< Previous

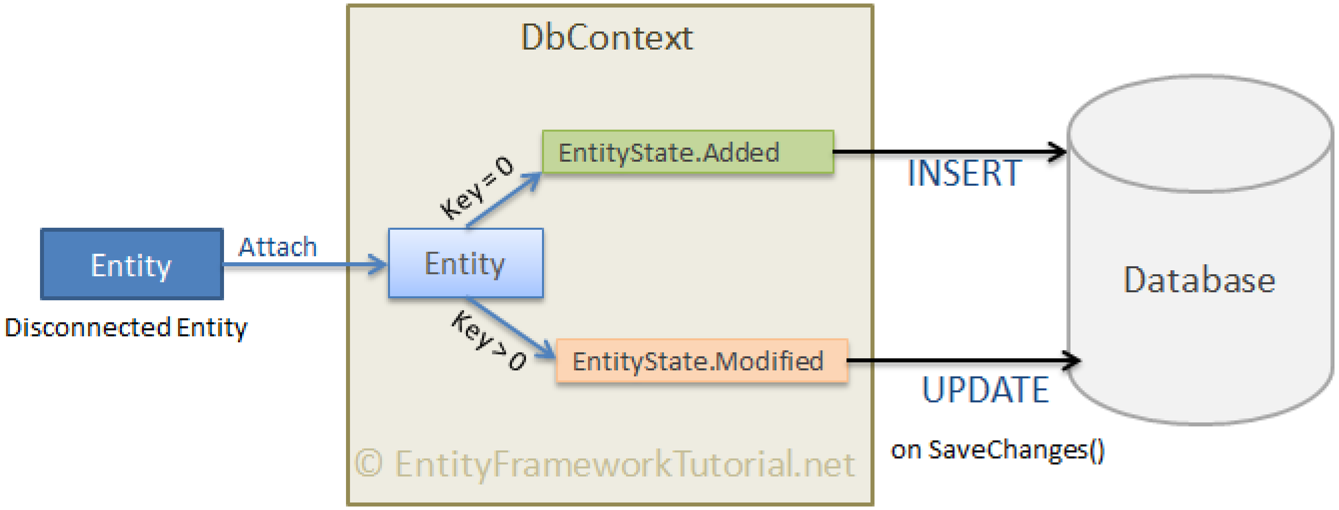
Next >

Saving a Disconnected Entity in EF 6

In this chapter, you will learn how to save a disconnected entity which is not being tracked by a context in EF 6.

Saving data in the disconnected scenario is a little bit different than in the connected scenario. In the disconnected scenario, an instance of `DbContext` is not aware of the disconnected entities because the entities were created or modified out of the scope of the current `DbContext` instance. So, you need to attach the disconnected entities to a context with an appropriate `EntityState` in order to perform INSERT or UPDATE operations in the database.

In the disconnected scenario, you need to find out whether an entity is new or existing and based on that you can set the `EntityState`. Here, if the key property value is zero then we will consider it a new entity and so we will set the Added state. If the key property value is greater than zero, then it means it is an existing entity and so we will set the Modified state.



The following example demonstrates saving a disconnected entity.

```
// disconnected new entity
var student = new Student(){ StudentName = "Bill" };

using (var context = new SchoolDBEntities())
{
    context.Entry(student).State = student.StudentId == 0? EntityState.Added : EntityState.Modified;

    context.SaveChanges();
}
```

In the above example, `studnet` is a disconnected entity object and `context` is not aware of its state. `context.Entry(student).State = student.StudentId == 0? EntityState.Added : EntityState.Modified;` sets the Added state if the value of the key property `StudentId` is zero, otherwise it sets the Modified state. The `SaveChanges()` method will build and execute the following INSERT command to the database.

```
exec sp_executesql N'INSERT [dbo].[Student]([StudentName], [StandardId])
VALUES (@0, NULL)
SELECT [StudentID] FROM [dbo].[Student]
WHERE @@ROWCOUNT > 0 AND [StudentID] = scope_identity(),@0='Bill'
```

The same way, if the value of `StudentId` is non-zero, then it will assign the Modified state and so, the `SaveChanges()` method will execute the UPDATE command.

```
// disconnected existing entity
var student = new Student(){ StudentId = 1, StudentName = "Steve" };

using (var context = new SchoolDBEntities())
{
    context.Entry(student).State = student.StudentId == 0? EntityState.Added : EntityState.Modified;

    context.SaveChanges();
}
```

In the above example, an object of `Student` entity includes the key property `StudentId` greater than zero, so it will be marked as Modified. This will execute the following UPDATE command in the database.

```
exec sp_executesql N'UPDATE [dbo].[Student]
SET [StudentName] = @0
WHERE @@ROWCOUNT > 0 AND [StudentID] = @1'N'@0 varchar(50),@1 int',@0='Steve',@1=1
```

Download EF 6 DB-First Demo Project from Github

< Previous

Next >

ENTITYFRAMEWORKTUTORIAL

Learn Entity Framework using simple yet practical examples on EntityFrameworkTutorial.net for free. Learn Entity Framework DB-First, Code-First and EF Core step by step. While using this site, you agree to have read and accepted our terms of use and privacy policy.

✉ feedback@entityframeworktutorial.net

TUTORIALS

- EF Basics
- EF 6 DB-First
- EF Core
- EF 6 Code-First

E-MAIL LIST

HOME PRIVACY POLICY ADVERTISE WITH US