2.17 Data Validation Using MetaData



This section will guide you to:

* Implement and test metadata validation

This guide has nine subsections, namely:

2.17.1 Creating an ASP.NET MVC project

2.17.2 Creating a Student table in a database

2.17.3 Adding EntityFramework using Nuget

2.17.4 Using EntityFramework to generate models and views

2.17.5 Creating StudentController to auto-create the views for CRUD operations

2.17.6 Creating a StudentMeta model to validate the Student class

2.17.7 Building the project

2.17.8 Publishing and running the project

2.17.9 Pushing the code to your GitHub repositories

**Step 2.17.1:** Creating an ASP.NET MVC project

* Open Visual Studio.
* From the top menu, select **File->New->Project**.
* In **Create A New Project** screen, select **ASP.NET Web Application(.Net Framework)** from the list of available project types and click on **Next.**
* Please select the project type where the **language** of the project is **C#.**
* Enter **Project Name** as **Phase3Section2.20** and click on **Create.**
* From the list of project sub-types, choose **Web MVC** and uncheck **Configure for HTTPS.** Click on **Create.**
* This will create the files for an ASP.NET MVC project.

**Step 2.17.2:** Creating a Student table in a database

* Open SQL Server Management Studio. In the login screen, make a note of the **Server Name** value as you will need to put in your ASP.NET application.
* In **Object Explorer**,right click **Databases** and choose **New Database**.
* Enter **Database name** as **School1** and click **Ok**.
* In **Object Explorer**,expand **School1->Tables.** Right click **Tables** and choose **New->Table**.
* For the first row enter ID as **Column Name**,int as **Data Type**, anduncheck **Allow Nulls**.
* In **Column Properties**,go to **Identity Specification** and expand it. Double click **Is Identity** to make it **Yes**.
* For the next row enter Name as **Column Name**,varchar(100) as **Data Type**, anduncheck **Allow Nulls**.
* For the next row enter Address as **Column Name**,varchar(100) as **Data Type**, anduncheck **Allow Nulls**.
* For the next row enter Email as **Column Name**,varchar(75) as **Data Type**, anduncheck **Allow Nulls**.
* For the next row enter Class as **Column Name**,varchar(5) as **Data Type**, anduncheck **Allow Nulls**.
* Click on the x icon to close the table grid tab. Click **Yes** on the save dialog.
* For **Enter a name** put Student and press **Ok**.
* In **Object Explorer**,expand **School1->Tables->Student.** Right click **Tables** and choose **Edit Top 200 Rows**.
* Add in a few rows of data with random values.
* Close the Management Studio.

**Step 2.17.3:** Adding EntityFramework using NuGet

* In the top menu, go to **Tools->Nuget Package Manager->Package Manager Console**.
* In the console, type **install-package entityframework** and press **Enter**.

**Step 2.17.4:** Using EntityFramework to generate models and views

* In the **Solution Explorer**,right click **Phase3Section2.34** and choose **Add->New Item**.
* Under **Visual C#**,select **Data** andchoose **ADO.NET Entity Data Model.** Enter **Name** as Student and click **Add**.
* In the next screen, click on **EF Designer from Database** and click **Next**.
* In the next screen, click on **New Connection**.
* This will open a popup window. In the **Server Name**,put the server name from the SQL Management Studio login screen.
* In **Select or enter a Database name**,choose **School1** and click **Ok**.
* This will close the popup window. In the current window, make sure **Save Connection settings in web.config** is checked and make a note of the value in the textbox eg.School1Entities.
* Click on **Next**.
* In the next screen, expand **Tables->dbo->Student,** check **Student**, andclick **Finish.**
* This will create all the entity files for the Student table.

**Step 2.17.5:** Creating StudentController to auto-create the views for CRUD operations.

* In **Solution Explorer**,expand **Controllers**,right click **Controllers** and choose **Add->Controller**.
* From the list of types, choose **MVC5 Controller with views**, **using Entity Framework** and click **Add**.
* In the next screen, choose **Model Class** as Student, **Data Context Class** as School1Entities, **Controller Name** as StudentsController and click **Add**.
* This will generate the code for the Controller for all CRUD operations.

**Step 2.17.6:** Creating a StudentMeta model to validate the Student class

* In **Solution Explorer**,right click **Models** and choose **Add->Class**.
* Enter Class Name as StudentMetaData.cs and click **Add**.
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Web;

**using** System.ComponentModel.DataAnnotations;

**namespace** Phase3Section3.\_34.Models

{

**public** **class** StudentMetaData

{

**public** **int** ID { **get**; **set**; }

[Required (ErrorMessage = "Name is required")]

[StringLength(50)]

[Display(Name = "Student Name")]

**public** **string** Name { **get**; **set**; }

[Required(ErrorMessage = "Class is required")]

[StringLength(5)]

[Display(Name = "Which Class ")]

**public** **string** Class { **get**; **set**; }

[Required(ErrorMessage = "Address is required")]

[StringLength(100)]

[Display(Name = "Address")]

**public** **string** Address { **get**; **set**; }

[Required(ErrorMessage = "Email is required")]

[StringLength(50)]

[Display(Name = "Student Email")]

[EmailAddress(ErrorMessage = "Invalid Email Address")]

**public** **string** Email { **get**; **set**; }

}

}

**Step 2.17.7:** Building the project

* From the top menu, choose **Build->Build Solution**.
* If any compile errors are shown, fix them as required.

**Step 2.17.8:** Publishing and running the project

* From the top menu, select **Debug->Start Without Debugging**.
* This will execute the program in the default browser.
* To see the student pages, go to the url : http://localhost:xxxx/students.

**Step 2.17.9:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master