



ASP .NET Core 3.0 {پیشرفته}

 Visual Studio 2019

مدرس: علیرضا ارومند
nikamooz;
آموزش برنامه نویسی و اجرای پروژه

www.NikAmooz.com

جلسه هشتم

معرفی علیرضا ارومند

۱. مدرس و مشاور ASP.NET Core و معماری‌های نرم‌افزاری (نیک آموز)
۲. مدیر فنی خبرگزاری نسیم
۳. کارشناس ارشد توسعه نرم افزار داتین (فناپ)
۴. کارشناس ارشد توسعه نرم افزار ارتباط فردا (بانک آینده)
۵. متخصص انجام پروژه‌های وب و .NET
۶. و...



Going deeper into the DbContext

چه خواهیم آموخت

۱. چگونه تغییرات تشخیص داده می‌شوند؟

۲. Change Tracker چیست؟

۳. کار با Raw Sql



خواص DbContext

۱. تنها سه خاصیت دارد

۲. ChangeTracker

۳. Database

۴. Model



چگونگی تشخیص تغییرات

۱. Entity های تحت ردیابی State دارند

۲. یک Enum از جنس EntityState

۳. تشخیص تغییر خاصیت با IsModified



چه دستوراتی State را تغییر می دهند؟

۱. نسبت به EF 6 بهینه سازی انجام شده



افزودن به دیتابیس



۱. استفاده از Add و AddRange

۲. وضعیت Entity را به Added

۳. وضعیت روابط وابسته به خود رابطه‌ها

حذف یک ردیف

۱. دستورات Remove و RemoveRange

۲. تغییر وضعیت به Deleted

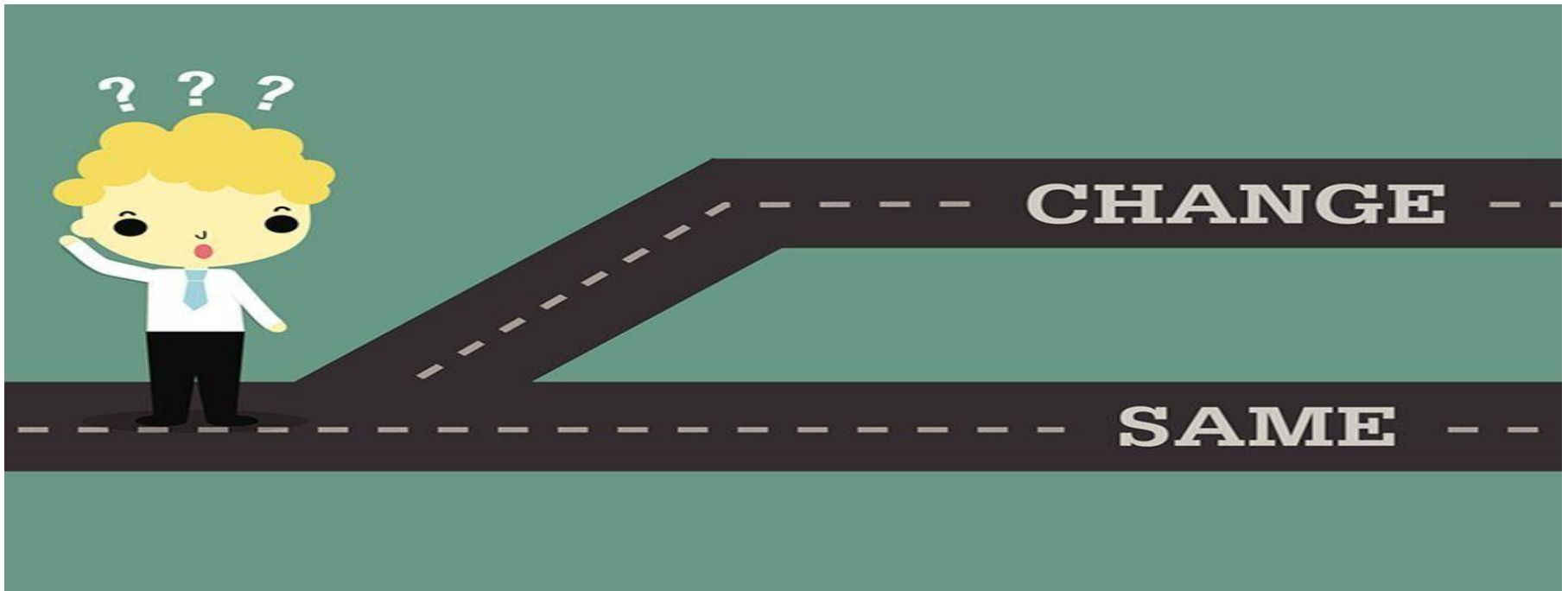
۳. وضعیت روابط بستگی به کلید اصلی



ویرایش اطلاعات

۱. تشخیص تغییرات DetectChange

۲. استفاده از Tracking snapshot



یک راه جدید



۱. سرعت پایین با داده‌های زیاد در روش سنتی
۲. `INotifyPropertyChanged` راهی جدید
۳. تغییر روش را باید با `Fluent` اعلام کنیم

همه چیز تغییر یافته

۱. Update و UpdateRange

۲. استفاده با Untracked Entity

۳. همه خواص و روابط را به روزرسانی



EVERYTHING

چگونگی ردیابی اشیا جدا

۱. استفاده از Attach و AttachRange

۲. فقط عمل اتصال را انجام می‌دهد

۳. وضعیت روابط به کلید اصلی مربوط



تنظیم مستقیم وضعیت

۱. استفاده از State برای تنظیم وضعیت
۲. امکان تنظیم IsModified هم وجود دارد



به روزرسانی روابط disconnected

۱. استفاده از TrackGraph
۲. امکانی جدید در EF Core
۳. به ازای کل گراف اجرا می‌شود.
۴. ارسال منطق تغییر وضعیت به شکل تابع

