

Swiftarm  
sig1

Generated by Doxygen 1.7.4

Fri Jun 8 2012 17:06:56



# Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Namespace Index</b>                        | <b>1</b> |
| 1.1      | Namespace List . . . . .                      | 1        |
| <b>2</b> | <b>Class Index</b>                            | <b>3</b> |
| 2.1      | Class List . . . . .                          | 3        |
| <b>3</b> | <b>File Index</b>                             | <b>5</b> |
| 3.1      | File List . . . . .                           | 5        |
| <b>4</b> | <b>Namespace Documentation</b>                | <b>7</b> |
| 4.1      | DownloadManager Namespace Reference . . . . . | 7        |
| 4.1.1    | Function Documentation . . . . .              | 8        |
| 4.1.1.1  | add . . . . .                                 | 8        |
| 4.1.1.2  | buildXML . . . . .                            | 8        |
| 4.1.1.3  | clearList . . . . .                           | 8        |
| 4.1.1.4  | dispatch . . . . .                            | 9        |
| 4.1.1.5  | downloadFirstInList . . . . .                 | 9        |
| 4.1.1.6  | getDownloadDirectory . . . . .                | 9        |
| 4.1.1.7  | getDownloads . . . . .                        | 9        |
| 4.1.1.8  | getIndexFromHash . . . . .                    | 9        |
| 4.1.1.9  | init . . . . .                                | 9        |
| 4.1.1.10 | pauseDownload . . . . .                       | 9        |
| 4.1.1.11 | removeFromDisk . . . . .                      | 10       |
| 4.1.1.12 | removeFromList . . . . .                      | 10       |
| 4.1.1.13 | resumeDownload . . . . .                      | 10       |
| 4.1.1.14 | setActiveDownload . . . . .                   | 10       |

|          |  |    |
|----------|--|----|
| 4.1.1.15 | <a href="#">setDownloadDirectory</a>             | 11 |
| 4.1.1.16 | <a href="#">startDownload</a>                    | 11 |
| 4.1.1.17 | <a href="#">startStream</a>                      | 11 |
| 4.1.1.18 | <a href="#">startStreamThread</a>                | 11 |
| 4.1.1.19 | <a href="#">stopStream</a>                       | 11 |
| 4.1.1.20 | <a href="#">switchDownload</a>                   | 12 |
| 4.1.1.21 | <a href="#">updateDownloadStatistics</a>         | 12 |
| 4.1.2    | <a href="#">Variable Documentation</a>           | 12 |
| 4.1.2.1  | <a href="#">active_download</a>                  | 12 |
| 4.1.2.2  | <a href="#">active_download_mutex</a>            | 12 |
| 4.1.2.3  | <a href="#">d_pid</a>                            | 12 |
| 4.1.2.4  | <a href="#">doc</a>                              | 12 |
| 4.1.2.5  | <a href="#">download_directory</a>               | 12 |
| 4.1.2.6  | <a href="#">downloaded</a>                       | 13 |
| 4.1.2.7  | <a href="#">downloads</a>                        | 13 |
| 4.1.2.8  | <a href="#">evcompl</a>                          | 13 |
| 4.1.2.9  | <a href="#">mutex</a>                            | 13 |
| 4.1.2.10 | <a href="#">streaming_thread</a>                 | 13 |
| 4.1.2.11 | <a href="#">thread</a>                           | 13 |
| 4.1.2.12 | <a href="#">uploaded</a>                         | 13 |
| 4.2      | <a href="#">HttpServer Namespace Reference</a>   | 13 |
| 4.2.1    | <a href="#">Function Documentation</a>           | 14 |
| 4.2.1.1  | <a href="#">handleRequest</a>                    | 14 |
| 4.2.1.2  | <a href="#">init</a>                             | 14 |
| 4.2.1.3  | <a href="#">sendResponse</a>                     | 14 |
| 4.2.1.4  | <a href="#">sendXMLResponse</a>                  | 14 |
| 4.2.2    | <a href="#">Variable Documentation</a>           | 15 |
| 4.2.2.1  | <a href="#">base</a>                             | 15 |
| 4.3      | <a href="#">SearchEngine Namespace Reference</a> | 15 |
| 4.3.1    | <a href="#">Function Documentation</a>           | 15 |
| 4.3.1.1  | <a href="#">getResults</a>                       | 15 |
| 4.3.1.2  | <a href="#">getResultWithHash</a>                | 15 |
| 4.3.1.3  | <a href="#">getResultWithName</a>                | 16 |
| 4.3.1.4  | <a href="#">search</a>                           | 16 |

---

|          |  |           |
|----------|--|-----------|
| 4.3.2    | Variable Documentation                 | 16        |
| 4.3.2.1  | search_results                         | 16        |
| <b>5</b> | <b>Class Documentation</b>             | <b>17</b> |
| 5.1      | Download Class Reference               | 17        |
| 5.1.1    | Detailed Description                   | 18        |
| 5.1.2    | Constructor & Destructor Documentation | 18        |
| 5.1.2.1  | Download                               | 18        |
| 5.1.2.2  | ~Download                              | 18        |
| 5.1.3    | Member Function Documentation          | 19        |
| 5.1.3.1  | calculateEstimatedTime                 | 19        |
| 5.1.3.2  | calculateRatio                         | 19        |
| 5.1.3.3  | getFilename                            | 19        |
| 5.1.3.4  | getID                                  | 19        |
| 5.1.3.5  | getRootHash                            | 19        |
| 5.1.3.6  | getStatistics                          | 19        |
| 5.1.3.7  | getStatus                              | 19        |
| 5.1.3.8  | getTrackerAddress                      | 19        |
| 5.1.3.9  | pause                                  | 20        |
| 5.1.3.10 | resume                                 | 20        |
| 5.1.3.11 | retry                                  | 20        |
| 5.1.3.12 | setDownloadAmount                      | 20        |
| 5.1.3.13 | setDownloadSpeed                       | 20        |
| 5.1.3.14 | setID                                  | 20        |
| 5.1.3.15 | setPeers                               | 21        |
| 5.1.3.16 | setProgress                            | 21        |
| 5.1.3.17 | setSeeders                             | 21        |
| 5.1.3.18 | setStatus                              | 21        |
| 5.1.3.19 | setUploadAmount                        | 21        |
| 5.1.3.20 | setUploadSpeed                         | 22        |
| 5.1.3.21 | start                                  | 22        |
| 5.1.3.22 | stop                                   | 22        |
| 5.1.4    | Member Data Documentation              | 22        |
| 5.1.4.1  | _filename                              | 22        |

