public class MyViewModel

{

public string Prop1 { get; set; }

public string Prop2 { get; set; }

}

// Index.cshtml - bound to MyViewModel

// This will render a text box that will read "Property 1". It will also render a hidden field

// containing "Property 2" but will not be displayed on the page. When you click the submit button,

// it will post back to the PostTest action on the Home controller. Notice that Prop3 wasn't used

// so it won't be included in the postback

@model MyProject.Models.MyViewModel

<div>

@using(Html.BeginForm("PostTest", "Home", FormMethod.Post))

{

@Html.TextBoxFor(m => m.Prop1);

@Html.HiddenFor(m => m.Prop2);

<button type="submit">GO!</button>

}

</div>

public class HomeController : Controller

{

public ActionResult Index()

{

// your index view is bound to MyViewModel so we'll hydrate that and send it to the view for rendering

var viewModel = new MyViewModel();

viewModel.Prop1 = "Property 1";

viewModel.Prop2 = "Property 2";

viewModel.Prop3 = "Property 3";

return View(vm);

}

public ActionResult PostTest(MyViewModel vm) // mvc will attempt to deserialize your form data into this object

{

var testProp1 = vm.Prop1; // "Property 1"

var testProp2 = vm.Prop2; // "Property 2"

var testProp3 = vm.Prop3; // null

}

}