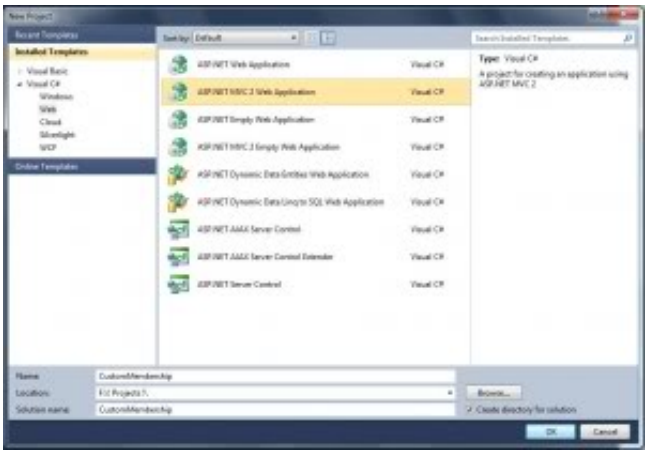
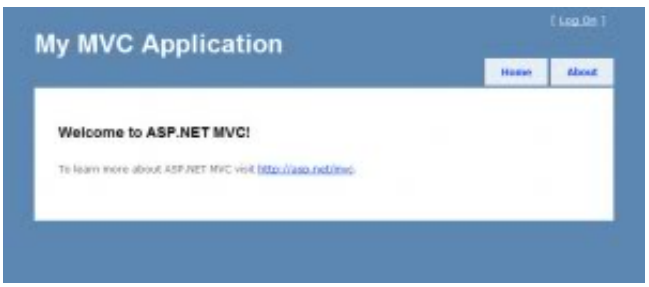
Реализация собственного Membership Provider (часть 1)

Недавно столкнулся с использованием Membership Provider. Многие руководства описывают исполозование встроенного в студию Web Site Administration Tool который работает с отдельной базой данных. Мне же нужно было, чтобы использовалась для этого собственную бд. Поискав и интернете, нашел статью довольно доступно это описывающие, правда она на английском и для asp.net MVC 2. В этой статье привожу ее перевод и адаптацию под MVC 3.

В этом руководстве я покажу как реализовать пользовательский Membership Provider в ASP.NET MVC 2 с помощью Microsoft Visual Web Developer 2010 Express. Это руководство будет состоять из нескольких частей (пока не знаю из скольки) и я постараюсь объяснить собственную реализацию Membership Provider, ролей и провайдеров профилей (membership, role and profile providers). Я буду использовать для этого собственную базу данных, Entity Framework, а также дополнительный класс, реализующий уровень между Membership Provider и Entity Framework.
Давайте начнем, создав новый проект и назвав его CustomMembership.

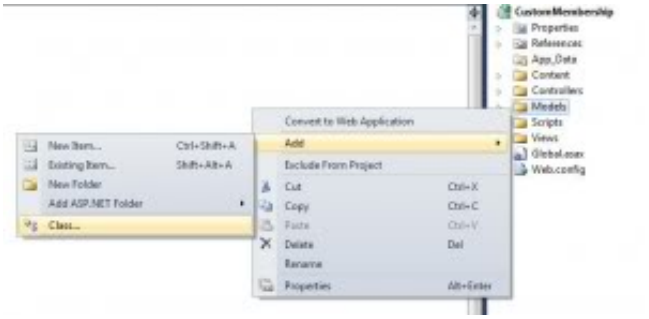


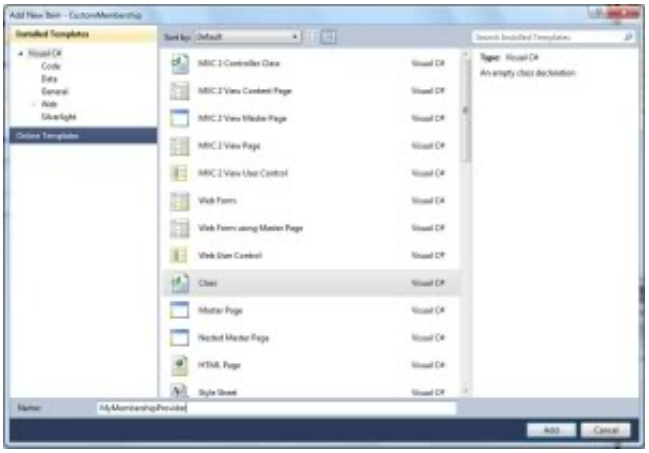
Т.к мы используем уже готовый шаблон ASP.NET MVC 2 Web Application, то мы можем сразу его запустить и увидеть:



Созданный проект уже настроен на использование ASPNETDB базы данных, расположенную в папке App_Data, и стандартного Membership Provider. Мы же хотим использовать собственную базу данных для хранения сведений о пользователях. И для этого мы должны реализовать собственных Membership Provider. Давайте начнем!
Сперва, создадим новый класс в папке Models и назовем его MyMembershipProvider.