---

|          |  |    |
|----------|--|----|
| 5.1.4.2  | <a href="#">_mutex</a>                                     | 22 |
| 5.1.4.3  | <a href="#">_root_hash</a>                                 | 22 |
| 5.1.4.4  | <a href="#">_stats</a>                                     | 22 |
| 5.1.4.5  | <a href="#">_status</a>                                    | 23 |
| 5.1.4.6  | <a href="#">_tracker</a>                                   | 23 |
| 5.2      | <a href="#">Download::downloadStats Struct Reference</a>   | 23 |
| 5.2.1    | <a href="#">Detailed Description</a>                       | 24 |
| 5.2.2    | <a href="#">Member Data Documentation</a>                  | 24 |
| 5.2.2.1  | <a href="#">download_amount</a>                            | 24 |
| 5.2.2.2  | <a href="#">download_percentage</a>                        | 24 |
| 5.2.2.3  | <a href="#">download_speed</a>                             | 24 |
| 5.2.2.4  | <a href="#">estimated</a>                                  | 24 |
| 5.2.2.5  | <a href="#">id</a>   | 24 |
| 5.2.2.6  | <a href="#">peers</a>                                      | 24 |
| 5.2.2.7  | <a href="#">ratio</a>                                      | 24 |
| 5.2.2.8  | <a href="#">seeders</a>                                    | 25 |
| 5.2.2.9  | <a href="#">upload_amount</a>                              | 25 |
| 5.2.2.10 | <a href="#">upload_speed</a>                               | 25 |
| 5.3      | <a href="#">FileNotFoundException Class Reference</a>      | 25 |
| 5.3.1    | <a href="#">Detailed Description</a>                       | 25 |
| 5.3.2    | <a href="#">Member Function Documentation</a>              | 25 |
| 5.3.2.1  | <a href="#">what</a>                                       | 25 |
| 5.4      | <a href="#">SearchEngine::result Struct Reference</a>      | 26 |
| 5.4.1    | <a href="#">Detailed Description</a>                       | 26 |
| 5.4.2    | <a href="#">Member Data Documentation</a>                  | 26 |
| 5.4.2.1  | <a href="#">filename</a>                                   | 26 |
| 5.4.2.2  | <a href="#">hash</a>                                       | 26 |
| 5.4.2.3  | <a href="#">tracker</a>                                    | 26 |
| 5.5      | <a href="#">Stream Class Reference</a>                     | 26 |
| 5.5.1    | <a href="#">Detailed Description</a>                       | 27 |
| 5.5.2    | <a href="#">Constructor &amp; Destructor Documentation</a> | 27 |
| 5.5.2.1  | <a href="#">Stream</a>                                     | 27 |
| 5.5.2.2  | <a href="#">~Stream</a>                                    | 28 |
| 5.5.3    | <a href="#">Member Function Documentation</a>              | 28 |

|          |  |           |
|----------|--|-----------|
| 5.5.3.1  | <a href="#">beginStreaming</a>                           | 28        |
| 5.5.3.2  | <a href="#">getEvent</a>                                 | 28        |
| 5.5.3.3  | <a href="#">getInstance</a>                              | 28        |
| 5.5.3.4  | <a href="#">getStatus</a>                                | 28        |
| 5.5.3.5  | <a href="#">getTrackerAddress</a>                        | 28        |
| 5.5.3.6  | <a href="#">init</a>                                     | 28        |
| 5.5.3.7  | <a href="#">readStreaming</a>                            | 28        |
| 5.5.3.8  | <a href="#">setStatus</a>                                | 29        |
| 5.5.3.9  | <a href="#">setTracker</a>                               | 29        |
| 5.5.3.10 | <a href="#">start</a>                                    | 29        |
| 5.5.3.11 | <a href="#">stop</a>                                     | 29        |
| 5.5.4    | <a href="#">Member Data Documentation</a>                | 29        |
| 5.5.4.1  | <a href="#">_evclose</a>                                 | 29        |
| 5.5.4.2  | <a href="#">_instance</a>                                | 29        |
| 5.5.4.3  | <a href="#">_mutex</a>                                   | 29        |
| 5.5.4.4  | <a href="#">_streaming</a>                               | 29        |
| 5.5.4.5  | <a href="#">_tracker</a>                                 | 30        |
| 5.6      | <a href="#">Download::time Struct Reference</a>          | 30        |
| 5.6.1    | <a href="#">Detailed Description</a>                     | 30        |
| 5.6.2    | <a href="#">Member Data Documentation</a>                | 30        |
| 5.6.2.1  | <a href="#">days</a>                                     | 30        |
| 5.6.2.2  | <a href="#">hours</a>                                    | 30        |
| 5.6.2.3  | <a href="#">minutes</a>                                  | 30        |
| 5.6.2.4  | <a href="#">seconds</a>                                  | 31        |
| <b>6</b> | <b><a href="#">File Documentation</a></b>                | <b>33</b> |
| 6.1      | <a href="#">include/Download.h File Reference</a>        | 33        |
| 6.1.1    | <a href="#">Define Documentation</a>                     | 34        |
| 6.1.1.1  | <a href="#">SECONDS_PER_DAY</a>                          | 34        |
| 6.1.1.2  | <a href="#">SECONDS_PER_HOUR</a>                         | 34        |
| 6.1.1.3  | <a href="#">SECONDS_PER_MINUTE</a>                       | 34        |
| 6.1.2    | <a href="#">Enumeration Type Documentation</a>           | 34        |
| 6.1.2.1  | <a href="#">Status</a>                                   | 34        |
| 6.2      | <a href="#">include/DownloadManager.h File Reference</a> | 34        |

|          |  |    |
|----------|--|----|
| 6.3      | include/Exceptions.h File Reference . . . . .    | 36 |
| 6.4      | include/HttpServer.h File Reference . . . . .    | 36 |
| 6.5      | include/SearchEngine.h File Reference . . . . .  | 37 |
| 6.6      | include/Stream.h File Reference . . . . .        | 38 |
| 6.7      | src/Download.cpp File Reference . . . . .        | 38 |
| 6.8      | src/DownloadManager.cpp File Reference . . . . . | 38 |
| 6.8.1    | Function Documentation . . . . .                 | 38 |
| 6.8.1.1  | downloadCallback . . . . .                       | 38 |
| 6.9      | src/HttpServer.cpp File Reference . . . . .      | 39 |
| 6.10     | src/Main.cpp File Reference . . . . .            | 39 |
| 6.10.1   | Function Documentation . . . . .                 | 39 |
| 6.10.1.1 | InstallHTTPGateway . . . . .                     | 39 |
| 6.10.1.2 | main . . . . .                                   | 39 |
| 6.11     | src/SearchEngine.cpp File Reference . . . . .    | 40 |
| 6.12     | src/Stream.cpp File Reference . . . . .          | 40 |
| 6.12.1   | Function Documentation . . . . .                 | 40 |
| 6.12.1.1 | closeCallback . . . . .                          | 40 |



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

|                                 |    |
|---------------------------------|----|
| <a href="#">DownloadManager</a> | 7  |
| <a href="#">HttpServer</a>      | 13 |
| <a href="#">SearchEngine</a>    | 15 |



## Chapter 2

# Class Index

### 2.1 Class List

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

|   |    |
|---|----|
| <a href="#">Download</a> . . . . .  | 17 |
| <a href="#">Download::downloadStats</a> (Struct for holding download statistics ) . . . . . | 23 |
| <a href="#">FileNotFoundException</a> . . . . .   | 25 |
| <a href="#">SearchEngine::result</a> . . . . .  | 26 |
| <a href="#">Stream</a> . . . . .  | 26 |
| <a href="#">Download::time</a> (Root hash needed to start swift download ) . . . . .        | 30 |



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

|                                     |    |
|-------------------------------------|----|
| include/Download.h . . . . .        | 33 |
| include/DownloadManager.h . . . . . | 34 |
| include/Exceptions.h . . . . .      | 36 |
| include/HttpServer.h . . . . .      | 36 |
| include/SearchEngine.h . . . . .    | 37 |
| include/Stream.h . . . . .          | 38 |
| src/Download.cpp . . . . .          | 38 |
| src/DownloadManager.cpp . . . . .   | 38 |
| src/HttpServer.cpp . . . . .        | 39 |
| src/Main.cpp . . . . .              | 39 |
| src/SearchEngine.cpp . . . . .      | 40 |
| src/Stream.cpp . . . . .            | 40 |



## Chapter 4

# Namespace Documentation

### 4.1 DownloadManager Namespace Reference

#### Functions

- void [setDownloadDirectory](#) (std::string dir)  
*Download thread pid.*
- void [setActiveDownload](#) ([Download](#) \*download)
- std::vector< [Download](#) > [getDownloads](#) ()
- std::string [getDownloadDirectory](#) ()
- void [init](#) ()
- void [startStream](#) (std::string tracker)
- void [stopStream](#) ()
- void \* [startStreamThread](#) (void \*arg)
- void [updateDownloadStatistics](#) ()
- std::string [buildXML](#) ()
- void \* [dispatch](#) (void \*arg)
- int [resumeDownload](#) (std::string download\_hash)
- int [startDownload](#) (const std::string download\_hash)
- int [getIndexFromHash](#) (const std::string download\_hash)
- void [downloadFirstInList](#) ()
- void [add](#) ([Download](#) \*download)
- void [pauseDownload](#) (const std::string download\_hash)
- void [switchDownload](#) (std::string hash)
- void [removeFromList](#) (const std::string download\_hash)
- void [removeFromDisk](#) (const std::string download\_hash)
- void [clearList](#) ()

