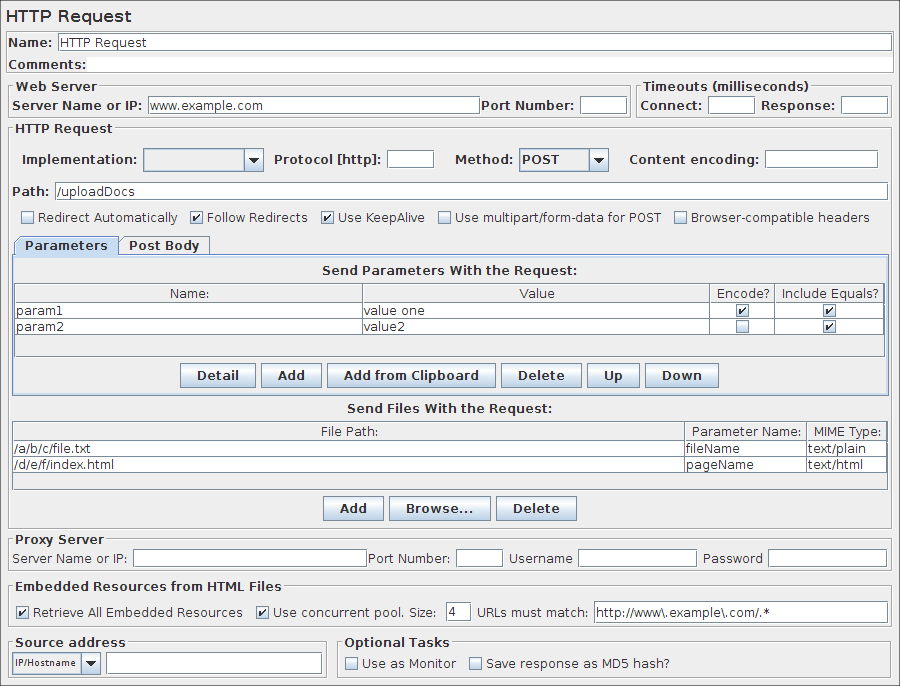
**Samplers**

Samplers perform the actual work of JMeter. Each sampler (except Test Action) generates one or more sample results. The sample results have various attributes (success/fail, elapsed time, data size etc.) and can be viewed in the various listeners.

