

Models



What is a Model?

- Model represents the domain the application focusses on
- They are plain C# classes
- They can be modelled with Entity relationships between them

MVC Support - Scaffolding

- MVC framework provided scaffolding creates Controllers and views for the standard index, edit, create and delete scenarios.

Controller Scaffolding Options

- Empty Controller
- Controller with Empty Read/Write Actions
- API Controller with Empty Read/Write Actions
- Controller with Read/Write Actions and Views, Using Entity Framework

Entity framework and Scaffolding

- EF is a ORM framework that understands how to map the objects to the database
- EF retrieves data from the database using LINQ
- EF with MVC supports Code First that enables this

Code First Conventions

- The model object's name is the name with which the database tables are created
- A property on the model object with the name ID is assumed as the primary key
- EF sets up auto incrementing (identity) key column in SQL Server

DbContext

- DbContext is the gateway to the database
- DbContext with the DbSet as shown below helps create the database tables

```
public class MusicStoreDB : DbContext
{
    public DbSet<Album> Albums { get; set; }
    public DbSet<Genre> Genres { get; set; }
    public DbSet<Artist> Artists { get; set; }
}
```

Configuring Connections

- Connection string name must match the name of the data context class

```
<connectionStrings>
```

```
<add name="MusicStoreDB"
```

```
connectionString="datasource=.\SQLEXPRESS;Integrated Security=SSPI;
```

```
initial catalog=MusicStore" providerName="System.Data.SqlClient" />
```

```
</connectionStrings>
```


Loading Related Objects

- Eager Loading Strategy

```
public ViewResult Index()  
{  
    var albums = db.Albums.Include(a => a.Genre).Include(a => a.Artist);  
    return View(albums.ToList());  
}
```

Loading Related Objects

- Lazy Loading Strategy

- In this type of loading, related entities are automatically loaded from the data source when you access a navigation property

Each navigation property must be declared as **public, virtual** (**Overridable** in Visual Basic), and not **sealed** (**NotOverridable** in Visual Basic) get accessor

Views and Scaffolding

- Index
- Edit
- Create
- Delete

The above views can be created as a result of scaffolding

Model Binding

- The scaffolding framework created input names to match the Model property names
- The Model Binder aggressively looks for properties in Route Data, Query String, Url and form values

Model Binding Security

- Over-Posting Attack
 - Aggressive search behaviour can lead to unwanted values being posted
 - This can be handled using the keyword Exclude