

Migrations

Code First Migrations

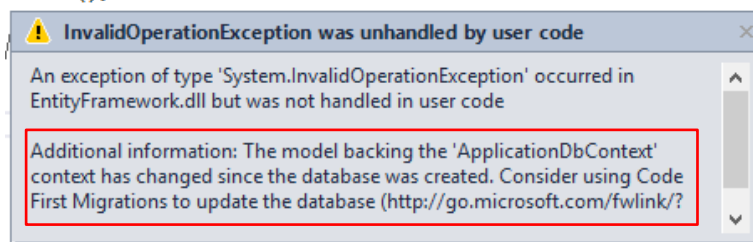
After we get our application to run, we might decide that we want to **change** one or more of our models.

Let's continue with our Book Demo App and see what we would have to do to make changes to a model at this point.

Let's add an ISBN property to the Book class.

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

Now when we run the application, we get the following exception.



To avoid having to start over and create the application all over again, we will use **Code First Migrations**.

Run the **Enable-Migrations** command in the NuGet Package Manager Console. If there is more than one context class in your application, use `Enable-Migrations -ContextTypeName BookDemoApp.Models.BookDemoAppContext`

```
Package Manager Console
Package source: nuget.org | Default project: BookDemoApp

PM> Enable-Migrations
Checking if the context targets an existing database...
Detected database created with a database initializer. Scaffolded migration
'201611280211515_InitialCreate' corresponding to existing database. To use an automatic migration
instead, delete the Migrations folder and re-run Enable-Migrations specifying the
EnableAutomaticMigrations parameter.
Code First Migrations enabled for project BookDemoApp.
PM> |
```

Now run the **Add-Migration AddISBN**. We name our migrations and we are calling this one AddISBN.

```
PM> Add-Migration AddISBN
Scaffolding migration 'AddISBN'.
The Designer Code for this migration file includes a snapshot of your current Code First model.
This snapshot is used to calculate the changes to your model when you scaffold the next migration.
If you make additional changes to your model that you want to include in this migration, then you
can re-scaffold it by running 'Add-Migration AddISBN' again.
PM> |
```

Run the **Update-Database** command.

Code First Migrations will compare the migrations in the **Migrations** folder with the ones that have been applied to the database. It will see that the **AddISBN** migration needs to be applied.

```
PM> Update-Database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [201611280240254_AddISBN].
Applying explicit migration: 201611280240254_AddISBN.
Running Seed method.
PM> |
```

Index

Create New

PublisherName	Title	ISBN	Price	
Pearson	Intermediate Algebra		115.95	Edit Details Delete
McGraw-Hill	College Algebra		104.99	Edit Details Delete
Pearson	Statistics		89.90	Edit Details Delete

Our application will now run again. For this change, we would need to go into all of our Books views and add the ISBN to our code. But that's pretty easy.

Notice that our books are still in the database but an empty field called ISBN was added for each record.

For more information, visit [https://msdn.microsoft.com/en-us/library/jj591621\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/jj591621(v=vs.113).aspx)

We now need to change all of the other Books views so that ISBN will be included.

One more thing: Very Important!

In the Edit and Create methods in the Books controller, add ISBN to the Bind attribute.

```
([Bind(Include = "BookID,Title,ISBN,PublisherID,Price")])
```