**ViewData und ViewBag in ASP.NET Core MVC**

**ViewData**

* Typ: ViewDataDictionary
* Zugriff per Schlüssel: ViewData["Name"]
* Keine IntelliSense-Unterstützung, keine Typprüfung

ViewData["Nachricht"] = "Willkommen auf unserer Seite!";

**ViewBag**

* Typ: dynamic
* Zugriff per Eigenschaft: ViewBag.Nachricht
* Intern basiert ViewBag ebenfalls auf ViewData

ViewBag.Nachricht = "Willkommen auf unserer Seite!";

|  |  |  |
| --- | --- | --- |
| Merkmal | ViewData | ViewBag |
| Typ | ViewDataDictionary | dynamic |
| Zugriff | Über Schlüssel (string) | Über dynamische Property |
| Typprüfung | Nein | Nein |
| IntelliSense | Nein | Eingeschränkt |
| Basisobjekt | ViewDataDictionary | ViewDataDictionary |

**Übung 1: Begrüßung über ViewBag**

Erstellen Sie im HomeController eine neue Action Begruessung, die folgende Daten übergibt:

* ViewBag.Name = Ihr eigener Name
* ViewBag.Zeit = aktuelle Uhrzeit (DateTime.Now.ToShortTimeString())

In der View soll ein Text erscheinen wie:  
„Hallo Max, es ist jetzt 09:30 Uhr.“

**Übung 2: Produktinfos über ViewData**

Erstellen Sie eine Action Produkt, die folgende Daten per ViewData übergibt:

* Produktname: "USB-Stick"
* Preis: 12.99
* Lagerbestand: true (ob auf Lager)

In der View soll stehen:  
„Produkt: USB-Stick – Preis: 12,99 € – Verfügbar: Ja“

**Merke**

* ViewData["Schlüssel"] ist wie ein Dictionary-Eintrag.
* ViewBag.Name ist wie eine dynamische Eigenschaft.
* Beides dient zur Datenübergabe vom Controller zur View – für kleine Datenmengen geeignet.
* **Für größere Datenmengen und überhaupt aus Gründen eines saubereren Programmier- und Struktur-Stills ViewModel verwenden**