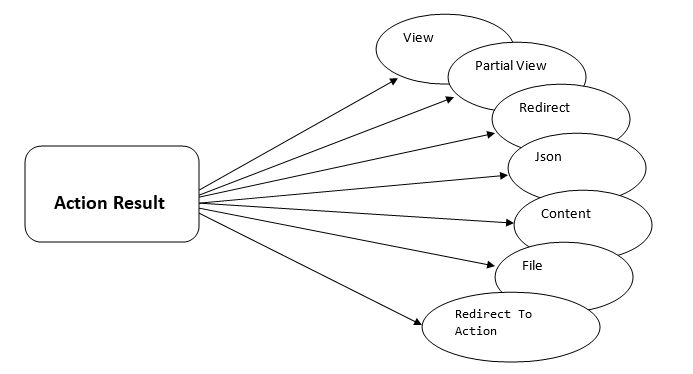
**All Types Of Action Methods In ASP.NET MVC 5**

### WHAT IS ACTION METHOD IN ASP.NET MVC 5?

All the public methods which are written inside a Controller are known as Action Method. When creating Action Method you must follow these rules.

**a.** Action method must be public.  
**b.** It cannot be overloaded.  
**c.** It cannot be a static method.  
**d.** Every controller has at least one default Action method Index() that returns the view page.  
**e.** ActionResult is a base class of all the result type action methods.

### TYPES OF ACTION METHOD

[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/2.jpg)

**ActionResult** is the base class of all the result type action method. There are following Result type action method in MVC.

**ViewResult** - Represents HTML and markup.  
**EmptyResult** - Represents no result.  
**RedirectResult** - Represents a redirection to a new URL.  
**JsonResult** - Represents a JavaScript Object Notation result that can be used in an AJAX application.  
**JavaScriptResult** - Represents a JavaScript script.  
**ContentResult** - Represents a text result.  
**FileContentResult** - Represents a downloadable file (with the binary content).  
**FilePathResult** - Represents a downloadable file (with a path).  
**FileStreamResult** - Represents a downloadable file (with a file stream).

**Programming Example**

**1. ViewResult**

1. // GET: Item
2. public **ViewResult** **Index**()
3. {
4. **ViewBag**.**ItemList** = "Computer Shop Item List Page";
5. return **View**();
6. }

**Output:** It will return Index View Page

**EmptyResult**

1. //GET: Item
2. public **EmptyResult** **Index**()
3. {
4. **ViewBag**.**ItemList** = "Computer Shop Item List Page";
5. return new **EmptyResult**();
6. }

**Output:** It will return a blank page with no result.

**RedirectResult**

Redirect Results is used for returning results to specific url. When you need to redirect to another action method, you can use RedirectResult Action Method.

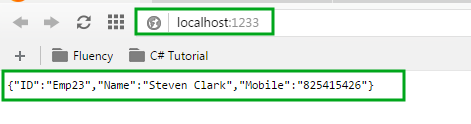
1. public **RedirectResult** **Index**()
2. {
3. return **Redirect**("Home/Contact");
4. }

When you will run this code, It will redirect you to **Contact** Page.

**JSONResult**

**JSON** Result returns simple text file format and key value pairs. Sometimes you may want to return data in JSON Format and that situation you JSONResult is the best option.

1. public **JsonResult** **Index**()
2. {
3. **Employee** emp = new **Employee**()
4. {
5. ID = "Emp23",
6. **Name** = "Steven Clark",
7. **Mobile** = "825415426"
8. };
9. return **Json**(emp, **JsonRequestBehavior**.**AllowGet**);
10. }
11. public class **Employee**
12. {
13. public string ID { get; set; }
14. public string **Name** { get; set; }
15. public string **Mobile** { get; set; }
16. }

**Output**[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-1.png)

**JavaScriptResult**

It returns java script that can be executed on the client browser. It sends javascript content in response to browser. This block allow you to execute java script on client at run time.

