

## Generalizing the health check configuration

Let's generalize the health check configuration that our microservices will use by creating a couple of new extension methods in our Common library.

In [Play.Common](#) repo

1. Copy the HealthChecks dir from Identity to Play.Common
2. Fix the namespace in MongoDBHealthCheck
3. Add HealthChecks\Extensions.cs:

```
namespace Play.Common.HealthChecks
{
    public static class Extensions
    {
        private const string MongoCheckName = "mongodb";
        private const string HealthEndpoint = "health";
        private const string ReadyTagName = "ready";
        private const string LiveTagName = "live";
        private const int DefaultSeconds = 3;

        public static IHealthChecksBuilder AddMongoDb(this IHealthChecksBuilder builder, TimeSpan?
        timeout = default)
        {
            return builder.Add(new HealthCheckRegistration(
                MongoCheckName,
                serviceProvider =>
                {
                    var configuration = serviceProvider.GetService<IConfiguration>();
                    var mongoDbSettings =
configuration.GetSection(nameof(MongoDbSettings)).Get<MongoDbSettings>();
                    return new MongoDBHealthCheck(mongoDbSettings.ConnectionString);
                },
                HealthStatus.Unhealthy,
                new[] { ReadyTagName },
                timeout ?? TimeSpan.FromSeconds(DefaultSeconds)));
        }

        public static void MapPlayEconomyHealthChecks(
            this IEndpointRouteBuilder endpoints)
        {
            endpoints.MapHealthChecks($"/{HealthEndpoint}/{ReadyTagName}", new HealthCheckOptions()
            {
                Predicate = (check) => check.Tags.Contains(ReadyTagName),
            });
        }
    }
}
```

```

    });
    endpoints.MapHealthChecks($"/{HealthEndpoint}/{LiveTagName}", new HealthCheckOptions()
    {
        Predicate = (_) => false
    });
}
}
}

```

4. Bump NuGet package version in README
5. Build and publish NuGet package
6. Commit and push.

[In Play.Identity repo](#)

7. Bump the Play.Common dependency version
8. Delete the HealthChecks directory
9. Update Startup:

```

public class Startup
{
    ...
    public void ConfigureServices(IServiceCollection services)
    {
        ...
        services.AddHealthChecks()
            .AddMongoDb();
    }

    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        ...
        app.UseEndpoints(endpoints =>
        {
            ...
            endpoints.MapRazorPages();
            endpoints.MapPlayEconomyHealthChecks();
        });
    }
}

```

10. Start the service

11. In Postman, confirm the health endpoints keep working

12. Commit and push.

Now that you know how to add health checks to your microservices, please move on to the next assignment where you will add health checks to the Catalog, Inventory and Trading services.