

## cURL 本家

- <http://curl.haxx.se/>

### multipart/form-data での POST 送信

```
curl -v --proxy proxy.example.com:8080 --user username:password --form 'file=@upload.csv' https://service.example.com/service/path
```

オプション	説明
-v	異常がないか表示情報を増加
--proxy	proxy.example.com:8080 プロキシサーバの指定
--user username:password	HTTP ベーシック認証で指定するユーザー ID とパスワード
--form 'file=@upload.csv'	POST multipart/form-data + 送信パラメータ指定
<a href="https://service.example.com/service/path">https://service.example.com/service/path</a>	送信先 URL

## 参考サイト

curl(1) で POST する際の --data と --form の違いについて / 刺身 ブーメランのはてなダイアリー

- <http://d.hatena.ne.jp/a666666/20110427/1303838381>

--data (-d, --data-ascii)

application/x-www-form-urlencoded 形式で POST する。

@/path/to/file のように value の先頭が @ ではじまっているとファイルを読み込んで改行文字を取り除く。

パラメータや @ つきで指定したファイルの中身はすべて URL エンコードされていることが期待される。

つまり curl(1) は URL エンコードしてくれない。

-d を複数回指定するとすべてのパラメータが & で連結される。

@ でファイルを指定する場合、-d 'file=@sale.txt' のようにすると中身が展開されないので注意 (file=@sale.txt という文字列が渡される)

--form (-F)

multipart/form-data 形式で POST する。

key=value 形式で指定する。

@ を value の先頭に置くとファイルのデータを「添付して」送る。

フォームでファイルアップロードした場合と同じ。

boundary とかも面倒みてくれる。

< を value の先頭に置くとファイルのデータを読み込んで「テキストフィールドの値として」送る。