

ASP.NETフレームワーク自由自在

インターフェイスの拡張の基礎と実践

>>> 小野 修司 ONO, Shuji
あおい情報システム株式会社

はじめに

ASP.NETは拡張性の高いフレームワークです。ASP.NETで提供されているWebフォーム（aspxファイル）やWebサービス（asmxファイル）といった機能はこのフレームワーク上に実装されています。同じフレームワーク上に独自の機能を実装し、組み込むことが可能です。また、ASP.NETにはWindows認証、フォーム認証、Passport認証といった認証方法が組み込まれています。これらも同じフレームワークをベースに構築されているものです。同様に独自の認証方法を実装して組み込むことも可能です。

レベル >>> Level

1 2 3 4 5

ツール >>> Tool

- Visual C# 2005 Express
- Visual Web Developer 2005 Express
- ASP.NET

ASP.NETのフレームワークはよくできていますので、これらの拡張を行なうための作業は難しいものではありません。フレームワークについて、ちょっとした知識さえあればよいのです。ASP.NETのフレームワークの仕組みを学び、それを実際に利用してみましょう。

基礎知識編

ASP.NETのフレームワークについて、まずそのフレームワークの特徴と、それを拡張する2つの要素(IHttpModuleインターフェイスとIHttpHandlerインターフェイス)についてきちんと理解しましょう。

言語 >>> Language

- Visual C#

サンプル >>> Sample

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、<http://www.shoeisha.com/mag/winddev/>からダウンロード可能です。

フレームワークとは

Webアプリケーションにおけるフレームワーク

最近のWebアプリケーションの開発においては、ある程度の規模以上の開発の場合、なんらかのフレームワークを利用することが一般的になっています。これは、Webアプリケーションがはっきりとした定型的部分を持っているからです。

Webアプリケーションでは、ブラウザからサーバーにリクエストが送られることによって処理が進められます。このときブラウザからサーバーに送られるリクエストは、HTTPという標準仕様に沿った形式のデータです。データの形式が決まっていますから、プログラムからそのデータにアクセスするための方法も決まった形になります。

処理の形式が決まっているのですから、その処理を行なう部分をフレームワークとして構築してしまえば、同じような処理の記述を何度も繰り返すような無駄な作業を省くことができます。

このように、Webアプリケーション

においてはブラウザからのリクエストという定型的な部分の存在が、それを扱うフレームワークを登場させる背景になったと考えてもいいでしょう。

フレームワークは「型枠」と訳せますが、ではこの枠の中には何が入るのでしょうか。

Webアプリケーションの定型的な処理の部分をフレームワークが受け持ってくれますので、残りは個々のアプリケーションごとに異なる部分となります。これを「ホットスポット」と呼びます。フレームワークを利用した開発では、このホットスポットの部分を実装することが作業の主な内容になります。

ASP.NETのフレームワークの特徴

ASP.NETを大きくとらえた場合、2つのホットスポットがあります。ひとつはブラウザから送られてくるすべてのリクエストを共通に扱うための処理を記述するホットスポット。そしてもうひとつが個々のリクエストを扱うための処理を記述するホットスポットです。

これらのホットスポットに対するプログラムの記述は、フレームワークによって用意された枠の中で行なう必要があります。ASP.NETはこのプログラムの記述のためにそれぞれインターフェイスを用意しています。すべてのリクエストを共通に扱うホットスポットに対するプログラムを記述するために用意されているのが「IHttpModule」インターフェイスです。また、個々のリクエストを扱うホットスポットに対するプログラムを記述するためには「IHttpHandler」インターフェイスが用意されています。

そしてASP.NETではこれらのイン

ターフェイスに対して、典型的なWebアプリケーションで必要とされる処理があらかじめ実装されて組み込まれています。それがIHttpModuleの実装である各種の認証処理であり、IHttpHandlerの実装であるWebフォーム (aspxファイル) やWebサービス (asmxファイル) です。

これらのインターフェイスを理解することが、ASP.NETのフレームワークを理解することにつながります。

IHttpModuleインターフェイス 組み込み済みのモジュール

IHttpModuleインターフェイスがどのように使われるのかを見るために、まず実際にASP.NETに組み込まれているクラスについて調べてみましょう。

そのためには、ASP.NETに対してHttpModuleを設定しているファイルを探します。ASP.NET 2.0の場合は「C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CONFIG」フォルダにあるWeb.configファイルの<httpModules>要素に以下のクラスが登録されています。

```
System.Web.Caching.OutputCacheModule
System.Web.SessionState.SessionStateModule
System.Web.Security
    .WindowsAuthenticationModule
System.Web.Security
    .FormsAuthenticationModule
System.Web.Security
    .PassportAuthenticationModule
System.Web.Security.RoleManagerModule
System.Web.Security.UrlAuthorizationModule
System.Web.Security.FileAuthorizationModule
System.Web.Security
    .AnonymousIdentificationModule
System.Web.Profile.ProfileModule
System.Web.Mobile.ErrorHandlerModule
System.ServiceModel.Activation.HttpModule
```

ドキュメントに記述が存在しないク

ラスもあるのですが、これらは大きく分けると以下のような役割を持っています。

- ・ユーザーの認証を行なう
Webアプリケーションを利用しようとしているユーザーが誰であるかをセットします。

WindowsAuthenticationModule、FormsAuthenticationModule、PassportAuthenticationModuleがこれにあたります。

- ・アクセス権の承認を行なう
ユーザーがリソースにアクセスしてよいかどうかを確認します。

UrlAuthorizationModule、FileAuthorizationModuleがこれにあたります。

- ・データの取得／保存を行なう
サーバー上で保存されているデータをプログラムから利用できるように取り出したり、処理の終了後に保存したりします。

SessionStateModule、RoleManagerModule、ProfileModuleがこれにあたります。

認証、承認、データ取得／保存といった処理はすべてのリクエストに対して共通に行なわれてはじめて有効に機能するものです。では具体的にはどのようにこれらの実装を行なうのでしょうか。それを知るためには、ASP.NETのフレームワーク上で発生するアプリケーションイベントについて理解する必要があります。

アプリケーションイベント

アプリケーションイベントというのは、ASP.NET上ですべてのリクエスト