

# Клиент службы HTTP

## 1.0

Создано системой Doxygen 1.9.1

|                                    |    |
|------------------------------------|----|
| 1 Иерархический список классов     | 1  |
| 1 Иерархический список классов     | 1  |
| 1.1 Иерархия классов . . . . .     | 1  |
| 2 Алфавитный указатель классов     | 1  |
| 2.1 Классы . . . . .               | 1  |
| 3 Список файлов                    | 2  |
| 3.1 Файлы . . . . .                | 2  |
| 4 Классы                           | 2  |
| 4.1 Класс Error . . . . .          | 2  |
| 4.1.1 Подробное описание . . . . . | 3  |
| 4.1.2 Конструктор(ы) . . . . .     | 3  |
| 4.2 Класс HTTPClient . . . . .     | 3  |
| 4.2.1 Подробное описание . . . . . | 4  |
| 4.2.2 Методы . . . . .             | 4  |
| 5 Файлы                            | 8  |
| 5.1 Файл Exception.h . . . . .     | 8  |
| 5.1.1 Подробное описание . . . . . | 9  |
| 5.2 Файл HTTPClient.h . . . . .    | 9  |
| 5.2.1 Подробное описание . . . . . | 10 |
| Предметный указатель               | 11 |

## 1 Иерархический список классов

### 1.1 Иерархия классов

Иерархия классов.

|                  |   |
|------------------|---|
| HTTPClient       | 3 |
| invalid_argument |   |
| Error            | 2 |

## 2 Алфавитный указатель классов

### 2.1 Классы

Классы с их кратким описанием.

|  |   |
|--|---|
| Error  |   |
| Класс для обработки ошибок                   | 2 |
| HTTPClient                                   |   |
| Класс, который реализует клиента службы HTTP | 3 |

## 3 Список файлов

### 3.1 Файлы

Полный список документированных файлов.

|                              |  |   |
|------------------------------|--|---|
| <a href="#">Exception.h</a>  | Описание класса <a href="#">Error</a>      | 8 |
| <a href="#">HTTPClient.h</a> | Описание класса <a href="#">HTTPClient</a> | 9 |

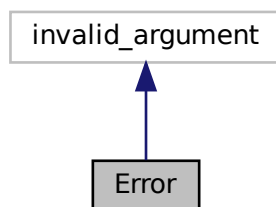
## 4 Классы

### 4.1 Класс Error

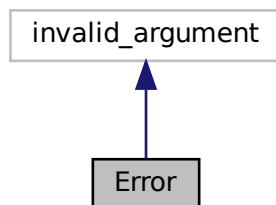
Класс для обработки ошибок

```
#include <Exception.h>
```

Граф наследования:Error:



Граф связей класса Error:



## Открытые члены

- [Error](#) ()=delete  
Запрещающий конструктор без параметров
- [Error](#) (const string what)  
Конструктор с параметром

## 4.1.1 Подробное описание

Класс для обработки ошибок

## 4.1.2 Конструктор(ы)

4.1.2.1 `Error()` `Error::Error (`  
`const string what )` [inline]

Конструктор с параметром

Аргументы

|                          |   |
|--------------------------|---|
| <code>std::string</code> | <code>what</code> - обычная строка, которая хранит описание ошибки. |
|--------------------------|---|

Данный конструктор перегружается вызовом конструктора базового класса с названием "invalid←  
\_argument"

Объявления и описания членов класса находятся в файле:

- [Exception.h](#)

## 4.2 Класс HTTPClient

Класс, который реализует клиента службы HTTP.

```
#include <HTTPClient.h>
```

## Открытые члены

- [HTTPClient](#) ()=default  
Конструктор по умолчанию
- [~HTTPClient](#) ()=default  
Деструктор по умолчанию
- `string` [get\\_server](#) ()  
Геттер для получения адреса сервера
- `string` [get\\_path](#) ()  
Геттер для получения пути файла на сервере

- void `set_url` (const string &url)  
Метод, выполняющий валидацию url-адреса и заполняющий атрибуты "server" и "path".
- void `set_file` (const string &f\_path)  
Метод, выполняющий валидацию файла, где должен сохраниться результат. Также метод заполняет атрибут "file\_path".
- int `request` ()  
Метод, выполняющий HTTP запрос и сохраняющий результат в файл, который был указан пользователем

#### Открытые статические члены

- static int `parse_headers` (const string &head\_buf\_str, int &http\_code, string &loc)  
Статический метод, выполняющий разбор заголовков

#### Закрытые члены

- void `connect` ()  
Устанавливает соединение с сервером
- void `save_to_file` ()  
Метод, записывающий полученные с сервера данные в файл, который был указан пользователем при запуске

#### Закрытые данные

- char `recv_buf` [1024 \* 4096]  
атрибут, хранящий буфер для приёма данных от сервера
- int `recv_count`  
атрибут, хранящий количество байт принятых данных в буфер "recv\_buf".
- string `server`  
атрибут, хранящий адрес сервера
- string `path`  
атрибут, хранящий путь к файлу на сервере
- string `file_path`  
атрибут, хранящий путь до файла, в котором будет сохранён результат
- int `client_sock`  
атрибут для сокета

#### 4.2.1 Подробное описание

Класс, который реализует клиента службы HTTP.

