In this video we will discuss, preventing unintended updates in mvc. Please [watch Part 19](http://csharp-video-tutorials.blogspot.com/2013/05/part-19-unintended-updates-in-mvc.html), before proceeding.

Modify **"Edit"** controller action method that is decorated with [HttpPost] attribute as shown below. This method is present in**"EmployeeController.cs"** file.  
[HttpPost]  
[ActionName("Edit")]  
public ActionResult Edit\_Post(int id)  
{  
    EmployeeBusinessLayer employeeBusinessLayer = new EmployeeBusinessLayer();  
  
    Employee employee = employeeBusinessLayer.Employees.Single(x => x.ID == id);  
    UpdateModel(employee, new string[] { "ID", "Gender", "City", "DateOfBirth" });  
              
    if (ModelState.IsValid)  
    {  
        employeeBusinessLayer.SaveEmployee(employee);  
  
        return RedirectToAction("Index");  
    }  
  
    return View(employee);  
}  
  
**Please note:**  
**1.** The name of the method is changed from **"Edit"** to **"Edit\_Post"**  
**2.** The method is decorated with [ActionName("Edit")] and [HttpPost] attributes. This indicates that, this method is going to respond to **"Edit"** action, when the form is posted to the server.  
**3.** The **"id"** of the employee that is being edited, is passed as a parameter to this method.

**4.** Using the **"id"** parameter we load the employee details(Id, Name, Gender, City & DateOfBirth) from the database.   
Employee employee = employeeBusinessLayer.Employees.Single(x => x.ID == id);  
**5.** We then call **UpdateModel**() function. This should automatically update **"Employee"**object with data from the posted form. We are also passing a string array as the second parameter. This parameter specifies the list of model properties to update. This is also called as **include list** or **white list**. Notice that, we did not include **"Name"** property in the list. This means, even if the posted form data contains value for **"Name"** property, it will not be used to update the **"Name"** property of the **"Employee"** object.  
UpdateModel(employee, new string[] { "ID", "Gender", "City", "DateOfBirth" });  
  
So, if we were to generate a post request using fiddler as we did in the previous session, **"Name"** property of the **"Employee"** object will not be updated.  
  
Alternatively, to exclude properties from binding, we can specify the exclude list as shown below.   
[HttpPost]  
[ActionName("Edit")]  
public ActionResult Edit\_Post(int id)  
{  
    EmployeeBusinessLayer employeeBusinessLayer = new EmployeeBusinessLayer();  
  
    Employee employee = employeeBusinessLayer.Employees.Single(x => x.ID == id);  
    UpdateModel(employee, null, null, new string[] { "Name" });  
  
    if (ModelState.IsValid)  
    {  
        employeeBusinessLayer.SaveEmployee(employee);  
  
        return RedirectToAction("Index");  
    }  
  
    return View(employee);  
}  
  
Notice that we are using a different overloaded version of **UpdateModel**() function. We are passing **"NULL"** for **"prefix"** and **"includeProperties"** parameters.  
UpdateModel<TModel>(TModel model, string prefix, string[] includeProperties, string[] excludeProperties)