations)
- [C# - Get Time Zone and UTC Offset from Latitude and Longitude GPS coordinates \(/post/2013/05/23/c-get-time-zone-and-utc-offset-from-latitude-and-longitude-gps-coordinates\)](/post/2013/05/23/c-get-time-zone-and-utc-offset-from-latitude-and-longitude-gps-coordinates)
- [C# - Get Time Zone and UTC Offset from Latitude and Longitude GPS coordinates \(/post/2013/05/23/c-get-time-zone-and-utc-offset-from-latitude-and-longitude-gps-coordinates\)](/post/2013/05/23/c-get-time-zone-and-utc-offset-from-latitude-and-longitude-gps-coordinates)

- MVC CheckboxList for property of type IList<Enum> (/post/2012/09/20/mvc-checkboxlist-for-property-of-type-ilst-enum)
- IIS URL Rewrite - Redirect multiple domains to one and http to https with a single rule (/post/2012/08/16/iis-url-rewrite-redirect-multiple-domains-to-one-and-http-to-https-with-a-single-rule)
- MVC3 redirect to mobile site and stop executing (/post/2012/07/06/mvc3-redirect-to-mobile-site-and-stop-executing)
- ASP.NET MVC3 + Altairis Web Security Toolkit (/post/2012/06/24/aspnet-mvc3-altairis-web-security-toolkit)
- MVC3 Partial View Validation (/post/2012/04/24/mvc3-partial-view-validation)
- Dependency Injection in 1 line with MVC3 and Ninject (/post/2012/03/15/dependency-injection-in-1-line-with-mvc3-and-ninject)
- Populate ASP.NET dropdown list with month names using LINQ (/post/2011/11/21/populate-aspnet-dropdown-list-with-month-names-using-linq)

37 Comments

Jason Watmore's Blog

 Login ▾ Recommend Tweet Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name


**Mehmet Uğur Güral** • 3 years ago

Hi Jason, thanks for the nice article and example. it was very useful to learn basics of it.

I have a question that bothers me ;

You integrated your uow class inside the data project, but you are accessing the uow directly from the web project. Is this the right thing to do if we look this situation from decoupling perspective ? I mean we are using the whole data project just for the uow class, and we are coupling them to each other directly.

Thanks again.

Supporter |  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)



Jason Watmore Mod → Mehmet Uğur Güral • 3 years ago

Hi Mehmet,

The uow class is implemented in the data layer because it's managing transactions at the database level, I don't think it would really fit anywhere else.

The web projects don't use the uow class directly, they access it via the IUnitOfWork interface in the BaseController (for MVC) and UnitOfWorkActionFilter (for Web API). The only reason the web projects need a reference to the data project is for configuring dependency injection, so Ninject knows where to find and inject the concrete implementation of the IUnitOfWork interface.

As far as I know there isn't a way of configuring DI without referencing the project that contains concrete implementations of your project's required interfaces, but happy to be corrected if I'm wrong.

Cheers,

Jason

1 ^ | v • Reply • Share ›



Matt → Jason Watmore • 2 years ago

Mehmet has a point on decoupling, but in a different place. Your IUnitOfWork interface defines ISession so it is always tightly coupled to Nhibernate

^ | v • Reply • Share ›

[Show more replies](#)



Shankar • a month ago

This is great help. I have a request for you.

it would be very helpful if you could provide a sample of similar pattern using Asp.Net Core MVC. I am trying to implement this pattern in Asp.Net Core MVC and having a hard time as I am new to DotNet Core MVC.

^ | v • Reply • Share ›



The Changelog • 10 months ago

Supporter Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)

Hi Jason, Thank for great article. [ethical ad by CodeFund \(https://codefund.app/invite/oSKfmPLO69o\)](https://codefund.app/invite/oSKfmPLO69o)

I would like to know how we can write complex joins using repository on multiple tables?

^ | v • Reply • Share ›



Mike Sigsworth • a year ago

Hi Jason, I found this super helpful today. Thanks for this! I wanted to chime in with one potential improvement however. Ordinarily, on GETs you won't need the transaction. So I modified the filter to check if the HTTP Method was PUT, POST, PATCH, or DELETE before beginning the transaction.

My filter uses Autofac's interfaces, but the body of the methods should still be helpful enough to anyone reading this in the future.

```
public class UnitOfWorkActionFilter : IAutofacActionFilter
{
    private static readonly List<string> HttpMethodsRequiringTransaction = new List<string> {
    private readonly IUnitOfWork _unitOfWork;
    private readonly ILog _log;

    public UnitOfWorkActionFilter(IUnitOfWork unitOfWork, ILog log)
    {
        _unitOfWork = unitOfWork;
        _log = log;
    }

    public Task OnActionExecutedAsync(HttpActionExecutedContext actionExecutedContext, Cance]
    {
        if (RequiresTransaction(actionExecutedContext.Request))
        {
            if (actionExecutedContext.Exception != null)
            {

```

see more

^ | v • Reply • Share ›



jorge • 2 years ago

Jason, considering the commit is only been executed at the end of any request, what if I need any Entity.Id (identity) in my domain layer ? thank you.

