Entity Framework
&
ASP.NET MVC

Bc. Radim Göth

Obsah

- Entity Framework (EF)
  - Úvodem
    - Co je to ORM
    - Historie EF
    - Srovnání s NHibernate
  - Demo
    - Code First, Database First, Model First
    - Vytváření a ukládání entit
    - Dotazy
    - Další funkce a vlastnosti

ADO.NET

- Sada tříd poskytujících nástroje pro manipulaci a přístup k datům (v relační DB, XML nebo aplikačních dat)
- Součást .NET Frameworku

Příklad práce s ADO.NET

```
SqlConnection conn = new SqlConnection(„conStr");

SqlCommand cmd = new SqlCommand("SELECT * FROM  Subscriptions WHERE
                                        UserName = @userName",conn);


cmd.Parameters.Add(new SqlParameter("@userName", hdnuserName.Value));

SqlDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
    lblMySubscription.Text +=(string)reader["SubscriptionName"];
}
// Release sources
cmd.Dispose(); reader.Close(); conn.Close();
```

Objektově-relační mapování a EntityFramework

„Entity Framework (EF) je nástroj pro objektově-relační mapování (ORM), který .NET vývojářům usnadňuje práci s entitně-relačními daty tím, že jim umožní s daty uloženými v klasické relační databázi manipulovat jako s objekty.“

Objektově-relační mapování

Entitně relační svět	Objektově orientovaný svět
Entita	Objekt
Atribut	Vlastnost
Vztah a jeho kardinalita	Asociace, agregace, kompozice a násobnost těchto vztahů

Historie EF

- EF 3.5 – .NET 3.5 SP1 a Visual Studio 2008 SP1
- EF 4 – .NET 4 a Visual Studio 2010
- EF 4.x
- EF 5 – .NET 4 a 4.5, Visual Studio 2012
- EF 6 – plné využití .NET 4.5
- EF 6.1 – nejnovější verze

NHibernate

- Začátek vývoje v roce 2005 firmou JBoss
- Open source – komunitní projekt
- Ze začátku velmi slibný
- Dnes již velmi málo aktivní komunita vývojářů
- Chybí projektové řízení
- Dnes již zaostává za EF

