

Apache Tutorial: CGI による動的コンテンツ

トピック

はじめに.....	1
CGI を許可するように Apache を設定する.....	1
CGI プログラムを書く.....	3
しかし、まだ動かない！.....	3
裏で何が起きているのか?.....	5
CGI モジュール/ライブラリ.....	6
更なる情報.....	6
URI References.....	7

はじめに

関連モジュール	関連ディレクティブ
<code>mod_alias</code>	<code>AddHandler</code>
<code>mod_cgi</code>	<code>Options</code>
	<code>ScriptAlias</code>

CGI (Common Gateway Interface) は、ウェブサーバが コンテンツ生成をする外部プログラムと協調して動作するための方法を 定義しています。そのプログラムはしばしば CGI プログラムや CGI スクリプトと呼ばれます。CGI は、ウェブサイトに動的な コンテンツを置くための最も簡単で一般的な方法です。このドキュメントは、Apache ウェブサーバで CGI を設定し、CGI プログラムを書き始めるための入門書となるでしょう。

CGI を許可するように Apache を設定する

CGI プログラムを正しく動作させるには、CGI を許可するように Apache の設定を行う必要があります。 これを行なうための方法がいくつかあります。

ScriptAlias

`ScriptAlias` ディレクティブを使用して、CGI プログラム用の特別な別ディレクトリを Apache に設定します。Apache は、このディレクトリ中の全てのファイルを CGI プログラムであると仮定します。そして、この特別なリソースがクライアントから要求されると、そのプログラムの実行を試みます。

`ScriptAlias` ディレクティブは以下のように使用します:

```
ScriptAlias /cgi-bin/ /usr/local/apache/cgi-bin/
```

デフォルト位置に Apache をインストールしたならば、この例はデフォルト状態の `httpd.conf` 設定ファイルに含まれています。`ScriptAlias` ディレクティブは、URL の前に付加するディレクトリを定義する `Alias` ディレクティブとかなり似ています。`Alias` と `ScriptAlias` は通常、`DocumentRoot` ディレクトリ外のディレクトリのために使用されます。`Alias` と `ScriptAlias` との差は、`ScriptAlias` が接頭辞で始まるすべての URL は CGI プログラムとみなされるという追加の意味を含んでいることです。従って、上記の例では、`/cgi-bin/` で始まるリソースへのあらゆるリクエストに対して、ディレクトリ `/usr/local/apache/cgi-bin/` から提供し、それらを CGI プログラムとして扱うよう Apache に示します。

例えば、URL `http://dev.rcbowen.com/cgi-bin/test.pl` が要求された場合、Apache は ファイル `/usr/local/apache/cgi-bin/test.pl` を実行し、その出力を返すことを試みます。もちろん、ファイルが存在し、実行可能であり、決められた方法で出力を返します。そうでなければ、Apache はエラーメッセージを返します。

ScriptAlias ディレクトリ外の CGI

CGI プログラムは、セキュリティ上の理由から ScriptAlias されたディレクトリに制限されることがしばしばあります。この方法により、CGI プログラムを使用できるユーザを管理者が厳しく制御することができます。しかしながら、適切なセキュリティ事前対策がとられるならば、CGI プログラムを任意のディレクトリで実行できないようにする理由はありません。例えば、ユーザに UserDir ディレクティブで 彼らのホームディレクトリ配下にウェブコンテンツを持たせたいとします。もし、彼らが CGI プログラムを持つことを望んでいても、メインの `cgi-bin` ディレクトリへのアクセスができない場合、CGI プログラムを実行することができる他の場所が必要になります。

CGI の実行を可能にするために Options を明示的に使用する

サーバのメインの設定ファイル中で Options ディレクティブを明示的に使用することで、特定のディレクトリ配下で CGI の実行を許可するように指定することができます:

```
<Directory /usr/local/apache/htdocs/somedir>
Options +ExecCGI
</Directory>
```

上記ディレクティブは、CGI ファイルの実行を可能にするよう Apache に伝えます。また、どのファイルが CGI ファイルかを サーバに伝える必要があります。次の AddHandler ディレクティブの例では、`cgi` または `pl` を拡張子に持つすべてのファイルを CGI プログラムとしてみなすことをサーバに伝えます:

```
AddHandler cgi-script cgi pl
```

