CS540 Library Proposal: Automatic cURL command generator

Bhuvana Sridhara

February 2021

1 Introduction

cURL is a command-line tool for getting or sending data including files using URL syntax.

2 The Problem

While working with API's, it is a little cumbersome to work directly with cURL, since some cURL requests can become complicated:

```
curl --location --request POST 'https://<REDACTED>/example/createExample?id=A2YJAFBMMDWYHO'
--header 'Authorization: Bearer 783545gfbsdhcfs8f6t378ry9347568345gbhvrdkslfsl
]cqt,mrtynrt89q4r0]bfuhgwyufgqwuer763485ka%^&Dbiowe8cht67a6%^*(4
g^hbjIUYT&^%bh&*^ghj876HGJ8967yGUT^tg*(9ukj&*hu&GUIIUui'
--header 'Content-Type: application/json'
--data-raw '{
"user": {
"imei": "user",
"name": "AAAAAAAAAA",
"type": "AAAAAA"
}
}
'|json_pp
```

Adding or removing parameters, finding and updating certain values, balancing the quotes and the brackets proves to be tricky.

3 Importance

- cURL is the most used command line tool for making API calls.
- cURL is also the main tool for describing API calls in the documentation

- cURL is the easiest way to share an API request.
- cURL is the most preferred way to test API's, API health, latency etc.

  An easier way to build a cURL command would benefit developers and testers alike.

4 Obstacles

There are non-programmatic ways to generate cURL commands from native requests. cURL provides a lot of options that have to be supported depending on the context from which it is being called.

5 Existing solutions

As of now, developers have the following ways to generate a cURL command:

- Build the cURL command manually: This approach might work for very simple requests. If the request involves simply typing curl at the command line, followed by the URL of the output to retrieve, with no headers, authorization, or body, this approach might work. For example: curl www.google.com But this is rarely the case. From the developer's point of view, if the API they just built contains multiple headers and numerous parameters, this approach is not the best way to build a corresponding cURL command as it is prone to errors and takes up a lot of valuable time.
- Online curl command line builder: There are websites that convert native code to cURL syntax.

  Example: https://curlbuilder.com, https://tools.w3cub.com/curl-builder, etc.

  The shortcomings of these converters are:

  - They do not provide a provision for the user to add 'options' which is a very powerful feature of the cURL request
  - The loss of a developer's concentration and time from leaving an IDE to look up a tool and provide is not insignificant.
- Postman: One can construct a request in Postman and convert it to cURL using the code snippet generator. Postman also offers a wide variety of other features and if you're not utilizing all of them, you may not want to create an account, install a Desktop app or a Web Browser app just to build cURL commands.
- cURL builder (PHP): https://github.com/alexkart/curl-builder This is a curl command generator that can be included in any PHP project which

can generate cURL commands automatically from PSR-7 server requests and manually by specifying options and URL. This implementation is specific to PHP.

6 Proposal

The proposed library builds up on the curl builder library described in the above section.

- This library can be included in any project and can generate cURL commands automatically from HTTP requests.
- The user can also manually specify URL, headers, options etc if they wish and generate a cURL command manually.
- This library will be available for use with multiple languages and support C-Restsharp, Java-OkHttp, Python-Requests, Python-http.client, Go-Native, etc.
- Example HTTP GET request in Python using http client:

  ```
  import http.client
  req = http.client.HTTPSConnection("<REDACTED>")
  payload = 'imei=xxx'
  headers = {
    'Authorization': 'Basic XXXX',
    'Content-Type': 'application/x-www-form-urlencoded'
  }
  req.request("GET", "/", payload, headers)
  res = req.getresponse()
  ```

  Generating the cURL command would be as easy as adding the following lines of code:

  ```
  import library

  curlGen.setRequest(req)
  curl = curlGen.generateCurl()
  #curl --request GET 'https://<REDACTED>' \
  #--header 'Authorization: Basic XXXX' \
  #--header 'Content-Type: application/x-www-form-urlencoded' \
  #--data-urlencode 'imei=xxx'
  ```

- The library also offers the user an option to manually add headers, url and options like:

  ```
  curlGen.setHeader("Authorization: Basic XXXX")
  curlGen.setUrl("'http://example.com');
  curlGen.addOption("-vv")
  ```

7 Benefits

- Save developer time: This library will put an end to juggling between multiple tools or the use of a Desktop/Web application to generate a cURL command
- Save developer effort: Since the code for the API being written and the cURL command generated are together, the user doesn't have to re-type the same headers and body. It also doesn't break the developer's flow during programming.
- Flexible: Anyone who isn't familiar with cURL's syntax can still generate a cURL command programmatically. Even if someone wants to generate a cURL command manually, they can follow the intuitive methods provided by the library to add tokens, headers, url etc.
- Extendable: The library can be used with multiple programming languages and can be extended to accomodate many more in the future as required.