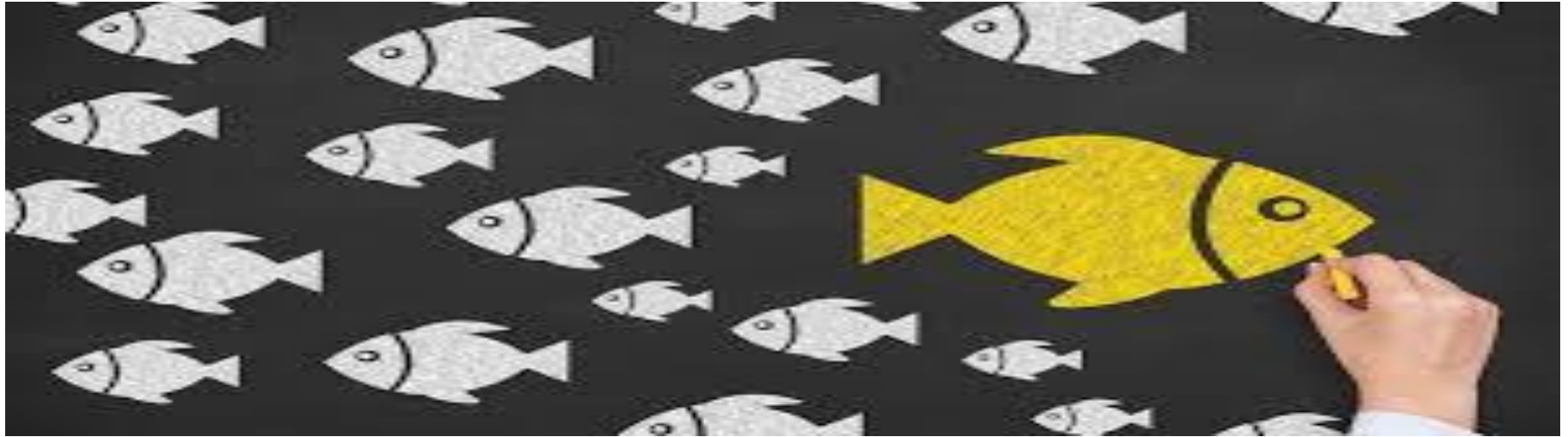


استفاده از changeTracker



۱. افزودن اطلاعات اضافه هنگام ذخیره
۲. ذخیره سازی سابقه تغییرات

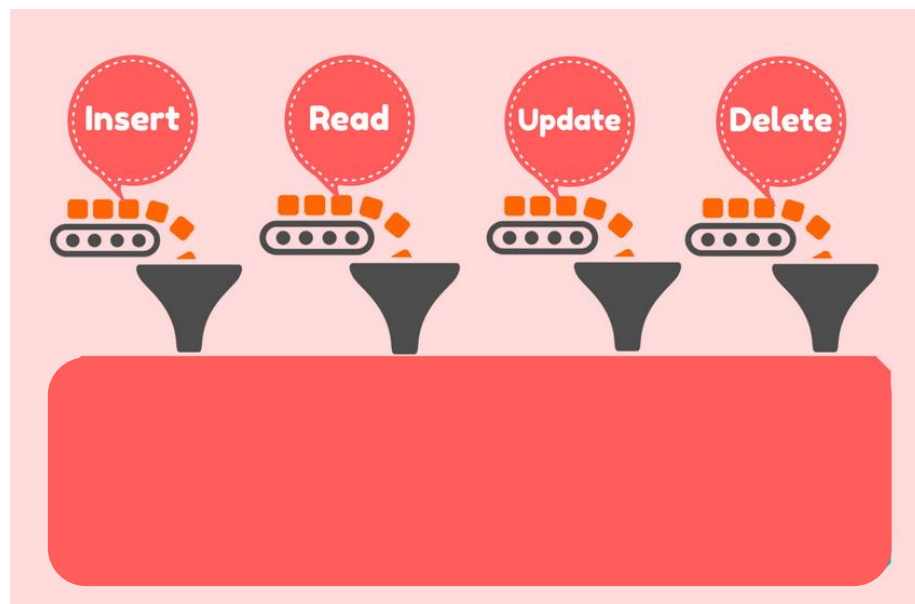
استفاده از changeTracker



۳. بررسی‌های امنیتی قبل از ذخیره سازی اطلاعات
۴. راه کلی بازنویسی SaveChanges

استفاده از RawSql

۱. برآورده نشدن تمام خواسته‌ها با LINQ
۲. نیاز به استفاده از کوئری موجود
۳. نیاز به بهبود پرفورمنس داریم



استفاده از دستور FromSql



۱. امکان استفاده از RawSql
۲. می‌توانیم SP را از این روش اجرا کنیم.

استفاده از دستور FromSql



۳. امکان استفاده با **Include** مشروط فراهم

۴. امکان **string interpolation** وجود دارد.

محدودیت‌ها

۱. فقط امکان استفاده با Entity های اصلی
۲. باید برای تمام خواص مقدار برگرداند



محدودیت‌ها

۳. نام ستون‌های نتیجه و Entity یکسان

۴. با وجود Model Level query محدودیت دارد



ExecuteSqlCommand

۱. بدون وابستگی به Entity Model
۲. به صورت non-query اجرا می شود.



استفاده از دستور Reload

۱. کاربرد همراه `ExecuteSqlCommand`
