Setup Kafka Website on Local Apache Server

This page provides a step-by-step guide to set up the contents of the Kafka website (https://kafka.apache.org/) as your local Apache HTTP Server's homepage. This is useful when you are contributing documentation changes, for which you need verify that any modified pages of yours still display and render correctly. Note that the `start-preview.sh` script does what is described here within a Docker container: **Most people will find `start-preview.sh` easier and more convenient than following this guide.** This guide is based on macOS, but should be easily adopted for other operating systems like Linux/Unix.

Table of Contents

---

1. Start the Apache HTTPD service

On macOS, the Apache HTTP Service should be installed but disabled by default. You can start the webserver with the command:

Start the webserver

```
$ sudo apachectl start
```

(If Apache HTTP Service is *not* installed, follow this page to download and install it.)

Now, verify your local HTTPD installation by accessing its default web page: In your browser, visit http://localhost to verify that the homepage renders successfully. By default, you should see the text "It Works!" from `/Library/WebServer/Documents/index.html.en`.

2. Make sure you have a local fork of the kafka-site git repository

Make sure you have cloned your forked kafka-site git repo (e.g. https://github.com/guozhang/kafka-site) from the official Apache git repo (https://github.com/apache/kafka-site) in your local directory. Then add the apache git repot as an additional remote repo.

For example, after these steps your local repo should look similar to:

Verify your local fork

```
$ pwd
/Users/guozhang/git/kafka-site

$ git remote -v
apache        https://github.com/apache/kafka-site.git (fetch)
apache        https://github.com/apache/kafka-site.git (push)
origin        https://github.com/guozhangwang/kafka-site.git (fetch)
origin        https://github.com/guozhangwang/kafka-site.git (push)

$ git checkout asf-site

$ git branch
* asf-site
```

3. Modify the Apache HTTPD configuration to point to your local fork

Now you need to edit the HTTPD config file and make the following changes.

Edit the webserver configuration

```
# The location of the configuration file may vary depending on the OS.
$ sudo vim /etc/apache2/httpd.conf
```

Step 1: Uncomment the following two lines so the "mod_include" and "mod_rewrite" modules are loaded on server startup.

```
LoadModule include_module libexec/apache2/mod_include.so
...
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
```

Step 2: Change `DocumentRoot` and `Directory` to point to the local clone of your forked kafka-site repo directory (e.g., "/Users/guozhang/git/kafka-site"). Also, change AllowOverride from "None" to "All". The three changed lines are marked below.

```
# File: /etc/apache2/httpd.conf
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/Users/guozhang/git/kafka-site"             # <<< THIS LINE IS CHANGED
<Directory "/Users/guozhang/git/kafka-site">              # <<< THIS LINE IS CHANGED

    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options FollowSymLinks Multiviews
    MultiviewsMatch Any

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   AllowOverride FileInfo AuthConfig Limit
    #
    AllowOverride All                                     # <<< THIS LINE IS CHANGED

    # Controls who can get stuff from this server.
    #
    Require all granted

</Directory>
```

4. Reload the Apache HTTPD configuration

Restart the webserver to make the config changes take effect:

Restart the webserver

```
$ sudo apachectl restart

# If the command above does not work (and you still see the default "It works!" page), try the following
variant.
$ sudo apachectl -k restart
```

In your browser, visit http://localhost/ again and refresh the page (Cmd+R on macOS). You should then see the complete Apache Kafka website, rendered from your local fork.

From now on, whenever you edited files in your local kafka-site fork, you should normally only need to refresh our browser (Cmd+R). If this is not sufficient, you can repeat the step above to restart the webserver.

Important Note: Your documentation changes should most likely be made in the main Apache Kafka code repository (https://github.com/apache/kafka) instead of the `kafka-site` repository, and in order to publish these changes to the Apache Kafka website (https://kafka.apache.org/) you should copy the modified documentation to the `kafka-site` repository into the respective release folders.

5. Shutdown the Apache HTTPD service

If you want to shutdown the web server, run the command:

Stop the webserver

```
$ sudo apachectl stop

# If the command above does not work (and the webserver is still running), try the following variant.
$ sudo apachectl -k stop
```