In this video, we will discuss **action selectors** in asp.net mvc

Actions are public methods in an mvc controller that responds to an URL request. You can **control or influence which action method gets invoked using action selectors**in mvc. Action selectors are attributes that can be applied to an action method in a controller.   
  
   
  
**ActionName selector:** This action selector is used when you want to invoke an action method with a different name, than what is already given to the action method.   
  
For example, the following URL request would invoke Index() action method in HomeController  
**/Home/Index**  
  
public class HomeController : Controller  
{  
    public string Index()  
    {  
        return "Index action method invoked";  
    }  
}  
  
If you want to invoke **Index**() action method, with the following URL  
**/Home/List**  
  
Then decorate the action method with ActionName attribute as shown below.  
public class HomeController : Controller  
{  
    [ActionName("List")]  
    public string Index()  
    {  
        return "Index action method invoked";  
    }  
}  
  
Now, if you navigate to **/Home/Index**, you will get an error - The resource cannot be found.   
  
At the moment, the **Index**() action method is returning a string, but if it returns a view, should the view be named - Index or List.?  
[ActionName("List")]  
public ActionResult Index()  
{  
    return View();  
}  
  
List should be the view name. If for some reason, you want to use **"Index"** as the view name, then modify the controller action method as shown below.  
[ActionName("List")]  
public ActionResult Index()  
{  
    return View("Index");  
}  
  
**AcceptVerbs selector:** Use this selector, when you want to control, the invocation of an action method based on the request type. In the example below, the "Edit" method that is decorated with GET acceptverb responds to the GET request, where as the other "Edit" method responds to POST request. The default is GET. So, if you don't decorate an action method with any accept verb, then, by default, the method responds to GET request.  
public class HomeController : Controller  
{  
    [AcceptVerbs(HttpVerbs.Get)]  
    public ActionResult Edit(int id)  
    {  
        Employee employee = GetEmployeeFromDB(id);  
        return View(employee);  
    }  
      
    [AcceptVerbs(HttpVerbs.Post)]  
    public ActionResult Save(Employee employee)  
    {  
        if (ModelState.IsValid)  
        {  
            // Save employee to the database  
            return RedirectToAction("Index");  
        }  
        return View(employee);  
    }  
}  
  
**HttpGet**and **HttpPost** attributes can be used as shown below. This is an alternative to using AcceptVerbs attribute.  
public class HomeController : Controller  
{  
    [HttpGet]  
    public ActionResult Edit(int id)  
    {  
        Employee employee = GetEmployeeFromDB(id);  
        return View(employee);  
    }  
      
    [HttpPost]  
    public ActionResult Save(Employee employee)  
    {  
        if (ModelState.IsValid)  
        {  
            // Save employee to the database  
            return RedirectToAction("Index");  
        }  
        return View(employee);  
    }  
}