Difference between Html.RenderPartial vs Html.Partial vs Html.RenderAction vs Html.Action in MVC

In this post, we are going to discuss about Html.RenderPartial vs Html.Partial, Html.RenderAction vs Html.Action. Let we split this stuff in two ways  based on binding the static and dynamic content like **Render partial**and**Action Partial** .

**Partial Vs RenderPartial**

* @Html.Partial(“Details”)  –  returns a String as an output
* @{ Html.RenderPartial(“Details”); } –  returns void as an output

**Action Vs RenderAction**

* @Html.Action(“ActionName”, “ControllerName”) – returns a String as an output
* @{ Html.RenderAction(“ActionName”, “ControllerName”); } –  returns void as an output

**Render vs Action partial**

Let us discuss about the main difference between RenderAction and RenderPartial. RenderPartial will render the view on the same controller. But RenderAction will execute the action method , then builds a model and returns a view result.

**Render Partial :** Use Html.Partial when you are **rendering static content.**

**Action Partial :** Use Html.Action when handling **dynamic Content/data.**

**Where to use Html.Partial and Html.Action**

**Html.Partial :**If we need to display a static menu for all users from the partial view, then we can use **Render partial.**

**Example:**

**View :**Here we call the partial view from the view (passing model as input).

@Html.Partial(“\_partialName”, model)

**Html.Action:**If we need to display different menu items based on the user level, then we can use **Action Partial.**

**Example:**

**Controller:**

public ActionResult CustomerDetail() {  
return PartialView(new CustomerDetailModel());  
}

**View**

@model Model  
@Html.Action(“CustomerDetail”)

**Difference :**

1) @Html.RenderPartial and @Html.RenderAction is the best practice to use in most cases because the output has been written in the same TextWriter object.

2) In contrast to this, @Html.Partial and @Html.Action methods create their own TextWriter instances at every time and buffer all their content into memory.

3) @Html.Action and @Html.RenderAction method is the best choice when you want to cache a partial view.  
@Html.Partial and Html.RenderPartial method is the best choice when you want to load new view.

4) The return type for @Html.Partial and Html.Action returns string type values  
@Html.RenderAction and Html.RenderPartial returns is void

5) @Html.Partial and @Html.Action will useful when assign the content into a variable for other purpose