Serving the right language via content negotiation

The problem Statement: I want the server to deliver the right document (variant) based on the browser's language preferences. Browsers send an Accept-Language header which lists their preferred language(s).

Apache

Before we configure our language negotiation parameters we need to setup things on the server. You can set this on the default `httpd.conf` configuration file (meaning that it'll work in all virtual hosts) or you can set in in each virtual host configuration (meaning it'll only work for that virtual host)

The two items are:

- `LanguagePriority` allows you the language precedence during content negotiation. If more than one version of a file is availab
- `ForceLanguagePriority` allows you to serve a result page rather than MULTIPLE CHOICES (Prefer) in case of a tie or NOT ACCEPTABLE (Fallback) in case no accepted languages matched the available variants.

```
<IfModule mod_negotiation.c>

LanguagePriority en-US en ca cs de el es pt pt-BR zh-CN zh-TW

ForceLanguagePriority Prefer Fallback

</IfModule>
```

MultiViews

MultiViews works as follows: if you request a resource named foo, and foo does not exist in the directory, Apache will search for all files named `foo.*`.

If we have this and the client's browser is configured with German as the primary language then the server will send `index.html.de` as the first option.

```
root@server1:~# ls -la /var/www/
total 20
drwxr-xr-x  2 root root 4096 Jul 20 00:13 .
drwxr-xr-x 14 root root 4096 Feb 14 18:43 ..
-rw-r--r--  1 root root  196 Jul 20 00:06 index.html.de
-rw-r--r--  1 root root  186 Jul 20 00:03 index.html.en
-rw-r--r--  1 root root  207 Jul 20 00:09 index.html.fr
root@server1:~#
```

MultiViews is a per-directory setting, i.e., it has to be set with an Options directive in a <Directory>, <Location>, or <Files> section in the Apache configuration. It has to be set explicitly with Options MultiViews. Using Options All does not set it.

The documentation for options lists valid values you can use in the directive.

```
<Directory /var/www/>
  Options Indexes FollowSymLinks MultiViews
  DirectoryIndex index.html
    AllowOverride None
    Order allow,deny
    allow from all
</Directory>
```

That's basically all we need for content negotiation. Using Language Priority, Force Language Priority and multiviews allows Apache to handle multiple versions of a page without us having to do anything on the client other than write the different versions of the page.

Nginx

The code is taken from this serverfault answer.

If the browser sends headers like this one: accept-language: en,en-US;q=0.8,ja;q=0.6 we can configure the server to handle content negotiation with the following block.

The configuration does the following:

1. Sets the variable $first_language to the value of the accept language header
2. Matches the string to the fir comma. This can be just the top level language code (en), the language and country code (en-US) or the language plus qualifier (en-US;q=0.8)
3. Set the default suffix to a language. In this case, English (en)
4. If the first language is Japanese (ja) then set the language suffix to Japanese

This will make the server render the page with the given language suffix.

```
set $first_language $http_accept_language; #1
if ($http_accept_language ~* '^(.+?),') {
    set $first_language $1;
}

s'^(.+?),'age_suffix 'en';
if ($first_language ~* 'ja') {
    set $language_suffix 'ja';
}
```

One shortcoming of this approach is that we need to have a list of our supported languages in hand as we prepare this configuration and every time we add a language we have to add if statements to the later block.

    The nginx_accept_language_module module parses the Accept-Language field in HTTP headers and chooses the most suitable locale for the user from the list of locales supported at your website.