## Variables

- static std::vector< [Download](#) > [downloads](#)
- static std::string [download\\_directory](#)  
*Vector containing all downloads.*
- static struct event [evcompl](#)
- static [Download](#) \* [active\\_download](#)
- static pthread\_t [streaming\\_thread](#)
- static pthread\_t [thread](#)
- static pthread\_mutex\_t [mutex](#)
- static pthread\_mutex\_t [active\\_download\\_mutex](#)  
*Mutex to make downloads vector thread safe.*
- static ticpp::Document \* [doc](#)  
*Mutex to make active\_download thread safe.*
- static double [downloaded](#)
- static double [uploaded](#)  
*Total amount of bytes downloaded this session.*
- static int [d\\_pid](#) = -1  
*Total amount of bytes uploaded this session.*

### 4.1.1 Function Documentation

#### 4.1.1.1 void DownloadManager::add ( [Download](#) \* [download](#) )

Adds a download to the list.

##### Parameters

|                   |                           |
|-------------------|---------------------------|
| <i>download,:</i> | The download to be added. |
|-------------------|---------------------------|

Definition at line 319 of file DownloadManager.cpp.

#### 4.1.1.2 std::string DownloadManager::buildXML ( )

Builds the XML response for the HTTP server, providing download statistics.

Definition at line 58 of file DownloadManager.cpp.

#### 4.1.1.3 void DownloadManager::clearList ( )

Removes all downloads from the list

Definition at line 434 of file DownloadManager.cpp.



**4.1.1.4 void \* DownloadManager::dispatch ( void \* *arg* )**

Dispatches swift::Channel::evbase to get into the main loop to download files.

**Parameters**

|               |                                    |
|---------------|------------------------------------|
| <i>arg</i> ,: | Unused argument of pthread_create. |
|---------------|------------------------------------|

Definition at line 213 of file DownloadManager.cpp.

**4.1.1.5 void DownloadManager::downloadFirstInList ( )**

Downloads first element in the download list.

Definition at line 367 of file DownloadManager.cpp.

**4.1.1.6 std::string DownloadManager::getDownloadDirectory ( )**

Returns the download directory as string.

Definition at line 34 of file DownloadManager.cpp.

**4.1.1.7 std::vector< Download > DownloadManager::getDownloads ( )**

Returns a copy of the vector downloads.

Definition at line 23 of file DownloadManager.cpp.

**4.1.1.8 int DownloadManager::getIndexFromHash ( const std::string *download\_hash* )**

Finds the index of a download in the list given a root hash.

**Parameters**

|                         |   |
|-------------------------|---|
| <i>download_hash</i> ,: | The root hash of a download to be searched. |
|-------------------------|---|

Definition at line 377 of file DownloadManager.cpp.

**4.1.1.9 void DownloadManager::init ( )**

Initialises the download manager.

Definition at line 6 of file DownloadManager.cpp.

**4.1.1.10 void DownloadManager::pauseDownload ( const std::string *download\_hash* )**

Pauses a download with the given root hash.

**Parameters**

|                                     |                                |
|-------------------------------------|--------------------------------|
| <i>download_</i> -<br><i>hash,:</i> | The root hash of the download. |
|-------------------------------------|--------------------------------|

Definition at line 231 of file DownloadManager.cpp.

#### 4.1.1.11 void DownloadManager::removeFromDisk ( const std::string *download\_hash* )

Removes a download from the hard disk given a root hash.

**Parameters**

|                                     |  |
|-------------------------------------|--|
| <i>download_</i> -<br><i>hash,:</i> | The root hash of the download to be removed. |
|-------------------------------------|--|

Definition at line 411 of file DownloadManager.cpp.

#### 4.1.1.12 void DownloadManager::removeFromList ( const std::string *download\_hash* )

Removes a download from the list given a root hash.

**Parameters**

|                                     |  |
|-------------------------------------|--|
| <i>download_</i> -<br><i>hash,:</i> | The root hash of the download to be removed. |
|-------------------------------------|--|

Definition at line 392 of file DownloadManager.cpp.

#### 4.1.1.13 int DownloadManager::resumeDownload ( std::string *download\_hash* )

Resumes a paused download given a root hash.

**Parameters**

|                                     |                                |
|-------------------------------------|--------------------------------|
| <i>download_</i> -<br><i>hash,:</i> | The root hash of the download. |
|-------------------------------------|--------------------------------|

Definition at line 250 of file DownloadManager.cpp.

#### 4.1.1.14 void DownloadManager::setActiveDownload ( Download \* *download* )

Sets the active download.

**Parameters**

|                   |   |
|-------------------|---|
| <i>download,:</i> | The download to be the active download. |
|-------------------|---|

Definition at line 42 of file DownloadManager.cpp.

4.1.1.15 void DownloadManager::setDownloadDirectory ( std::string *dir* )

[Download](#) thread pid.

Sets the directory where swift will download files to.

#### Parameters

|               |   |
|---------------|---|
| <i>dir</i> ,: | The directory where swift will download files to. |
|---------------|---|

Definition at line 15 of file DownloadManager.cpp.

4.1.1.16 int DownloadManager::startDownload ( const std::string *download\_hash* )

Starts a download given a root hash.

#### Parameters

|                            |                                |
|----------------------------|--------------------------------|
| <i>download_ - hash</i> ,: | The root hash of the download. |
|----------------------------|--------------------------------|

Definition at line 276 of file DownloadManager.cpp.

4.1.1.17 void DownloadManager::startStream ( std::string *tracker* )

Starts the streaming thread.

#### Parameters

|                   |   |
|-------------------|---|
| <i>tracker</i> ,: | The tracker from which we stream content. |
|-------------------|---|

Definition at line 467 of file DownloadManager.cpp.

4.1.1.18 void \* DownloadManager::startStreamThread ( void \* *arg* )

Starts streaming.

#### Parameters

|               |                                    |
|---------------|------------------------------------|
| <i>arg</i> ,: | Unused argument of pthread_create. |
|---------------|------------------------------------|

Definition at line 457 of file DownloadManager.cpp.

4.1.1.19 void DownloadManager::stopStream ( )

Stops streaming.

Definition at line 449 of file DownloadManager.cpp.

#### 4.1.1.20 void DownloadManager::switchDownload ( std::string *hash* )

Switches active download to another download with the given root hash.

##### Parameters

|                |  |
|----------------|--|
| <i>hash</i> ,: | The root hash of the download to be switched to. |
|----------------|--|

Definition at line 305 of file DownloadManager.cpp.

#### 4.1.1.21 void DownloadManager::updateDownloadStatistics ( )

Updates the download statistics.

Definition at line 51 of file DownloadManager.cpp.

### 4.1.2 Variable Documentation

#### 4.1.2.1 Download\* DownloadManager::active\_download [static]

Definition at line 26 of file DownloadManager.h.

#### 4.1.2.2 pthread\_mutex\_t DownloadManager::active\_download\_mutex [static]

Mutex to make downloads vector thread safe.

Definition at line 30 of file DownloadManager.h.

#### 4.1.2.3 int DownloadManager::d\_pid = -1 [static]

Total amount of bytes uploaded this session.

Definition at line 36 of file DownloadManager.h.

#### 4.1.2.4 ticpp::Document\* DownloadManager::doc [static]

Mutex to make active\_download thread safe.

Definition at line 32 of file DownloadManager.h.

#### 4.1.2.5 std::string DownloadManager::download\_directory [static]

Vector containing all downloads.

