

# Entity Framework


EF 5.0

**Code First: Data Annotations, Fluent API**

# Добрый день!

- Инструктор: Олег Кулыгин

[www.linkedin.com/in/kulygin](http://www.linkedin.com/in/kulygin)



**Олег Кулыгин**


Тренер-консультант, эксперт по построению сервис-ориентированных и облачных приложений.

Сертифицированный специалист Microsoft (MCTS, MCPD, MCT).

Основные направления:



- C#
- ASP .NET MVC
- WCF
- Azure
- Entity Framework
- ADO.NET
- UML
- Patterns of Design
- ООД


**Microsoft CERTIFIED Trainer**



Microsoft Certified Technology Specialist



OLEG KULYGIN  
Has successfully completed the requirements to be recognized as a Microsoft Certified Technology Specialist: .NET Framework & Data Access.






Microsoft Certified Professional Developer



OLEG KULYGIN  
Has successfully completed the requirements to be recognized as a Microsoft Certified Professional Developer: Web Developer 4.





Microsoft Certified Trainer

OLEG KULYGIN  
Has successfully completed the requirements to be recognized as a Trainer.



# Code First

## Обзор технологии

Code First – это техника создания БД и модели на основе существующих классов.

Генерация БД и модели сущностей Entity Data Model (EDM) происходит после построения проекта.

Модели в EDMX (XML) виде не существует!

# Code First

## Настройка модели



**Если не существует XML (EDMX)  
– как настраивать модель?**

# Code First

## Настройка модели

1

### Data Annotations

```
[MaxLength(20), MinLength(5)]  
public string Name
```

2

### Fluent API

```
modelBuilder.Entity<Post>()  
    .Property(p => p.Title).HasMaxLength(10);
```

# Code First

## Набор атрибутов Data Annotations

```
■ System.ComponentModel.DataAnnotations
└─ { } System.ComponentModel.DataAnnotations
    ▶ AssociatedMetadataTypeTypeDescriptionProvider
    ▶ AssociationAttribute
    ▶ BindableTypeAttribute
    ▶ CompareAttribute
    ▶ ConcurrencyCheckAttribute
    ▶ KeyAttribute
    ▶ MaxLengthAttribute
    ▶ MetadataTypeAttribute
    ▶ MinLengthAttribute
    ▶ PhoneAttribute
    ▶ RangeAttribute
    ▶ RegularExpressionAttribute
    ▶ RequiredAttribute
    ▶ ScaffoldColumnAttribute
    ▶ ScaffoldTableAttribute
    ▶ StringLengthAttribute
    ▶ TimestampAttribute
```

```
└─ { } System.ComponentModel.DataAnnotations.Schema
    ▶ ColumnAttribute
    ▶ ComplexTypeAttribute
    ▶ DatabaseGeneratedAttribute
    ▶ DatabaseGeneratedOption
    ▶ ForeignKeyAttribute
    ▶ InversePropertyAttribute
    ▶ NotMappedAttribute
    ▶ TableAttribute
```

# Code First

## DataAnnotations: **Key**Attribute

### Соглашения для ключевого свойства

1. Свойство с именем **Id**



```
public partial class Item
{
    public Guid Id { get; set; }
}
```

1. Свойство с именем  
**[имя\_класса]Id**



```
public partial class Item
{
    public Guid ItemId { get; set; }
}
```

Необходим атрибут **[Key]**, если  
имя ключевого свойства произвольно!

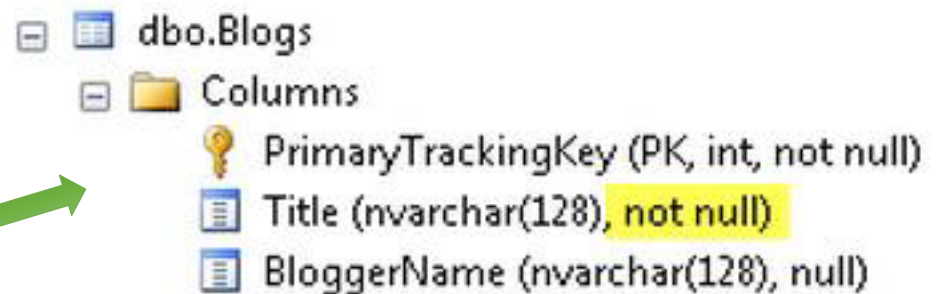
```
public partial class Item
{
    [Key]
    public Guid GlobalItemKey { get; set; }
}
```