^ | v • Reply • Share ›



Jason Watmore Mod → jorge • 2 years ago

Hi Jorge, the entity id is available as soon as an entity is created and before the transaction is committed to the database. For example a product id is set as soon as

Supporter

The CodeFund - Create the code you need, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)

^ | v • Reply • Share ›

campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)



Guest • 2 years ago

Can I use it with EF6? do you have an example?

^ | v • Reply • Share ›



Shan Sundaram • 2 years ago

Cascade Save / Update Issues

Hi Jason

I am new to NHibernate, just found your article and made it to work as it is. It is very useful for a starter. Thanks for this. I am trying to work with parent child relations , one to many and many to many etc. I am have lot of issues. As you used Automapping, how to instruct these relations and make it save as one commit. For example, I have Store and Employee one to many relation, I want to simply create all one in commit. Can you please suggest or give some references how to achieve this

Thanks

^ | v • Reply • Share ›



Jason Watmore Mod → **Shan Sundaram** • 2 years ago

Hi Shan,

You can override any of the fluent nhibernate automapping rules with a class like the [/UnitOfWorkExample.Data/MappingOverrides/ProductOverrides.cs](#) in the example.

To cascade all saves and updates from stores to employees you could do it like this:

```
public class StoreOverrides : IAutoMappingOverride<Store>
{
    public void Override(AutoMapping<Store> mapping)
    {
        mapping.HasMany(x => x.Employees).Cascade.SaveUpdate();
    }
}
```

Just to be clear though, all database actions performed within a single http request using

this [UnitOfWork](#) code will happen within a single database transaction, so you can perform

multiple separate data changes via the service layer and they will all be grouped into a single database commit at the end of the request.

Supporter



Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)

ethical ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

Cheers,
Jason

^ | v • Reply • Share ›



Adam Bull • 3 years ago

Hi Jason,

Good post. I'm trying to implement this in Web API 2 and have gotten to the last line of this post and realize that of course OnActionExecuting and OnResultExecuted are not available on the ApiController class. Do you know what the Web API equivalent of these methods are?

Thanks,

A

^ | v • Reply • Share ›



Jason Watmore Mod ➔ Adam Bull • 3 years ago

Hi Adam,

Yeah I actually had to do exactly that recently on a Web API 2 project, I used a custom action filter to hook into the OnActionExecuting and OnActionExecuted events.

I've added a WebApi project to the example solution that shows how it all works and updated the post.

Cheers,
Jason

^ | v • Reply • Share ›



Adam Bull ➔ Jason Watmore • 3 years ago

Great stuff, thank you!

^ | v • Reply • Share ›



phiree yuen • 3 years ago

Supporter **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=1511_ethical) ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)
there will be an error "Session is close()" when I refresh the page.

^ | v • Reply • Share ›



Jason Watmore Mod → phiree yuen • 3 years ago

The session should always be open when Commit() is called since a new session is opened at the beginning of each request and Commit() is called at the end of each request, it works correctly in the example and never throws an exception for me.

However I did find online recommendations to use Session.Dispose() instead of Session.Close() as it does a bit of extra clean up work and also checks if the session is closed first (<http://stackoverflow.com/qa...>). So I've updated the example to use Dispose() instead of Close() and also made some other minor tweaks that I detailed at the top of the post.

Cheers,
Jason

^ | v • Reply • Share ›



Ünal Akyüz • 3 years ago

I would suggest you to add Service Pattern also into your project. You will separate business and database concerns by this way.

^ | v • Reply • Share ›



Jason Watmore Mod → Ünal Akyüz • 3 years ago

Hi Unal, the business and database concerns are already completely separate.

All business logic code in the example is stored in the Domain project which is completely separate from the data access code which is all stored in the Data project.

^ | v • Reply • Share ›



Ibad • 3 years ago

Do I need to implement Mapping Override class for each of my model class?

^ | v • Reply • Share ›



Jason Watmore Mod → Ibad • 3 years ago

Supporter

The Changelog Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)

Hi Ibad, no you don't need a Mapping Override class for all of your entity/model classes, fluent nhibernate automapping is configured in the example to automatically map all entity

classes (classes that implement the IEntity interface).

You only need to add override classes where you want to override the default automapping configuration, for example if you wanted to map the Product.Name property to an nvarchar(4000) column in SQL Server (the default for string properties is nvarchar(255)) you would do this:

```
public class ProductOverrides : IAutoMappingOverride<product>
{
    public void Override(AutoMapping<product> mapping)
    {
        mapping.Map(x => x.Name).Length(4000);
    }
}
```

^ | v • Reply • Share ›



Ünal Akyüz → Ibad • 3 years ago

yes, of course.

