

URL からファイルシステム上の位置へのマップ

この文書は Apache がリクエストの URL から送信するファイルの ファイルシステム上の位置を決定する方法を説明します。

トピック

関連するモジュールとディレクティブ.....	1
DocumentRoot.....	1
DocumentRoot 外のファイル.....	1
ユーザディレクトリ.....	2
URL リダイレクション.....	3
リバースプロキシ.....	3
リライトエンジン.....	4
File Not Found.....	4
URI References.....	5

関連するモジュールとディレクティブ

関連モジュール	関連ディレクティブ
<code>mod_alias</code>	<code>Alias</code>
<code>mod_proxy</code>	<code>AliasMatch</code>
<code>mod_rewrite</code>	<code>CheckSpelling</code>
<code>mod_userdir</code>	<code>DocumentRoot</code>
<code>mod_speling</code>	<code>ErrorDocument</code>
<code>mod_vhost_alias</code>	<code>Options</code>
	<code>ProxyPass</code>
	<code>ProxyPassReverse</code>
	<code>Redirect</code>
	<code>RedirectMatch</code>
	<code>RewriteCond</code>
	<code>RewriteMatch</code>
	<code>ScriptAlias</code>
	<code>ScriptAliasMatch</code>
	<code>UserDir</code>

DocumentRoot

リクエストに対してどのファイルを送信するかを決定するときの Apache のデフォルトの動作は、リクエストの URL-Path (URL のホスト名と ポート番号の後に続く部分) を取り出して設定ファイルで指定されている `DocumentRoot` の最後に追加する、というものです。ですから、`DocumentRoot` の下のディレクトリやファイルがウェブから見える基本のドキュメントの木構造を なします。

Apache にはサーバが複数のホストへのリクエストを受け取る バーチャルホスト¹ の機能もあります。この場合、それぞれのバーチャルホストに対して違う `DocumentRoot` を指定することができます。また、`mod_vhost_alias` モジュールにより提供されるディレクティブを使って、送信するためのコンテンツの場所をリクエストされた IP アドレスやホスト名から動的に決めることもできます。

DocumentRoot 外のファイル

ファイルシステム上の、厳密には `DocumentRoot` の下にはない部分へのウェブアクセスを許可する必要がある場合がよくあります。Apache はこのために複数の方法を用意しています。Unix システムでは、ファイルシステムの他の部分をシンボリックリンクを使って `DocumentRoot` の下に持ってくることができます。セキュリティ上の理由により、Apache は該当するディレクトリの `Options` の設定に `FollowSymLinks` か `SymLinksIfOwnerMatch` がある場合にのみシンボリックリンクをたどります。

代替の方法として、`Alias` ディレクティブを使ってファイルシステムの任意の部分をウェブの空間にマップできます。たとえば、

```
Alias /docs /var/web
```

という設定のときは、URL `http://www.example.com/docs/dir/file.html` には `/var/web/dir/file.html` が送信されます。`ScriptAlias` も、対象となっているパスが CGI スクリプトとして扱われるという追加の効果以外は同じように動作します。

もっと柔軟な設定が必要な状況では、`AliasMatch` ディレクティブや `ScriptAliasMatch` ディレクティブを使って強力な正規表現に基づいたマッチと置換を行なうことができます。たとえば、

```
ScriptAliasMatch ^/~([a-zA-Z0-9]*)/cgi-bin/(.*) /home/$1/cgi-bin/$2
```

は `http://example.com/~user/cgi-bin/script.cgi` へのリクエストを `/home/user/cgi-bin/script.cgi` というパスへマップし、このマップの結果としてのファイルを CGI スクリプトとして扱います。

ユーザディレクトリ

伝統的に Unix システムではユーザ `user` のホームディレクトリを `~user/` として参照できます。`mod_userdir` モジュールはこの概念をウェブに拡張して、それぞれのユーザのホームディレクトリのファイルを以下のような URL を使ってアクセスできるようにします。

```
http://www.example.com/~user/file.html
```

セキュリティの観点から、ウェブからユーザのホームディレクトリへ直接アクセスできるようにすることは適切ではありません。ですから、`UserDir` ディレクティブにはユーザのホームディレクトリの下、ウェブファイルの置かれているディレクトリを指定します。

デフォルトの設定の `Userdir public_html` を使うと、上の URL は `/home/user/public_html/file.html` というようなファイルにマップされます。ここで、`/home/user/` は `/etc/passwd` で指定されているユーザのホームディレクトリです。

`Userdir` には、`/etc/passwd` にホームディレクトリの位置が書かれていないシステムでも使うことのできる他の形式もあります。

中にはシンボル `~` (%7e のように符号化されることが多い) を格好が悪いと思って、ユーザのディレクトリを表すために別の文字列の使用を好む人がいます。`mod_userdir` はこの

URL からファイルシステム上の位置へのマップ

機能をサポートしていません。しかし、ユーザのホームディレクトリが規則的な構成のときは、`AliasMatch` を使って望みの 効果を達成することができます。たとえば、`http://www.example.com/upages/user/file.html` が `/home/user/public_html/file.html` にマップされるようにするには、以下のように `AliasMatch` ディレクティブを使います:

```
AliasMatch ^/upages/([a-zA-Z0-9]*)/?(.*) /home/$1/public_html/$2
```

