

Apache モジュール mod_negotiation

| | |
|----------|------------------------------------|
| 説明: | コンテンツネゴシエーション ¹ 機能を提供する |
| ステータス: | Base |
| モジュール識別子 | negotiation_module |
| : | |
| ソースファイル: | mod_negotiation.c |

概要

コンテンツネゴシエーション、より正確にはコンテンツの選択機能は、複数用意されているドキュメントから、クライアントの能力に一番合ったドキュメントを選択する機能です。この実装は二つあります。

- タイプマップ (type-map ハンドラで扱われるファイル)。これは variants を含んでいるファイルを明示的に指定します。
- MultiViews の探索 (MultiViews Option で使用するようになります)。サーバが暗黙の内にファイル名のパターンマッチを行ない、その結果から選択します。

トピック

| | |
|---------------------|---|
| タイプマップ..... | 1 |
| MultiViews..... | 3 |
| URI References..... | 5 |

ディレクティブ

| | |
|----------------------------|---|
| CacheNegotiatedDocs..... | 3 |
| ForceLanguagePriority..... | 3 |
| LanguagePriority..... | 4 |

参照

- [DefaultLanguage](#)
- [AddEncoding](#)
- [AddLanguage](#)
- [AddType](#)
- [MultiViewsMatch](#)

タイプマップ

タイプマップは RFC 822 のメールヘッダと同じ書式です。ドキュメントの記述が空行で分離されて書かれていて、ハッシュ文字 ('#') で始まる行はコメントとして扱われます。ドキュメントの説明は複数のヘッダレコードから構成されます。レコードは、続きの行が空白で始まっていると複数の行にまたがります。最初の空白が消去されて、前の行とつなげて 1 行として扱われます。ヘッダレコードはキーワード名の後に値が続くという形式で、キーワード名は常にコロンで終わります。空白はヘッダ名と値の間、値のトークンの間に入れることができます。使用可能なヘッダは以下のとおりです:

Content-Encoding:

ファイルのエンコーディング。Apache は [AddEncoding](#) ディレクティブ で定義されたエンコーディングだけを認識します。通常 compress されたファイルのための

Apache モジュール mod_negotiation

x-compress と gzip されたファイルのための x-gzip を含みます。エンコーディングの比較をするときは、接頭辞 x- は無視されます。

Content-Language:

インターネット標準の言語タグ (RFC 1766) で定義されている言語の種類。例えば、en は英語を表します。

Content-Length:

ファイルの長さ (バイト数)。このヘッダがない場合、ファイルの実際の長さが使用されます。

Content-Type:

ドキュメントの MIME メディアタイプ、オプションなパラメータ付き。パラメータの構文は name=value で、メディアタイプや他のパラメータとはセミコロンで分離されます。共通のパラメータは以下のとおり:

level

メディアタイプのバージョンを示す整数。text/html では 2 がデフォルトで、その他の場合は 0 がデフォルトです。

qs

クライアントの能力に関係なく、variant を他と比較したときの相対的な「品質」で、0.0 から 1.0 の範囲の浮動点小数。例えば、写真を表現しようとしているときは普通は JPEG ファイルの方が ASCII ファイルよりも高い品質になります。しかし、リソースが ASCII アートで表現されているときは、ASCII ファイルの方が JPEG ファイルよりも高い品質になります。このように、qs はリソース毎に特有の値を取ります。

例:

```
Content-Type: image/jpeg; qs=0.8
```

URI:

(指定のメディアタイプ、コンテンツエンコーディングの) variant のファイルの uri。これは、マップファイルからの相対 URL として解釈されます。同じサーバに存在しなければならず、クライアントが直接リクエストしたときにアクセスを許可されるものでなければなりません。

Body:

Apache 2.0 で新設されたこの Body ヘッダを使って、リソースの実際の内容をタイプマップファイルに書くことができます。このヘッダは本文の内容の区切りとなる文字列で始まる必要があります。タイプマップファイルの続く行は、区切り文字列が見つかるまで、リソースの本文になります。

例:

```
Body:----xyz----
<html>
<body>
<p>Content of the page.</p>
</body>
</html>
----xyz----
```

MultiViews

MultiViews 探索は、Multiviews Options ディレクティブにより有効になります。サーバが /some/dir/foo へのリクエストを受け取り、/some/dir/foo が存在 しない場合、サーバはディレクトリを読んで、foo.* にあてはまる全てのファイルを探し、事実上それらのファイルをマップするタイプマップを作ります。そのとき、メディアタイプとコンテンツエンコーディングは、そのファイル名を直接指定したときと同じものが割り当てられます。それからクライアントの要求に一番合うものを選び、そのドキュメントを返します。

