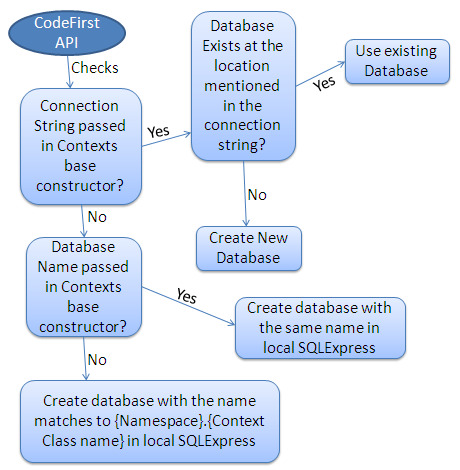
# [DB Initialization（数据库初始化）[EF Code-First系列]](http://www.cnblogs.com/caofangsheng/p/5021829.html)

前面的例子中，我们已经看到了Code-First自动为我们创建数据库的例子。

这里我们将要学习的是，当初始化的时候，Code-First是怎么决定数据库的名字和服务的呢？？？

下面的图，解释了这一切！！！

这个图解释了，数据库初始化的流程，是基于我们在上下文类中的构造函数中传递的参数。



在上面的图中，context类中的base构造器中，可以填入下面的参数：

* 无参数（No Parameter）
* 数据库的名字（Database Name）
* 连接字符串的名字（Connection String Name）

### **无参数**

如果你没有在base构造器中指定任何的参数，Code-First将会以你本地的服务器名字，也就是：{Namespace}.{Context class name}.例如下面代码中：code-First将会将会一个名字为EF1.DbContextClass的数据库。

using System;using System.Collections.Generic;using System.Data.Entity;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF1

{

public class DbContextClass:DbContext

{

public DbContextClass()

: base()

{ }

public DbSet<Student> Studnets { get; set; }

public DbSet<Standard> Standards { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

Database.SetInitializer(new DropCreateDatabaseIfModelChanges<DbContextClass>());

base.OnModelCreating(modelBuilder);

}

}

}

IMG_259

### **数据库名字**

你同样可以指定数据库的名字在base构造器中，例如：

using System;using System.Collections.Generic;using System.Data.Entity;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF1

{

public class DbContextClass:DbContext

{

public DbContextClass()

: base("MYDBHAHA")

{ }

public DbSet<Student> Studnets { get; set; }

public DbSet<Standard> Standards { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

Database.SetInitializer(new DropCreateDatabaseIfModelChanges<DbContextClass>());

base.OnModelCreating(modelBuilder);

}

}

}

然后生成的数据库是：

IMG_262

### **连接字符串**

using System;using System.Collections.Generic;using System.Data.Entity;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF1

{

public class DbContextClass:DbContext

{

public DbContextClass()

: base("ConnectionString")

{ }

public DbSet<Student> Studnets { get; set; }

public DbSet<Standard> Standards { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

Database.SetInitializer(new DropCreateDatabaseIfModelChanges<DbContextClass>());

base.OnModelCreating(modelBuilder);

}

}

}

或者这样写：

using System;using System.Collections.Generic;using System.Data.Entity;using System.Linq;using System.Text;using System.Threading.Tasks;

namespace EF1

{

public class DbContextClass:DbContext

{

public DbContextClass()

: base("name=ConnectionString")

{ }

public DbSet<Student> Studnets { get; set; }

public DbSet<Standard> Standards { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

Database.SetInitializer(new DropCreateDatabaseIfModelChanges<DbContextClass>());

base.OnModelCreating(modelBuilder);

}

}

}

上面任选一种方式，改动之后，我们在配置文件中，添加下面的节点：

<connectionStrings>

<add name="ConnectionString" connectionString="server=.;database=EFCodeFirstDB;uid=sa;pwd=Password\_1" providerName="System.Data.SqlClient"/>

</connectionStrings>

这个节点，我一般都是加在这个位置：

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<configSections>

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false"/>

</configSections> <connectionStrings>

<add name="ConnectionString" connectionString="server=.;database=EFCodeFirstDB;uid=sa;pwd=Password\_1" providerName="System.Data.SqlClient"/>

</connectionStrings>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5"/>

</startup>

<entityFramework>

<defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">

<parameters>

<parameter value="v11.0"/>

</parameters>

</defaultConnectionFactory>

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer"/>

</providers>

</entityFramework>

</configuration>

改好之后，我们运行程序，在数据库管理工具中就生成了我们想要的数据库：



这就是三种方式的初始化数据库了。