Progress Check Exercise 2

Creating a Controller

In this exercise, you will learn how to update the controller to implement simple functionality of the Music Store application. That controller will define action methods to handle each of the following specific requests:

- A listing page of the music genres in the Music Store
- A browse page that lists all of the music albums for a particular genre
- A details page that shows information about a specific music album

For the scope of this exercise, those actions will simply return a string by now.

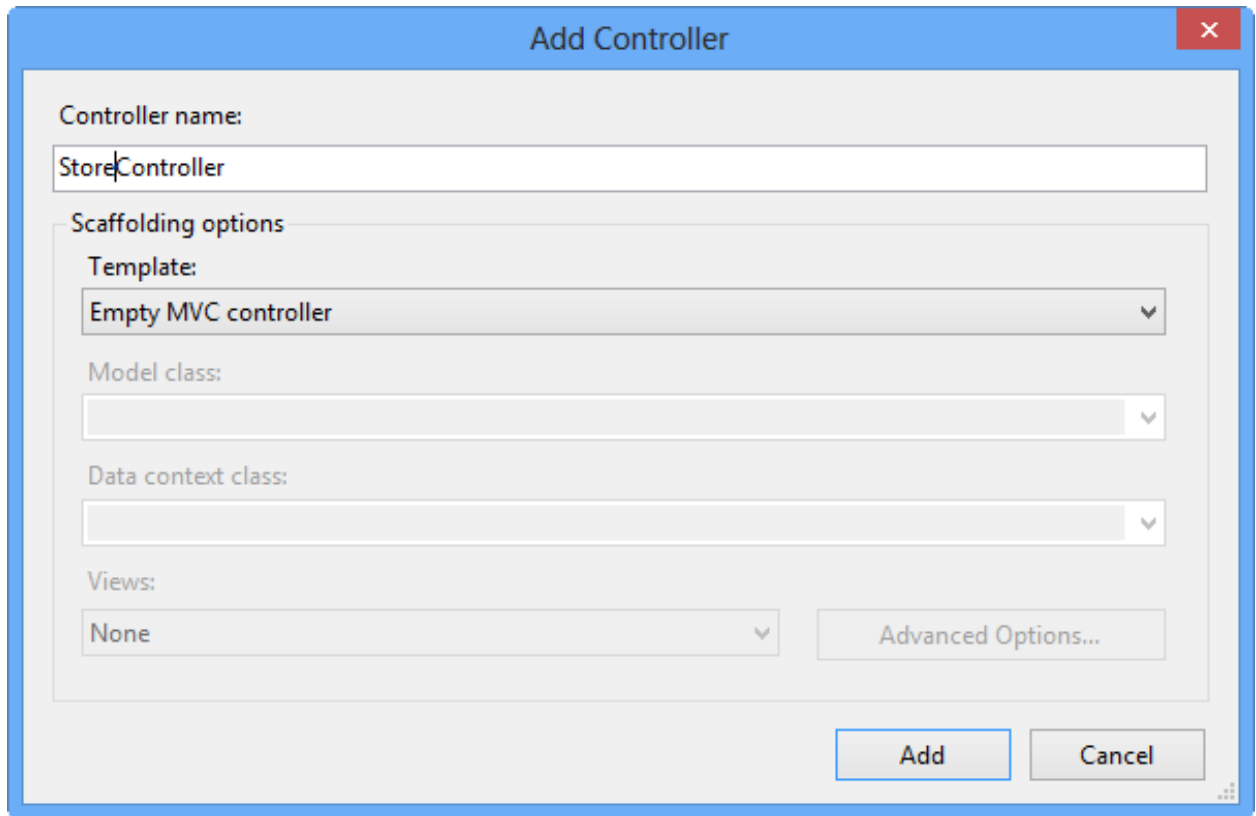**Task 1 - Adding a New StoreController Class**

In this task, you will add a new Controller.

1. If not already open, start **VS Express for Web 2012**.
2. In the **File** menu, choose **Open Project**. In the Open Project dialog, browse to **Source\Ex02-CreatingAController\Begin**, select **Begin.sln** and click **Open**. Alternatively, you may continue with the solution that you obtained after completing the previous exercise.
    1. If you opened the provided **Begin** solution, you will need to download some missing NuGet packages before continue. To do this, click the **Project** menu and select **Manage NuGet Packages**.
    2. In the **Manage NuGet Packages** dialog, click **Restore** in order to download missing packages.
    3. Finally, build the solution by clicking **Build** | **Build Solution**.

        Note

        One of the advantages of using NuGet is that you don't have to ship all the libraries in your project, reducing the project size. With NuGet Power Tools, by specifying the package versions in the Packages.config file, you will be able to download all the required libraries the first time you run the project. This is why you will have to run these steps after you open an existing solution from this lab.

