

必ずできる!

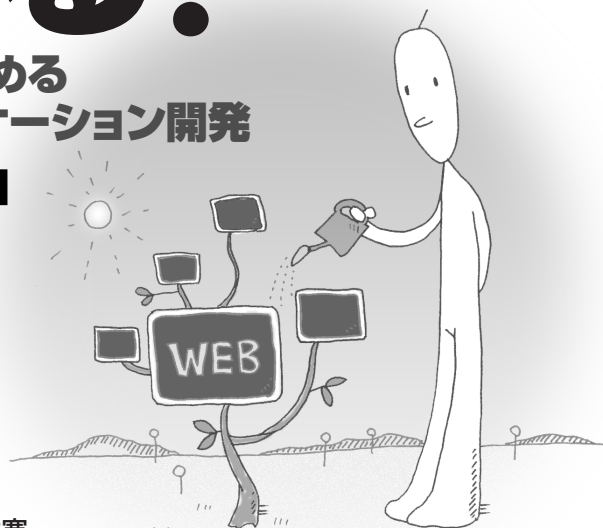
基礎固め ゼロからはじめる Webアプリケーション開発

ASP.NET

第11回

Global.asaxで アプリケーション共通の 機能を追加しよう

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

Global.asaxとは?

Global.asaxは「ASP.NETアプリケーションファイル」とも呼ばれ、アプリケーションレベルのイベント（アプリケーションの開始/終了など）を処理したり、アプリケーション共通で利用可能なオブジェクトをまとめて宣言するために使用します。必ず仮想ディレクトリの「直下」に配置してください（Web.configのように、アプリケーション内の任意のフォルダに分散配置することはできません）。

古いASPの時代には「Global.asa」という名前で提供されていたものに相当しますが、ASP.NETでは扱えるイベントの数も増え、大幅に機能強化されています（表1）。

イベントの種類は、大きく「リクエストイベント」と「条件付きイベント」とに分類されます。リクエストイベントは、アプリケーション内のページがリクエストされるたびに“必ず”発生しますが、条件付きイベントは、アプ

リケーションの起動、あるいはエラーの発生などの“ある特定の条件のもと”でのみ発生するイベントになります。

もっとも、これらのイベントすべてを常に利用しなければならないというわけではありません。必要に応じて、最低限のイベントプロシージャだけを定義すればよいのです。そもそもアプリケーションイベントを処理する必要がないという場合には、Global.asax自体を省略することも可能です。

Global.asaxはアイデア次第でさまざまな機能をアプリケーションに提供してくれますが、本稿ではその中でも特に重要と思われる機能を「アクセスロギング」「共通データの事前読み込み」「エラー情報のメール通知」などのサンプルを例に示してみることにしましょう。

アクセスログを記録する

まずは、リクエストイベントの中でももっとも代表的なApplication_On

表1：Global.asaxで扱うことができるイベント

カテゴリ	イベント名	概要
リクエスト イベント	Application_BeginRequest	HTTPハンドラがリクエスト処理を開始する前に発生
	Application_AuthenticateRequest	認証の準備ができたタイミングで発生
	Application_AuthorizeRequest	ユーザー認証が完了したタイミングで発生
	Application_ResolveRequestCache	リクエストを（ページ実行ではなく）キャッシュ処理するタイミングで発生（ただし、キャッシュ処理の有無に関わらず、イベントは認証終了のタイミングで発生）
	Application_AcquireRequestState	リクエストに関連づいた状態（セッション状態など）を取得するタイミングで発生（独自セッション管理を行ないたい場合は、本イベントを利用）
	Application_PreRequestHandlerExecute	HTTPハンドラがページの実行を開始する直前に発生
	Application_PostRequestHandlerExecute	HTTPハンドラがページの実行を完了した直後に発生
	Application_ReleaseRequestState	すべての実行を完了したタイミングで発生（本イベントのタイミングで、セッションなどの状態を任意のストアに保存）
	Application_UpdateRequestCache	実行完了後に発生（出力キャッシュを更新）
	Application_EndRequest	すべてのリクエスト処理が完了したタイミングで発生
	Application_PreSendRequestHeaders	HTTPヘッダをクライアントに送信する直前に発生
	Application_PreSendRequestContent	HTTPコンテンツをクライアントに送信する直前に発生
条件付き イベント	Application_Start	アプリケーションが初回起動した1回のみ発生。アプリケーションステイトや静的変数を初期化するためなどに使用。Application_BeginRequestイベントに先立って発生
	Session_Start	ユーザーセッションが初回起動した1回のみ発生。セッション変数などを初期化するために使用
	Session_End	ユーザーセッションがクローズしたタイミングで発生。セッション変数にプールされた情報をデータベースなどに最終的に保存するためなどに使用
	Application_End	アプリケーションが破棄されたタイミングで発生。アプリケーション変数にプールされた情報をデータベースなどに最終保存するためなどに使用
	Application_Disposed	共通言語ランタイムがASP.NETアプリケーションを最終的にメモリから破棄するタイミングで発生。このイベントを利用するケースはほとんどない
	Application_Error	アプリケーション内で処理されない例外（エラー）が発生した場合にのみ発生。エラーログの記録や管理者への通知、エラーページの生成などのために使用

COLUMN リクエストイベントが発生する順番

表1でも示したように、Global.asaxで対応しているリクエストイベントの数は12にものぼります。利用にあたっては、それぞれのイベントがどのような順番で発生するのかを知っておくことが重要です（図A）。

ただし、Application_OnPreSendRequestHeaderイベントとApplication_OnPreSendRequestContentイベントの発生順序は、常に図Aの順番になるわけではない点に注意してください。バッファリング処理を有効（デフォルト）にしている場合、この2つのイベントは図の通り、リクエスト処理をすべて終了したタイミングで発生しますが、バッファリング処理が無効である場合、ヘッダ／コンテンツの出力は処理のたびに行なわれるので、リクエストイベント内の任意のタイミングで発生することになります。

図A：リクエストイベントの発生順序

