

УДК 681.324:621.325

В.О. Мартовицький, Л.Л. Колодочкин

Харківський університет Повітряних Сил імені Івана Кожедуба, Харків

## СТВОРЕННЯ КРОС-ПЛАТФОРМНОЇ СИСТЕМИ ЗАХИСТУ WEB-СЕРВІСІВ І ДОДАТКІВ НА ОСНОВІ XML-ФАЙЛІВ ДЛЯ ТЕХНОЛОГІЇ ASP.NET

Стаття присвячена проблемі створення крос-платформеної системи захисту Web-сервісів і додатків створених на базі ASP.NET. Розглянута архітектура системи безпеки користувача, яка побудована на основі класів Membership API і Roles API.

**Ключові слова:** Xml-Файл, .NET Framework, Web-технології, база даних.

### Вступ

Microsoft .NET Framework – це платформа для створення, розгортання і запуску Web-сервісів та додатків. Вона надає високопродуктивну, засновану на стандартах, багато яzikове середовище, яке дозволяє інтегрувати існуючі застосування з додатками і сервісами наступного покоління, а також вирішувати завдання розгортання і використання інтернет-додатків. .NET Framework складається з трьох основних частин - загальномовного середовища виконання (common language runtime), ієрархічної безлічі уніфікованих бібліотек класів і компонентну версію ASP, звану ASP.NET [1].

У кінці 2014 року випущена наступна версія Visual Studio 2015, спеціальна безкоштовна редакція Visual Studio Community 2013, Visual Studio 2013 update 4, відкривається початковий код .NET Server Core, випущена наступна версія .NET 2015, оновлені інструменти кросс-платформеної розробки на базі Apache Cordova, в Visual Studio з'явився емулятор ОС Android. Це дійсно новий етап в розвитку інструментів розробки Microsoft [2]. У зв'язку з цим стає актуальною розробка призначених для користувача системи безпеки на основі Membership API і Roles API для Web-сервісів і додатків. Модель профілів ідеально стикується з Web-сторінками ASP.NET. Проте вона не занадто гнучка в плані конфігурації. Рішення про створення власного постачальника профілів може бути прийняте з кількох причин:

- необхідно зберігати інформацію профілю в джерелі даних, відмінному від SQL Server, такому як база даних Oracle, MySQL або XML-файлах;

- необхідно забезпечити доступність до даних профілів іншим застосуванням. Розбір інформації в полях PropertyValuesString і PropertyValuesBinary стомливий, багатий наслідками помилками і недостатньо гнучкий. Якщо треба використовувати цю інформацію в інших запитах або додатках, краще зберегти інформацію профілів в таблиці бази даних, розбитої на окремі поля;

- треба реалізувати додаткову логіку при збереженні або витяганні даних профілю. Наприклад,

можна застосувати перевірку достовірності, кешування, протоколювання, шифрування і стискування. (В деяких випадках можна отримати усі ці кошти, просто розширивши клас ProfileBase, який інкапсулює установки профілю, замість того, щоб створювати повністю новий ProfileProvider.)

### Виклад основного матеріалу

ASP.NET включає інтерфейси Membership API і Roles API, які надають готову платформу управління користувачами і ролями. Цю платформу можна розширити за допомогою механізму постачальників, які реалізують доступ до сховища даних, що лежить в основі [3]. Замінити реалізацію за умовчанням, що працює з SQL Server, можна за рахунок побудови власних постачальників членства і ролей. Це надає можливість заміни сховища, що лежить в основі, інформації про користувачів і ролі, не зачіпаючи Web-додатків. Ця модель може бути розширена за рахунок користувацьких постачальників.

Призначений для користувача постачальник завжди базується на нижчому рівні в багаторівневій моделі, що забезпечується інтерфейсами членства і ролей ASP.NET. Важливо знати, що всі інші засновані на постачальниках API-інтерфейси в ASP.NET структуровані аналогічним чином. Тому реалізація призначених для користувача постачальників для Profiles API або механізмів персоналізації ASP.NET дуже схожа. Служби членства і ролей не залежать один від одного. Тому постачальники членства і ролей мають різні базові класи; на додаток інформацію про членство і ролі можна зберігати в різних системах заднього плану. Хорошим прикладом може служити служба Roles Service з Windows-аутентифікацією. Реалізація призначених для користувача постачальників досить прямолінійна, але вимагає деякого часу, оскільки вимагається реалізувати безліч методів і властивостей. Наступний постачальник створений та призначений для користувача постачальник членства і ролей, що працює з XML-файлом і джерелом даних, що лежить в основі. Файли XML – не занадто хороше рішення для масштабованих застосувань, але може виявитися непоганою альтернативою, якщо розробляється про-

сте кросс-платформене застосування, яке повинне розміщуватися на сайті постачальника, що не має доступу до бази даних SQL Server.

Перед створенням призначеного для користувача постачальника слід продумати загальну структуру усього рішення. Мета полягає в тому, щоб зберегти функціональність, що лежить в основі, наскільки можливо простою. Це дозволить зосередитися на дійсній реалізації постачальників членства і ролей. У разі XML простий спосіб завантажувати і зберігати дані у файлах XML – це XML-серіалізація. Вона дає можливість зберігати повний граф об'єкту у файлі одним викликом функції і читати його також одним викликом іншої функції.