.htaccess files

.htaccess ファイルは、ディレクトリ毎にディレクティブを指定する方法です。Apache は、リソースを提供するときに、提供しているファイルが置かれているディレクトリ中の .htaccess というファイルを参照します。そのファイルを発見したら、その中で発見されたディレクティブが適用されます。.htaccess ファイルは、AllowOverride ディレクティブの指定により使えるようになります。AllowOverride ディレクティブは、.htaccess ファイルで設定できるディレクティブのタイプを指定します。AllowOverride ディレクティブの指定がない場合、まったく使えません。CGI の実行を許可するために必要となる ディレクティブを指定可能にするには、以下の設定がサーバのメインの設定で必要になります:

```
AllowOverride Options
```

.htaccess ファイルでは、次のディレクティブが必要となります:

```
Options +ExecCGI
```

この設定では、このディレクトリにおける CGI プログラムの実行を許可するよう Apache

に伝えます。

CGI プログラムを書く

「通常の」プログラミングと CGI プログラミングの間には主に二つの違いがあります。

一つは、CGI プログラムのすべての出力には MIME-type ヘッダを付けなければなりません。これはどのような種類のコンテンツを受け取っているかをクライアントに示す HTTP ヘッダです。ほとんどの場合では、次のように出力します：

```
Content-type: text/html
```

もう一つは、出力を HTML か、ブラウザが表示することができる何か他の形式にする必要があります。大抵の場合は HTML でしょうが、GIF イメージや他の非 HTML コンテンツを出力する CGI プログラムを書くこともあるでしょう。

これら二点以外では、CGI プログラムを書くことは、あなたが書いている他のプログラムとよく似ているでしょう。

最初の CGI プログラム

次に示すのは、ブラウザに 1 行印字する CGI プログラムの例です。以下を入力し、first.pl というファイルに保存し、それを cgi-bin ディレクトリに置いてください。

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "Hello, World.";
```

Perl に精通していなくても、何が起こるかを理解することはできるでしょう。1 行目は、/usr/bin/perl で見つけられるインタプリタにこのファイルを供給することでこのプログラムが実行されることを Apache に（シェル上で実行しようとしているならば、そのシェルに）示します。2 行目は、前述したとおり content-type の定義を印字します。これには復帰改行の二つの組を後に付加します。これにより、ヘッダの終りに空行が置かれ、HTTP ヘッダの終りとボディの始まりを示します。3 行目は、`Hello, World.` という文字列を印字し、これで終りとなります。

好みのブラウザを開き、アドレス

```
http://www.example.com/cgi-bin/first.pl
```

あるいはファイルを置いたロケーションを指定すると、Hello, World. という 1 行がブラウザウィンドに現れるでしょう。それはあまりエキサイティングなことではありません。しかし、これがうまく動けば、他のどのようなものでも動かすことができるようになります。

しかし、まだ動かない！

ウェブから CGI プログラムへのアクセスを行なったとき、ブラウザで見る可能性がある四

Apache Tutorial: CGI による動的コンテンツ

つの基本的なことがあります:

CGI プログラムの出力

素晴らしい！それはすべてがうまく動いたことを意味します。

CGI プログラムのソースコード、または "POST Method Not Allowed" というメッセージ

これは、CGI プログラムを処理できるよう Apache を適切に設定していなかったことを意味します。「CGI を許可するように Apache を設定する」の章を読み直し、あなたが何を間違えたかを探してみてください。

メッセージが "Forbidden" で始まっている

これはパーミッションの問題ということの意味します。Apache のエラーログと、後述の「ファイルのパーミッション」の章をチェックしてください。

"Internal Server Error" というメッセージ

Apache のエラーログをチェックすると、"Premature end of script headers" というログが記録されていると思います。そして、おそらく CGI プログラムによって生成されたエラーメッセージも記録されているでしょう。この場合、CGI プログラムが適切な HTTP ヘッダを出力できない原因を知るために、以下の各章でチェックしてみてください。

ファイルのパーミッション

サーバはあなたの権限で実行されていないのを忘れないように。つまり、起動するとき、サーバは特権をもたないユーザ - 通常 ``nobody`` や ``www`` の権限で実行されます。したがって、あなたが所有するファイルを実行するには別のパーミッションが必要となります。通常、``nobody`` が実行するのに十分なパーミッションを与える方法は、ファイルに誰でも実行可能とするパーミッションを与えることです:

