

# ASP.NETの ユーザー認証をマスターする

知ってるようで意外に知らない、  
認証ノウハウ大公開

大澤 文孝

OSAWA, Fumitaka

## Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

## Level



## Samples

## ASP.NETにおける ユーザー認証

業務アプリケーションでは、特定のユーザーしか利用させないようにするため、ユーザー認証を必要とする場面も多いでしょう。

そこで本稿では、ASP.NETにおけるユーザー認証について説明します。

## ASP.NETにおける ユーザー認証の基本

ASP.NETでは、Web.configファイルのauthentication要素とauthorization要素で、ユーザー認証方式とユーザー権限を決めます。

### ■ authentication要素

authentication要素は、認証方式を決めるもので、mode属性を使って、次のいずれかの認証方式を指定できます。デフォルトは、「Windows」です。

```
<authentication mode="Windows" />
```

### ①Windows

IISの認証をそのまま引き継ぐ。たとえば、IISで匿名アクセスをしている場合には、ASP.NETユーザー（NT AUTHORITY\NETWORKSERVICEユーザー）、それ以外の場合には、IISが実際に認証したユーザー。

### ②None

認証しない。

### ③Forms

フォーム認証する。

### ④Passport

.NET Passport認証する。

### ■ authorization要素

authorization要素は、どのようなユーザーにアクセス権を与えるのかを指定するものです。allow要素で許可、deny要素で禁止を示します。

デフォルトでは、次のように、すべてのユーザーがアクセスできるようになっています。

```
<authorization>
<allow users="*" />
</authorization>
```

ここで、「\*」は全ユーザーを示します。

## フォーム認証の基本

ASP.NETでユーザー認証をするとき、幅広く使われるのがフォーム認証です。

フォーム認証とは、図1のように、ユーザーが最初にアクセスしてきたときに、ログインページに誘導します<sup>[注1]</sup>。

そしてログインページで正しいユーザー名とパスワードを入力したならば、ユーザーが正しいことの証拠としてCookieを与えます。これがフォーム認証におけるログイン処理となります。

一度ユーザーにCookieを与えれば、Cookieの有効期間を過ぎるまで、同じWebアプリケーション（同一の仮想ディレクトリに保存されている.aspxファイル群）にアクセスできます。

## フォーム認証を使ってみる

では実際に、フォーム認証を使ってみましょう。

フォーム認証を使うには、Web.configファイルを、リスト1のように記述します。

### ①フォーム認証の設定

どのようにしてフォーム認証をするのかを決めるのが、forms要素です（表1）。

注1) ログインページへの誘導行程では、要求してきたURLのWebフォームが実在するかどうかのチェックはされません。実在しないWebフォームを要求した場合でも、ログインページに遷移します。

図1：フォーム認証の仕組み

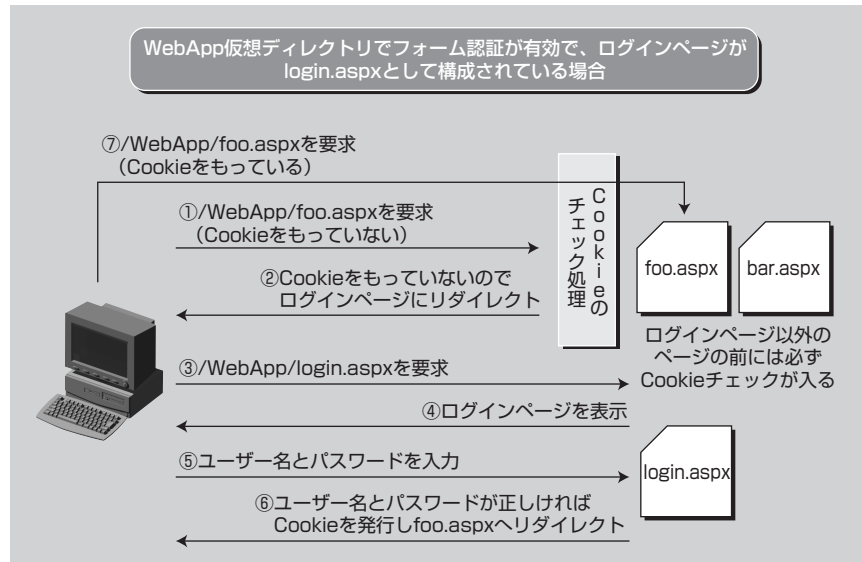


表1のうち、slidingExpirationについて補足しておきます。

slidingExpirationがFalseであるときには、ユーザーがログインして、timeout属性で指定した時間が経過するとCookieが無効になります。

たとえばtimeout属性で「30」を指定してあれば、ユーザーがログインして30分経過し、たとえユーザーがWebアプリケーションで何か処理をしていても、そこですぐにCookieが切れ、ログインページに誘導されます。

それに対してslidingExpiration属性をTrueにすると、ユーザーがどこかのWebフォームにアクセスした時点で、

Cookieのタイムアウト時間が延長されます。

すなわち、slidingExpiration属性をTrueにすることで、「最後にユーザーがWebアプリケーションのいずれかのWebフォームにアクセスしてからtimeout属性で指定した時間までCookieが有効」という動作になります。

ほとんどの場合、ユーザーはWebアプリケーションでの作業中に、突如タイムアウトになって、ログインページに誘導されたら怒るでしょう。ですから、slidingExpiration属性はTrueにするのが一般的と言えます。

表1：forms要素で決められる項目

項目	内容
name	ユーザーに発行するCookieの名前
loginUrl	ログインページを提供するWebフォーム
protection	Cookieの暗号化の設定
timeout	タイムアウト時間 (分)
path	送信するCookieに設定するURLパス
requireSSL	SSLによる暗号化を必須とするかどうか
slidingExpiration	アクセスするたびにタイムアウト時間を延ばすかどうか