URL リダイレクション

上の節で説明した設定用のディレクティブは Apache に ファイルシステムの特定の場所からコンテンツを取ってきて クライアントに送り返すようにします。ときには、その代わりに クライアントにリクエストされたコンテンツは別の URL にあることを 知らせ、クライアントが新しい URL へ新しいリクエストを行なうように する方が望ましいことがあります。これはリダイレクションと 呼ばれていて、`Redirect` ディレクティブにより実装されています。たとえば、`DocumentRoot` の下のディレクトリ `/foo/` が新しいディレクトリ `/bar/` に移動したときは、以下のようにしてクライアントが新しい場所のコンテンツをリクエストするように 指示することができます:

```
Redirect permanent /foo/ http://www.example.com/bar/
```

これは、`/foo/` で始まるすべての URL-Path を、`www.example.com` サーバの `/bar/` が `/foo/` に置換されたものにリダイレクトします。サーバは自分自身のサーバだけでなく、どのサーバにでもクライアントを リダイレクトすることができます。

Apache はより複雑な書き換えの問題のために、`RedirectMatch` ディレクティブを 提供しています。たとえば、サイトのホームページを違うサイトにリダイレクト するけれど、他のリクエストはそのまま扱う、というときは以下の設定を 使います:

```
RedirectMatch permanent ^/$ http://www.example.com/startpage.html
```

あるいは、一時的にサイトのすべてのページを一つのページへリダイレクトする ときは、以下を使います:

```
RedirectMatch temp .* http://www.example.com/startpage.html
```

リバースプロキシ

Apache は遠隔地にあるドキュメントをローカルのサーバの URL 空間に 持つてくることもできます。この手法はリバースプロキシと呼ばれています。ウェブサーバが遠隔地のドキュメントを取得してクライアントに送り返すのが プロキシサーバの動作のように見えるからです。クライアントにはドキュメントが リバースプロキシサーバから送られてきているように見える点が通常の プロキシとは異なります。

次の例では、クライアントが `/foo/` ディレクトリの下にある ドキュメントをリクエストすると、サーバが `internal.example.com` の `/bar/` ディレクトリから取得して、さもローカルサーバからの ドキュメントのようにしてクライアントに返します。

URL からファイルシステム上の位置へのマップ

```
ProxyPass /foo/ http://internal.example.com/bar/  
ProxyPassReverse /foo/ http://internal.example.com/bar/
```

`ProxyPass` ディレクティブは サーバが適切なドキュメントを取得するように設定し、`ProxyPassReverse` ディレクティブは `internal.example.com` からのリダイレクトがローカルサーバの適切なディレクトリを指すように書き換えます。ただし、ドキュメントの中のリンクは書き換えられない、ということは知っておいてください。ですから、`internal.example.com` への絶対パスによるリンクでは、クライアントがプロキシサーバを抜け出して `internal.example.com` に 直接リクエストを送る、ということになります。

リライトエンジン

より一層強力な置換が必要なときは、`mod_rewrite` が提供するリライトエンジンが役に立つでしょう。このモジュールにより提供されるディレクティブは ブラウザの種類、リクエスト元の IP アドレスなどのリクエストの特徴を使って送り返すコンテンツの場所を決めます。さらに、`mod_rewrite` は外部のデータベースファイルやプログラムを使ってリクエストの扱い方を決めることもできます。リライトエンジンは上で挙げられている三つのマッピングすべてを行なうことができます: 内部のリダイレクト (エイリアス)、外部のリダイレクト、プロキシです。`mod_rewrite` を使う多くの実用的な例は URL リライトガイド² で説明されています。

File Not Found

必ず、リクエストされた URL に対応するファイルがファイルシステムに 無いという場合が発生します。これが起こるのにはいくつかの理由があります。場合によっては、ドキュメントを別の場所に移動した結果であることがあります。この場合は、クライアントにリソースの新しい位置を知らせるために URL リダイレクションを使うのが最善の方法です。そうすることによって、リソースは新しい位置に移動しているけれども、古いブックマークやリンクが動作し続けるようにすることができます。

“File Not Found” エラーのもう一つのよくある理由は、ブラウザへの直接入力や HTML リンクからの偶発的な URL の入力間違いです。Apache はこの問題を改善するために、`mod_spelling` モジュール (意図的な綴り間違い) (訳注: 正しくは `spelling`) を提供しています。このモジュールが使用されているときは、“File Not Found” エラーを横取りして、似たファイル名のリソースを探します。もし一つだけ見つかった場合は `mod_spelling` はクライアントに正しい位置を知らせるために HTTP リダイレクトを送ります。もし複数の「近い」ファイルが見つかった場合は、それら 代替となりえるもののリストがクライアントに表示されます。

`mod_spelling` の非常に有用な機能は、大文字小文字を区別せずに ファイル名を比較するものです。これは URL と unix の ファイルシステムが両方とも大文字小文字を区別するものである、ということを知ることができないシステムで役に立ちます。ただし、時折の URL 訂正程度で済まず、`mod_spelling` をより多く使用すると、サーバにさらなる負荷がかかります。すべての「正しくない」リクエストの後に URL のリダイレクトとクライアントからの新しいリクエストがくること になりますから。

コンテンツの位置を決めようとするすべての試みが失敗すると、Apache は、HTTP ステータスコード 404 (file not found) と共に エラーページを返します。このエラーページの外観は `ErrorDocument` ディレクティブで制御され、カスタムエラーレスポンス³ と国際化

URL からファイルシステム上の位置へのマップ

サーバエラーレスポンス⁴ で説明されているように、柔軟な設定を行なうことができます。

URI References

- [1] <http://httpd.apache.org/docs-2.1/vhosts/>
- [2] <http://httpd.apache.org/docs-2.1/misc/rewriteguide.html>
- [3] <http://httpd.apache.org/docs-2.1/custom-error.html>
- [4] http://httpd.apache.org/docs-2.1/misc/custom_errordocs.html