

Управление состоянием и безопасность в ASP.Net

УПРАВЛЕНИЕ СОСТОЯНИЕМ

Варианты

- На клиенте
 - в запросе
 - в странице (скрытые поля)
 - в cookies
- На сервере
 - сессия
- [ASP.NET State Management Overview](#)

В запросе

- Формируем строку вида
<http://localhost/Url?State=1>
- Все ссылки содержат параметр, значение которого меняется (или передается неизменным)
- Работает только в Get-запросах

В странице

- Добавление скрытого поля на страницу
 - `<input id="State" name="State" type="hidden" value="0" />`
- В MVC
 - поле в модель
 - на страницу `Html.HiddenFor`
- Передается только при Post-запросах

В Cookie

- Чтение
 - `Request.Cookies[StateCookieName]`
- Добавление / обновление
 - `Response.AppendCookie(stateCookie)`
 - `Response.SetCookie(stateCookie)`

Сессия на сервере

- Доступ
 - Session[< имя_переменной >]
- Поддержка
 - куки
 - URL

Хранение сессии

- Время жизни
- Место хранения
 - в памяти
 - в отдельной службе
 - в базе
 - свое

УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ

Задачи

- Аутентификация
- Авторизация
- ~~Защита данных~~
 - ~~— от изменений~~
 - ~~— от доступа~~

Идентификация пользователя

- Интерфейсы

- IIdentity

- IPrincipal

```
public interface IIdentity
{
    string Name { get; }
    string AuthenticationType { get; }
    bool IsAuthenticated { get; }
}

public interface IPrincipal
{
    bool IsInRole(string role);
    IIdentity Identity { get; }
}
```

- Доступ

- System.Threading.Thread.CurrentPrincipal

- User.Identity (контроллер MVC)

Аутентификация

- На уровне протокола
 - Anonymous
 - Basic
 - Digest
 - Windows
- На уровне приложения
 - Form

Form-аутентификация

- Класс FormsAuthentication
 - SetAuthCookie
 - RedirectFromLoginPage
 - RedirectToLoginPage
 - RenewTicketIfOld
 - SignOut

Управление аутентификацией

```
<authentication mode="Forms">  
  <forms loginUrl="/Account/Logon" />  
</authentication>
```

```
<authorization>  
  <deny users="?" />  
</authorization>
```

```
[Authorize]  
public ActionResult Index()  
{  
    return View();  
}
```

Membership провайдеры

- Стандартные
 - System.Web.Security.ActiveDirectoryMembershipProvider
 - System.Web.Security.SqlMembershipProvider
- Собственный
 - наследник от System.Web.Security.MembershipProvider

```
<membership>
  <providers>
    <add name="AspNetSqlMembershipProvider"
        type="System.Web.Security.SqlMembershipProvider, System.Web"
        connectionStringName="ApplicationServices"
        applicationName="/" />
  </providers>
</membership>
```

Role провайдеры

- Стандартные
 - System.Web.Security.AuthorizationStoreRoleProvider
 - System.Web.Security.SqlRoleProvider
 - System.Web.Security.WindowsTokenRoleProvider
- Собственный
 - наследник от System.Web.Security.RoleProvider

```
<roleManager enabled="true">
  <providers>
    <clear/>
    <add name="AspNetSqlRoleProvider"
        type="System.Web.Security.SqlRoleProvider"
        connectionStringName="ApplicationServices"
        applicationName="/" />
    <add name="AspNetWindowsTokenRoleProvider"
        type="System.Web.Security.WindowsTokenRoleProvider"
        applicationName="/" />
  </providers>
</roleManager>
```


Использование

```
if (Membership.ValidateUser(userAndPassword.User, userAndPassword.Password))
{
    FormsAuthentication.SetAuthCookie(userAndPassword.User, false);
    return Redirect(ReturnUrl);
}
```