

MVC 5

(Model-View-Controller)

State Management

Uvod u upravljanje stanjima (State Management)

State management – to je proces spremanja informacija između dva zahtjeva (request-a) i postback-a prema serveru

Bez State Management-a

Molimo, unesite svoje login podatke:

Ime:

Vedran

Prezime:

Zdešić

POŠALJI

Izvinite! Ne znamo tko ste Vi zapravo!



Sa State Management-om

Molimo, unesite svoje login podatke:

Ime:

Vedran

Prezime :

Zdešić

POŠALJI

Bok Mirko !



Opcije kod State Management-a (I)

Podaci koji opisuju stanje mogu biti spremljeni na dva mjesta:

Na strani servera:

- Application state
- Session state
- User profile
- Caching
- TempData
- ... ili baza podataka

Na strani klijenta:

- Cookies
- Query String

Opcije kod State Management-a (II)

NAPOMENA: Postoje objekti **ViewBag**, **ViewData**, **TempData**, ali oni se ne mogu koristiti za trajno spremanje informacija o stanju.

Oni se uništavaju kod svakog novog Request-a (zahtjeva za stranicom). Koriste se za spremanje informacija u komunikaciji između controllera, viewa i parcijalnog viewa kod renderiranja pojedine HTML stranice.

ViewBag, ViewData, TempData

	ViewData	ViewBag	TempData
Tip	Key-Value Dictionary Collection	Dynamic Object ViewBag je wrapper (omotač) oko ViewData	Key-Value Dictionary Collection
Opis	ViewData je Dictionary objekt i svojstvo ControllerBase klase	ViewBag je Dynamic svojstvo ControllerBase klase	TempData je Dictionary objekt i svojstvo ControllerBase klase
Brzina	ViewData je brži nego ViewBag kod korištenja	ViewBag je sporiji nego ViewData kod korištenja	
Verzije	ViewData je uveden u MVC 1.0 verziji dostupan je u MVC 1.0 i višim verzijama	ViewBag je uveden u MVC 3.0 i dostupan je u MVC 3.0 i višim verzijama	TempData je uveden u MVC1.0 i dostupan je u MVC 1.0 i višim verzijama
Korištenje	Prosljeđuje parametar od Controllera do Viewa	Prosljeđuje parametar od Controllera do Viewa	Prosljeđuje parametar od Controllera do Controllera. Koristan za prosljeđivanje error i validacijskih poruka
Konverzija/Cast	Konverzija tipa (Cast/Type Conversion) je obavezna kod enumeriranja	Zato što je ViewBag Dynamic tipa, nije potrebna konverzija tipa kod enumeriranja	Konverzija tipa (Cast/Type Conversion) je obavezna kod enumeriranja
Redirekcija	Postaje null kod redirekcije	Postaje null kod redirekcije	TempData se koristi za prosljeđivanje podataka između dva uzastopna requesta. Podaci su sačuvani kod redirekcije.
Vijek trajanja	Postoji samo tijekom trenutnog requesta	Postoji samo tijekom trenutnog requesta	Postoji tijekom redirekcije

Opcije za spremanje stanja sesije (Session State)

- Stanje sesije može biti konfigurirano u različitim modovima:
 - **InProc** — podaci se čuvaju u memoriji (default)
 - **StateServer** — podaci se čuvaju u vanjskom State Server-u
 - **SQLServer** — podaci se čuvaju u SQL Server bazi
 - **Custom** — prilagođena implementacija
 - **Off** — podaci se ne čuvaju
- Osim za **InProc**, svi drugi modovi zahtjevaju od podataka koji se spremaju da su serializabilni, da ih je moguće serijalizirati
- **InProc** mod nije dobar mod u slučaju web farmi

Korištenje SQL Server za spremanje podataka sesije

- Podatke o sesiji možemo spremiti u SQL Server bazu podataka
- Baza će biti nazivaa ASPState
 - Kako bi kreirali bazu pokrenite **aspnet_regsql.exe -ssadd**
- Za korištenje baze, konfigurirajte web site ovako:

```
<configuration>  
<system.web>  
  <sessionState mode="SQLServer"  
    sqlConnectionString="Integrated Security=SSPI;  
    data source=SampleSqlServer;" />  
</system.web>  
</configuration>
```

Global.asax datoteka

- Global.asax datoteka poznata i kao ASP.NET aplikacijska datoteka sadrži kod koji se izvršava kao odgovor na aplikacijske događaje (evente)
- Metode koje ovdje možemo naći su:
 - Application_Start
 - Application_End
 - Application_BeginRequest
 - Application_EndRequest
 - Session_Start
 - Session_End
 - Application_Error
 - Application_AuthenticateRequest
 - ...

Debugging

- Neke općenite napomene tijekom debugiranja global.asax metoda:
 - Session_End event je podržan jedino u modu InProc
 - Session objekt se uništava nakon definiranog timeouta (1 minuta u donjem primjeru) unutar web.config datoteke:

```
<system.web>  
...  
  <sessionState mode="InProc" timeout="1" />  
</system.web>
```
 - Session_End se neće izvršiti ako ne spremimo ništa unutar trenutne sesije
 - Session_End se neće sam od sebe pokrenuti ako zatvorimo preglednik
 - Ne možete napraviti Response.Redirect unutar Session_End
 - Kako bi upravljali bolje stanjem sesije koristite metode poput:
 - Session.Abandon();
 - Session.Clear();
 - Application_BeginRequest se može izvršiti više od jednom za svaki request jer aplikacija dohvaća dodatne resurse u odvojenim requestima
 - Potrebno je pripaziti na IIS Express proces. Ponekada ga treba ugasiti kako bi bili sigurni da sve kreće od nultog stanja