#### 4.2.2 Методы

##### 4.2.2.1 `connect()` void HTTPClient::connect ( ) [private]

Устанавливает соединение с сервером

## Исключения

|       |  |
|-------|--|
| Error | в случае ошибки в открытии сокета и в случае ошибки с установкой соединения с сервером |
|-------|--|

## 4.2.2.2 get\_path() string HTTPClient::get\_path ( )

Геттер для получения пути файла на сервере

Возвращает

std::string, в которой путь файла на сервере

## 4.2.2.3 get\_server() string HTTPClient::get\_server ( )

Геттер для получения адреса сервера

Возвращает

std::string, в которой хранится адрес сервера

4.2.2.4 parse\_headers() int HTTPClient::parse\_headers (   
const string & head\_buf\_str,   
int & http\_code,   
string & loc ) [static]

Статический метод, выполняющий разбор заголовков

Аргументы

|             |   |
|-------------|---|
| std::string | head_buf_str - строка с данными, полученная от сервера. |
| http_code   | - целочисленный http код ответа                         |
| std::string | loc -значение заголовка "Location"                      |

Возвращает

размер заголовков в байтах

#### 4.2.2.5 request() int HTTPClient::request ( )

Метод, выполняющий HTTP запрос и сохраняющий результат в файл, который был указан пользователем

## Исключения

|       |                                     |
|-------|-------------------------------------|
| Error | в случае ошибок в работе с сервером |
|-------|-------------------------------------|

## Возвращает

возвращает ответа сервера (http код)

## 4.2.2.6 save\_to\_file() void HTTPClient::save\_to\_file ( ) [private]

Метод, записывающий полученные с сервера данные в файл, который был указан пользователем при запуске

Записывает определённо количество байт из буфера "recv\_buf" в файл "file\_path". Данный процесс происходит циклично. Запись данных в файл происходит до тех пор, пока на сокете не закончатся данные

## Исключения

|       |   |
|-------|---|
| Error | в случае ошибки при записи данных файл и в случае ошибки в обмене данных с сервером |
|-------|---|

4.2.2.7 set\_file() void HTTPClient::set\_file (   
const string & f\_path )

Метод, выполняющий валидацию файла, где должен сохраниться результат. Также метод заполняет атрибут "file\_path".

## Аргументы

|             |  |
|-------------|--|
| std::string | f_path - путь до файла, который указал пользователь при запуске. |
|-------------|--|

## Исключения

|            |                              |
|------------|------------------------------|
| Error,если | файл не найден или не открыт |
|------------|------------------------------|

4.2.2.8 set\_url() void HTTPClient::set\_url (   
const string & url )

Метод, выполняющий валидацию url-адреса и заполняющий атрибуты "server" и "path".



## Аргументы

|                          |  |
|--------------------------|--|
| <code>std::string</code> | url - url-адрес, который указал пользователь при запуске |
|--------------------------|--|

## Исключения

|                           |                        |
|---------------------------|------------------------|
| <code>Error</code> , если | url-адрес некорректный |
|---------------------------|------------------------|

Объявления и описания членов классов находятся в файлах:

- [HTTPClient.h](#)
- [HTTPClient.cpp](#)

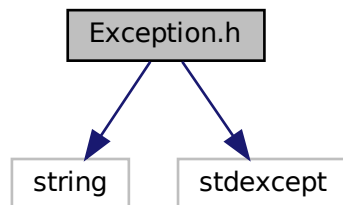
## 5 Файлы

### 5.1 Файл Exception.h

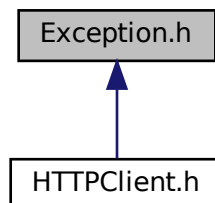
Описание класса [Error](#).

```
#include <string>
#include <stdexcept>
```

Граф включаемых заголовочных файлов для Exception.h:



Граф файлов, в которые включается этот файл:



## Классы

- class [Error](#)  
Класс для обработки ошибок

## 5.1.1 Подробное описание

Описание класса [Error](#).

Автор

Соколов Д.А.

Версия

1.0

Дата

12.05.21

Авторство

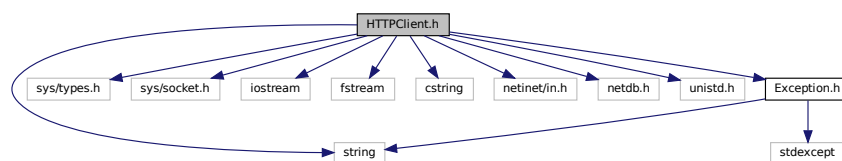
ИБСТ ПГУ

## 5.2 Файл HTTPClient.h

Описание класса [HTTPClient](#).

```
#include <string>
#include <sys/types.h>
#include <sys/socket.h>
#include <iostream>
#include <fstream>
#include <cstring>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>
#include "Exception.h"
```

Граф включаемых заголовочных файлов для HTTPClient.h:



## Классы

- class [HTTPClient](#)

Класс, который реализует клиента службы HTTP.

### 5.2.1 Подробное описание

Описание класса [HTTPClient](#).

#### Автор

Соколов Д.А.

#### Версия

1.0

#### Дата

12.05.21

#### Авторство

ИБСТ ПГУ

## Предметный указатель

- connect
  - [HTTPClient](#), [4](#)
- Error, [2](#)
  - Error, [3](#)
- Exception.h, [8](#)
- get\_path
  - [HTTPClient](#), [5](#)
- get\_server
  - [HTTPClient](#), [5](#)
- [HTTPClient](#), [3](#)
  - connect, [4](#)
  - get\_path, [5](#)
  - get\_server, [5](#)
  - parse\_headers, [5](#)
  - request, [5](#)
  - save\_to\_file, [7](#)
  - set\_file, [7](#)
  - set\_url, [7](#)
- [HTTPClient.h](#), [9](#)
- parse\_headers
  - [HTTPClient](#), [5](#)
- request
  - [HTTPClient](#), [5](#)
- save\_to\_file
  - [HTTPClient](#), [7](#)
- set\_file
  - [HTTPClient](#), [7](#)
- set\_url
  - [HTTPClient](#), [7](#)