۲. استفاده برای بارگذاری مجدد `Entity`
۳. یکسانسازی وضعیت `Entity` با دیتابیس



استفاده از ADO.NET

۱. امکان استفاده همزمان با EF Core
۲. دریافت connection از EF
۳. انجام کارها کاملاً دستی
۴. از دست دادن هوشمندی EF



Microsoft

ADO.NET

Context.Model

۱. روش دسترسی به اطلاعات پایگاه داده
۲. استفاده برای تولید کدهای خودکار
۳. نسبت به EF 6 پیشرفت چشمگیر



رفع مشکلات Connection

۱. رایجترین مشکل کار با دیتابیس
۲. دفعات تلاش برای اتصال به دیتابیس
۳. در صورت timeout چه کاری انجام دهیم؟



Transaction و پیاده سازی آن در EF

۱. اصل همه یا هیچ در تغییرات داده‌ها
۲. پیش فرض کارها Transactional
۳. می‌توان تراکنش‌ها را دستی مدیریت کرد



مدیریت تراکنش با Execution Strategy

۱. در تنظیمات Strategy را معلوم کنیم
۲. یک نمونه از strategy می‌سازیم
۳. اجرای کد داخل strategy
۴. پیاده سازی strategy سفارشی



شبکه‌های اجتماعی نیک آموز

اطلاع رسانی سریع کارگاه‌های نسبتاً رایگان،

کوپن‌های تخفیف، مقالات، فیلم و دوره‌های نیک آموز



Instagram



Telegram