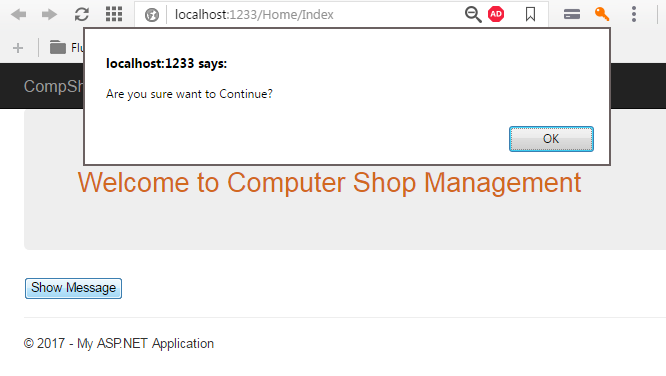
**Example:**  
**Step 1:** Add this **JavaScriptResult()** method in home controller.

1. [**HttpGet**]
2. public **JavaScriptResult** **WarningMessage**()
3. {
4. var msg = "alert('Are you sure want to Continue?');";
5. return new **JavaScriptResult**() { **Script** = msg };
6. }

**Step 2:** Open **Index.cshtml** and add the following highlighted code.

1. @{
2. **ViewBag**.**Title** = "Computer Shop Management";
3. }
4. <script src="~/Scripts/jquery-1.10.2.js"></script>
5. <div class="jumbotron">
6. <h2 style="color:chocolate">**Welcome** to **Computer** **Shop** **Management**</h2>
7. </div>

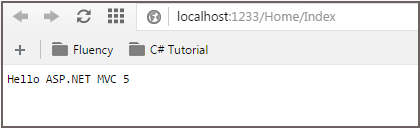
10. <script>
11. $(document).ready(function () {
12. $("button").click(function () {
13. $.getScript("/Home/WarningMessage");
14. });
15. });
16. </script>
18. <button>**Show** **Message**</button>

**Output**[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-1-1.png)

**Content Result**

It returns user-defined content type. It is useful when you want to send some plain text message to browser screen.

1. public **ContentResult** **Index**()
2. {
3. return **Content**("Hello ASP.NET MVC 5", "text/plain", **System**.**Text**.**Encoding**.UTF8);
4. }

**Output**[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-2-2.png)

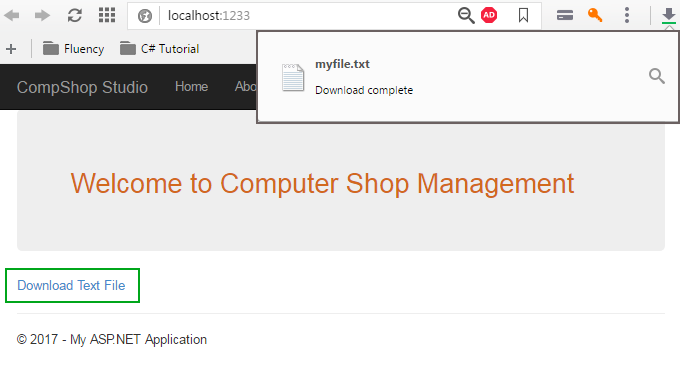
**File Result**

It represents the content of the file.  
**Step 1:** Add following code in HomeControllers

1. [**HttpGet**]
2. public **FileResult** **Download**()
3. {
4. byte[] fileBytes = **System**.IO.**File**.**ReadAllBytes**(@"D:\folder\myfile.txt");
5. string filename = "myfile.txt";
6. return **File**(fileBytes, **System**.**Net**.**Mime**.**MediaTypeNames**.**Application**.**Octet**, filename);
7. }