Definition at line 23 of file DownloadManager.h.

4.1.2.6 `double DownloadManager::downloaded` `[static]`

Definition at line 34 of file DownloadManager.h.

4.1.2.7 `std::vector<Download> DownloadManager::downloads` `[static]`

Definition at line 21 of file DownloadManager.h.

4.1.2.8 `struct event DownloadManager::evcompl` `[static]`

Definition at line 25 of file DownloadManager.h.

4.1.2.9 `pthread_mutex_t DownloadManager::mutex` `[static]`

Definition at line 29 of file DownloadManager.h.

4.1.2.10 `pthread_t DownloadManager::streaming_thread` `[static]`

Definition at line 27 of file DownloadManager.h.

4.1.2.11 `pthread_t DownloadManager::thread` `[static]`

Definition at line 28 of file DownloadManager.h.

4.1.2.12 `double DownloadManager::uploaded` `[static]`

Total amount of bytes downloaded this session.

Definition at line 35 of file DownloadManager.h.

## 4.2 HttpServer Namespace Reference

### Functions

- static void [sendXMLResponse](#) (std::string msg, struct evhttp\_request \*req, struct evbuffer \*buf)
- static void [sendResponse](#) (struct evhttp\_request \*req, struct evbuffer \*buf, const char \*message)
- static void [handleRequest](#) (struct evhttp\_request \*req, void \*arg)
- int [init](#) ()

## Variables

- struct event\_base \* [base](#)

## 4.2.1 Function Documentation

**4.2.1.1** static void HttpServer::handleRequest ( struct evhttp\_request \* *req*, void \* *arg* )  
[static]

The HTTP GET request handler.

### Parameters

|               |                                   |
|---------------|-----------------------------------|
| <i>req</i> ,: | The request struct from libevent. |
| <i>arg</i> ,: | Ignored argument.                 |

Definition at line 53 of file HttpServer.cpp.

**4.2.1.2** int HttpServer::init ( )

Initialises the web server.

Definition at line 195 of file HttpServer.cpp.

**4.2.1.3** static void HttpServer::sendResponse ( struct evhttp\_request \* *req*, struct evbuffer \* *buf*, const char \* *message* ) [static]

Sends the HTTP response.

### Parameters

|                   |                                |
|-------------------|--------------------------------|
| <i>req</i> ,:     | The received HTTP request.     |
| <i>buf</i> ,:     | Buffer used to send the reply. |
| <i>message</i> ,: | The message to be sent.        |

Definition at line 36 of file HttpServer.cpp.

**4.2.1.4** static void HttpServer::sendXMLResponse ( std::string *msg*, struct evhttp\_request \* *req*, struct evbuffer \* *buf* ) [static]

Sends the HTTP XML response.

### Parameters

|               |                                |
|---------------|--------------------------------|
| <i>msg</i> ,: | The xml message to be sent.    |
| <i>req</i> ,: | The received HTTP request.     |
| <i>buf</i> ,: | Buffer used to send the reply. |

Definition at line 13 of file HttpServer.cpp.

#### 4.2.2 Variable Documentation

##### 4.2.2.1 struct event\_base\* HttpServer::base

Definition at line 4 of file HttpServer.cpp.

### 4.3 SearchEngine Namespace Reference

#### Classes

- struct [result](#)

#### Functions

- std::vector< struct [result](#) > [search](#) (std::string search\_term)
- std::vector< struct [result](#) > [getResults](#) ()
- struct [result](#) [getResultWithHash](#) (std::string hash)
- struct [result](#) [getResultWithName](#) (std::string name)

#### Variables

- static std::vector< struct [result](#) > [search\\_results](#)

#### 4.3.1 Function Documentation

##### 4.3.1.1 std::vector< struct SearchEngine::result > SearchEngine::getResults ( )

Returns the current list of results.

Definition at line 43 of file SearchEngine.cpp.

##### 4.3.1.2 struct SearchEngine::result SearchEngine::getResultWithHash ( std::string hash ) [read]

Returns the result with a certain hash.

Definition at line 54 of file SearchEngine.cpp.

4.3.1.3 **struct SearchEngine::result** SearchEngine::getResultWithName ( *std::string name* )  
[read]

Returns the result with a certain filename.

Definition at line 71 of file SearchEngine.cpp.

4.3.1.4 **std::vector< struct SearchEngine::result >** SearchEngine::search ( *std::string search\_term* )

Generates a new list of result based on the search term and returns the list.

#### Parameters

|                      |                             |
|----------------------|-----------------------------|
| <i>search_term</i> : | Keyword used to find files. |
|----------------------|-----------------------------|

Definition at line 8 of file SearchEngine.cpp.

## 4.3.2 Variable Documentation

4.3.2.1 **std::vector<struct result>** SearchEngine::search\_results [static]

Definition at line 20 of file SearchEngine.h.



## Chapter 5

# Class Documentation

### 5.1 Download Class Reference

```
#include <Download.h>
```

#### Classes

- struct [downloadStats](#)  
*Struct for holding download statistics.*
- struct [time](#)  
*Root hash needed to start swift download.*

#### Public Member Functions

- void [retry](#) ()  
*Struct holding the statistics of the download.*
- void [start](#) ()
- void [stop](#) ()
- void [pause](#) ()
- void [resume](#) ()
- const int [getID](#) ()
- const int [getStatus](#) ()
- std::string [getTrackerAddress](#) ()
- std::string [getFilename](#) ()
- std::string [getRootHash](#) ()
- struct [downloadStats](#) [getStatistics](#) ()
- void [setDownloadSpeed](#) (double speed)
- void [setUploadSpeed](#) (double speed)
- void [calculateRatio](#) ()
- void [setProgress](#) (double percentage)

- void [setUploadAmount](#) (double amount)
- void [setDownloadAmount](#) (double amount)
- void [setSeeders](#) (int amount)
- void [setPeers](#) (int amount)
- void [calculateEstimatedTime](#) ()
- void [setID](#) (int id)
- void [setStatus](#) (int status)
- [Download](#) (std::string tracker, std::string root\_hash, std::string filename)
- [~Download](#) ()

### Protected Attributes

- pthread\_mutex\_t [\\_mutex](#)
- volatile int [\\_status](#)  
*Mutex to prevent download thread and main thread from accessing same data at the same time.*
- std::string [\\_filename](#)  
*Current status of the download.*
- std::string [\\_tracker](#)  
*Name of the download.*
- std::string [\\_root\\_hash](#)  
*Trackers seeding this download.*
- [downloadStats](#) [\\_stats](#)

### 5.1.1 Detailed Description

Definition at line 25 of file Download.h.

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 [Download::Download](#) ( std::string *tracker*, std::string *root\_hash*, std::string *filename* ) [inline]

Constructor.

Definition at line 95 of file Download.h.

#### 5.1.2.2 [Download::~~Download](#) ( ) [inline]

Destructor.

Definition at line 107 of file Download.h.

### 5.1.3 Member Function Documentation

#### 5.1.3.1 void Download::calculateEstimatedTime ( )

Calculates estimated download time.

Definition at line 204 of file Download.cpp.

#### 5.1.3.2 void Download::calculateRatio ( )

Calculates upload/download ratio.

Definition at line 115 of file Download.cpp.

#### 5.1.3.3 std::string Download::getFilename ( )

Getter for the filename.

Definition at line 316 of file Download.cpp.

#### 5.1.3.4 const int Download::getID ( )

Getter for the download ID.

Definition at line 262 of file Download.cpp.

#### 5.1.3.5 std::string Download::getRootHash ( )

Getter for the root hash.

Definition at line 323 of file Download.cpp.

#### 5.1.3.6 Download::downloadStats Download::getStatistics ( ) [read]

Getter for the download statistics.

Definition at line 282 of file Download.cpp.

#### 5.1.3.7 const int Download::getStatus ( )

Getter for the status.