3. Add the new controller. To do this, right-click the **Controllers** folder within the Solution Explorer, select **Add** and then the **Controller** command. Change the **Controller Name** to *StoreController*, and click **Add**.

Add Controller Dialog

Task 2 - Modifying the StoreController's Actions

In this task, you will modify the Controller methods that are called **actions**. Actions are responsible for handling URL requests and determining the content that should be sent back to the browser or user that invoked the URL.

1. The **StoreController** class already has an **Index** method. You will use it later in this Lab to implement the page that lists all genres of the music store. For now, just replace the **Index** method with the following code that returns a string "Hello from Store.Index()":

   (Code Snippet - *ASP.NET MVC 4 Fundamentals - Ex2 StoreController Index*)

   C#

- ```
  public string Index()
  {
    return "Hello from Store.Index()";
  }
  ```

- Add **Browse** and **Details** methods. To do this, add the following code to the **StoreController**:

(Code Snippet - *ASP.NET MVC 4 Fundamentals - Ex2 StoreController BrowseAndDetails*)

C#

```
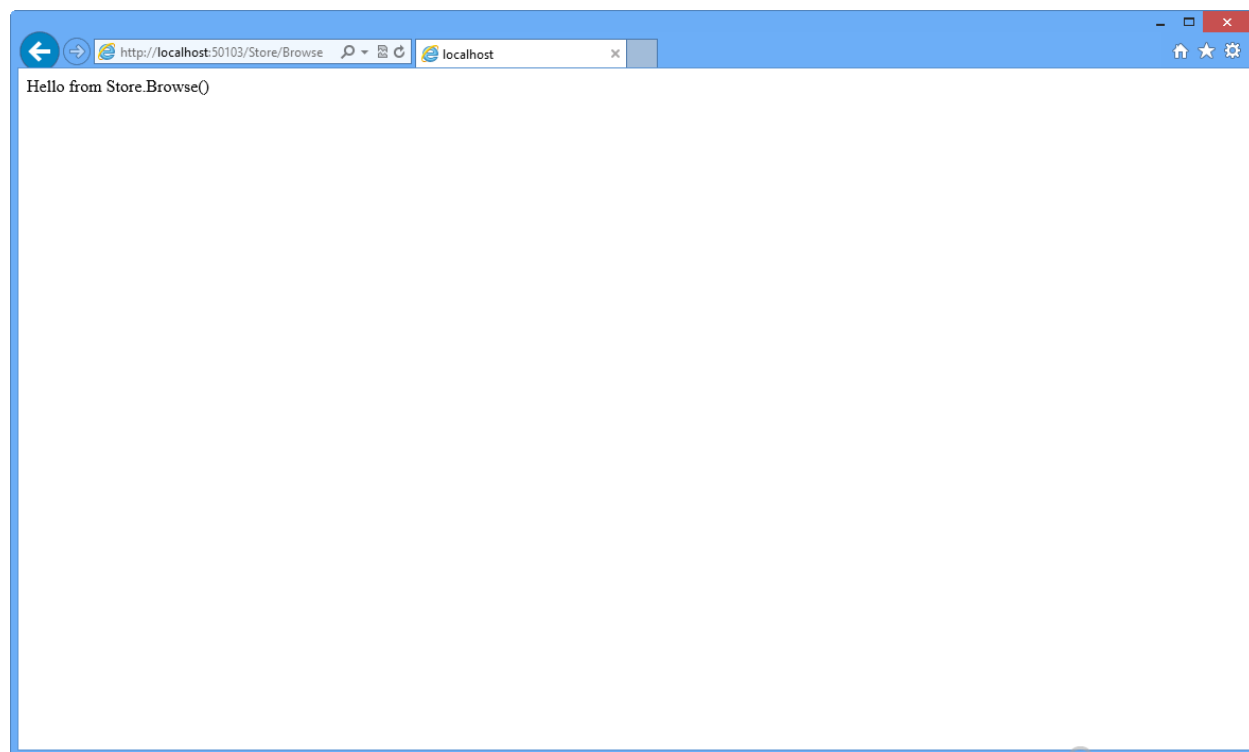2. // GET: /Store/Browse
3. public string Browse()
4. {
5.   return "Hello from Store.Browse()";
6. }
7.
8. // GET: /Store/Details
9. public string Details()
10. {
11.   return "Hello from Store.Details()";
12. }
13.
```

Task 3 - Running the Application

In this task, you will try out the Application in a web browser.

1. Press **F5** to run the Application.
2. The project starts in the **Home** page. Change the URL to verify each action's implementation.
   1. **/Store**. You will see **"Hello from Store.Index()"**.
   2. **/Store/Browse**. You will see **"Hello from Store.Browse()"**.
   3. **/Store/Details**. You will see **"Hello from Store.Details()"**.

Browsing /Store/Browse

3. Close the browser.