

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

### آموزش ایجاد وقفه (interception)

مدرس: مهندس افشین رفوآ

## آموزش ایجاد وقفه (interception)

### Interception

خوب در این قسمت چگونگی انجام **Intercept** در **EF** زمانی که دستورات پایگاه داده اجرا می شود را یاد خواهید گرفت.

**EF 6** توانایی انجام **intercept** کردن **context** را با استفاده از **IDbCommandInterceptor** را قبل و بعد از اجرا کردن عملیات **ExecuteNonQuery**، **ExecuteScalar**، **ExecuteReader** روی پایگاه داده را فراهم می کند.

نخست **IDbCommandInterceptor** را به طریق زیر پیاده سازی کنید.

```
class EFCommandInterceptor: IDbCommandInterceptor
{
    public void
    NonQueryExecuted(System.Data.Common.DbCommand command,
    DbCommandInterceptionContext<int> interceptionContext)
    {
        LogInfo("NonQueryExecuted",
        String.Format(" IsAsync: {0}, Command Text: {1}",
        interceptionContext.IsAsync, command.CommandText));
    }
    public void
    NonQueryExecuting(System.Data.Common.DbCommand command,
    DbCommandInterceptionContext<int> interceptionContext)
    {
        LogInfo("NonQueryExecuting",
        String.Format(" IsAsync: {0}, Command Text: {1}",
        interceptionContext.IsAsync, command.CommandText));
    }
    public void
    ReaderExecuted(System.Data.Common.DbCommand command,
    DbCommandInterceptionContextt interceptionContext)
    {
        LogInfo("ReaderExecuted",
        String.Format(" IsAsync: {0}, Command Text: {1}",
        interceptionContext.IsAsync, command.CommandText));
    }
    public void
```

```

ReaderExecuting(System.Data.Common.DbCommand command,
DbCommandInterceptionContext interceptionContext)
    {
        LogInfo("ReaderExecuting",
String.Format(" IsAsync: {0}, Command Text: {1}",
interceptionContext.IsAsync, command.CommandText));
    }
    public void
ScalarExecuted(System.Data.Common.DbCommand command,
DbCommandInterceptionContext<object> interceptionContext)
    {
        LogInfo("ScalarExecuted",
String.Format(" IsAsync: {0}, Command Text: {1}",
interceptionContext.IsAsync, command.CommandText));
    }
    public void
ScalarExecuting(System.Data.Common.DbCommand command,
DbCommandInterceptionContext<object> interceptionContext)
    {
        LogInfo("ScalarExecuting",
String.Format(" IsAsync: {0}, Command Text: {1}",
interceptionContext.IsAsync, command.CommandText));
    }
    private void LogInfo(string command, string
commandText)
    {
        Console.WriteLine("Intercepted on: {0}
:- {1} ", command, commandText);
    }
}

```

همان طور که در کد بالا می بینید **IDbCommandInterceptor** برای اجرا شدن 6 متد را فراهم می کند.

اکنون برای پیکربندی **interceptor** شما می توانید این را با استفاده از فایل **Config** یا با کدنویسی این را انجام دهید.

Config file:

```

<entityframework>
  <interceptors>
    <interceptor
type="EF6DBFirstTahlildadeh.EFCommandInterceptor,
EF6DBFirstTahlildadeh">
    </interceptor>
  </interceptors>

```

```
</entityframework>  
Code-based config:  
    public class FE6CodeConfig : DbConfiguration  
    {  
        public FE6CodeConfig()  
        {  
            this.AddInterceptor(new  
EFCommandInterceptor());  
        }  
    }
```

خوب اکنون ما می توانیم هر زمانی که **DbContext** دستورات **ExecuteScalar**، **ExecuteNonQuery**، **ExecuteReader** را اجرا می کند، آنها را مشاهده کنیم.