There are several ways available to pass data **from a controller to a view** in an mvc application.   
**1.** ViewBag or ViewData  
**2.** Dynamic type  
**3.** Strongly typed view

**The following are the advantages of using strongly typed views**. We get  
**1.** Intellisense and  
**2.** Compile-time error checking  
  
With ViewBag and Dynamic type, we don't have these advantages.

**Using ViewBag:**  
Notice that, the employee object is stored in ViewBag.  
In HomeController.cs  
public ActionResult Details(int id)  
{  
    SampleDBContext db = new SampleDBContext();  
    Employee employee = db.Employees.Single(x => x.Id == id);  
    ViewBag.EmployeeData = employee;  
    return View();  
}  
  
We want to display **employee FullName** and **Gender**. Notice that, as we are typing **FullName** and **Gender** properties, we don't get **intellisense**. Also, if we mis-spell **FullName** or **Gender** properties, we will not get any compilation errors. We will come to know about these errors only at runtime.  
**In Details.cshtml View**  
<div class="display-label">  
    @Html.DisplayName("FullName")  
</div>  
<div class="display-field">  
    @ViewBag.EmployeeData.FullName  
</div>  
<div class="display-label">  
    @Html.DisplayName("Gender")  
</div>  
<div class="display-field">  
    @ViewBag.EmployeeData.Gender  
</div>  
  
**Using Dynamic Type:**  
In HomeController.cs  
public ActionResult Details(int id)  
{  
    SampleDBContext db = new SampleDBContext();  
    Employee employee = db.Employees.Single(x => x.Id == id);  
    return View(employee);  
}  
  
**In Details.cshtml View**  
@model dynamic  
<div class="display-label">  
    @Html.DisplayName("FullName")  
</div>  
<div class="display-field">  
    @Model.FullName  
</div>  
<div class="display-label">  
    @Html.DisplayName("Gender")  
</div>  
<div class="display-field">  
    @Model.Gender  
</div>  
  
With dynamic type also, we don't get intellisense and compile-time error checking.  
  
**Using Strongly Typed View:** No change is required in the controller action method. Make the following change to **Details.cshtml** view. Notice that the view is strongly typed against Employee model class. We get **intellisense** and if we **mis-spell** a property name, we get to know about it at compile time.  
@model MVCDemo.Models.Employee  
<div class="display-label">  
    @Html.DisplayName("FullName")  
</div>  
<div class="display-field">  
    @Model.FullName  
</div>  
<div class="display-label">  
    @Html.DisplayName("Gender")  
</div>  
<div class="display-field">  
    @Model.Gender  
</div>  
  
**Please Note:** We discussed enabling compile time error checking in views in [Part 50](http://csharp-video-tutorials.blogspot.com/2013/07/part-50-detect-errors-in-views-at.html) of the [MVC tutorial](http://csharp-video-tutorials.blogspot.com/p/aspnet-mvc-tutorial-for-beginners.html).