^ | v • Reply • Share ›



Nazmi Altun • 3 years ago

Hello Jason,

Nice article , and I liked your answers to the questions below. I would also like to ask a few questions ;

- 1. What is the reason that you prefer using Ninject as your dependency injection framework? Is there any specific reason?
- 2. Shouldn't UnitOfWork class be database agnostic? As in your definition, your UnitOfWork class is configured for MS-SQL database. In case, we want to use another dbms (for example MySql) , we need to change the code of UnitOfWork. I think it is quite undesirable way to do it. Wouldn't it be better to make UnitOfWork class db agnostic and manage that configuration outside of the class definition?

Thanks In Advance,

^ | v • Reply • Share ›

Supporter **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)



Jason Watmore Mod → Nazmi Altun <https://codefund.app/invoke/oSKfmPLO69o>

Hi Nazmi

 Nazmi Altun,

1. I like Ninject because it allows me to setup dependency injection with just a few lines of code (on top of what is added via nuget). I've played around a bit with other DI containers but haven't looked at them in a while so they may offer the same features now. At the end of the day they all do pretty much the same thing so I think it comes down to personal preference.

2. Yes it would be possible to add extra logic to the UoW class to select which database to use based on a configuration setting, but my main goal was to keep the code simple and practical without too much over-engineering. In my project I don't think the benefit of being able to switch db from a config file would be worth adding the extra code to maintain. Also in my experience switching database providers is extremely rare for a production application and when it does happen there are a lot of code changes and testing required, it's never simply changing a config setting.

Cheers,
Jason

^ | v • Reply • Share ›



Pedro René González → Nazmi Altun • 3 years ago

point 2, I think if we add a method in another class looking for the type of database engine for select the type configuration, there are few and are less than 10 to make a simple search ... but just think out loud ... Cheers

^ | v • Reply • Share ›



Pedro René González • 3 years ago

hi.Might make an example of implementation of a Controller ??? thx

^ | v • Reply • Share ›



Jason Watmore Mod → Pedro René González • 3 years ago

Hi Pedro, there's a complete working example project at the [GitHub repository](#) linked to at the top of the post, it includes the implementation of a controller.

^ | v • Reply • Share ›

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)



Pedro René González → Jason Watmore • 3 years ago

Hi Pedro,

```
mi Factory is:
static UnitOfWork()
{
    _sessionFactory = Fluently.Configure()
        .Database(MsSqlConfiguration.MsSql2008.ConnectionString(
            x => x.FromConnectionStringWithKey("DBExpediente")))
        .Mappings(x => x.AutoMappings.Add(
            AutoMap.AssemblyOf<expediente>(new DefaultAutomappingConfiguration()))
        .UseOverridesFromAssemblyOf<expedientemap>())
        .ExposeConfiguration(config => new SchemaUpdate(config).Execute(false, true))
        .BuildSessionFactory();
}
```

i need Add more entity, I need to do something else?

^ | v • Reply • Share ›

[Show more replies](#)

[Show more replies](#)



Damian • 3 years ago

Hi, great article - it was really helpful for me. I just wonder what are the advantages of using IUnitOfWork in Repository constructor instead of using UnitOfWork without downcasting? Am I missing here any hidden scenario or IoC limitation? Best regards, Damian.

^ | v • Reply • Share ›



Jason Watmore Mod ➔ Damian • 3 years ago

Hi Damian, thanks for the feedback. No there isn't any advantage to injecting the IUnitOfWork interface over the UnitOfWork class in the example. I guess I'm just in the habit of using interfaces in constructors for IoC, but either way is fine here since I'm casting to the concrete class straight away anyway.

Cheers,
Jason

Supporter




The ChangeLog • Hear from the hackers, leaders, & innovators of software development

(https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)



Jaime Sangrean • 4 years ago



Jaime Sangcap

• 4 years ago

How to deal with multiple db context? Lets say I have SalesDbContext and MembershipDbContext? How to tell that which context to use? Or am I doing it wrong? And basically what Im doing. after calling saveChanges, I publish events which event handlers might use different context. Example: on membership context, there is MemberRegistered which will be publish by MembershipDbContext after saveChanges, then there might be MemberRegisteredEventHandler on Sales to create SalesPerson that might use the SalesDbContext.


^

|

▼

• Reply

• Share



Jason Watmore

Mod

➔ Jaime Sangcap • 4 years ago

Hi Daskul, as far as I know there isn't any mechanism built into entity framework or nhibernate to manage a transaction (unit of work) across multiple dbcontexts/sessions, unless possibly if they're connecting to the same db in which case I would question why multiple dbcontexts are required.