Definition at line 272 of file Download.cpp.

#### 5.1.3.8 std::string Download::getTrackerAddress ( )

Getter for the tracker address.

Definition at line 309 of file Download.cpp.

#### 5.1.3.9 void Download::pause ( )

Pauses downloading and uploading.

Definition at line 57 of file Download.cpp.

#### 5.1.3.10 void Download::resume ( )

Resumes a paused download.

Definition at line 71 of file Download.cpp.

#### 5.1.3.11 void Download::retry ( )

Struct holding the statistics of the download.

Deletes the downloaded content and try again from the beginning.

Definition at line 19 of file Download.cpp.

#### 5.1.3.12 void Download::setDownloadAmount ( double *amount* )

Setter for download amount.

##### Parameters

|                  |                         |
|------------------|-------------------------|
| <i>amount</i> ,: | Amount to be set in Kb. |
|------------------|-------------------------|

Definition at line 165 of file Download.cpp.

#### 5.1.3.13 void Download::setDownloadSpeed ( double *speed* )

Setter for download speed.

##### Parameters

|                 |                      |
|-----------------|----------------------|
| <i>speed</i> ,: | The speed in Kb/sec. |
|-----------------|----------------------|

Definition at line 90 of file Download.cpp.

#### 5.1.3.14 void Download::setID ( int *id* )

Setter for download ID.

##### Parameters

|              |                                       |
|--------------|---------------------------------------|
| <i>id</i> ,: | The download id given by swift::Open. |
|--------------|---------------------------------------|

Definition at line 239 of file Download.cpp.

#### 5.1.3.15 void Download::setPeers ( int *amount* )

Setter for amount of peers.

##### Parameters

|                  |                                |
|------------------|--------------------------------|
| <i>amount</i> ,: | The amount of peers to be set. |
|------------------|--------------------------------|

Definition at line 192 of file Download.cpp.

#### 5.1.3.16 void Download::setProgress ( double *percentage* )

Setter for download progress.

##### Parameters

|                      |   |
|----------------------|---|
| <i>percentage</i> ,: | The percentage to be set between 0 and 100. |
|----------------------|---|

Definition at line 139 of file Download.cpp.

#### 5.1.3.17 void Download::setSeeders ( int *amount* )

Setter for amount of seeders.

##### Parameters

|                  |                                  |
|------------------|----------------------------------|
| <i>amount</i> ,: | The amount of seeders to be set. |
|------------------|----------------------------------|

Definition at line 178 of file Download.cpp.

#### 5.1.3.18 void Download::setStatus ( int *status* )

Setter for status

##### Parameters

|                  |                           |
|------------------|---------------------------|
| <i>status</i> ,: | The new status to be set. |
|------------------|---------------------------|

Definition at line 250 of file Download.cpp.

#### 5.1.3.19 void Download::setUploadAmount ( double *amount* )

Setter for upload amount.

##### Parameters

|                  |                         |
|------------------|-------------------------|
| <i>amount</i> ,: | Amount to be set in Kb. |
|------------------|-------------------------|

Definition at line 152 of file Download.cpp.

#### 5.1.3.20 void Download::setUploadSpeed ( double *speed* )

Setter for upload speed.

##### Parameters

|                 |                      |
|-----------------|----------------------|
| <i>speed</i> ,: | The speed in Kb/sec. |
|-----------------|----------------------|

Definition at line 103 of file Download.cpp.

#### 5.1.3.21 void Download::start ( )

Starts downloading and uploading.

Definition at line 29 of file Download.cpp.

#### 5.1.3.22 void Download::stop ( )

Stops the download and removes all content from disk.

Definition at line 6 of file Download.cpp.

### 5.1.4 Member Data Documentation

#### 5.1.4.1 std::string Download::\_filename [protected]

Current status of the download.

Definition at line 32 of file Download.h.

#### 5.1.4.2 pthread\_mutex\_t Download::\_mutex [protected]

Definition at line 28 of file Download.h.

#### 5.1.4.3 std::string Download::\_root\_hash [protected]

Trackers seeding this download.

Definition at line 34 of file Download.h.

#### 5.1.4.4 downloadStats Download::\_stats [protected]

Definition at line 60 of file Download.h.

#### 5.1.4.5 volatile int Download::\_status [protected]

Mutex to prevent download thread and main thread from accessing same data at the same time.

Definition at line 30 of file Download.h.

#### 5.1.4.6 std::string Download::\_tracker [protected]

Name of the download.

Definition at line 33 of file Download.h.

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

- include/Download.h
- src/Download.cpp

## 5.2 Download::downloadStats Struct Reference

Struct for holding download statistics.

```
#include <Download.h>
```

### Public Attributes

- int [id](#)
- double [download\\_speed](#)  
*Download id needed to check the stats.*
- double [upload\\_speed](#)  
*Current download speed.*
- double [ratio](#)  
*Current upload speed.*
- double [download\\_percentage](#)  
*Download speed divided by upload speed.*
- double [upload\\_amount](#)  
*Download progress in percentage.*
- double [download\\_amount](#)  
*Uploaded amount in bytes.*
- int [seeders](#)  
*Downloaded amount in bytes.*
- int [peers](#)  
*Number of seeders uploading this file.*
- struct [time\\_estimated](#)  
*Number of peers connected to us for this file.*

### 5.2.1 Detailed Description

Struct for holding download statistics.

Definition at line 45 of file Download.h.

### 5.2.2 Member Data Documentation

#### 5.2.2.1 `double Download::downloadStats::download_amount`

Uploaded amount in bytes.

Definition at line 52 of file Download.h.

#### 5.2.2.2 `double Download::downloadStats::download_percentage`

[Download](#) speed divided by upload speed.

Definition at line 50 of file Download.h.

#### 5.2.2.3 `double Download::downloadStats::download_speed`

[Download](#) id needed to check the stats.

Definition at line 47 of file Download.h.

#### 5.2.2.4 `struct time Download::downloadStats::estimated`

Number of peers connected to us for this file.

Definition at line 57 of file Download.h.

#### 5.2.2.5 `int Download::downloadStats::id`

Definition at line 46 of file Download.h.

#### 5.2.2.6 `int Download::downloadStats::peers`

Number of seeders uploading this file.

Definition at line 55 of file Download.h.

#### 5.2.2.7 `double Download::downloadStats::ratio`

Current upload speed.

Definition at line 49 of file Download.h.



#### 5.2.2.8 int Download::downloadStats::seeders

Downloaded amount in bytes.

Definition at line 54 of file Download.h.

#### 5.2.2.9 double Download::downloadStats::upload\_amount

[Download](#) progress in percentage.

Definition at line 51 of file Download.h.

#### 5.2.2.10 double Download::downloadStats::upload\_speed

Current download speed.

Definition at line 48 of file Download.h.

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

- include/[Download.h](#)

## 5.3 FileNotFoundException Class Reference

```
#include <Exceptions.h>
```

### Public Member Functions

- virtual const char \* [what](#) () const throw ()

#### 5.3.1 Detailed Description

Definition at line 8 of file Exceptions.h.

#### 5.3.2 Member Function Documentation

5.3.2.1 virtual const char\* FileNotFoundException::what ( ) const throw () [inline, virtual]

Definition at line 10 of file Exceptions.h.

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

- include/[Exceptions.h](#)

## 5.4 SearchEngine::result Struct Reference

```
#include <SearchEngine.h>
```

### Public Attributes

- std::string [filename](#)
- std::string [tracker](#)
- std::string [hash](#)

### 5.4.1 Detailed Description

Definition at line 14 of file SearchEngine.h.

### 5.4.2 Member Data Documentation

#### 5.4.2.1 std::string SearchEngine::result::filename

Definition at line 15 of file SearchEngine.h.

#### 5.4.2.2 std::string SearchEngine::result::hash

Definition at line 17 of file SearchEngine.h.