**Step 2:** Add Action Link in **Index.cshtml** page.

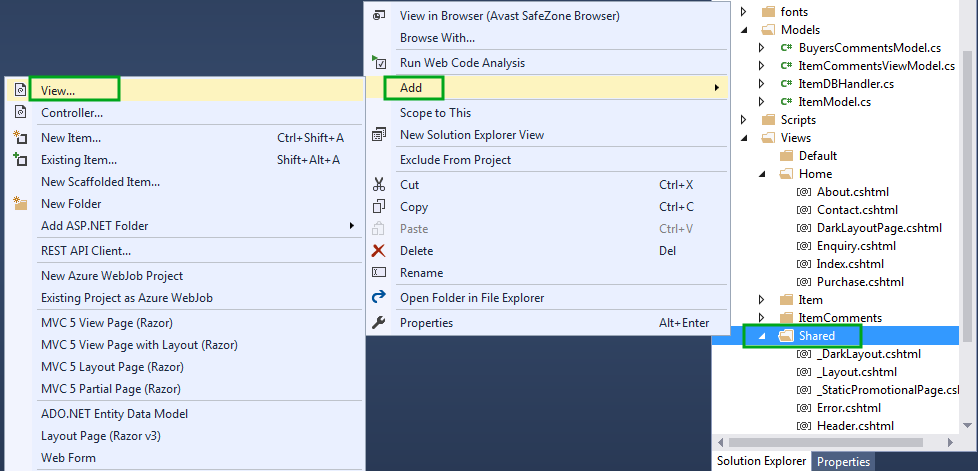
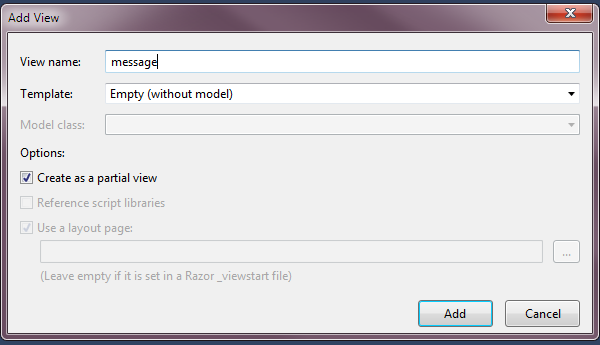
1. @Html.**ActionLink**("Download Text File","Download","Home")

**Output**[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-3-2.png)

**PartialViewResult**

Returns HTML from Partial view.

**Step 1:** Add a partial page. Go to **Solution Explorer**  **Shared**. **Right-click** on it and select **Add**  **View**.

[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-4-1.png)**Step 2:** Create partial view page as described in this picture.[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-5-3.png)**Step 3:** Add the following code in **message.cshtml** page.

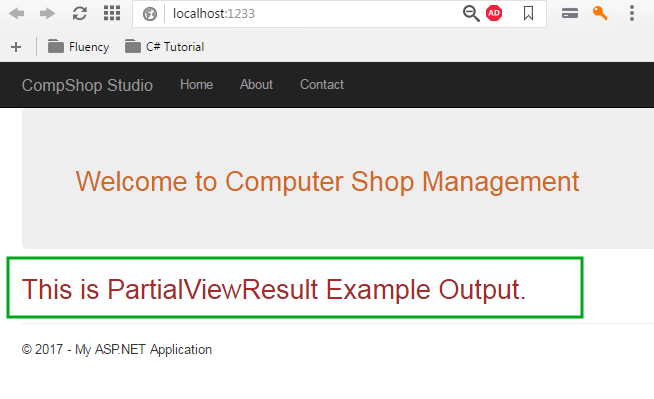
## **This is PartialViewResult Example Output.**

**Step 4:** Open **HomeController.cs** and add following line of code.

1. [**HttpGet**]
2. public **PartialViewResult** messagepage()
3. {
4. return **PartialView**("message");
5. }

**Step 5:** Go to **Index.cshtml** and add this code.

1. @{
2. **Html**.**RenderAction**("messagepage", "Home");
3. }

**Output**[](https://www.completecsharptutorial.com/wp-content/uploads/2017/09/New-Picture-6-3.png)