2.9 Using Custom Model Binding



This section will guide you to:

* Implement and test custom model binder

This guide has eight subsections, namely:

2.9.1 Create an ASP.NET MVC project to show custom model binding

2.9.2 Create a StudentModel Model

2.9.3 Create StudentEditView View to edit student profile

2.9.4 Change Index.cshtml to link to Student Profile form

2.9.5 Make changes to HomeController to bind StudentModel to StudentEditView

2.9.6 Build the project

2.9.7 Publish and run the project

2.9.8 Push the code to your GitHub repositories

**Step 2.9.1:** Create an ASP.NET MVC project to show custom Model Binding

* Open Visual Studio.
* From the top menu, select **File->New->Project.**
* In **Create A New Project** screen, select **ASP.NET Core Web Application** from the list of available project types and click on **Next.**
* Enter **Project Name** as **Phase3Section2.18** and click on **Create.**
* From the list of project sub-types, choose **Web Application (Model-View-Controller)** and uncheck **Configure for HTTPS.** Click on **Create.**
* This will create the files for an ASP.NET MVC project.

**Step 2.9.2:** Create a StudentModel Model

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

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Threading.Tasks;

**namespace** Phase3Section2.\_18.Models

{

**public** **class** StudentModel

{

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

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

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

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

}

}

**Step 2.9.3:** Create StudentEditView View to edit student profile

* In **Solution Explorer**,expand **Views->Home.** Right click **Home** and choose **Add->View**.
* Enter **View Name as** StudentEditView and click **Add.**
* Enter the following script:

@model StudentModel

@{

ViewData["Title"] = "StudentEditView";

}

<h2>StudentEditView</h2>

<div **class**="container">

<div **class**="row">

<div **class**="col-sm-6">

@**using** (Html.BeginForm())

{

@Html.EditorForModel("StudentModel")

<button **class**="btn btn-default">Save</button>

}

</div>

</div>

</div>

**Step 2.9.4:** Change Index.cshtml to link to Student Profile form

* In **Solution Explorer**,expand **Views->Home** and double-click index.cshtml.
* Enter the following script:

@{

ViewData["Title"] = "Home Page";

}

<**div** class="container">

<**br** />

<**div** class="row">

<**div** class="col-sm-12">

<**a** href="/Home/StudentEditView">Student Edit Form using Custom Model Binder</**a**>

</**div**>

</**div**>

</**div**>

**Step 2.9.5:** Make changes to HomeController to bind StudentModel to StudentEditView

* In **Solution Explorer**,expand **Controllers** and double click HomeController.
* Enter the following code:

**using** System;

**using** System.Collections.Generic;

**using** System.Diagnostics;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** Microsoft.AspNetCore.Mvc;

**using** Phase3Section2.\_17.Models;

**namespace** Phase3Section2.\_17.Controllers

{

**public** **class** HomeController : Controller

{

**public** IActionResult Index()

{

**return** View();

}

**public** IActionResult About()

{

ViewData["Message"] = "Your application description page.";

**return** View();

}

**public** IActionResult Contact()

{

ViewData["Message"] = "Your contact page.";

**return** View();

}

**public** IActionResult Privacy()

{

**return** View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = **true**)]

**public** IActionResult Error()

{

**return** View(**new** ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

**public** IActionResult StudentEditView()

{

**return** View();

}

[HttpPost]

**public** ContentResult StudentEditView(StudentModel model)

{

String result = model.Name + "," + model.Address + "," + model.WhichClass + "," + model.TotalMarks.ToString();

**return** Content("Form was submitted:\n" + result);

}

}

}

**Step 2.9.6:** Build the project

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

**Step 2.9.7:** Publish and run 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 2.9.8:** 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