In this video, we will discuss **childactiononly**attribute in asp.net mvc. Let us understand this with an example.    
  
   
  
**Step 1:** Create a blank asp.net mvc 4 application  
  
**Step 2:** Add HomeController. Copy and paste the following code.  
public class HomeController : Controller  
{  
    // Public action method that can be invoked using a URL request  
    public ActionResult Index()  
    {  
        return View();  
    }  
  
    // This method is accessible only by a child request. A runtime   
    // exception will be thrown if a URL request is made to this method  
    [ChildActionOnly]  
    public ActionResult Countries(List<String> countryData)  
    {  
        return View(countryData);  
    }  
}  
  
   
  
**Step 3:** Right click on the **"Countries()"** action method and add **"Countries"** view. This view will render the given list of strings as an un-ordered list.  
  
@model List<string>  
@foreach (string country in Model)  
{  
    <ul>  
        <li>  
            <b>  
                @country  
            </b>  
        </li>  
    </ul>  
}  
  
**Step 4:** Right click on the **"Index()"** action method and add **"Index"** view.  Copy and paste the following code. Notice that, to invoke childaction, we are using Action() HTML Helper.  
  
<h2>Countries List</h2>  
@Html.Action("Countries", new { countryData = new List<string>() { "US", "UK", "India" } })  
  
**Please Note:** Child actions can also be invoked using **"RenderAction()"** HTMl helper as shown below.  
@{  
    Html.RenderAction("Countries", new { countryData = new List<string>() { "US", "UK", "India" } });  
}  
  
**Points to remember about "ChildActionOnly" attribute**  
**1.** Any action method that is decorated with [ChildActionOnly] attribute is a child action method.

**2.** Child action methods will not respond to URL requests. If an attempt is made, a runtime error will be thrown stating - Child action is accessible only by a child request.  
  
**3.** Child action methods can be invoked by making child request from a view using "Action()" and "RenderAction()" html helpers.  
  
**4.** An action method doesn’t need to have [ChildActionOnly] attribute to be used as a child action, but use this attribute to prevent if you want to prevent the action method from being invoked as a result of a user request.  
  
**5.** Child actions are typically associated with partial views, although this is not compulsory.  
  
**6.** Child action methods are different from NonAction methods, in that NonAction methods cannot be invoked using Action() or RenderAction() helpers. We discussed NonAction methods in [Part 70](http://csharp-video-tutorials.blogspot.com/2013/08/part-70-authorize-and-allowanonymous.html) of [ASP.NET MVC tutorial](http://www.youtube.com/playlist?list=PL6n9fhu94yhVm6S8I2xd6nYz2ZORd7X2v) series.  
  
**7.** Using child action methods, it is possible to cache portions of a view. This is the main advantage of child action methods. We will cover this when we discuss [OutputCache] attribute.