いまからだって遅くない！ASP.NETを完全マスター

冬休み短期集中講座

# ASP.NETで作るWebアプリケーション

最終回

## セキュリティ対策とパフォーマンスの向上

西沢 直木 *NISHIZAWA, Naoki*
http://www.nishi2002.com/

**Technology Tools**

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

**Level**

★★☆☆☆

**Samples**

### 動くだけでは使えない

Webアプリケーションはエラーなしに動けばよいというだけではなく、状況に応じてセキュリティ対策やパフォーマンス向上のための施策が必要となります。今回はこれらの対策を見ていきましょう。

### セキュリティ対策

Webアプリケーションでのセキュリティ対策には、アクセスを制限したり、フォームからの入力値をエスケープするなどのパターンがあります。また、整備が進みつつある法律についても知っておいたほうがよいでしょう。

### アクセス制限

アクセスを制限するには、ログイン画面を設置してユーザーを認証します。ASP.NETで利用可能な認証方式には、

- Windows認証
- フォーム認証
- Passport認証

があります。このうち、Windows認証とフォーム認証、さらにASP.NET 2.0で認証機能を作成するためのコントロールを見ていきましょう。

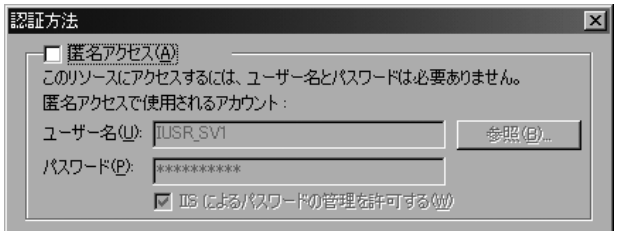
#### Windows認証

Windows認証はIISの認証機能を使うもので、ユーザー情報はサーバーに登録されたWindowsアカウントを使うので特別な準備なく使用できます。次のように3種類の認証形式が用意されています。

- 基本認証
- ダイジェスト認証
- 統合Windows認証

「基本認証」は多くのブラウザで利用可能ですが、ユーザー名とパスワードが暗号化されず（Base64エンコード）送信される点に注意を要します。よってSSL環境で運用することが必要です。

図1：匿名アクセスを無効にする




「ダイジェスト認証」は基本認証と同様の機能ですが、ユーザー情報が暗号化されて送信されるという点でセキュリティ面では優れています。ただし、HTTP 1.1準拠のブラウザを使う必要があります。

「統合Windows認証」は主にイントラネットで使われる形式で、Windowsのユーザー情報が認証に使われるのでネットワーク上をパスワードが送信されないという点で安全です。ただ、利用可能なブラウザがInternet Explorerに限られ、プロキシ経由では使用できないこともあります。

統合Windows認証を利用するには次の方法があります。ディレクトリ全体のファイルにアクセス制限をかける場合は、IISの設定画面で匿名アクセスを無効（図1）にして認証形式を指定します。このディレクトリにアクセスするとユーザー情報の入力を促すダイアログが表示されます（図2）。

また、ディレクトリ内のASPXページにアクセス制限をかける場合はWeb.configのauthenticationセクションで認証形式として「Windows」を指定し、authorizationセクションで匿名アクセスを無効（<deny users="?" />）にします。

## ■フォーム認証

IISの認証機能を使わない場合、ログイン画面や認証後の画面遷移ロジックを自分で用意します。ASP.NETではWeb.configにユーザーのリダイレクト先やユーザー情報を指定可能です。リスト1のコードは、認証されていないユーザーをlogin.aspxに移動させ、ユーザー名「test」、パスワード「pass1234」という値と照合するための定義例です。

protection属性はCookieの設定で、既定値は「All」（暗号化と改ざん検知）です。

パスワード形式（passwordFormat）に「Clear」を指定して「pass1234」のようなパスワードを記述することもできますが、セキュリティ面を考慮して上の例のような暗号化

図2：認証ダイアログ


リスト1：フォーム認証を使用する

```
<authentication mode="Forms">
  <forms name="logintest"
   loginUrl="login.aspx" protection="All">
    <credentials passwordFormat="MD5">
      <user name="test"
       password="b4af804009cb036a4ccdc33431ef9ac9"/>
    </credentials>
  </forms>
</authentication>
<authorization>
  <deny users="?" />
</authorization>
```

も可能です。さらに高いセキュリティを確保するなら、credentials要素にユーザー情報を記述せず、データベースに保存しておくほうが良いでしょう。

## ■セキュリティ関連コントロール

ログイン画面に加えてユーザー登録機能や、ユーザーがパスワードを忘れたときに再通知する機能が必要となることもあります。ASP.NET 2.0ではセキュリティ関連コントロールによって、これらの機能を簡単に作成可能です。

たとえば、ログイン画面を作成するLoginコントロールの簡単なコードは以下のとおりです。これにより、図3のようなログイン画面が作成されます。

```
<asp:Login ID="Login1" Runat="server">
</asp:Login>
```

ログインに必要なユーザー情報を登録する機能はCreateUserWizardコントロールで作成可能です。コントロールの