Entity Framework Performance Cheat Sheet V1

Before starting a Project:

1. Consider using proxy entities
   - Lower memory footprint
   - No detect changes
   - Don't use AutoMappers
2. Disable lazy loading
   - This pattern is unhealthy
3. Define your performance goals

During your everyday coding:

1. Enumerate early
   Enumerate deliberately
2. Test performance goals
3. Only optimise if you don't reach your goals
4. Use .AsNoTracking for read-only data
5. Keep the scope of your context small
   - RequestScope is good
   - SingletonScope is bad

When you are not meeting your performance goals:

1. Profile your queries (EF profiler)
2. Enumerate in the correct places
3. Simplify your queries
4. Let EF track fewer items at once

If you need to squeeze every last drop of performance:

1. Consider caching data for read
2. Use update without selects

```
using (var context = new MyContext())
{
      var dummy = new Test{Id= 1};
      context.Tests.Attach(dummy);
      dummy.Something = "Hello World";
      context.SaveChanges();
}
```

3. If Snapshot Tracking with many tracked entities consider disabling AutoDetectChanges
   This will have some negative impacts:
   - Fixups won't be applied
   - You will need to manually call Detect changes before saving