У момент створення екземпляра серіалізатора XmlSerializer йому необхідно повідомити тип, який потрібно серіалізувати і де серіалізувати.

Оскільки такі класи, як MembershipUser, не надають доступу до деякої інформації – наприклад, до пароля застосовувати їх з XML-серіалізацією безпосередньо не можна. XML-серіалізація вимагає, щоб усі властивості, що зберігаються, і члени були загальнодоступними. Таким чином, потрібно буде створити власні представлення користувачів і ролей у вигляді службових класів для сховища заднього плану. Ці класи ніколи не передаватимуться в додаток, який просто покладається на існуючі класи членства. (Буде передбачена деяка проста логіка відображення між внутрішнім представленням користувача і класом MembershipUser). На рис. 1 показано проектне рішення для створеного користувача постачальника. Класи SimpleUser і SimpleRole забезпечують можливість XML-серіалізації. Хоча це вимагає певної логіки відображення для підтримки MembershipUser, вся реалізація значно полегшується. UserStore і RoleStore – службові класи для інкапсуляції доступу до XML-файлу. Ці класи включають функції для завантаження і збереження XML-файлів разом з деякими базовими службовими функціями пошуку інформації в сховищі. Модель включає класи XmlMembershipProvider і XmlRoleProvider. Клас XmlMembershipProvider наслідує базову функціональність від MembershipProvider, тоді як XmlRoleProvider від RoleProvider. Обидва базові класи визначено в просторі імен System.Web.Security.

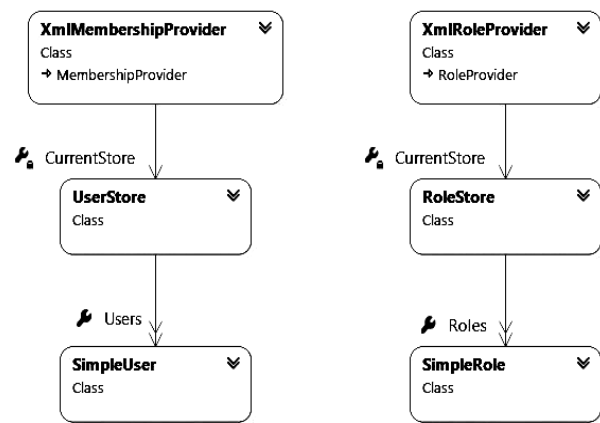


Рис. 1. UML-діаграма класів створеного користувача провайдера

## Висновки

Дана реалізація користувача провайдера для управління безпекою Web-сервісів і додатків дозволяє:

- зберігати інформацію профілю в джерелі даних, відмінному від SQL Server, такому як XML-файли;
- можливість розміщення простих Web-сервісів і додатків на базі ASP.NET на інших платформах відмінних від Windows;
- коли треба реалізувати додаткову логіку при збереженні або витяганні даних профілю. Наприклад, можна застосувати перевірку достовірності, хешування, протоколювання, шифрування і стискування.

## Список літератури

1. Годитейн С. Оптимізація приложених на платформі .NET / С. Годитейн, Д. Зурбалева, И. Флатов. – ДМК Пресс, 2014. – 522 с.
2. .NET Server Core, кросс-платформенная разработка, Visual Studio 2015 и другие анонсы Microsoft Connect(). [Електрон. ресурс]. – Режим доступу <http://www.itshop.ru/.NET-Server-Core-kross-platformennaya-razrabotka-Visual-Studio-2015-i-drugie-anonsy-Microsoft-Connect/19i37350>.
3. Фримен А. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов / А. Фримен. – Вильямс, 2013. – 688 с.

Надійшла до редколегії 14.04.2015

Рецензент: д-р техн. наук, проф. І.В. Рубан, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.

## СОЗДАНИЕ КРОСС-ПЛАТФОРМЕННОЙ СИСТЕМЫ ЗАЩИТЫ WEB-СЕРВИСОВ И ПРИЛОЖЕНИЙ НА ОСНОВЕ XML-ФАЙЛОВ ДЛЯ ТЕХНОЛОГИИ ASP.NET

В.А. Мартовицкий, Л.Л. Колодочкин

Статья посвящена проблеме создания кросс-платформенной системы защиты Web-сервисов и приложений созданных на базе ASP.NET. Рассмотрена архитектура системы безопасности пользователя, которая построена на основе Membership API и Roles API.

**Ключевые слова:** Xml-файл, .NET Framework, Web-технологии, база данных.

## CROSS-PROTECTION SYSTEM PLATFORMENYII WEB-SERVICES AND APPLICATIONS BASED ON XML-FILES FOR THE TECHNOLOGY ASP.NET

V.A. Martovytskyi, L.L. Kolodochkyn

The article is devoted to the problem of the cross-platformenyii protection system Web-services and applications created based on ASP.NET. Considered architecture security guide, which is based on OS-new Membership API and Roles API.

**Keywords:** Xml-File, .NET Framework, Web-technology database.