**Jmeter Samplers Includes**

1. **Http Request:** This sampler lets you send an HTTP/HTTPS request to a web server. It also lets you control whether or not Jmeter parse HTML files for images and other embedded resources and sends HTTP requests to retrieve them.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Parameters**   |  |  | | --- | --- | | **Attribute** | **Description** | | Name | Descriptive name for this sampler that is shown in the tree. | | Server | Domain name or IP address of the web server. e.g. www.example.com. [Do not include the http:// prefix.] Note: in JMeter 2.5 (and later) if the "Host" header is defined in a Header Manager, then this will be used as the virtual host name. | | Port | Port the web server is listening to. Default: 80 | | Connect Timeout | Connection Timeout. Number of milliseconds to wait for a connection to open. | | Response Timeout | Response Timeout. Number of milliseconds to wait for a response. Note that this applies to each wait for a response. If the server response is sent in several chunks, the overall elapsed time may be longer than the timeout. A [Duration Assertion](about:blank../usermanual/component_reference.html#Duration_Assertion) can be used to detect responses that take too long to complete. | | Server (proxy) | Hostname or IP address of a proxy server to perform request. [Do not include the http:// prefix.] | | Port | Port the proxy server is listening to. | | Username | (Optional) username for proxy server. | | Password | (Optional) password for proxy server. (N.B. this is stored unencrypted in the test plan) | | Implementation | Java, HttpClient3.1, HttpClient4. If not specified (and not defined by HTTP Request Defaults), the default depends on the value of the JMeter property jmeter.httpsampler , failing that, the HttpClient4 implementation is used. | | Protocol | HTTP, HTTPS or FILE. Default: HTTP | | Method | GET, POST, HEAD, TRACE, OPTIONS, PUT, DELETE, PATCH (not supported for JAVA implementation). With HttpClient4, the following methods related to WebDav are also allowed: COPY, LOCK, MKCOL, MOVE, PROPFIND, PROPPATCH, UNLOCK, REPORT, MKCALENDAR. | | Content Encoding | Content encoding to be used (for POST, PUT, PATCH and FILE). This character encoding to be used, and is not related to the Content-Encoding HTTP header. | | Redirect Automatically | Sets the underlying http protocol handler to automatically follow redirects, so they are not seen by JMeter, and thus will not appear as samples. Should only be used for GET and HEAD requests. The HttpClient sampler will reject attempts to use it for POST or PUT. **Warning: see below for information on cookie and header handling.** | | Follow Redirects | This only has any effect if "Redirect Automatically" is not enabled. If set, the JMeter sampler will check if the response is a redirect and follow it if so. The initial redirect and further responses will appear as additional samples. The URL and data fields of the parent sample will be taken from the final (non-redirected) sample, but the parent byte count and elapsed time include all samples. The latency is taken from the initial response (versions of JMeter after 2.3.4 - previously it was zero). Note that the HttpClient sampler may log the following message: "Redirect requested but followRedirects is disabled" This can be ignored. In versions after 2.3.4, JMeter will collapse paths of the form '/../segment' in both absolute and relative redirect URLs. For example http://host/one/../two => http://host/two. If necessary, this behaviour can be suppressed by setting the JMeter property httpsampler.redirect.removeslashdotdot=false | | Use KeepAlive | JMeter sets the Connection: keep-alive header. This does not work properly with the default HTTP implementation, as connection re-use is not under user-control. It does work with the Apache HttpComponentsHttpClient implementations. | | Use multipart/form-data for HTTP POST | Use a multipart/form-data or application/x-www-form-urlencoded post request | | Browser-compatible headers | When using multipart/form-data, this suppresses the Content-Type and Content-Transfer-Encoding headers; only the Content-Disposition header is sent. | | Path | The path to resource (for example, /servlets/myServlet). If the resource requires query string parameters, add them below in the "Send Parameters With the Request" section. **As a special case, if the path starts with "http://" or "https://" then this is used as the full URL.** In this case, the server, port and protocol fields are ignored; parameters are also ignored for GET and DELETE methods. Also please note that the path is not encoded - apart from replacing spaces with %20 - so unsafe characters may need to be encoded to avoid errors such as URISyntaxException. | | Send Parameters With the Request | The query string will be generated from the list of parameters you provide. Each parameter has a *name* and *value* , the options to encode the parameter, and an option to include or exclude an equals sign (some applications don't expect an equals when the value is the empty string). The query string will be generated in the correct fashion, depending on the choice of "Method" you made (ie if you chose GET or DELETE, the query string will be appended to the URL, if POST or PUT, then it will be sent separately). Also, if you are sending a file using a multipart form, the query string will be created using the multipart form specifications. **See below for some further information on parameter handling.**  Additionally, you can specify whether each parameter should be URL encoded. If you are not sure what this means, it is probably best to select it. If your values contain characters such as the following then encoding is usually required.:   * ASCII Control Chars * Non-ASCII characters * Reserved characters:URLs use some characters for special use in defining their syntax. When these characters are not used in their special role inside a URL, they need to be encoded, example : '$', '&', '+', ',' , '/', ':', ';', '=', '?', '@' * Unsafe characters: Some characters present the possibility of being misunderstood within URLs for various reasons. These characters should also always be encoded, example : ' ', '', '>', '#', '%', ... | | File Path: | Name of the file to send. If left blank, JMeter does not send a file, if filled in, JMeter automatically sends the request as a multipart form request.  If it is a POST or PUT or PATCH request and there is a single file whose 'Parameter name' attribute (below) is omitted, then the file is sent as the entire body of the request, i.e. no wrappers are added. This allows arbitrary bodies to be sent. This functionality is present for POST requests after version 2.2, and also for PUT requests after version 2.3. **See below for some further information on parameter handling.** | | Parameter name: | Value of the "name" web request parameter. | | MIME Type | MIME type (for example, text/plain). If it is a POST or PUT or PATCH request and either the 'name' attribute (below) are omitted or the request body is constructed from parameter values only, then the value of this field is used as the value of the content-type request header. | | Retrieve All Embedded Resources from HTML Files | Tell JMeter to parse the HTML file and send HTTP/HTTPS requests for all images, Java applets, JavaScript files, CSSs, etc. referenced in the file. See below for more details. | | Use as monitor | For use with the [Monitor Results](about:blank../usermanual/component_reference.html#Monitor_Results) listener. | | Save response as MD5 hash? | If this is selected, then the response is not stored in the sample result. Instead, the 32 character MD5 hash of the data is calculated and stored instead. This is intended for testing large amounts of data. | | Embedded URLs must match: | If present, this must be a regular expression that is used to match against any embedded URLs found. So if you only want to download embedded resources from http://example.com/, use the expression: http://example\.com/.\* | | Use concurrent pool | Use a pool of concurrent connections to get embedded resources. | | Size | Pool size for concurrent connections used to get embedded resources. | | Source address type | *[Only for HTTP Request with HTTPClient implementation]*  To distinguish the source address value, select the type of these:   * Select *IP/Hostname* to use a specific IP address or a (local) hostname * Select *Device* to pick the first available address for that interface which this may be either IPv4 or IPv6 * Select *Device IPv4* To select the IPv4 address of the device name (like eth0, lo, em0, etc.) * Select *Device IPv6* To select the IPv6 address of the device name (like eth0, lo, em0, etc.) | | Source address field | *[Only for HTTP Request with HTTPClient implementation]*  This property is used to enable IP Spoofing. It override the default local IP address for this sample. The JMeter host must have multiple IP addresses (i.e. IP aliases, network interfaces, devices). The value can be a host name, IP address, or a network interface device such as "eth0" or "lo" or "wlan0". If the property **httpclient.localaddress**is defined, that is used for all HttpClient requests. | | | |

