

Configuring a custom database health check

Let's now configure our custom MongoDB health check in our Identity microservice.

[In Play.Identity repo](#)

1. Update Startup:

```
public class Startup
{
    ...
    public void ConfigureServices(IServiceCollection services)
    {
        ...

        services.AddHealthChecks()
            .Add(new HealthCheckRegistration(
                "mongodb",
                serviceProvider =>
                {
                    var mongoClient = new MongoClient(mongoDbSettings.ConnectionString);
                    return new MongoDBHealthCheck(mongoClient);
                },
                HealthStatus.Unhealthy,
                new[] { "ready" },
                TimeSpan.FromSeconds(3)));
    }

    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        ...
        app.UseEndpoints(endpoints =>
        {
            ...
            endpoints.MapRazorPages();
            endpoints.MapHealthChecks("/health/ready", new HealthCheckOptions()
            {
                Predicate = (check) => check.Tags.Contains("ready"),
            });
            endpoints.MapHealthChecks("/health/live", new HealthCheckOptions()
            {
                Predicate = (_) => false
            });
        });
    }
}
```

2. Ensure all infra containers have started

3. Start Identity service

In Postman

4. In Settings, ensure Request timeout in ms is 0

5. Do a GET for <https://localhost:5003/health/live>

6. Do a GET for <https://localhost:5003/health/ready>

7. Stop mongo container

8. Do a GET for <https://localhost:5003/health/live>

Notice it's still healthy

9. Do a GET for <https://localhost:5003/health/ready> again

Notice it's unhealthy

10. Restart mongo container

11. Do a GET for <https://localhost:5003/health/ready> again

Notice it's back to healthy

12. Stop rabbitmq container

13. Do a GET for <https://localhost:5003/health/ready> again

Notice it's back to unhealthy

MassTransit already includes health checks

14. Commit and push.

In the next lesson you will create K8s health probes that take advantage of the new health checks.