# Code First

## DataAnnotations: **Required**Attribute

### Задание обязательности значения

```
public partial class Blog
{
    [Key]
    public int PrimaryTrackingKey { get; set; }
    [Required]
    public string Title { get; set; }
}
```



The screenshot shows a web form titled "blog". It contains two input fields: "Title" and "BloggerName". The "Title" field is highlighted with a red border, and a red error message "The Title field is required." is displayed next to it. The "BloggerName" field is empty. At the bottom of the form, there is a "Create" button.



# Code First

## DataAnnotations: **MinLength** и **MaxLength**

### Задание допустимой длины значения

```
public partial class Blog
{
    [Key]
    public int PrimaryTrackingKey { get; set; }
    [Required]
    public string Title { get; set; }
    [MinLength(5), MaxLength(10)]
    public string BloggerName { get; set; }
}
```

Columns

- PrimaryTrackingKey (PK, int, not null)
- Title (nvarchar(128), not null)
- BloggerName (nvarchar(10), null)

Blog

Title  
A New Blog

BloggerName  
Julie the Blogger

Create

# Code First

## DataAnnotations: **NotMappedAttribute**

### Вычисляемое свойство

```
public partial class Blog
{
    [Key]
    public int PrimaryTrackingKey { get; set; }
    [Required]
    public string Title { get; set; }
    [MinLength(5), MaxLength(10)]
    public string BloggerName { get; set; }
    [NotMapped]
    public int BlogCode
    {
        get
        {
            return Title.Substring(0, 1) + ":" + BloggerName.Substring(0, 1);
        }
    }
}
```

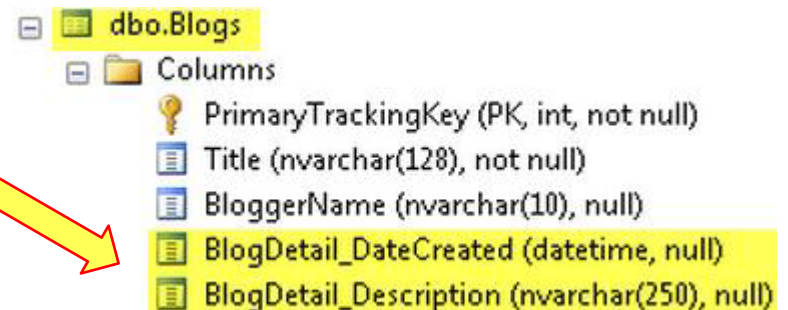
**Поле не сохраняется в БД.**

# Code First

## DataAnnotations: **ComplexTypeAttribute**

### Денормализация БД

```
public partial class Blog
{
    [Key]
    public int PrimaryTrackingKey { get; set; }
    [Required]
    public string Title { get; set; }
    [MinLength(5), MaxLength(10)]
    public string BloggerName { get; set; }
    public BlogDetails BlogDetail { get; set; }
}
[ComplexType]
public partial class BlogDetails
{
    public DateTime? DateCreated { get; set; }
    [MaxLength(250)]
    public string Description { get; set; }
}
```



# Code First

## DataAnnotations: **ConcurrencyCheckAttribute**

Защита от одновременного редактирования

```
public partial class Blog
{
    [Key]
    public int PrimaryTrackingKey { get; set; }
    [Required]
    public string Title { get; set; }
    [ConcurrencyCheck, MaxLength(10)]
    public string BloggerName { get; set; }
}
```

UPDATE ..... where (([PrimaryTrackingKey] = @4) and ([BloggerName] = @5))  
@4=1,@5=N'Alex'