1. **JDBC Request:**This sampler lets you send a JDBC Request (an SQL query) to a database.

Before using this you need to set up a[**JDBC Connection Configuration**](http://jmeter.apache.org/usermanual/component_reference.html#JDBC_Connection_Configuration) Configuration element

If the Variable Names list is provided, then for each row returned by a Select statement, the variables are set up with the value of the corresponding column (if a variable name is provided), and the count of rows is also set up. For example, if the Select statement returns 2 rows of 3 columns, and the variable list is A,,C, then the following variables will be set up:

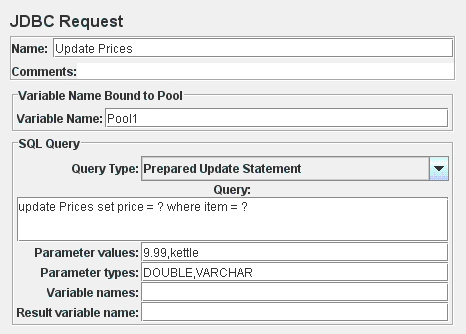
|  |
| --- |
| A\_#=2 (number of rows)  A\_1=column 1, row 1  A\_2=column 1, row 2  C\_#=2 (number of rows)  C\_1=column 3, row 1  C\_2=column 3, row 2 |

|  |  |
| --- | --- |
| |  | | --- | |  | |

If the Select statement returns zero rows, then the A\_# and C\_# variables would be set to 0, and no other variables would be set.

Old variables are cleared if necessary - e.g. if the first select retrieves 6 rows and a second select returns only 3 rows, the additional variables for rows 4, 5 and 6 will be removed.

