

C++ Requests



Curl for People

github.com/whoshuu/cpr

HTTP at a high level

- ❖ Verbs
 - ❖ GET me this page
 - ❖ PUT this data here
 - ❖ DELETE this user
- ❖ That's it

Isn't that naïve?

- ❖ Authentication
- ❖ Headers
- ❖ Cookies, sessions, proxies, etc.

HTTP should be as simple as a
print statement

- Kenneth Reitz

Let's try it

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"});

    std::cout << r.text << std::endl;
}
```

Compare this to

```
#include <curl/curl.h>
#include <iostream>
#include <string>

size_t write(void* ptr, size_t size, size_t nmemb, std::string* data) {
    data->append((char*) ptr, size * nmemb);
    return size * nmemb;
}

int main(int argc, char** argv) {
    auto curl = curl_easy_init();
    curl_easy_setopt(curl, CURLOPT_URL, "https://httpbin.org/get");
    curl_easy_setopt(curl, CURLOPT_NOPROGRESS, 1L);
    curl_easy_setopt(curl, CURLOPT_USERAGENT, "curl/7.42.0");
    curl_easy_setopt(curl, CURLOPT_TCP_KEEPALIVE, 1L);

    std::string response_text;
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write);
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response_text);

    curl_easy_perform(curl);
    curl_easy_cleanup(curl);
    curl = NULL;

    std::cout << response_text << std::endl;
}
```

And that's the easy interface

```
#include <Poco/Net/.....>
#include <iostream>
#include <string>

int main(int argc, char** argv) {
    Poco::URI uri("https://httpbin.org/get");
    Poco::Net::HTTPClientSession session(uri.getHost(), uri.getPort());
    std::string path(uri.getPathAndQuery());
    Poco::Net::HTTPRequest request(Poco::Net::HTTPRequest::HTTP_GET,
                                   path,
                                   Poco::Net::HTTPMessage::HTTP_1_1);
    session.sendRequest(request);

    Poco::Net::HTTPResponse response;
    std::istream& in_stream = session.receiveResponse(response);
    std::ostringstream out_stream;
    Poco::StreamCopier::copyStream(in_stream, out_stream);

    std::cout << out_stream.str() << std::endl;
}
```

Seven nouns participating in several APIs

```
uri          : getHost(), getPort(), getPathAndQuery()
session      : sendRequest(), receiveResponse()
path         : request constructor
request      : sendRequest()
response     : receiveResponse()
in_stream    : copy_stream()
out_stream   : copy_stream()
```

An HTTP API for People

- ❖ Objects participate in a single request verb
- ❖ Steps to response are minimal
- ❖ Is useful in 95% of use cases

Query parameters

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Parameters{{"key", "value"}});

    std::cout << r.text << std::endl;
}
```

Authentication

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Authentication{"user", "pass"});

    std::cout << r.text << std::endl;
}
```

Digest Authentication

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"} ,
                      Digest{"user", "pass"});

    std::cout << r.text << std::endl;
}
```

Custom headers

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Headers{{"Accept", "test/html"}});

    std::cout << r.text << std::endl;
}
```

Url encoded POSTs

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Post(Url{"https://httpbin.org/post"}, 
                       Payload{{"key", "value"}, 
                               {"hello", "world"}});

    std::cout << r.text << std::endl;
}
```

Multipart POSTs

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Post(Url{"https://httpbin.org/post"},
                       Multipart{{"key", "long-string"}});

    std::cout << r.text << std::endl;
}
```

File POSTs

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Post(Url{"https://httpbin.org/post"},
                       Multipart{{"key", "long-string"}, {"name", File{"path"}}});
    std::cout << r.text << std::endl;
}
```

More complex requests

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/post"} ,
                      Authentication{"user", "pass"} ,
                      Headers{{"Accept", "test/html"}},
                      Parameters{{"key", "value"}});

    std::cout << r.text << std::endl;
}
```

And with some template magic...

Orderless arguments

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Authentication{"user", "pass"},
                      Headers{{"Accept", "test/html"}},
                      Parameters{{"key", "value"}));

    std::cout << r.text << std::endl;
}
```

Orderless arguments

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Headers{{"Accept", "test/html"}},
                      Authentication{"user", "pass"},
                      Parameters{{"key", "value"}});

    std::cout << r.text << std::endl;
}
```

Orderless arguments

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Headers{{"Accept", "test/html"}},
                      Parameters{{"key", "value"}},
                      Authentication{"user", "pass});

    std::cout << r.text << std::endl;
}
```

Orderless arguments

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"},
                      Authentication{"user", "pass"},
                      Parameters{{"key", "value"}},
                      Headers{{"Accept", "text/html"}});

    std::cout << r.text << std::endl;
}
```

An API is a rung on an abstraction ladder

Sessions

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    Session session;
    session.SetUrl("https://httpbin.org/get");
    session.SetParameters({{"key", "value"}});
    auto r = session.Get();

    std::cout << r.text << std::endl;
}
```

Asynchronous requests

```
#include <cpr.h>
#include <iostream>
#include <future>
#include <vector>

int main(int argc, char** argv) {
    std::vector<std::future<Response>> responses;
    auto url = Url{"https://httpbin.org/get"};
    for (int i = 0; i < 1000; ++i) {
        responses.emplace_back(cpr::GetAsync(url));
    }
    for (auto& r : responses) {
        std::cout << r.get().text << std::endl;
    }
}
```

Asynchronous callbacks

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto future = cpr::GetCallback([](Response r) {
        return r.text;
    }, Url{"https://httpbin.org/get"});

    std::cout << future.get() << std::endl;
}
```

```
#include <cpr.h>
#include <iostream>

int main(int argc, char** argv) {
    auto r = cpr::Get(Url{"https://httpbin.org/get"});

    std::cout << r.text << std::endl;
}
```

Developers are People too

This is an API for People

C++ Requests



Curl for People

github.com/whoshuu/cpr

Thanks.