EF6 – Podporované DB

- Microsoft SQL Server
- Microsoft SQL Server Compact Edition

- MySQL
- PostgreSQL
- Oracle

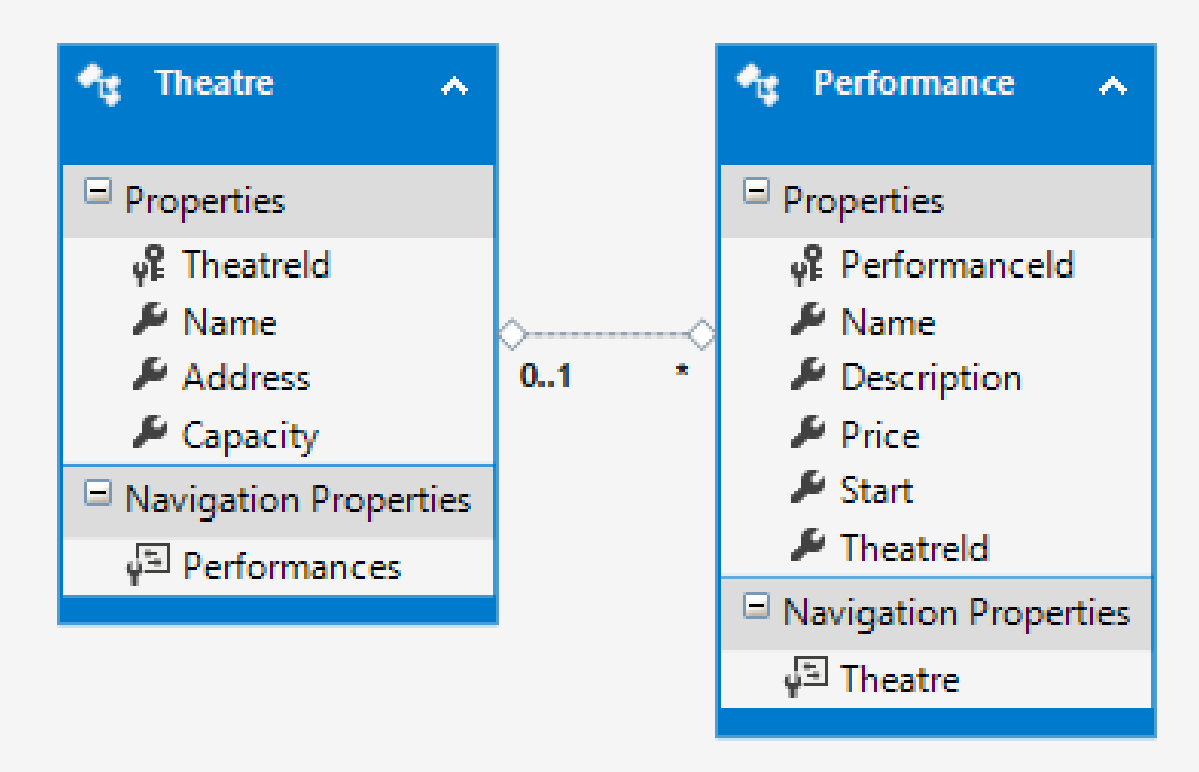
- A další poskytovatelé

http://msdn.microsoft.com/en-us/data/jj730568

EF – jak začít

- Code First
  - Vytvoření POCO tříd a databázového kontextu
- Database First
  - Vytvoření kontextu a tříd z existující databáze
- Model First
  - Vytvoření modelu v designéru Visual Studia

Projekt – Informační systém Národního divadla Brno



Code First

1. Vytvoření POCO tříd
2. Přidání cizích klíčů a navigačních vlastností
3. Přidání anotací vlastnostem
4. Vytvoření databázového kontextu

- První užití DB kontextu vytvoří databázi na localdb nebo SQLEXPRESS

Krok 1 – POCO třídy

```
public class Theatre
{
    public int TheatreId { get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public string Phone { get; set; }
}
```

Krok 1 – POCO třídy

```
public class Performance
{
    public int PerformanceId { get; set; }
    public string Description { get; set; }
    public string Name { get; set; }
    public Nullable<DateTime> Start { get; set; }
    public Nullable<decimal> Price { get; set; }
}
```

Krok 2 – navigační vlastnosti

```
public class Theatre
{
    public int TheatreId { get; set; }
    public string Name { get; set; }
    public string Address { get; set; }
    public string Phone { get; set; }

    public virtual List<Performance> Performances
                                        { get; set; }
}
```

Krok 2 – navigační vlastnosti

```
public class Performance
{
    public int PerformanceId { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public Nullable<DateTime> Start { get; set; }
    public Nullable<decimal> Price { get; set; }

    // Cizí klíč
    public int TheatreId { get; set; }
    public virtual Theatre Theatre { get; set; }
}
```

Krok 3 – Anotace

```
public class Theatre
{
    [Key]
    public int TheatreId { get; set; }

    [Required(ErrorMessage="Name is required")]
    [MaxLength(200,ErrorMessage="Maximum length is 200 chars")]
    public string Name { get; set; }

    [MinLength(5, ErrorMessage="Minimum length is 5 chars.")]
    public string Address { get; set; }

    //.....
}
```

Anotace

- System.ComponentModel.DataAnnotations;
- Princip Conventions Over Configurations
- Pokud dodržujeme konvence, často nejsou anotace potřeba. Např. [Key], [ForeignKey]
- Anotace pro Code First na msdn:

http://msdn.microsoft.com/en-us/data/jj591583

Krok 4 – DatabaseContext

```
using System.Data.Entity;

public class NationatTheatreContext : DbContext
{
    public DbSet<Theatre> Theatres { get; set; }
    public DbSet<Performance> Performances { get; set; }
}
```

Database First

Postup:

1. Vytvořím databázi
2. V databázi vytvořím tabulky, triggery,…
3. Do projektu ve Visual Studiu přidám ADO.NET Entity Data Model
4. Pomocí wizzardu vytvořím model z existující databáze

Model First

Postup:

1. Do projektu ve Visual Studiu přidám ADO.NET Entity Data Model
2. Ve wizzardu zvolím vytvoření prázdného modelu
3. Pomocí grafického návrháře vytvořím model

Code First vs. Database First

- Rychlé, jednoduché
- Nemusím opustit VS
- Čistší kód

- Nemám plnou kontrolu nad databází
- Jsou případy, kdy stejně musím opustit VS a použít SQL

Code First

- Úplná kontrola na databází (efektivita)

- Musím umět DDL v SQL

- Vygenerovaný, méně čistý kód
- Nutnost udržování edmx souboru

Database First

Vytváření a ukládání objektů

```
using (NationatTheatreContext db =
                              new NationatTheatreContext())
{
        Theatre mahen = new Theatre();
        mahen.Name = "Mahenovo divadlo";
        mahen.Address = "Malinovského nám. 1";
        mahen.Phone = "789456123";

        db.Theatres.Add(mahen);
        db.SaveChanges();
}
```

Dotazy

```
using(NationatTheatreContext db =
                    new NationatTheatreContext())
{
    var p = (from a in db.Performances
             where a.Name.Contains(“Králova")
             select a).ToList();
}
```

Líné načítání

```
var theatre = (from a in db.Theatres
               where a.Name.Contains("Mahen")
               select a).FirstOrDefault();
```

- Pokud budu chtít manipulovat s `theatre.Performances`, způsobím vyvolání dalšího dotazu.