#### 5.4.2.3 std::string SearchEngine::result::tracker

Definition at line 16 of file SearchEngine.h.

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

- include/[SearchEngine.h](#)

## 5.5 Stream Class Reference

```
#include <Stream.h>
```

### Public Member Functions

- [~Stream](#) ()
- void [start](#) ()
- void [stop](#) ()
- void [init](#) ()

- void [setTracker](#) (std::string)
- void [beginStreaming](#) ()
- bool [readStreaming](#) ()
- const int [getStatus](#) ()
- std::string [getTrackerAddress](#) ()
- struct event \* [getEvent](#) ()
- void [setStatus](#) (int status)

### Static Public Member Functions

- static [Stream](#) \* [getInstance](#) ()

### Protected Attributes

- std::string [\\_tracker](#)
- pthread\_mutex\_t [\\_mutex](#)  
*Trackers seeding this stream.*
- struct event [\\_evclose](#)  
*Mutex to prevent download thread and main thread from accessing same data at the same time.*
- volatile bool [\\_streaming](#)  
*Event used by libevent to stop a stream.*

### Private Member Functions

- [Stream](#) ()  
*Singleton instance of [Stream](#) class.*

### Static Private Attributes

- static [Stream](#) \* [\\_instance](#)  
*Boolean to determine whether a stream is opened or not.*

#### 5.5.1 Detailed Description

Definition at line 12 of file Stream.h.

#### 5.5.2 Constructor & Destructor Documentation

##### 5.5.2.1 [Stream::Stream](#) ( ) [inline, private]

Singleton instance of [Stream](#) class.

Constructor, made private to implement Singleton pattern.

Definition at line 25 of file Stream.h.

#### 5.5.3.2 `Stream::~~Stream ( ) [inline]`

Destructor.

Definition at line 35 of file Stream.h.

### 5.5.3 Member Function Documentation

#### 5.5.3.1 `void Stream::beginStreaming ( )`

Sets stream state to "begin streaming".

Definition at line 16 of file Stream.cpp.

#### 5.5.3.2 `event * Stream::getEvent ( ) [read]`

Returns the event to start the libevent loop.

Definition at line 59 of file Stream.cpp.

#### 5.5.3.3 `Stream * Stream::getInstance ( ) [static]`

Returns Singleton instance of own class.

Definition at line 66 of file Stream.cpp.

#### 5.5.3.4 `const int Stream::getStatus ( )`

#### 5.5.3.5 `std::string Stream::getTrackerAddress ( )`

#### 5.5.3.6 `void Stream::init ( )`

Initialises stream state and mutex.

Definition at line 8 of file Stream.cpp.

#### 5.5.3.7 `bool Stream::readStreaming ( )`

Returns stream state.

Definition at line 26 of file Stream.cpp.

5.5.3.8 void Stream::setStatus ( int *status* )

5.5.3.9 void Stream::setTracker ( std::string *tracker* )

Sets the tracker address of the stream.

#### Parameters

|                  |                                |
|------------------|--------------------------------|
| <i>tracker</i> , | The tracker address to be set. |
|------------------|--------------------------------|

Definition at line 52 of file Stream.cpp.

5.5.3.10 void Stream::start ( )

Starts the stream.

Definition at line 86 of file Stream.cpp.

5.5.3.11 void Stream::stop ( )

Stops the stream.

Definition at line 76 of file Stream.cpp.

### 5.5.4 Member Data Documentation

5.5.4.1 struct event Stream::\_evclose [protected]

Mutex to prevent download thread and main thread from accessing same data at the same time.

Definition at line 16 of file Stream.h.

5.5.4.2 Stream \* Stream::\_instance [static, private]

Boolean to determine whether a stream is opened or not.

Definition at line 20 of file Stream.h.

5.5.4.3 pthread\_mutex\_t Stream::\_mutex [protected]

Trackers seeding this stream.

Definition at line 15 of file Stream.h.

5.5.4.4 volatile bool Stream::\_streaming [protected]

Event used by libevent to stop a stream.

Definition at line 17 of file Stream.h.

#### 5.5.4.5 `std::string Stream::_tracker` [protected]

Definition at line 14 of file Stream.h.

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

- include/[Stream.h](#)
- src/[Stream.cpp](#)

## 5.6 Download::time Struct Reference

Root hash needed to start swift download.

```
#include <Download.h>
```

### Public Attributes

- int [days](#)
- int [hours](#)
- int [minutes](#)
- int [seconds](#)

### 5.6.1 Detailed Description

Root hash needed to start swift download.

Struct for holding time data.

Definition at line 37 of file Download.h.

### 5.6.2 Member Data Documentation

#### 5.6.2.1 `int Download::time::days`

Definition at line 38 of file Download.h.

#### 5.6.2.2 `int Download::time::hours`

Definition at line 39 of file Download.h.

#### 5.6.2.3 `int Download::time::minutes`

Definition at line 40 of file Download.h.

**5.6.2.4 int Download::time::seconds**

Definition at line 41 of file Download.h.

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

- include/[Download.h](#)





## Chapter 6

# File Documentation

### 6.1 include/Download.h File Reference

```
#include <iostream>
#include <ctime>
#include <string>
#include <cstdio>
#include <cstdlib>
#include "swift.h"
```

#### Classes

- class [Download](#)
- struct [Download::time](#)  
*Root hash needed to start swift download.*
- struct [Download::downloadStats](#)  
*Struct for holding download statistics.*

#### Defines

- #define [SECONDS\\_PER\\_MINUTE](#) (60)
- #define [SECONDS\\_PER\\_HOUR](#) (SECONDS\_PER\_MINUTE \* SECONDS\_PER\_MINUTE)
- #define [SECONDS\\_PER\\_DAY](#) (SECONDS\_PER\_HOUR \* 24)

#### Enumerations

- enum [Status](#) {

```
READY, PAUSED, DOWNLOADING, UPLOADING,  
STOPPED, SWITCHING }
```

### 6.1.1 Define Documentation

#### 6.1.1.1 #define SECONDS\_PER\_DAY (SECONDS\_PER\_HOUR \* 24)

Definition at line 14 of file Download.h.

#### 6.1.1.2 #define SECONDS\_PER\_HOUR (SECONDS\_PER\_MINUTE \* SECONDS\_PER\_MINUTE)

Definition at line 13 of file Download.h.

#### 6.1.1.3 #define SECONDS\_PER\_MINUTE (60)

Definition at line 12 of file Download.h.

### 6.1.2 Enumeration Type Documentation

#### 6.1.2.1 enum Status

Enumerator:

```
READY  
PAUSED  
DOWNLOADING  
UPLOADING  
STOPPED  
SWITCHING
```

Definition at line 16 of file Download.h.

