

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش پیکربندی مبتنی بر کد

مدرس: مهندس افشین رفوآ

## آموزش پیکربندی مبتنی بر کد

Entity Framework 6 چیزی به اسم **code based configuration** معرفی کرده است. بنابراین شما می

توانید تنظیمات مرتبط با **entity framework** با استفاده از کد که قبلا در قسمت از فایل **app.config** پیکربندی شده است را پیکربندی کرد. هرچند فایل **app.config** مقدم تر از **code-based configuration** است. به عبارتی دیگر یک مورد از پیکربندی در هر دو قسمت کد و فایل **config** تنظیم شده است و سپس تنظیمات موجود در فایل **config** استفاده شده است.

خوب بگذارید ببینیم که چگونه با استفاده از **Entity Framework6**، **code-based configuration** را پیاده سازی کنیم.

ابتدا شما باید کلاسی را که از کلاس **DbConfiguration** مشتق گرفته شده است را ایجاد کنید. (**System.Data.Entity.DbConfiguration**)

```
public class FE6CodeConfig : DbConfiguration
{
    public FE6CodeConfig()
    {
        //define configuration here
    }
}
```

شما اکنون می توانید **codeConfigurationType** را در فایل **config** همان طور که در زیر نشان داده شده است را **set** کنید.

```
<entityframework codeconfigurationtype="EF6DBFirstTahlildadeh.FE6CodeConfig, EF6DBFirstTahlildadeh">
</entityframework>
```

یا اینکه شما می توانید ویژگی **DbConfigurationType** را در کلاس **context** برای تنظیم کردن کلاس **code-based configuration** استفاده کنید.

```
[DbConfigurartionType(typeof(FE6CodeConfig))]
public partial class SchoolEntities : DbContext
{
    public SchoolEntities()
```

```

        : base("name=SchoolDBEntities")
    { }
}

```

نکته

EF از داشتن کلاس های چندگانه **configuration** در **AppDomain** یکسان پشتیبانی نمی کند. از شما از این ویژگی استفاده کنید که کلاس های **configuration** مختلفی را برای دو **contexts** را **set** کنید، خطایی روی خواهد داد.

اکنون شما می توانید با استفاده از متد های مختلف **DbConfiguration** را در سازنده همانطور که در زیر نشان داده شده است را استفاده کنید.

چگونگی انجام دادن تنظیمات با استفاده از **code-based configuration** و فایل **config** در زیر نشان داده شده است.

Set **default** connection factory:

Code-based configuration:

```

public class FE6CodeConfig : DbConfiguration
{

```

```

    public FE6CodeConfig()
    {

```

```

        this. SetDefaultConnectionFactory(new
System. Data. Entity. Infrastructure. SqlConnectionFactory());
    }
}

```

config file:

```

<entityframework>
<defaultconnectionfactory type="System. Data. Entity.
Infrastructure. SqlConnectionFactory, EntityFramework" />
</entityframework>

```

Set Database Provider:

Code-based configuration:

```

public class FE6CodeConfig : DbConfiguration
{

```

```

    public FE6CodeConfig()
    {

```

```

        this. SetProviderServices("System. Data.
SqlClient"،

```

```

                System. Data. Entity. SqlServer.
SqlProviderServices. Instance);
    }
}

```

```

    }
    config file:
    <entityframework>
    <providers>
    <provider invariantname="System. Data.
    SqlClient" type="System. Data. Entity. SqlServer.
    SqlProviderServices, EntityFramework. SqlServer" />
    </providers>
  </entityframework>
  Set Database Initializers

```

شما می توانید راه انداز های پایگاه داده (فقط برای **code-first**) با استفاده از **code-based configuration** همان طور که در زیر نشان داده شده است را تنظیم کنید.

```

public class FE6CodeConfig : DbConfiguration
{
    public FE6CodeConfig()
    {
        this. SetDatabaseInitializer(new
CustomDBInitializer());
    }
}
config file:
<entityframework>
<contexts>
<context type="EF6DBFirstTahlildadeh.
SchoolDBEntities, EF6DBFirstTutorials">
<databaseinitializer
type="EF6DBFirstTutorials. CustomDBInitializer ,
EF6DBFirstTutorials">
</databaseinitializer>
</context>
</contexts>
</entityframework>

```

نمونه پروژه **Code-based configuration** را می توانید از اینجا دانلود کنید.