SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [Ftp](http://docs.google.com/classsf_1_1Ftp.htm)
* [ListingResponse](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm)

[Public Types](#_gjdgxs) | [Public Member Functions](#_30j0zll) | [List of all members](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse-members.htm)

sf::Ftp::ListingResponse Class Reference

[Network module](http://docs.google.com/group__network.htm)

Specialization of FTP response returning a filename lisiting. [More...](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm#details)

#include <[Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm)>

Inheritance diagram for sf::Ftp::ListingResponse:



| Public Types | |
| --- | --- |
| enum | [Status](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3b) {  [RestartMarkerReply](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba07e06d3326ba2d078583bef93930d909) = 110,  [ServiceReadySoon](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba22413357ade6b586f6ceb0d704f35075) = 120,  [DataConnectionAlreadyOpened](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bafa52d19bc813d69055f4cc390d4a76ca) = 125,  [OpeningDataConnection](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba794ebe743688be611447638bf9e49d86) = 150,  [Ok](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baa956e229ba6c0cdf0d88b0e05b286210) = 200,  [PointlessCommand](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba38adc424f1adcd332745de8cd3b7737a) = 202,  [SystemStatus](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba9bdd02ae119b8be639e778859ee74060) = 211,  [DirectoryStatus](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba8729460a695013cc96330e2fced0ae1f) = 212,  [FileStatus](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baebddfc7997dca289c83068dff3f47dce) = 213,  [HelpMessage](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba840fd2a1872fd4310b046541f57fdeb7) = 214,  [SystemType](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba78391f73aa11f07f1514c7d070b93c08) = 215,  [ServiceReady](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baea2ee2007d7843c21108bb686ef03757) = 220,  [ClosingConnection](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bab23931490fc2d1df3081d651fe0f4d6e) = 221,  [DataConnectionOpened](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3badc78ed87d5bddb174fa3c16707ac2f2d) = 225,  [ClosingDataConnection](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bac723ebc8a38913bbf0d9504556cbaaa6) = 226,  [EnteringPassiveMode](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba48314fc47a72ad0aacdea93b91756f6e) = 227,  [LoggedIn](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba54a88210386cb72e35d737813a221754) = 230,  [FileActionOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baf988b69b0a5f55f8122da5ba001932e0) = 250,  [DirectoryOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba06d26e95a170fc422af13def415e0437) = 257,  [NeedPassword](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba9249e3fe9818eb93f181fbbf3ae3bc56) = 331,  [NeedAccountToLogIn](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba9e048185f253f6eb6f5ff9e063b712fa) = 332,  [NeedInformation](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba02e6f05964ecb829e9b6fb6020d6528a) = 350,  [ServiceUnavailable](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba43022ddf49b68a4f5aff0bea7e09e89f) = 421,  [DataConnectionUnavailable](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba757b89ff1f236941f7759b0ed0c28b88) = 425,  [TransferAborted](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba7cfefcc586c12ba70f752353fde7126e) = 426,  [FileActionAborted](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baf822d1b0abf3e9ae7dd44684549d512d) = 450,  [LocalError](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bae54e84baaca95a7b36271ca3f3fdb900) = 451,  [InsufficientStorageSpace](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba5d9f3666222c808553c27e4e099c7c6d) = 452,  [CommandUnknown](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba75bdf0b6844fa9c07b3c25647d22c269) = 500,  [ParametersUnknown](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baf4c7c88815981bbb7c3a3461f9f48b67) = 501,  [CommandNotImplemented](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba2ca4834c756c81b924ebed696fcba0a8) = 502,  [BadCommandSequence](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bad0c7ab07f01c1f7af16a1852650d7c47) = 503,  [ParameterNotImplemented](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba8807473b8590e1debfb3740b7a3d081c) = 504,  [NotLoggedIn](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bafcfbaff2c6fed941b6bcbc0999db764e) = 530,  [NeedAccountToStore](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba1af0f173062a471739b50d8e0f40d5f7) = 532,  [FileUnavailable](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba3f8f931e499936fde6b750d81f5ecfef) = 550,  [PageTypeUnknown](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bad220bc12dc45593af6e5079ea6c532c3) = 551,  [NotEnoughMemory](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baf418e54753e0b8f9cb0325dd618acd14) = 552,  [FilenameNotAllowed](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba03254aba823298179a98056e15568c5b) = 553,  [InvalidResponse](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba59e041e4ef186e8ae8d6035973fc46bd) = 1000,  [ConnectionFailed](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3ba51aa367cc1e85a45ea3c7be48730e990) = 1001,  [ConnectionClosed](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3bad1e5dcf298ce30c528261435f1a2eb53) = 1002,  [InvalidFile](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3baed2c74a9f335dee1463ca1a4f41c6478) = 1003  } |
|  | Status codes possibly returned by a FTP response. [More...](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3b) |
|  | |

| Public Member Functions | |
| --- | --- |
|  | [ListingResponse](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm#aefc1b85e59ee0c3ee180666b4a4631e4) (const [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) &response, const std::vector< char > &data) |
|  | Default constructor. |
|  | |
| const std::vector< std::string > & | [getListing](http://docs.google.com/classsf_1_1Ftp_1_1ListingResponse.htm#a5f0771b52a966bf25b33a70602b6f97f) () const |
|  | Return the array of directory/file names. |
|  | |
| bool | [isOk](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a4dadbe0fe0a3ef2d571a017e1645e675) () const |
|  | Check if the status code means a success. |
|  | |
| [Status](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3b) | [getStatus](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#ac7f937b3883d1c4fbc75c003a1786aaa) () const |
|  | Get the status code of the response. |
|  | |
| const std::string & | [getMessage](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#a0015675c528a4a84a671484b9e5499d6) () const |
|  | Get the full message contained in the response. |
|  | |