## 6.2 include/DownloadManager.h File Reference

```
#include <vector>  
#include <string>  
#include <iostream>  
#include <cstdio>  
#include <cstdlib>  
#include <float.h>
```

```
#include "Stream.h"
#include "Download.h"
#include "swift.h"
#include "ticpp.h"
#include "Exceptions.h"
```

## Namespaces

- namespace [DownloadManager](#)

## Functions

- void [DownloadManager::setDownloadDirectory](#) (std::string dir)  
*Download thread pid.*
- void [DownloadManager::setActiveDownload](#) ([Download](#) \*download)
- std::vector< [Download](#) > [DownloadManager::getDownloads](#) ()
- std::string [DownloadManager::getDownloadDirectory](#) ()
- void [DownloadManager::init](#) ()
- void [DownloadManager::startStream](#) (std::string tracker)
- void [DownloadManager::stopStream](#) ()
- void \* [DownloadManager::startStreamThread](#) (void \*arg)
- void [DownloadManager::updateDownloadStatistics](#) ()
- std::string [DownloadManager::buildXML](#) ()
- void \* [DownloadManager::dispatch](#) (void \*arg)
- int [DownloadManager::resumeDownload](#) (std::string download\_hash)
- int [DownloadManager::startDownload](#) (const std::string download\_hash)
- int [DownloadManager::getIndexFromHash](#) (const std::string download\_hash)
- void [DownloadManager::downloadFirstInList](#) ()
- void [DownloadManager::add](#) ([Download](#) \*download)
- void [DownloadManager::pauseDownload](#) (const std::string download\_hash)
- void [DownloadManager::switchDownload](#) (std::string hash)
- void [DownloadManager::removeFromList](#) (const std::string download\_hash)
- void [DownloadManager::removeFromDisk](#) (const std::string download\_hash)
- void [DownloadManager::clearList](#) ()

## Variables

- static std::vector< [Download](#) > [DownloadManager::downloads](#)
- static std::string [DownloadManager::download\\_directory](#)  
*Vector containing all downloads.*
- static struct event [DownloadManager::evcompl](#)
- static [Download](#) \* [DownloadManager::active\\_download](#)
- static pthread\_t [DownloadManager::streaming\\_thread](#)

- static pthread\_t [DownloadManager::thread](#)
- static pthread\_mutex\_t [DownloadManager::mutex](#)
- static pthread\_mutex\_t [DownloadManager::active\\_download\\_mutex](#)  
*Mutex to make downloads vector thread safe.*
- static ticpp::Document \* [DownloadManager::doc](#)  
*Mutex to make active\_download thread safe.*
- static double [DownloadManager::downloaded](#)
- static double [DownloadManager::uploaded](#)  
*Total amount of bytes downloaded this session.*
- static int [DownloadManager::d\\_pid](#) = -1  
*Total amount of bytes uploaded this session.*

### 6.3 include/Exceptions.h File Reference

```
#include <exception>
#include <string>
```

#### Classes

- class [FileNotFoundException](#)

### 6.4 include/HttpServer.h File Reference

```
#include <iostream>
#include <string>
#include <cstdio>
#include <cstdlib>
#include <event2/event.h>
#include <event2/http.h>
#include <event2/buffer.h>
#include <event2/util.h>
#include <event2/event-config.h>
#include <event2/thread.h>
#include "Exceptions.h"
#include "Download.h"
#include "Stream.h"
#include "DownloadManager.h"
```

```
#include "SearchEngine.h"
#include "swift.h"
```

## Namespaces

- namespace [HttpServer](#)

## Functions

- static void [HttpServer::sendXMLResponse](#) (std::string msg, struct evhttp\_request \*req, struct evbuffer \*buf)
- static void [HttpServer::sendResponse](#) (struct evhttp\_request \*req, struct evbuffer \*buf, const char \*message)
- static void [HttpServer::handleRequest](#) (struct evhttp\_request \*req, void \*arg)
- int [HttpServer::init](#) ()

## 6.5 include/SearchEngine.h File Reference

```
#include <vector>
#include <string>
#include <iostream>
#include "HttpServer.h"
#include "Exceptions.h"
```

## Classes

- struct [SearchEngine::result](#)

## Namespaces

- namespace [SearchEngine](#)

## Functions

- std::vector< struct result > [SearchEngine::search](#) (std::string search\_term)
- std::vector< struct result > [SearchEngine::getResults](#) ()
- struct result [SearchEngine::getResultWithHash](#) (std::string hash)
- struct result [SearchEngine::getResultWithName](#) (std::string name)

## Variables

- static std::vector< struct result > [SearchEngine::search\\_results](#)

## 6.6 include/Stream.h File Reference

```
#include <iostream>
#include <string>
#include <ctime>
#include <cstdio>
#include <cstdlib>
#include "swift.h"
```

## Classes

- class [Stream](#)

## 6.7 src/Download.cpp File Reference

```
#include "Download.h"
```

## 6.8 src/DownloadManager.cpp File Reference

```
#include "DownloadManager.h"
```

## Functions

- void [downloadCallback](#) (int fd, short event, void \*arg)

### 6.8.1 Function Documentation

6.8.1.1 void [downloadCallback](#) ( int *fd*, short *event*, void \* *arg* )

Callback to keep all downloads and uploads running.

#### Parameters

|                 |                                    |
|-----------------|------------------------------------|
| <i>fd</i> ,:    | The file descriptor used by swift. |
| <i>event</i> ,: | The event it gets from libevent.   |
| <i>arg</i> ,:   | Unused argument from libevent.     |

Definition at line 165 of file DownloadManager.cpp.

## 6.9 src/HttpServer.cpp File Reference

```
#include "HttpServer.h"
```

### Namespaces

- namespace [HttpServer](#)

### Variables

- struct event\_base \* [HttpServer::base](#)

## 6.10 src/Main.cpp File Reference

```
#include <iostream>
#include "ticpp.h"
#include "swift.h"
#include "DownloadManager.h"
#include "HttpServer.h"
```

### Functions

- bool [InstallHTTPGateway](#) (struct event\_base \*evbase, swift::Address addr, uint32\_t chunk\_size, double \*maxspeed)
- int [main](#) ()

#### 6.10.1 Function Documentation

**6.10.1.1** bool [InstallHTTPGateway](#) ( struct event\_base \* *evbase*, swift::Address *addr*, uint32\_t *chunk\_size*, double \* *maxspeed* )

Define the InstallHTTPGateway method in httpgw.cpp.

**6.10.1.2** int [main](#) ( )

Application main loop.

Definition at line 16 of file Main.cpp.

## 6.11 src/SearchEngine.cpp File Reference

```
#include "SearchEngine.h"
```

## 6.12 src/Stream.cpp File Reference

```
#include "Stream.h"
```

### Functions

- void [closeCallback](#) (int fd, short event, void \*arg)

### 6.12.1 Function Documentation

6.12.1.1 void [closeCallback](#) ( int *fd*, short *event*, void \* *arg* )

Libevent loop for streaming files, also used to close stream.

#### Parameters

|                 |                                    |
|-----------------|------------------------------------|
| <i>fd</i> ,:    | The file descriptor used by swift. |
| <i>event</i> ,: | The event it gets from libevent.   |
| <i>arg</i> ,:   | Argument taken by callback.        |

Definition at line 39 of file Stream.cpp.



# Index

- ~Download
  - Download, [18](#)
- ~Stream
  - Stream, [28](#)
- \_evclose
  - Stream, [29](#)
- \_filename
  - Download, [22](#)
- \_instance
  - Stream, [29](#)
- \_mutex
  - Download, [22](#)
  - Stream, [29](#)
- \_root\_hash
  - Download, [22](#)
- \_stats
  - Download, [22](#)
- \_status
  - Download, [22](#)
- \_streaming
  - Stream, [29](#)
- \_tracker
  - Download, [23](#)
  - Stream, [30](#)
- active\_download
  - DownloadManager, [12](#)
- active\_download\_mutex
  - DownloadManager, [12](#)
- add
  - DownloadManager, [8](#)
- base
  - HttpServer, [15](#)
- beginStreaming
  - Stream, [28](#)
- buildXML
  - DownloadManager, [8](#)
- calculateEstimatedTime
  - Download, [19](#)
- calculateRatio
  - Download, [19](#)
- clearList
  - DownloadManager, [8](#)
- closeCallback
  - Stream.cpp, [40](#)
- d\_pid
  - DownloadManager, [12](#)
- days
  - Download::time, [30](#)
- dispatch
  - DownloadManager, [8](#)
- doc
  - DownloadManager, [12](#)
- Download, [17](#)
  - ~Download, [18](#)
  - \_filename, [22](#)
  - \_mutex, [22](#)
  - \_root\_hash, [22](#)
  - \_stats, [22](#)
  - \_status, [22](#)
  - \_tracker, [23](#)
  - calculateEstimatedTime, [19](#)
  - calculateRatio, [19](#)
  - Download, [18](#)
  - getFilename, [19](#)
  - getID, [19](#)
  - getRootHash, [19](#)
  - getStatistics, [19](#)
  - getStatus, [19](#)
  - getTrackerAddress, [19](#)
  - pause, [20](#)
  - resume, [20](#)
  - retry, [20](#)
  - setDownloadAmount, [20](#)
  - setDownloadSpeed, [20](#)
  - setID, [20](#)
  - setPeers, [21](#)
  - setProgress, [21](#)
  - setSeeders, [21](#)

