My Project

Generated by Doxygen 1.9.8

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 curling::Request Class Reference

Handles HTTP requests using libcurl.

`#include <curling.hpp>`

Public Types

- enum class Method { GET , POST , PUT , DELETE }

  Enumerates the supported HTTP methods.

Public Member Functions

- Request ()

  Constructor for the Request class.
- ~Request ()
- void setMethod (Method m)

  Sets the HTTP method for the request.
- void setURL (const std::string &URL)

  Sets the URL for the request.
- void addArg (const std::string &arg)
- void addHeader (const std::string &header)
- void setBody (const std::string &body)
- void **send** ()

  Sends the HTTP request using libcurl.
- const std::string & getResponse () const

  Retrieves the response body from the last sent request.
- const std::map< std::string, std::string > & getResponseHeadersMap () const
- void reset ()

Static Public Attributes

- static std::atomic< int > instances = 0

3.1.1 Detailed Description

Handles HTTP requests using libcurl.

This class provides functionality to perform HTTP operations such as GET, POST, PUT, and DELETE. It manages the setup and execution of these requests with libcurl.

3.1.2 Member Enumeration Documentation

3.1.2.1 Method

```
enum class curling::Request::Method  [strong]
```

Enumerates the supported HTTP methods.

Enumerator

GET	Represents an HTTP GET request.
POST	Represents an HTTP POST request.
PUT	Represents an HTTP PUT request.
DELETE	Represents an HTTP DELETE request.

3.1.3 Constructor & Destructor Documentation

3.1.3.1 Request()

```
curling::Request::Request ( )
```

Constructor for the Request class.

Initializes a new instance of the Request class and increments the instances counter.

3.1.3.2 ~Request()

```
curling::Request::~Request ( )
```

```
   @brief Destructor for the Request class.

   Decrements the instances counter and performs any necessary
```

cleanup.

3.1.4 Member Function Documentation

3.1.4.1 addArg()

```
void curling::Request::addArg (
            const std::string & arg )
```

```
   @brief Adds an argument to the query string of the request.

   @param arg The key-value pair formatted as a string (e.g.,
```

"key=value").

3.1.4.2 addHeader()

```
void curling::Request::addHeader (
            const std::string & header )
```

```
   @brief Adds a header to the request.

   @param header The header string to be added (e.g., "Content-Type:
```

text/html").

3.1.4.3 getResponse()

```
const std::string & curling::Request::getResponse ( ) const
```

Retrieves the response body from the last sent request.

Returns

A constant reference to the response string.

3.1.4.4 getResponseHeadersMap()

```
const std::map< std::string, std::string > & curling::Request::getResponseHeadersMap ( ) const
```

```
   @brief Retrieves a map of response headers from the last sent
```

request.

```
   @return A constant reference to the map containing response
```

headers.

3.1.4.5 reset()

```
void curling::Request::reset ( )
```

```
   @brief Resets the internal state for reuse of this Request
```

instance.

3.1.4.6 setBody()

```
void curling::Request::setBody (
            const std::string & body )
```

```
   @brief Sets the body of the HTTP request, applicable for POST and
```

PUT methods.

```
   @param body The body content as a string.
```

3.1.4.7 setMethod()

```
void curling::Request::setMethod (
            Method m )
```

Sets the HTTP method for the request.

Parameters

m	The method to set (e.g., Method::GET, Method::POST).

3.1.4.8 setURL()

```
void curling::Request::setURL (
            const std::string & URL )
```

Sets the URL for the request.

Parameters

URL	The URL string to be used in the HTTP request.

3.1.5 Member Data Documentation

3.1.5.1 instances

```
std::atomic<int> curling::Request::instances = 0  [inline], [static]
```

```
   @var instances
   @brief Static atomic counter to track the number of Request
```

instances.

The documentation for this class was generated from the following file:

- include/curling.hpp

Chapter 4

File Documentation

4.1 curling.hpp

```
00001 #ifndef CURLING_HPP
00002 #define CURLING_HPP
00003
00004 #include <iostream>
00005 #include <string>
00006 #include <map>
00007 #include <sstream>
00008 #include <atomic>
00009 #include <curl/curl.h>
00010
00011 namespace curling {
00012
00022 class Request {
00023 public:
00028     enum class Method {
00029         GET,
00030         POST,
00031         PUT,
00032         DELETE
00033     };
00034
00040     inline static std::atomic<int> instances = 0;
00041
00048     Request();
00049
00056     ~Request();
00057
00063     void setMethod(Method m);
00064
00070     void setURL(const std::string& URL);
00071
00078     void addArg(const std::string& arg);
00079
00086     void addHeader(const std::string& header);
00087
00094     void setBody(const std::string& body);
00095
00099     void send();
00100
00106     const std::string& getResponse() const;
00107
00115     const std::map<std::string, std::string>& getResponseHeadersMap()
00116 const;
00117
00122     void reset();
00123
00124 private:
00125     Method method;
00126     CURL* curlHandle;
00127     struct curl_slist* list;
00128     std::string url, args, response, body, cookieFile, cookieJar;
00129     std::map<std::string, std::string> responseHeadersMap;
00130
00151     static size_t WriteCallback(void* contents, size_t size, size_t nmemb,
00152
00153         void* userp);
00154
00174     static size_t HeaderCallback(char* buffer, size_t size, size_t nitems,
```

```
00175
00176         void* userdata);
00177
00185     void clean();
00186
00194     void updateURL();
00195 };
00196
00197 } // namespace curling
00198
00199 #endif // CURLING_HPP
```

Index