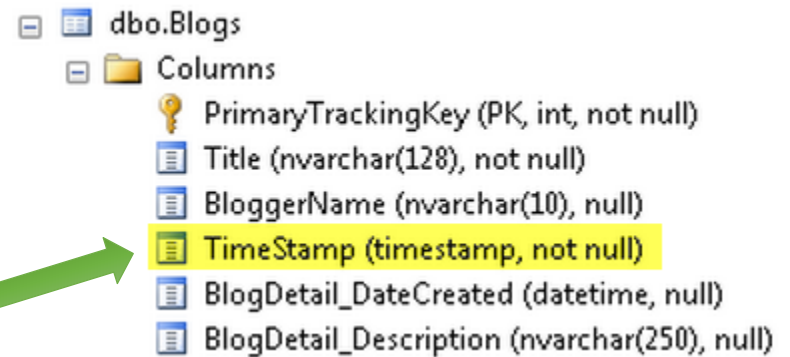
**DbUpdateConcurrencyException**

# Code First

## DataAnnotations: **TimeStampAttribute**

### Защита от одновременного редактирования

```
public partial class Blog
{
    [Key]
    public int PrimaryTrackingKey { get; set; }
    [Required]
    public string Title { get; set; }
    [MinLength(5), MaxLength(10)]
    public string BloggerName { get; set; }
    [TimeStamp]
    public Byte[] TimeStamp { get; set; }
}
```



При обнаружении атрибута `TimeStamp` EF включает `ConcurrencyCheck` и `DatabaseGeneratedPattern=Computed`

# Code First

## DataAnnotations: **TableAttribute** и **ColumnAttribute**

### Переименование таблиц/колонок

```
[Table("InternalBlogs")]
```

```
public partial class Blog
```

```
{
```

```
    [Key]
```

```
    public int PrimaryTrackingKey { get; set; }
```

```
    [Required]
```

```
    public string Title { get; set; }
```

```
    [MinLength(5), MaxLength(10)]
```

```
    public string BloggerName { get; set; }
```

```
    public BlogDetails BlogDetail { get; set; }
```

```
}
```

```
[ComplexType]
```

```
public partial class BlogDetails
```

```
{
```

```
    public DateTime? DateCreated { get; set; }
```

```
    [Column("BlogDescription", TypeName="ntext")]
```

```
    public string Description { get; set; }
```

```
}
```



dbo.InternalBlogs



Columns



PrimaryTrackingKey (PK, int, not null)



Title (nvarchar(128), not null)



BloggerName (nvarchar(10), null)



TimeStamp (timestamp, not null)



BlogDetail\_DateCreated (datetime, null)



BlogDescription (ntext, null)



# Code First

## FluentAPI: **ModelBuilder** vs **Configuration**

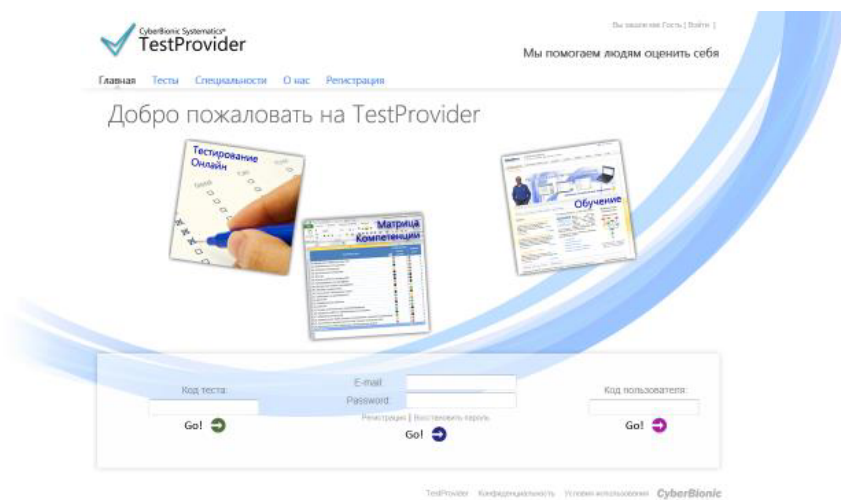
```
public class AttendeeContext : DbContext
{
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        modelBuilder.Entity<Attendee>. [configure entity]

        modelBuilder.Entity<Attendee>.Property(p=>p.Lastname). [configure entity]
    }
}
```

```
public class AttendeeConfiguration : EntityTypeConfiguration<Attendee>
{
    public AttendeeConfiguration ()
    {
        HasKey(p=>p.EntityTrackingID); [configure entity]
        ToTable("AttendeesList");
        Property(p=>p.Lastname). [configure entity]
    }
}
```

