# Reusing common code via Nuget

## (Demo prep)

- Remove package sources:
  dotnet nuget remove source "Package source 1"
- Reopen Postman

## Refactoring common code

1. Refactor into a new Extensions.cs:

```
namespace Play.Catalog.Service.Repositories
{
    public static class Extensions
    {
        public static IServiceCollection AddMongo(this IServiceCollection services)
        {
            BsonSerializer.RegisterSerializer(new GuidSerializer(BsonType.String));
            BsonSerializer.RegisterSerializer(new DateTimeOffsetSerializer(BsonType.String));

            services.AddSingleton(serviceProvider =>
            {
                var configuration = serviceProvider.GetService<IConfiguration>();
                var mongoDbSettings = configuration.GetSection(nameof(MongoDbSettings)).Get<MongoDbSettings>();
                var serviceSettings = configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();

                var mongoClient = new MongoClient(mongoDbSettings.ConnectionString);
                return mongoClient.GetDatabase(serviceSettings.ServiceName);
            });

            return services;
        }

        public static IServiceCollection AddMongoRepository<T>(this IServiceCollection services, string
collectionName)
            where T : IEntity
        {
            services.AddSingleton<IRepository<T>>(serviceProvider =>
            {
                var database = serviceProvider.GetService<IMongoDatabase>();
                return new MongoRepository<T>(database, collectionName);
            });

            return services;
        }
```

```
    }
}
```

2. Update Startup.cs again:

```
public void ConfigureServices(IServiceCollection services)
{
    serviceSettings = Configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();

    services.AddMongo()
            .AddMongoRepository<Item>("items");
```

In the next lesson we will see how to move our common classes to a new shared library that can be used by all of our microservices.