CacheNegotiatedDocs ディレクティブ

| | |
|---------|-----------------------------------------------|
| 説明: | コンテンツネゴシエーションされたドキュメントをプロキシサーバが キャッシュできるようにする |
| 構文: | CacheNegotiatedDocs on off |
| デフォルト: | CacheNegotiatedDocs off |
| コンテキスト: | サーバ設定ファイル, バーチャルホスト |
| ステータス: | Base |
| モジュール: | mod_negotiation |
| 互換性: | バージョン 2.0で構文が変わりました |

このディレクティブが設定されていると、コンテンツネゴシエーション をした結果のドキュメントのキャッシュを許可します。これは、プロキシの後ろにいるクライアントが能力に一番合った ドキュメントではなく、キャッシュをより効果的にするものを得る可能性があるということです。

このディレクティブは HTTP/1.0 ブラウザからのリクエスト のみに適用されます。HTTP/1.1 は、交渉されたドキュメントのキャッシュに対してずっとよい制御が可能なので、このディレクティブは HTTP/1.1 のリクエストには影響しません。

2.0 より前のバージョンでは、CacheNegotiatedDocs は引数を取らず、ディレクティブが存在することで on の動作をしていました。

ForceLanguagePriority ディレクティブ

| | |
|---------|--------------------------------------------------------------|
| 説明: | 要求に合う単独のドキュメントが見つからなかったときに行なうことを指定 |
| 構文: | ForceLanguagePriority None Prefer Fallback [Prefer Fallback] |
| デフォルト: | ForceLanguagePriority Prefer |
| コンテキスト: | サーバ設定ファイル, バーチャルホスト, ディレクトリ, .htaccess |
| 上書き: | FileInfo |
| ステータス: | Base |
| モジュール: | mod_negotiation |
| 互換性: | バージョン 2.0.30 以降で使用可能 |

ForceLanguagePriority ディレクティブは 要求に合うドキュメントを一つだけ返すことがで

きないときに、`LanguagePriority` ディレクティブを使ってネゴシエーションの結果を返します。

`ForceLanguagePriority Prefer` は、同等の選択肢がいくつかあるときに、HTTP の 300 (MULTIPLE CHOICES) を返す代わりに、`LanguagePriority` を使って一つだけドキュメントを返すようにします。以下のディレクティブが指定されていて、ユーザの `Accept-Language` ヘッダでは `en` と `de` の品質が共に `.500` (同じくらい許容) であるときは、最初にマッチする variant の `en` が送られます。

```
LanguagePriority en fr de
ForceLanguagePriority Prefer
```

`ForceLanguagePriority Fallback` は、HTTP 406 (NOT ACCEPTABLE) の代わりにドキュメントを送ります。以下のディレクティブが指定されていて、ユーザの `Accept-Language` は `es` 言語のみを許可していて、さらにそのような variant がないときには、以下の `LanguagePriority` のリストの最初の variant が送れます。

```
LanguagePriority en fr de
ForceLanguagePriority Fallback
```

`Prefer` と `Fallback` の両方のオプションを同時に指定することができます。ですから、複数の variant があるときは `LanguagePriority` の最初の variant が送られ、クライアントの許容言語に合う variant がないときは存在するドキュメントで最初のもものが送られる、という様にすることができます。

LanguagePriority ディレクティブ

| | |
|---------|---------------------------------------------------------|
| 説明: | クライアントが優先度を示さなかったときの言語の variant の優先度を指定 |
| 構文: | <code>LanguagePriority MIME-lang [MIME-lang] ...</code> |
| コンテキスト: | サーバ設定ファイル, バーチャルホスト, ディレクトリ, <code>.htaccess</code> |
| 上書き: | <code>FileInfo</code> |
| ステータス: | <code>Base</code> |
| モジュール: | <code>mod_negotiation</code> |

`LanguagePriority` は、`MultiViews` リクエストを扱うときに、クライアントが優先順位を提供していない場合の言語の優先順位を設定します。`MIME-lang` のリストが優先度の降順に並びます。例:

```
LanguagePriority en fr de
```

`foo.html` がリクエストされ、`foo.html.fr` と `foo.html.de` が両方存在し、ブラウザが言語の優先順位を提供してない場合は `foo.html.fr` が返されます。

このディレクティブは他の方法で「最善」の言語が決定できないときか、`ForceLanguagePriority` ディレクティブが `None` 以外のときにのみ効果があることに注意してください。HTTP/1.1 リクエストが正しく実装されている場合には、このディレクティブ

は無効になります。

URI References

- [1] <http://httpd.apache.org/docs-2.1/content-negotiation.html>