# Q&A





Перейти к тестированию  
[www.TestProvider.com](http://www.TestProvider.com)

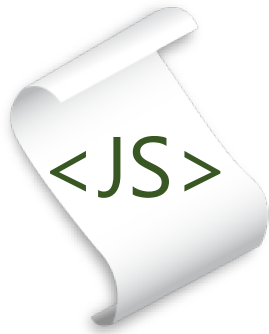
## Тестирование IT-специалистов

**TestProvider** обеспечивает надежную и объективную оценку технических знаний и опыта работы IT-специалиста с программными продуктами Microsoft.

## Сертификация IT-специалистов

Подтвердите ваш практический опыт работы с технологиями **Microsoft**, получив сертификацию, соответствующую той работе, которую вы выполняете сейчас или желаете получить в будущем.

Компании **Microsoft**, **CyberBionic Systematics** и **Intel** на базе портала [TestProvider](http://TestProvider) компании CyberBionic Systematics с использованием платформы Microsoft Azure совместно с Министерством науки и образования Украины проводят Всеукраинское дистанционное мониторинговое исследование уровня сформированности у выпускников учебных заведений навыков использования информационно-коммуникативных технологий в практической деятельности.



В последнее время JavaScript приобретает в жизни веб разработчика все больше и больше значения. Если ранее его называли языком для домохозяек, сейчас он является полноценным интерпретируемым объектно-ориентированным языком, который по своему богатству не уступает остальным. Возможности, которые скрывает в себе этот язык, поражают Вас каждый день новыми красивыми фото галереями, симпатичными шустрыми эффектами, выпадающими пунктами меню на ярких сайтах и многим другим. А ведь он все развивается и развивается дальше.

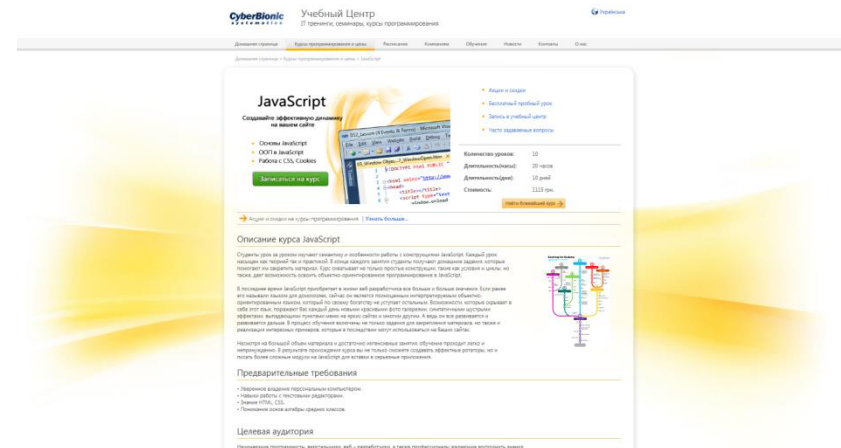
В курс JavaScript включены не только задания для закрепления материала, но также и реализация интересных примеров, которые в последствии могут использоваться на Ваших сайтах.

## Описание курса:

Курс JavaScript состоит из 10-ти уроков позволяющих начинающим разработчикам освоить все основные концепции JavaScript. Примеры курса составлены в простой форме и будут понятными, как разработчикам с опытом, так и начинающим программистам. Курс охватывает не только простые конструкции, такие как условия и циклы, но также, дает возможность освоить объектно-ориентированное программирование в JavaScript.

## Длительность:

20 часов/10 дней.



Узнать более подробно о курсе на сайте:

[www.edu.cbssystematics.com](http://www.edu.cbssystematics.com)

# **CyberBionic** **s y s t e m a t i c s**

*Coevolution of humans and machines.*