## Detailed Description

Specialization of FTP response returning a filename lisiting.

Definition at line [221](http://docs.google.com/Ftp_8hpp_source.htm#l00221) of file [Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm).

## Member Enumeration Documentation

| | enum [sf::Ftp::Response::Status](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3b) | | --- | | inherited |
| --- | --- | --- |

Status codes possibly returned by a FTP response.

**Enumerator:**

| *RestartMarkerReply* | Restart marker reply. |
| --- | --- |
| *ServiceReadySoon* | Service ready in N minutes. |
| *DataConnectionAlreadyOpened* | Data connection already opened, transfer starting. |
| *OpeningDataConnection* | File status ok, about to open data connection. |
| *Ok* | Command ok. |
| *PointlessCommand* | Command not implemented. |
| *SystemStatus* | System status, or system help reply. |
| *DirectoryStatus* | Directory status. |
| *FileStatus* | File status. |
| *HelpMessage* | Help message. |
| *SystemType* | NAME system type, where NAME is an official system name from the list in the Assigned Numbers document. |
| *ServiceReady* | Service ready for new user. |
| *ClosingConnection* | Service closing control connection. |
| *DataConnectionOpened* | Data connection open, no transfer in progress. |
| *ClosingDataConnection* | Closing data connection, requested file action successful. |
| *EnteringPassiveMode* | Entering passive mode. |
| *LoggedIn* | User logged in, proceed. Logged out if appropriate. |
| *FileActionOk* | Requested file action ok. |
| *DirectoryOk* | PATHNAME created. |
| *NeedPassword* | User name ok, need password. |
| *NeedAccountToLogIn* | Need account for login. |
| *NeedInformation* | Requested file action pending further information. |
| *ServiceUnavailable* | Service not available, closing control connection. |
| *DataConnectionUnavailable* | Can't open data connection. |
| *TransferAborted* | Connection closed, transfer aborted. |
| *FileActionAborted* | Requested file action not taken. |
| *LocalError* | Requested action aborted, local error in processing. |
| *InsufficientStorageSpace* | Requested action not taken; insufficient storage space in system, file unavailable. |
| *CommandUnknown* | Syntax error, command unrecognized. |
| *ParametersUnknown* | Syntax error in parameters or arguments. |
| *CommandNotImplemented* | Command not implemented. |
| *BadCommandSequence* | Bad sequence of commands. |
| *ParameterNotImplemented* | Command not implemented for that parameter. |
| *NotLoggedIn* | Not logged in. |
| *NeedAccountToStore* | Need account for storing files. |
| *FileUnavailable* | Requested action not taken, file unavailable. |
| *PageTypeUnknown* | Requested action aborted, page type unknown. |
| *NotEnoughMemory* | Requested file action aborted, exceeded storage allocation. |
| *FilenameNotAllowed* | Requested action not taken, file name not allowed. |
| *InvalidResponse* | [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) is not a valid FTP one. |
| *ConnectionFailed* | Connection with server failed. |
| *ConnectionClosed* | Connection with server closed. |
| *InvalidFile* | Invalid file to upload / download. |

Definition at line [74](http://docs.google.com/Ftp_8hpp_source.htm#l00074) of file [Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::Ftp::ListingResponse::ListingResponse | ( | const [Response](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm) & | *response*, |
| --- | --- | --- | --- |
|  |  | const std::vector< char > & | *data* |
|  | ) |  |  |

Default constructor.

Parameters

| response | Source response |
| --- | --- |
| data | Data containing the raw listing |

## Member Function Documentation

| const std::vector<std::string>& sf::Ftp::ListingResponse::getListing | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Return the array of directory/file names.

ReturnsArray containing the requested listing

| | const std::string& sf::Ftp::Response::getMessage | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the full message contained in the response.

ReturnsThe response message

| | [Status](http://docs.google.com/classsf_1_1Ftp_1_1Response.htm#af81738f06b6f571761696291276acb3b) sf::Ftp::Response::getStatus | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Get the status code of the response.

ReturnsStatus code

| | bool sf::Ftp::Response::isOk | ( |  | ) | const | | --- | --- | --- | --- | --- | | inherited |
| --- | --- | --- | --- | --- | --- | --- |

Check if the status code means a success.

This function is defined for convenience, it is equivalent to testing if the status code is < 400.

ReturnsTrue if the status is a success, false if it is a failure

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

* [Ftp.hpp](http://docs.google.com/Ftp_8hpp_source.htm)

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::