If you have to implement a UoW across separate databases I'd recommend searching for how to implement distributed transactions.


^

|

▼

• Reply

• Share



Vinicius Stutz

• 4 years ago

Hi, Jason! I'm a newbie with NHibernate and did my transaction control out of a Unit of Work the way you did, but I found very interesting. A question: what is the benefit to using Fluent NHibernate instead of just using NHibernate? Thanks for the great article! Sorry my bad English ...


^

|

▼

• Reply

• Share



Jason Watmore


Mod

➔ Vinicius Stutz • 4 years ago

Oi Vinicius, thanks for the feedback. Fluent NHibernate is just my personal preference, I like the mapping interface in FNH better than the one that comes with NHibernate (mapping-by-code / loquacious). Plus when I started using NHibernate it was before mapping-by-code was released so FNH was the only option for configuring mappings in C# code.

That being said you could do the same thing with either FNH, mapping-by-code or configuring mappings in xml files (the old fashioned way), it all comes down to what you prefer.

Supporter

 **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

Espero que isso te ajude :)

Espero que isso te ajude: ./

1 ^ | v • Reply • Share ›



Luís Gabriel Nascimento Simas → Jason Watmore • 4 years ago

Jason.... obrigado :D

^ | v • Reply • Share ›

[Show more replies](#)



Alexey Zimarev • 4 years ago




I would make some changes:

- 1) Register SessionFactory in the container
- 2) Remove Update on the repository, why would you need that?
- 3) Change BeginTransaction and Commit to something like Start and Finish
- 4) Add Fail for rollback


^ | v • Reply • Share ›

ABOUT

I'm a web developer in Sydney Australia and the technical lead at Point Blank Development (<https://www.pointblankdevelopment.com.au>), I've been building websites and web applications in Sydney since 1998.

Find me on:  (https://twitter.com/jason_watmore)  (<https://github.com/cornflourblue>)
 (<https://www.youtube.com/channel/UCc46Wo9z8S3xSDhw9vdxvtg/>)

Support me on Patreon (<https://www.patreon.com/jasonwatmore>)

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development ([https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?](https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151)
campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

MONTHS

2019

May (/posts/2019/05) (1)

April (/posts/2019/04) (6)

March (/posts/2019/03) (1)

February (/posts/2019/02) (4)

January (/posts/2019/01) (1)

2018

2017

2016

2015

2014


2013

2012

2011

TAGS

Angular 2, Angular 4, Angular 5, Angular 6, Angular 7, Angular Directive, Angular UI Router, AngularJS, Animation, ASP.NET, ASP.NET Core, ASP.NET Web API, Authentication and Authorization, AWS, Basic Authentication, Bootstrap, C#, Chai, CKEditor, CSS3, DDD, Design Patterns, Dynamic LINQ, ELMAH, ES6, Exchange, Facebook, Fluent NHibernate, Google Analytics, Google API, Google Maps API, Google Plus, Heroku, HTML5, HTTP, IIS, Insecure Content, Instagram API, Ionic Framework, iOS, iPhone, JavaScript, jQuery, JWT, LinkedIn, LINQ, Login, MEAN Stack, Mocha, Modal, MongoDB, Moq, MVC, MVC5, NGINX, ngMock, NHibernate, Ninject, NodeJS, npm, Pagination, Pinterest, Razor Pages, React, Redmine, Redux, Registration, Repository, RxJS, Security, Shell Scripting, Sinon, SinonJS, TDD, Terraform, Twitter, TypeScript, Ubuntu, Umbraco, Unit of Work, Unit Testing, URL Rewrite, Validation, Vue, Vuex, Webpack, Windows Server 2008,

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)

 Adobe Stock




(<https://srv.carbonads.net/ads/click/x/GTND42QMCAADC23YCES4YKQMCWSILKQUCEADEZ3JCWBI52JNCTAI52QKC6BIP27YCESDTK3EHJNCLSIZ?segment=placement:jasonwatmore;>)

ads via Carbon (http://carbonads.net/?utm_source=jasonwatmore&utm_medium=ad_via_link&utm_campaign=in_unit&utm_term=carbon)

Limited time offer:
Get 10 free Adobe Stock images.
(<https://srv.carbonads.net/ads/click/x/GTND42QMCAADC23YCES4YKQMCWSILKQUCEADEZ3JCWBI52JNCTAI52QKC6BIP27YCESDTK3EHJNCLSIZ?segment=placement:jasonwatmore;>)

Powered by MEANie (<https://jasonwatmore.com/meanie>)

© 2019 JasonWatmore.com

Supporter  **The Changelog** Hear from the hackers, leaders, & innovators of software development (https://codefund.app/impressions/2b13afa3-12ce-42ab-a0da-2177d4ba9679/click?campaign_id=151) *ethical* ad by CodeFund (<https://codefund.app/invite/oSKfmPLO69o>)