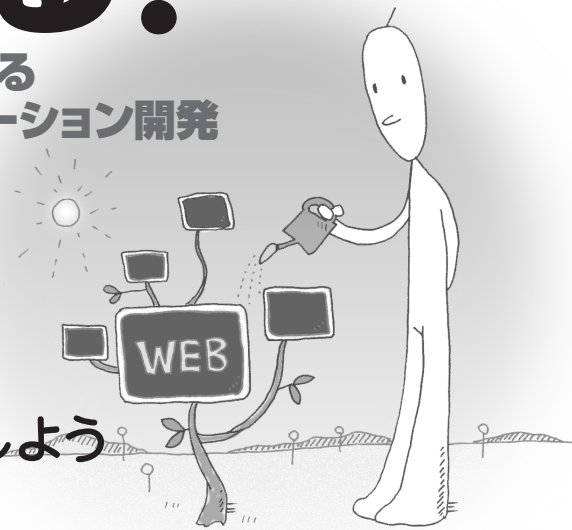


必ずできる!

基礎固め ゼロからはじめる Webアプリケーション開発

ASP.NET



第8回

フォーム認証でアプリケーションを共有しよう

山田 祥寛 *YAMADA, Yoshihiro*
<http://www.wings.msn.to/>

Technology Tools

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

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥ASPディレクトリに収録されています。

¥BEGINASPNET
 「グループスケジュール管理」アプリケーション

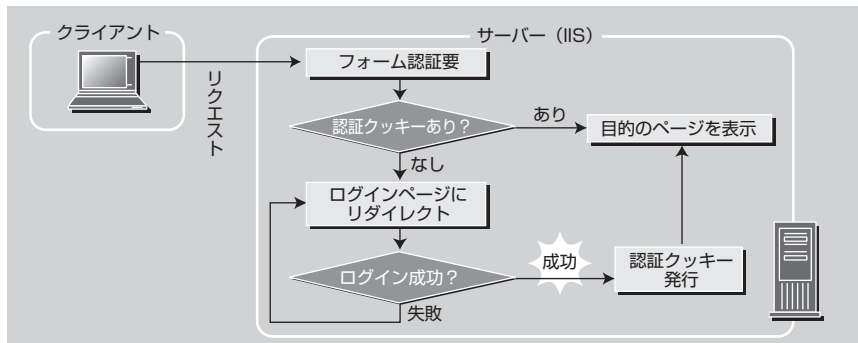
はじめに

前回までは、あくまでひとりのユーザーがアプリケーションを占有するという前提で、スケジュールアプリケーションにさまざまな機能を追加してきました。しかし、Web上でスケジュールを管理するということは、複数のユーザーでスケジュール情報を共有してこそ意味があります。情報を共有するには、まず、アプリケーション内で現在のユーザーを識別できるようにする必要があります。この識別の役割を担うのが、認証機能です。

ASP.NETでは「Windows認証」や「フォーム認証」「パスポート認証」のような認証手段に標準で対応していますが、本稿では、その中でももっとも使い出があるフォーム認証にフォーカスしてご紹介したいと思います。

フォーム認証は、「認証クッキー」(チケット)という仕組みを利用して、認証がすでに行なわれているかどうかを決定します(図1)。認証が必要なページにアクセスした場合(図2)、ASP.NETはまずクライアントが「認証済み情報を含む」クッキー(すなわち「認証クッキー」)を持っているかどうかを判断します。持っている場合には目的

図1：フォーム認証までの流れ



必ずできる!

基礎固め ゼロからはじめる
Webアプリケーション開発

ASP.NET



図2：アプリケーションにアクセスすると、認証ページを表示

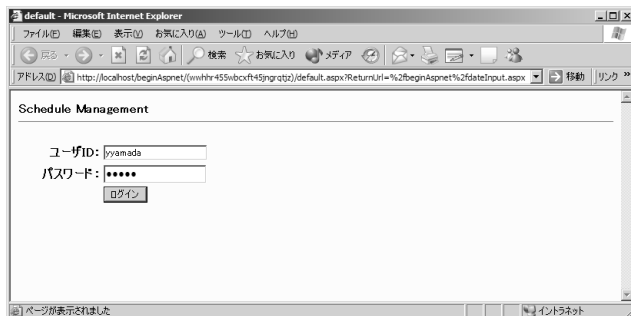
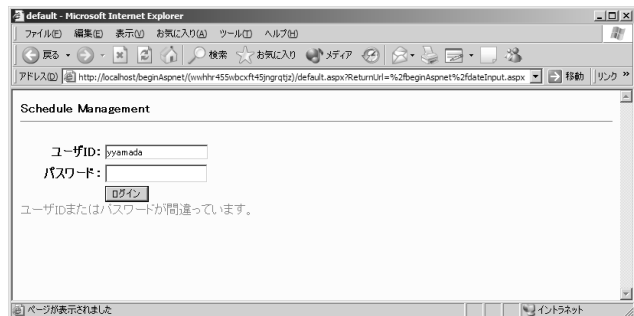


図3：ログインに失敗した場合はエラー表示



のページをそのまま表示し、持っていない場合にはあらかじめ指定されたログインページにリダイレクトするというわけです。ログインが成功した場合は目的のページを表示し、失敗した場合はエラーを表示します (図3)。

ザー情報の格納先を選びません。標準では「Web.config」という構成 (設定) ファイルでユーザー情報を管理しますが、必要に応じて、データベースサーバーやXMLファイルなどを自由に選択することができます。

※ 認証の方法と基本情報を設定する
プロジェクトルート直下には、すでにデフォルトのWeb.configが用意されています。コードエディタを開き、「authentication」で検索すると、以下のような記述が見つかりますので、この記述をリスト1のように書き換えてください。

フォーム認証の特性

フォーム認証には、以下のような特性があります。

特性 1 ログイン画面の

カスタマイズが自由自在

Windows認証などでは、ユーザーIDやパスワードの入力時に標準のダイアログボックスを使うしか選択肢がありません。しかし、フォーム認証では自分でログインページを構築することができるので、アプリケーションのパナーアイコンやユーザー未登録時の案内、あるいは、アプリケーション使用上の規約など、ログイン時に必要な情報を自由に盛り込めます。

特性 2 ユーザー情報の格納先を

選ばない

後述しますが、フォーム認証はユー

特性 3 ログアウト機能が

標準で用意されている

Windows認証では、ログアウト機能を実装できません。つまり、いったんログインしたユーザーの情報は、ブラウザを閉じるまで保持され続けるということです。これは、意図せずして認証済みの状態が維持されてしまうという意味で、セキュリティ上好ましいことではありません。しかし、フォーム認証ではユーザーが明示的にログアウトを行なうことができます。

基本的な フォーム認証の実現

まず、フォーム認証の構築は、プロジェクトルートのWeb.configに対して、認証の方法とユーザー情報、アクセスポリシーを設定するところから始まります。

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

<authentication>要素のmode属性を「Forms」に設定した場合 (フォーム認証を有効化した場合)、<authentication>要素配下にはフォーム認証に必要な一連の設定を定義する必要があります (表1)。

※ アクセスポリシーの定義を行なう

ただし、<authentication>要素はただ単に認証の種類と認証方法について定義しているにすぎません。認証を通過した“どのユーザー”が、アプリケーション内の“どのリソース”にアクセス可能なのかについては、同じようにWeb.config内の<authorization>要素で定義する必要があります。

先述の<authentication>要素は“必ず”プロジェクトルート直下のWeb.