**Note:** The latency time is set from the time it took to acquire a connection.



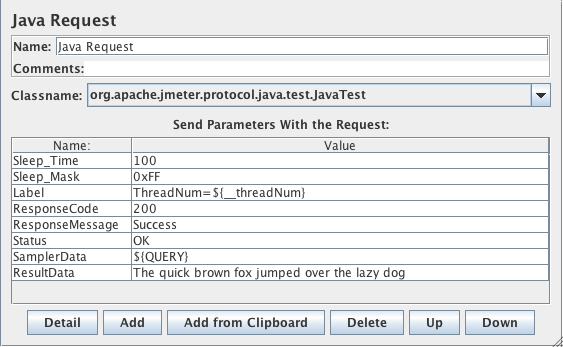
**Parameters**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Attribute** | **Description** | | Name | Descriptive name for this sampler that is shown in the tree. | | Variable Name | Name of the JMeter variable that the connection pool is bound to. This must agree with the 'Variable Name' field of a JDBC Connection Configuration. | | Query Type | Set this according to the statement type:   * Select Statement * Update Statement - use this for Inserts and Deletes as well * Callable Statement * Prepared Select Statement * Prepared Update Statement - use this for Inserts and Deletes as well * Commit * Rollback * Autocommit(false) * Autocommit(true) * Edit - this should be a variable reference that evaluates to one of the above | | SQL Query | SQL query. Do not enter a trailing semi-colon. There is generally no need to use { and } to enclose Callable statements; however they mey be used if the database uses a non-standard syntax. [The JDBC driver automatically converts the statement if necessary when it is enclosed in {}]. For example:   * select \* from t\_customers where id=23 * CALL SYSCS\_UTIL.SYSCS\_EXPORT\_TABLE (null,?, ?, null, null, null)   + Parameter values: tablename,filename   + Parameter types: VARCHAR,VARCHAR   The second example assumes you are using Apache Derby. | | Parameter values | Comma-separated list of parameter values. Use ]NULL[ to indicate a NULL parameter. (If required, the null string can be changed by defining the property "jdbcsampler.nullmarker".) The list must be enclosed in double-quotes if any of the values contain a comma or double-quote, and any embedded double-quotes must be doubled-up, for example:  "Dbl-Quote: "" and Comma: ," | | Parameter types | Comma-separated list of SQL parameter types (e.g. INTEGER, DATE, VARCHAR, DOUBLE) or integer values of Constants when for example you use custom database types proposed by driver (-10 for OracleTypes.CURSOR for example). These are defined as fields in the class java.sql.Types, see for example: [Javadoc for java.sql.Types](http://docs.oracle.com/javase/6/docs/api/java/sql/Types.html). [Note: JMeter will use whatever types are defined by the runtime JVM, so if you are running on a different JVM, be sure to check the appropriate document] **If the callable statement has INOUT or OUT parameters, then these must be indicated by prefixing the appropriate parameter types, e.g. instead of "INTEGER", use "INOUT INTEGER".**  If not specified, "IN" is assumed, i.e. "DATE" is the same as "IN DATE". If the type is not one of the fields found in java.sql.Types, versions of JMeter after 2.3.2 also accept the corresponding integer number, e.g. since OracleTypes.CURSOR == -10, you can use "INOUT -10". There must be as many types as there are placeholders in the statement. | | Variable Names | Comma-separated list of variable names to hold values returned by Select statements, Prepared Select Statements or CallableStatement. Note that when used with CallableStatement, list of variables must be in the same sequence as the OUT parameters returned by the call. If there are less variable names than OUT parameters only as many results shall be stored in the thread-context variables as variable names were supplied. If more variable names than OUT parameters exist, the additional variables will be ignored | | Result Variable Name | If specified, this will create an Object variable containing a list of row maps. Each map contains the column name as the key and the column data as the value. Usage: columnValue = vars.getObject("resultObject").get(0).get("Column Name"); | | Handle ResultSet | Defines how ResultSet returned from callable statements be handled:   * Store As String (default) - All variables on Variable Names list are stored as strings, will not iterate through a ResultSets when present on the list. * Store As Object - Variables of ResultSet type on Variables Names list will be stored as Object and can be accessed in subsequent tests/scripts and iterated, will not iterate through the ResultSet * Count Records - Variables of ResultSet types will be iterated through showing the count of records as result. Variables will be stored as Strings. | | | |

1. **JAVA Request:**This sampler lets you control a java class that implements the**org.apache.jmeter.protocol.java.sampler.javasamplerclient**interface. By writing your own implementation of this interface, you can use JMeter to harness multiple threads, input parameter control, and data collection.

The pull-down menu provides the list of all such implementations found by JMeter in its classpath. The parameters can then be specified in the table below - as defined by your implementation. Two simple examples (JavaTest and SleepTest) are provided.

The Java Test example sampler can be useful for checking test plans, because it allows one to set values in almost all the fields. These can then be used by Assertions, etc. The fields allow variables to be used, so the values of these can readily be seen.



