

# .NET Framework

# ネームスペースの旅

すべての根源よりすべてのクラスに向けて

Episode

## 4 System.Web ネームスペース編

最終回

グレースシティ株式会社  
アドバイザリースタッフ  
矢沢 久雄 YAZAWA, Hisao



### はじめに



.NET Frameworkで作成できるアプリケーションには、2種類のユーザーインターフェイスがあります。Windows フォームとWebフォームです。Windows フォームとは、独立したウィンドウをもつWindowsアプリケーションのことで（これに関しては、前回の連載で説明しました）。Webフォームとは、Webブラウザ上で動作するWebアプリケーションのことで

.NET Frameworkが登場する前は、WindowsアプリケーションとWebアプリケーションの開発スタイルがまったく異なっていました。「スタンドアロン

アプリケーション」→「クライアント/サーバーアプリケーション」→「Webアプリケーション」という進化に伴って、さまざまな技術が追加されていったからです。.NET Frameworkでは、両者をほとんど同じスタイルで開発できます。WebフォームにWebコントロールを貼り付け、プロパティを設定し、イベントにコードを記述すればよいのです。ただし、Webアプリケーションには、Windowsアプリケーションにない概念もあります。今回は、Webアプリケーションを作成するためのクラスを提供するSystem.Webで始まる名前のネームスペースを旅します。



### System.Web ネームスペース



.NET Frameworkのクラスライブラリには、System.Webで始まる名前のネームスペースがいくつかあります。これらは、WebアプリケーションやXML Webサービスなど、Web関連のプログラムを作成するためのクラスを提供します。主なネームスペースを表1に示しておきましょう。

これらのネームスペースの中で、最初に注目してほしいのは、System.Webネームスペースです。このネームスペースは、WebサーバーとWebブラウザの基本的な通信を実現するクラスを提供します。主なクラスを表2に示します。

.NET Frameworkでは、Windowsアプリケーションと同様の開発スタイルでWebアプリケーションを作成できますが、両者には根本的な違いがひとつあります。それは、Webアプリケーションが、WebサーバーとWebブラウザの通信によって動作していることです。このことから、Webアプリケーション

#### 本稿で前提となるもの

OS Windows 2000 Professional (SP2) 以降

開発環境 Visual Studio.NET  
.NET Framework 1.0.3705.288 (SP2)





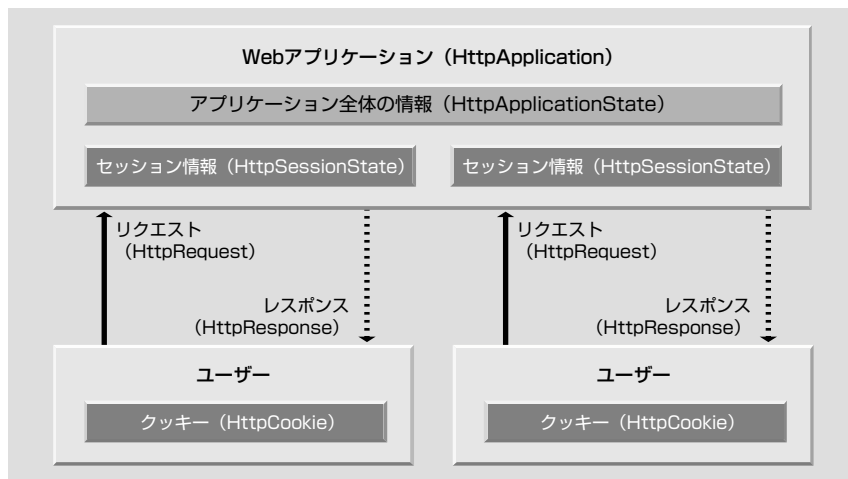
表1：Web関連のクラスを提供するネームスペース

ネームスペース名	提供するクラスの機能
System.Web	WebサーバーとWebブラウザの基本的な通信、クッキーの操作、ファイルの転送など
System.Web.UI System.Web.UI.WebControls System.Web.UI.HtmlControls	Webアプリケーションのユーザーインターフェイスの構築
System.Web.Services	XML Webサービスの作成
System.Web.Caching	Webアプリケーションのキャッシュ機能
System.Web.Configuration	Webアプリケーションの設定
System.Web.Security	Webアプリケーションのセキュリティ機能

表2：System.Webネームスペースの主なクラス

クラス名	機能
HttpRequest	WebブラウザからWebサーバーに送られたリクエストを取得する
HttpResponse	WebサーバーからWebブラウザにレスポンスを送る
HttpCookie	クッキーを操作する
HttpApplicationState	アプリケーション全体の状態を保持
HttpSessionState	個々のセッションの状態を保持
HttpApplication	すべてのWebアプリケーションに共通する機能を提供する

図1：Webアプリケーションの基本的な機能を提供するクラス



には、リクエスト、レスポンス、セッションなど、Windowsアプリケーションにはない概念がいくつかあります。

WebブラウザからWebサーバーに送られる要求を「リクエスト」と呼びます。WebサーバーからWebブラウザに返される応答を「レスポンス」と呼びます。これらは、それぞれHttpRequest

クラスとHttpResponseクラスで操作できます。

WebブラウザとWebサーバーの通信は、1回のリクエストとレスポンスで完結してしまいます。すなわち、1回ごとに新規接続の状態を繰り返しているのです。ただし、ショッピングサイトのように、Webページが切り換わっても

同じユーザーが接続している情報を記録したい場合もあるでしょう。そのためには「クッキー」や「セッション変数」を使います。クッキーとは、WebサーバーがWebブラウザにファイルとして記録する情報のことです。クッキーは、HttpCookieクラスを使って操作できます。

Webサーバーに配置されたWebアプリケーションは、ひとつの「アプリケーション」です。ただし、ひとつのWebアプリケーションは、複数のユーザーからアクセスされます。個々のユーザーとの接続のことを「セッション」と呼びます。アプリケーションは、HttpApplicationStateクラスで操作でき、セッションは、HttpSessionStateクラスで操作できます。HttpSessionStateクラスが個々のセッションごとの情報(セッション変数)をもち、HttpApplicationStateクラスがすべてのセッション(すなわちアプリケーション全体)の情報をもつわけです。それらをまとめるのが、HttpApplicationクラスというわけです(図1)。



## System.Web.UI.Pageクラス



Visual Studio.NETを使ってWebアプリケーション(ASP.NET Webアプリケーション)のプロジェクトを作成してみましょう。Webアプリケーションのテンプレートとなるコードが自動生成されます(リスト1)。

ここで注目してほしいのは、ひとつのWebアプリケーション(ひとつのWebページ)がSystem.Web.UI.Pageクラス