Vypnutí líného načítání

1. Nedělat navigační vlastnosti virtual

2. Nastavení v konstruktoru kontextu

```
public NationatTheatreContext()
{
      this.Configuration.LazyLoadingEnabled = false;
}
```

Dychtivé načítání

```
var t2 = db.Theatres
          .Where(x => x.Name.Contains("Mahen"))
          .Include(x => x.Performances)
          .FirstOrDefault();
```

- Include mi zajistí, že výsledný dotaz bude sestaven tak, aby se s objektem typu `theatre` načetla i všechna jeho vystoupení.

EF – závěr

- Jednoduchý framework
- Jednoduché použití
- Relativně bezpečný

- MSDN data developement:

http://msdn.microsoft.com/en-us/data/default

ASP.NET MVC

ASP.NET

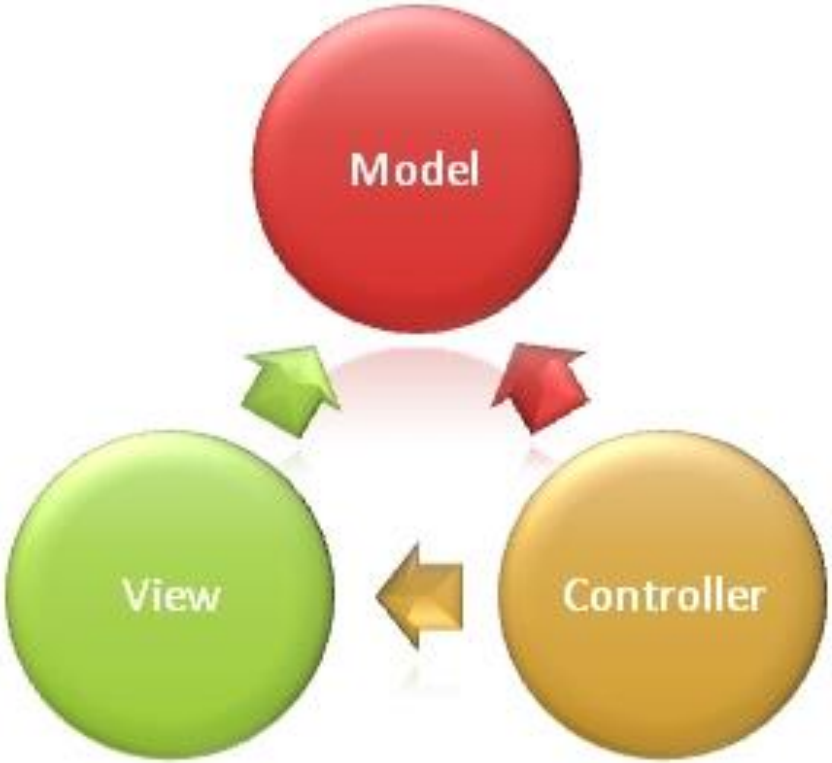
- Prvně pouze ASP
  - Skriptovací nástroj
  - Poslední verze v roce 2000

- Poté ASP.NET
  - Platforma pro tvorbu webových aplikací a služeb
  - Součást .NET Frameworku
  - Typy projektů
    - MVC
    - WebForms
    - A další…

Http

- Bezstavový protokol
- Spolehlivé doručení (pracuje nad TCP)
- Funguje na principu request – response
- Definuje několik metod, hlavní jsou:
  - GET
  - POST
- http status kódy
- Zabezpečená verze https

MVC architektura



Zdroj: www.w3schools.com

MVC

- Model
  - DB + Entity framework
- Controller
  - System.Web
  - System.Web.Mvc
- View
  - ASPX
  - Razor (.cshtml)

Controller

- Convension over Configuration

```
public class HomeController : Controller
{
    // GET: www.mojedomena.cz/Home/Search
    public ActionResult Search()
    {
        ViewBag.Message = "Best search tool";
        return View();
    }
}
```

View – Razor

```
<h2>@ViewBag.Message</h2>

@using(Html.BeginForm("Search","Home", FormMethod.Get))
{
      <input type="text" name="substr" />
      <input type="submit" value="Search" />
}
```

View – Html

```
<h2>Best search tool</h2>

<form action="/Home/Search" method="get">
     <input type="text" name="substr" />
     <input type="submit" value="Search" />
</form>
```

View – ASPX

```
<% using (Html.BeginForm("Search","Performances",
                                    FormMethod.Get)) { %>
    <input type="text" name="substr" />
    <input type="submit" value="Search" />
<% } %>
```

Formuláře

- GET / POST
- Pro každou metodu akci v kontroleru
- Lze využít Scaffolding templates

Zdroje

- Professional ASP.NET MVC 4 / Jon Galloway
- Oficiální ASP.NET MVC stránka:
  http://www.asp.net/mvc