**Parameters**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Attribute** | **Description** | | Name | Descriptive name for this sampler that is shown in the tree. | | Classname | The specific implementation of the JavaSamplerClient interface to be sampled. | | Send Parameters with Request | A list of arguments that will be passed to the sampled class. All arguments are sent as Strings. See below for specific settings. | | | |

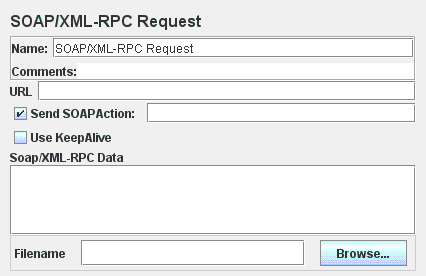
The following parameters apply to the **SleepTest** and **JavaTest** implementations:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Attribute** | **Description** | | Sleep\_time | How long to sleep for (ms) | | Sleep\_mask | How much "randomness" to add: The sleep time is calculated as follows: totalSleepTime = SleepTime + (System.currentTimeMillis() % SleepMask) | | | |

The following parameters apply additionaly to the **JavaTest** implementation:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Attribute** | **Description** | | Label | The label to use. If provided, overrides Name | | ResponseCode | If provided, sets the SampleResultResponseCode. | | ResponseMessage | If provided, sets the SampleResultResponseMessage. | | Status | If provided, sets the SampleResult Status. If this equals "OK" (ignoring case) then the status is set to success, otherwise the sample is marked as failed. | | SamplerData | If provided, sets the SampleResultSamplerData. | | ResultData | If provided, sets the SampleResultResultData. | | | |

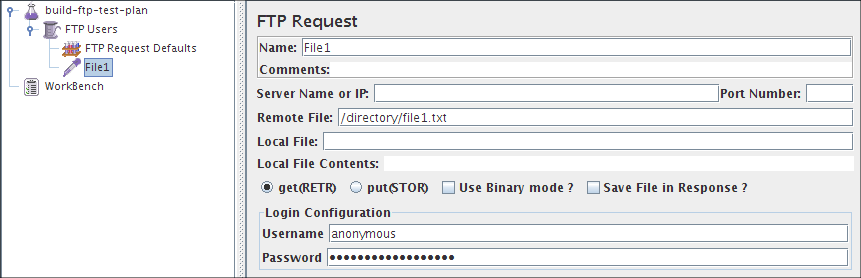
1. **SOAP/XML-RPC Request:**This sampler lets you send a SOAP request to a webservice. It can also be used to send XML-RPC over HTTP. It creates an HTTP POST request, with the specified XML as the POST content. To change the "Content-type" from the default of "text/xml", use a HeaderManager. Note that the sampler will use all the headers from the HeaderManager. If a SOAP action is specified, that will override any SOAPaction in the HeaderManager. The primary difference between the soap sampler and webservicesampler is the soap sampler uses raw post and does not require conformance to SOAP 1.1.



**Parameters**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Attribute** | **Description** | | Name | Descriptive name for this sampler that is shown in the tree. | | URL | The URL to direct the SOAP request to. | | Send SOAP action | Send a SOAP action header? (overrides the Header Manager) | | Use KeepAlive | If set, sends Connection: keep-alive, else sends Connection: close | | Soap/XML-RPC Data | The Soap XML message or XML-RPC instructions. Not used if the filename is provided. | | Filename | If specified, then the contents of the file are sent, and the Data field is ignored | | | |

1. **FTP Sampler:**This controller lets you send an FTP "retrieve file" or "upload file" request to an FTP server. If you are going to send multiple requests to the same FTP server, consider using a [FTP Request Defaults](http://jmeter.apache.org/usermanual/component_reference.html#FTP_Request_Defaults) Configuration Element so you do not have to enter the same information for each FTP Request Generative Controller. When downloading a file, it can be stored on disk (Local File) or in the Response Data, or both.



**Parameters**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | **Attribute** | **Description** | | Name | Descriptive name for this sampler that is shown in the tree. | | Server Name or IP | Domain name or IP address of the FTP server. | | Port | Port to use. If this is >0, then this specific port is used, otherwise JMeter uses the default FTP port. | | Remote File: | File to retrieve or name of destination file to upload. | | Local File: | File to upload, or destination for downloads (defaults to remote file name). | | Local File Contents: | Provides the contents for the upload, overrides the Local File property. | | get(RETR) / put(STOR) | Whether to retrieve or upload a file. | | Use Binary mode ? | Check this to use Binary mode (default Ascii) | | Save File in Response ? | Whether to store contents of retrieved file in response data. If the mode is Ascii, then the contents will be visible in the Tree View Listener. | | Username | FTP account username. | | Password | FTP account password. N.B. This will be visible in the test plan. | | | |