NAME
>   CURLOPT_HTTPPOST – specify the multipart formpost content

SYNOPSIS
>   #include <curl/curl.h>
>
>   CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HTTPPOST,
>   struct curl_httppost *formpost);

DESCRIPTION
>   Tells libcurl you want a multipart/formdata HTTP POST to be made and you instruct what data to pass on
>   to the server in the *formpost* argument. Pass a pointer to a linked list of curl_httppost structs as parameter.
>   The easiest way to create such a list, is to use *curl_formadd(3)* as documented. The data in this list must
>   remain intact until you close this curl handle again with *curl_easy_cleanup(3)*.
>
>   Using POST with HTTP 1.1 implies the use of a "Expect: 100-continue" header. You can disable this
>   header with *CURLOPT_HTTPHEADER(3)*.
>
>   When setting *CURLOPT_HTTPPOST(3)*, it will automatically set *CURLOPT_NOBODY(3)* to 0.

DEFAULT
>   NULL

PROTOCOLS
>   HTTP

EXAMPLE
>   /* Fill in the file upload field. This makes libcurl load data from
>   the given file name when curl_easy_perform() is called. */
>   curl_formadd(&formpost,
>   &lastptr,
>   CURLFORM_COPYNAME, "sendfile",
>   CURLFORM_FILE, "postit2.c",
>   CURLFORM_END);
>
>   /* Fill in the filename field */
>   curl_formadd(&formpost,
>   &lastptr,
>   CURLFORM_COPYNAME, "filename",
>   CURLFORM_COPYCONTENTS, "postit2.c",
>   CURLFORM_END);
>
>   /* Fill in the submit field too, even if this is rarely needed */
>   curl_formadd(&formpost,
>   &lastptr,
>   CURLFORM_COPYNAME, "submit",
>   CURLFORM_COPYCONTENTS, "send",
>   CURLFORM_END);

AVAILABILITY
>   As long as HTTP is enabled

RETURN VALUE
>   Returns CURLE_OK if HTTP is enabled, and CURLE_UNKNOWN_OPTION if not.

SEE ALSO
>   **CURLOPT_POSTFIELDS**(3), **CURLOPT_POST**(3),