

Manejo de Estado

Introducción

Como las aplicaciones web no guardan un estado entre las peticiones, es decir que cada vez que el usuario realiza una petición la página se carga completamente de nuevo existen distintas formas de persistir datos entre peticiones de usuarios:

- objetos de aplicación

- campos ocultos

- objetos de sesión

- objetos de cache

- cookies, etc.

La diferencia entre las mismas está dada por la visibilidad, manejo de datos, y tiempo de persistencia. [+ info](#)

Cómo elegir cual usar?

Necesitamos saber:

- Quien necesita los datos?
- Por cuanto tiempo se necesita que se persistan?
- Que tan grande son los datos?

Objetos contenedores

Método de persistencia	Quien necesita?	Cuanto tiempo?	Cantidad de datos?
Application	Todos	Hasta que la siguiente aplicación comience.	Casi cualquier tamaño.
Cookie	Usuario	El tiempo que se desee si no se borra.	Poco, datos simples.
Form Post	Usuario	Para la próxima petición.	Cualquier tamaño. Los datos viajan con la pagina.
QueryString	Usuario o grupo de usuarios.	Para la próxima petición.	Poco, datos simples.
Session	Usuario	Mientras el usuario este activo, más el timeout.	Casi cualquier tamaño

Método de persistencia	Quien necesita?	Cuanto tiempo?	Cantidad de datos?
Cache	Todos los usuarios.	Por el tiempo que se necesite.	Todo tipo.
Context	Usuario	Solo en la peticion.	Gran tamaño.
ViewState	Usuario	Un formulario Web	Minimo, estos datos son enviados y recibidos con cada página.
Config file	Todos	Hasta que el archivo de configuración sea actualizado.	Puede contener gran cantidad de datos.

QueryString

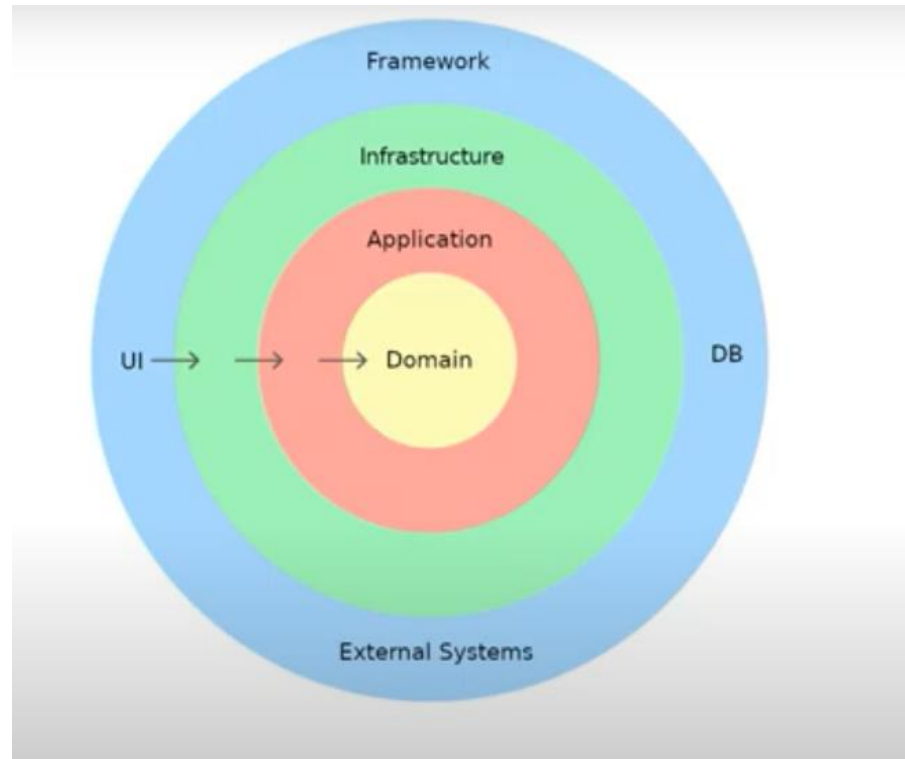
Los datos guardados en el objeto QueryString son utilizados por un solo usuario.

Es un buen modo de enviar información, en este caso se envía al método ver el parámetro cédula con el valor 123123.

`Localhost/profesor/ver?cedula=123123`

Arquitectura

Dado que Net Core es una Arquitectura Limpia, que es independiente del contexto, para poder usar Session tengo que inyectarlo.



Session

Sus datos pertenecen a un usuario en particular.

Viven mientras el usuario continúe haciendo peticiones, más un un timeout (por defecto 20 min.)

Pueden guardar grandes o pequeñas cantidades de datos.

Su uso debe de tratar de restringirse si se desea un buen rendimiento con muchos usuarios.

Configurar el estado de sesión

- En el archivo program.cs
 - Hay que habilitar el servicio

```
// builder.WebApplication.CreateBuilder(args);  
  
// Add services to the container.  
builder.Services.AddControllersWithViews();  
builder.Services.AddSession();
```

- Agregar a la aplicación que va a usar el servicio

```
app.UseRouting();  
app.UseSession();
```

- Luego se utiliza en el controlador o en las vistas. En el ejemplo guarda en dos variables de sesión email y rol

```
//  
HttpContext.Session.SetString("email", email);  
HttpContext.Session.SetString("rol", "admin");  
//
```

Establecer y obtener valores de Session

Se tiene acceso al estado de sesión desde la clase PageModel de Razor Pages o desde la clase Controller de MVC con HttpContext.Session. Esta propiedad es una implementación de ISession. [+ info](#)

La implementación ISession proporciona varios métodos de extensión para establecer y recuperar valores de cadena y enteros. Los nuevos métodos de extensión se encuentran en el espacio de nombres Microsoft.AspNetCore.Http.

Métodos de extensión ISession:

- Get(ISession, String)
- GetInt32(ISession, String)
- GetString(ISession, String)
- SetInt32(ISession, String, Int32)
- SetString(ISession, String, String)

Establecer y obtener valores de Session

```
// Guarda valores
HttpContext.Session.SetString("LogueadoNombre", "Juan");
HttpContext.Session.SetInt32("Logueado", 1);
```

```
// Recupera valores
if (HttpContext.Session.GetString("LogueadoNombre") == null)
{
    return RedirectToAction("index");
}
return View(new Mensual());
```

Establecer y obtener valores de Session

```
@if (Context.Session.GetString("LogeadoNombre") == null)
{
    <li class="nav-item"><a class="nav-link text-dark" href="/usuario/login">Login</a></li>
}
else
{
    <li class="nav-item"><a class="nav-link text-dark" href="/usuario/Logout">Salir</a></li>
}
```

Para utilizarlo en la vista

```
@using Microsoft.AspNetCore.Http
```