3 способа загрузки файла на сервер с помощью curl

Продемонстрируем curl запрос на примере, с учетом следующих ограничений.  
1. Файл надо загрузить методом POST  
2. Необходима авторизация  
3. На сервере установлен сертификат ssl

Способ 1, командная строка: Загрузка файла непосредственно через командную строку выглядит так:

curl -i -u user:password -X POST --form datafile=@/path/to/some/file.txt https://localhost:443/page --insecure

Разберем подробнее:  
-X — указываем метод POST  
-u — логин и пароль пользователя  
—insecure — при работе по https часто приходится тестировать в тестовом окружении где не стоит проверенный сертификат. Этой опцией отказываемся от проверки.  
Чтобы ещё более подробно узнать о тонкостях команды, используйте curl —help

Способ 2, PHP: Продемонстрируем как загрузить файл

// очень полезно предварительно сделать проверку на наличие библиотеки curl для php

if (!in\_array ('curl', get\_loaded\_extensions())) {

exit("CURL is NOT installed on this server");

}

$ch = curl\_init();

//указываем url адрес на сервере

curl\_setopt($ch, CURLOPT\_URL, 'https://localhost:443/page');

// чтобы возвращало

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

// указываем метод POST

curl\_setopt($ch, CURLOPT\_POST, 1);

// этот параметр для указания логина и пароля

curl\_setopt($ch, CURLOPT\_USERPWD, "login:password");

curl\_setopt($ch, CURLOPT\_POSTFIELDS, [

'some\_more\_data' => 'здесь можно передать дополнительные параметры, например если эмулируется отправка формы',

'datafile' => curl\_file\_create('/path/to/file.txt' , mime\_content\_type($path\_to\_file), basename('/path/to/file'))

]

);

// проверка peer для ssl отключена, смотрите также CURLOPT\_SSL\_VERIFYHOST для полного игнорирования ssl сертификата

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, 0);

// все опции выставлены - выполняем запрос

$result = curl\_exec($ch);

// распечатываем ответ сервера

var\_dump($result);

// закрываем соединение

curl\_close($ch);

Практически всё прокомментировано в коде, однако отмечу функцию curl\_file\_create , один сервер постоянно ругался что файл как-то не так указан или отсутсвует. Это вылечилось добавлением мим типа, т.е. функция curl\_file\_create создает дополнительные заголовки для корректной отправки файла.

Способ 3, Python : Наконец дошли до python, здесь воспользуемя отличной библиотекой requests

import requests

auth = ('user', 'password')

data = [

('some\_more\_field', 'some\_field\_data')

]

# curl request

files = {'datafile': open('/path/to/file.txt', 'rb')}

r = requests.post('https://localhost:443/page', files=files, data=data, auth=auth)

print(r.text)

В указанном примере открываем файл в режиме бинарного чтения, и дальше сразу делаем запроc. В запросе указаны параметры авторизации, метод, страница и файл. Если файлов несколько — просто расширьте словарь files.  
Всё просто, удачи с curl!