- setStatus, 21
- setUploadAmount, 21
- setUploadSpeed, 22
- start, 22
- stop, 22
- Download.h
  - DOWNLOADING, 34
  - PAUSED, 34
  - READY, 34
  - SECONDS\_PER\_DAY, 34
  - SECONDS\_PER\_HOUR, 34
  - SECONDS\_PER\_MINUTE, 34
  - Status, 34
  - STOPPED, 34
  - SWITCHING, 34
  - UPLOADING, 34
- Download::downloadStats, 23
  - download\_amount, 24
  - download\_percentage, 24
  - download\_speed, 24
  - estimated, 24
  - id, 24
  - peers, 24
  - ratio, 24
  - seeders, 24
  - upload\_amount, 25
  - upload\_speed, 25
- Download::time, 30
  - days, 30
  - hours, 30
  - minutes, 30
  - seconds, 30
- download\_amount
  - Download::downloadStats, 24
- download\_directory
  - DownloadManager, 12
- download\_percentage
  - Download::downloadStats, 24
- download\_speed
  - Download::downloadStats, 24
- downloadCallback
  - DownloadManager.cpp, 38
- downloaded
  - DownloadManager, 12
- downloadFirstInList
  - DownloadManager, 9
- DOWNLOADING
  - Download.h, 34
- DownloadManager, 7
  - active\_download, 12
  - active\_download\_mutex, 12
  - add, 8
  - buildXML, 8
  - clearList, 8
  - d\_pid, 12
  - dispatch, 8
  - doc, 12
  - download\_directory, 12
  - downloaded, 12
  - downloadFirstInList, 9
  - downloads, 13
  - evcompl, 13
  - getDownloadDirectory, 9
  - getDownloads, 9
  - getIndexFromHash, 9
  - init, 9
  - mutex, 13
  - pauseDownload, 9
  - removeFromDisk, 10
  - removeFromList, 10
  - resumeDownload, 10
  - setActiveDownload, 10
  - setDownloadDirectory, 10
  - startDownload, 11
  - startStream, 11
  - startStreamThread, 11
  - stopStream, 11
  - streaming\_thread, 13
  - switchDownload, 11
  - thread, 13
  - updateDownloadStatistics, 12
  - uploaded, 13
- DownloadManager.cpp
  - downloadCallback, 38
- downloads
  - DownloadManager, 13
- estimated
  - Download::downloadStats, 24
- evcompl
  - DownloadManager, 13
- filename
  - SearchEngine::result, 26
- FileNotFoundException, 25
  - what, 25
- getDownloadDirectory
  - DownloadManager, 9
- getDownloads

- DownloadManager, 9
- getEvent
  - Stream, 28
- getFilename
  - Download, 19
- getId
  - Download, 19
- getIndexFromHash
  - DownloadManager, 9
- getInstance
  - Stream, 28
- getResults
  - SearchEngine, 15
- getResultWithHash
  - SearchEngine, 15
- getResultWithName
  - SearchEngine, 15
- getRootHash
  - Download, 19
- getStatistics
  - Download, 19
- getStatus
  - Download, 19
  - Stream, 28
- getTrackerAddress
  - Download, 19
  - Stream, 28
- handleRequest
  - HttpServer, 14
- hash
  - SearchEngine::result, 26
- hours
  - Download::time, 30
- HttpServer, 13
  - base, 15
  - handleRequest, 14
  - init, 14
  - sendResponse, 14
  - sendXMLResponse, 14
- id
  - Download::downloadStats, 24
- include/Download.h, 33
- include/DownloadManager.h, 34
- include/Exceptions.h, 36
- include/HttpServer.h, 36
- include/SearchEngine.h, 37
- include/Stream.h, 38
- init
  - DownloadManager, 9
  - HttpServer, 14
  - Stream, 28
- InstallHTTPGateway
  - Main.cpp, 39
- main
  - Main.cpp, 39
- Main.cpp
  - InstallHTTPGateway, 39
  - main, 39
- minutes
  - Download::time, 30
- mutex
  - DownloadManager, 13
- pause
  - Download, 20
- PAUSED
  - Download.h, 34
- pauseDownload
  - DownloadManager, 9
- peers
  - Download::downloadStats, 24
- ratio
  - Download::downloadStats, 24
- readStreaming
  - Stream, 28
- READY
  - Download.h, 34
- removeFromDisk
  - DownloadManager, 10
- removeFromList
  - DownloadManager, 10
- resume
  - Download, 20
- resumeDownload
  - DownloadManager, 10
- retry
  - Download, 20
- search
  - SearchEngine, 16
- search\_results
  - SearchEngine, 16
- SearchEngine, 15
  - getResults, 15
  - getResultWithHash, 15
  - getResultWithName, 15

- search, 16
- search\_results, 16
- SearchEngine::result, 26
  - filename, 26
  - hash, 26
  - tracker, 26
- seconds
  - Download::time, 30
- SECONDS\_PER\_DAY
  - Download.h, 34
- SECONDS\_PER\_HOUR
  - Download.h, 34
- SECONDS\_PER\_MINUTE
  - Download.h, 34
- seeders
  - Download::downloadStats, 24
- sendResponse
  - HttpServer, 14
- sendXMLResponse
  - HttpServer, 14
- setActiveDownload
  - DownloadManager, 10
- setDownloadAmount
  - Download, 20
- setDownloadDirectory
  - DownloadManager, 10
- setDownloadSpeed
  - Download, 20
- setID
  - Download, 20
- setPeers
  - Download, 21
- setProgress
  - Download, 21
- setSeeders
  - Download, 21
- setStatus
  - Download, 21
  - Stream, 28
- setTracker
  - Stream, 29
- setUploadAmount
  - Download, 21
- setUploadSpeed
  - Download, 22
- src/Download.cpp, 38
- src/DownloadManager.cpp, 38
- src/HttpServer.cpp, 39
- src/Main.cpp, 39
- src/SearchEngine.cpp, 40
- src/Stream.cpp, 40
- start
  - Download, 22
  - Stream, 29
- startDownload
  - DownloadManager, 11
- startStream
  - DownloadManager, 11
- startStreamThread
  - DownloadManager, 11
- Status
  - Download.h, 34
- stop
  - Download, 22
  - Stream, 29
- STOPPED
  - Download.h, 34
- stopStream
  - DownloadManager, 11
- Stream, 26
  - ~Stream, 28
  - \_evclose, 29
  - \_instance, 29
  - \_mutex, 29
  - \_streaming, 29
  - \_tracker, 30
  - beginStreaming, 28
  - getEvent, 28
  - getInstance, 28
  - getStatus, 28
  - getTrackerAddress, 28
  - init, 28
  - readStreaming, 28
  - setStatus, 28
  - setTracker, 29
  - start, 29
  - stop, 29
  - Stream, 27
- Stream.cpp
  - closeCallback, 40
- streaming\_thread
  - DownloadManager, 13
- switchDownload
  - DownloadManager, 11
- SWITCHING
  - Download.h, 34
- thread
  - DownloadManager, 13
- tracker

---

- SearchEngine::result, [26](#)
- updateDownloadStatistics
  - DownloadManager, [12](#)
- upload\_amount
  - Download::downloadStats, [25](#)
- upload\_speed
  - Download::downloadStats, [25](#)
- uploaded
  - DownloadManager, [13](#)
- UPLOADING
  - Download.h, [34](#)
- what
  - FileNotFoundException, [25](#)