```
chmod a+x first.pl
```

また、もしあなたのプログラムが他のファイルを読み書きするならば、それらのファイルは、これが可能となる正しいパーミッションを持っている必要があります。

これに対する例外は、サーバが suexec¹ を使用するよう設定されている場合です。suexec は、CGI プログラムが置かれているバーチャルホストまたはユーザの ホームディレクトリによって、異なるユーザ権限で実行されるようにします。suexec はとても厳しいパーミッションのチェックがあり、そのチェックを通過できないと "Internal Server Error" となり、その CGI プログラムの実行は失敗します。この場合、どのセキュリティチェックが失敗しているのかを知るために suexec ログファイルをチェックする必要があります。

パス情報

コマンドラインからプログラムを実行するとき、意識しなくてもシェルに渡される情報があります。例えば、参照するファイルのためにどこを検索したらよいかをシェルに伝えるパスがあります。

プログラムが CGI プログラムとしてウェブサーバによって実行されるとき、それはパスを持ちません。CGI プログラム内で呼び出すあらゆるプログラム (例えば、`sendmail` のようなもの) は、フルパスで指定する必要があるでしょう。それにより、CGI プログラムを実行しようとしたとき、シェルはそのようなプログラムを見つけることができます。

同様なことは、スクリプトのインタプリタ (しばしば perl) へのパスで、CGI プログラム

の 1 行目に次のように示されます:

```
#!/usr/bin/perl
```

これがインタプリタへの実際のパスであることを確実にしておきます。

構文エラー

CGI プログラムが失敗するのは大抵、プログラム自身に問題がある場合です。一度 CGI の使い方を理解し、前述の二つの誤りを犯していないならば、まず間違いなくそうでしょう。ブラウザを通してテストを行う前に必ず、コマンドラインからプログラムの実行を試さない。これにより、大抵の問題が起こらなくなります。

エラーログ

エラーログは友達です。全てのうまくいかないことは、エラーログにメッセージを生成します。必ずそれを最初に見るべきです。もし、あなたがウェブサイトを主催している場所がエラーログの参照を許していないならば、きっと他のサイトで主催するべきです。エラーログの読み方を学ぶことで、ほとんど全ての問題が迅速に確認され、迅速に解決されるということが分かるでしょう。

裏で何が起きているのか?

CGI プログラミングに熟達すると、裏で起きていることについて更に理解することは有益になるでしょう。ブラウザとサーバがどのように相互通信するかについては特にそうです。なぜなら、`Hello, World.` を印字するプログラムを書くことはまことに結構ですが、それは特に有益ではありません。

環境変数

環境変数は、あなたがコンピュータを使うときに辺りに存在している値です。それらは、パス (コマンドをタイプしたときに実行する実際のファイルを探し出すところ)、ユーザ名、端末型などのような便利なものです。通常の、毎日の環境変数の完全なリストを調べるには、コマンドプロンプトで `env` を入力します。

CGI の処理中、サーバとブラウザも環境変数を設定し、それにより相互に通信することができますようになります。その環境変数は、ブラウザタイプ (Netscape, IE, Lynx)、サーバタイプ (Apache, IIS, WebSite)、実行されている CGI プログラムの名前などのようなものです。

これらの変数は CGI プログラムが使用することができます。そして、それはクライアントとサーバの通信の話の半分です。必要な変数の完全なリストは <http://hoohoo.ncsa.uiuc.edu/cgi/env.html> にあります。

以下の単純な Perl CGI プログラムは、渡される全ての環境変数を表示します。同様のプログラムは、Apache ディストリビューションの `cgi-bin` ディレクトリに二つ含まれています。いくつかの変数が必須であり、いくつかは任意であることに注意してください。そして、公式のリストにはないいくつかの変数が表示されているかもしれません。さらに、Apache はデフォルトで用意されている基本的なものにあなた自身の環境変数を加える²ための、多くの異なる方法を用意してします。

