**ViewBag in ASP.NET Core MVC**

ViewBag is a wrapper around ViewData. With ViewData we use string keys to store and retrieve data. With **ViewBag**we use dynamic properties instead of string keys.   
  
  
**Using ViewBag to pass Data from a Controller to a View**  
  
We want to pass Employee model data and a Title for the view page from the Details()action method of the HomeController to the Details.cshtml view. So modify the Details() action method in the HomeController as shown below.   
  
Notice, we are using dynamic properties (Title and Properties) on the **ViewBag**instead of string keys.

public ViewResult Details()  
{  
    Employee model = \_employeeRepository.GetEmployee(1);  
  
    // To store the page title and empoyee model object in the   
    // ViewBag we are using dynamic properties PageTitle and Employee  
    ViewBag.PageTitle = "Employee Details";  
    ViewBag.Employee = model;  
  
    return View();  
}

**Accessing ViewBag in a View**  
  
To access the **ViewBag**data passed by the Details() action method of the HomeController, modify the code in Details.cshtml view file as shown below. Notice, we are using the same dynamic properties PageTitle and Employee to access the **ViewBag**data.

<html>  
<head>  
    <title></title>  
</head>  
<body>  
    <h3>@ViewBag.PageTitle</h3>  
  
    <div>  
        Name : @ViewBag.Employee.Name  
    </div>  
    <div>  
        Email : @ViewBag.Employee.Email  
    </div>  
    <div>  
        Department : @ViewBag.Employee.Department  
    </div>  
</body>  
</html>

**ViewData v/s ViewBag**

* Both **ViewData**and **ViewBag**can be used to pass data from a Controller to a View
* **ViewBag**is a wrapper around **ViewData**
* Both of them create a loosely typed view
* With **ViewData**we use string keys to store and retrieve data from the **ViewData**dictionary
* With **ViewBag**we use dynamic properties to store and retrieve data
* Both **ViewData**keys and **ViewBag**dynamic properties are resolved dynamically at runtime.
* Both **ViewData**and **ViewBag**does not provide compile-time type checking and as a result we do not get intellisense.
* Since we do not have intellisense, the speed at which we write code is reduced and the chances of mis-spelling and making typographical errors are also high.
* We will only come to know about these errors at run time.
* For this reason we usually do not use **ViewData**or **ViewBag**.
* The preferred approach to pass data from a controller to a view is by using a strongly typed model object.
* Using a strongly typed model object creates a **strongly typed view**.