Затем, создадим новый класс в папке Models и назовем его MyMembershipProvider.



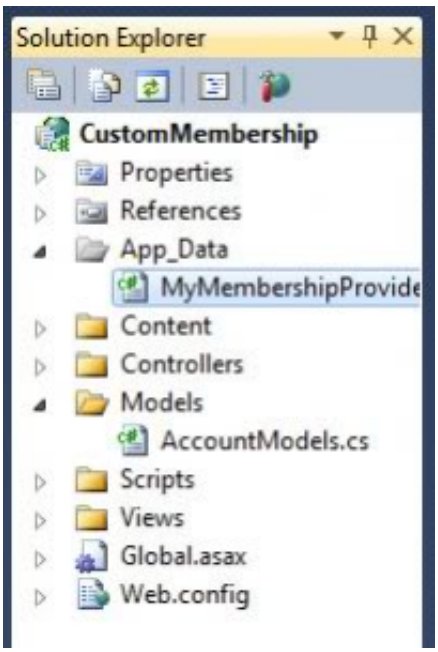


Наследуем наш класс от MembershipProvider (из пространства имен System.Web.Security).

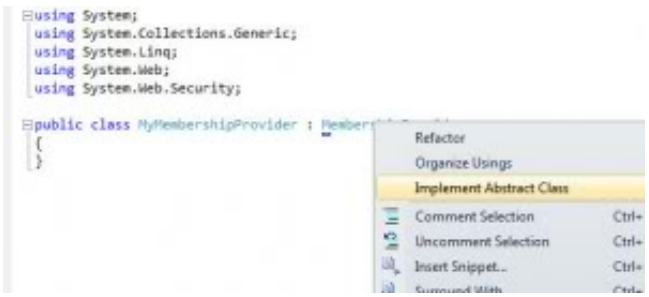
```
01.
using System;
02.
using System.Collections.Generic;
03.
using System.Linq;
04.
using System.Web;
05.
using System.Web.Security;
06.

07.
public class MyMembershipProvider : MembershipProvider
08.
{
```

```
09.
}
```

Переместим наш файл с классом в папку App_Data, просто перетащив ее из папки Models.



Снова откроем этот файл с классом и реализуем все абстрактные члены класса MembershipProvider. Для этого щелкаем правной кнопкой мыши на названии класса **MembershipProvider** и выбираем пункт **Implement Abstract Class**.



Это создаст переопределяемые методы для нашего класса.

Мы только что создали наш собственный Membership Provider. Хоть он и не реализует еще никакой функциональности, давайте продолжим и настроем наш сайт, на его использование.

Для этого открываем файл Web.config, в корне сайта, и изменяем значение узла membership на использование нашего собственного провайдера.

```
01.
<membership defaultProvider="CustomMembershipProvider">
02.
<providers>
03.
<clear />
04.
<add name="CustomMembershipProvider"
type="MyMembershipProvider"
05.
connectionStringName="ApplicationServices"
06.
enablePasswordRetrieval="false"
07.
enablePasswordReset="true"
08.
requiresQuestionAndAnswer="false"
09.
requiresUniqueEmail="false"
10.
maxInvalidPasswordAttempts="5"
11.
minRequiredPasswordLength="6"
12.
minRequiredNonalphanumericCharacters="0"
13.
```

```
passwordAttemptWindow = "10"
applicationName = "/" />
</providers>
</membership>
```

Если мы запустим наше приложение сейчас и попытаемся войти, используя ссылку вверху страницы, мы получим следующую ошибку:

Которая нам всего лишь говорит, что мы не реализовали еще ни одного метода. Давайте изменим метод ValidateUser, чтобы он возвращал true и снова попробуем зайти на сайт.

```
public override bool ValidateUser(string username, string
password)
```

```
password)
2.
{
3.
return username == password;
4.
}
```

Теперь у нас есть возможность заходить на сайт так долго, как долго у нас будут совпадать логин и пароль. Давайте посмотрим на это.

На этом все. В следующей части мы создадим собственную базу данных и класс UserRepository.

Продолжение см. Реализация собственного Membership Provider (часть 2)

www.gotdotnet.ru

http://www.gotdotnet.ru/blogs/zhidkov/9955/

http://goo.gl/XlQ4T