```
#!/usr/bin/perl
print "Content-type: text/html\r\n\r\n";
foreach $key (keys %ENV) {
print "$key --> $ENV{$key}<br>";
}
```

STDIN と STDOUT

サーバとクライアント間のもう一つの通信は、標準入力 (STDIN) と標準出力 (STDOUT) を通じて行なわれます。通常の文脈において、STDIN はキーボードやプログラムが動作するために与えられるファイルを意味し、STDOUT は通常コンソールまたはスクリーンを意味します。

ウェブフォームから CGI プログラムへ POST したとき、フォームのデータは特別なフォーマットで束ねられ、STDIN を通じて、CGI プログラムに引き渡されます。プログラムはデータがキーボード もしくはファイルから来ていたかのように処理することができます。

「特別なフォーマット」はとても単純です。フィールド名と値はイコール (=) で結ばれます。そして値の組はアンパサンド (&) で結ばれます。スペース、アンパサンド、イコールのような面倒な文字は、それらが動作を駄目にしないようにその文字に相当する 16 進に変換されます。全データ文字列は、以下のようになります:

```
name=Rich%20Bowen&city=Lexington&state=KY&sidekick=Squirrel%20Monkey
```

時々、このような文字列が URL に付加されるのを見るでしょう。その場合、サーバは QUERY_STRING という環境変数にその文字列を入れます。それは GET リクエストと呼ばれます。HTML フォームでは、データを渡すために GET と POST のどちらを使用するかを、FORM タグの METHOD 属性の設定で指定します。

CGI プログラムは、その文字列を役に立つ情報に分割する責任があります。幸いにも、そのデータ処理を助けるライブラリやモジュールが存在します。これらは、CGI プログラムの他の面でも同様に役に立ちます。

CGI モジュール/ライブラリ

CGI プログラムを書くとき、面倒な仕事の大部分をしてくれる コードライブラリまたはモジュールを使うことを検討すべきです。これはエラーを減らし、早い開発につながります。

Perl で CGI プログラムを書いているなら、モジュールは CPAN³ で提供されています。この目的のための最も普及しているモジュールは CGI.pm です。CGI::Lite も検討しましょう。これは、ほとんどのプログラム において必要とするすべての機能の最小セットの実装です。

C で CGI プログラムを書いているなら、いろいろな オプションがあります。これらの内の一つは <http://www.boutell.com/cgic/> で提供されている CGIC ライブラリです。

更なる情報

Apache Tutorial: CGI による動的コンテンツ

CGI に関する情報はウェブで数多く提供されています。CGI の問題については Usenet の `comp.infosystems.www.authoring.cgi` で、他のユーザと論議することができます。HTML Writers Guide の `-servers` メーリングリストは、あなたの質問に回答してくれる偉大なリソースです。 <http://www.hwg.org/lists/hwg-servers/> で更に多くを探し出すことができます。

そしてもちろん、おそらく CGI プログラムの動作に関する詳細の全てが記述されている CGI の仕様を読むべきです。オリジナルバージョンを NCSA⁴ で、アップデートされたドラフトを Common Gateway Interface RFC プロジェクト⁵で参照することができます。

CGI の問題について、加わっているメーリングリストまたはニュース グループに質問を送るとき、起こったもの、起こってほしいこと、実際に起こったことがどう違うか、使用しているサーバ、CGI プログラムを記述している言語に関する十分な情報と、可能であれば問題のコードを提供するようにしてください。そうすることで、問題がより簡単に見つかるようになります。

Apache のソースコードにおいて問題を発見したことを確信していない限り、CGI の問題に関する質問を Apache バグデータベースに送るべきでないことに注目してください。

URI References

- [1] <http://httpd.apache.org/docs-2.1/suexec.html>
- [2] <http://httpd.apache.org/docs-2.1/env.html>
- [3] <http://www.cpan.org/>
- [4] <http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>
- [5] <http://web.golux.com/coar/cgi/>