added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

`public abstract class HttpURLConnection`
`extends URLConnection` (https://developer.android.com/reference/java/net/URLConnection.html)

java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)
↳ java.net.URLConnection (https://developer.android.com/reference/java/net/URLConnection.html)
↳ java.net.HttpURLConnection

Known direct subclasses
HttpsURLConnection (https://developer.android.com/reference/javax/net/ssl/HttpsURLConnection.h

HttpsURLConnection (https://developer.android.com/reference/javax/net/ssl/HttpsURLConnection.html)	**HttpsURLCon** extends **HttpURLConn** with support fc specific featur

A URLConnection with support for HTTP-specific features. See the spec (http://www.w3.org/pub/WWW/Protocols/) for details.

Uses of this class follow a pattern:

1. Obtain a new **`HttpURLConnection`** by calling **`URL.openConnection()`** (https://developer.android.com/reference/java/net/URL.html#openConnection()) and casting the result to **`HttpURLConnection`**.
2. Prepare the request. The primary property of a request is its URI. Request headers may also include metadata such as credentials, preferred content types, and session cookies.
3. Optionally upload a request body. Instances must be configured with **`setDoOutput(true)`** (https://developer.android.com/reference/java/net/URLConnection.html#setDoOutput(boolean)) if they include a request body. Transmit data by writing to the stream returned by **`URLConnection.getOutputStream()`** (https://developer.android.com/reference/java/net/URLConnection.html#getOutputStream()).
4. Read the response. Response headers typically include metadata such as the response body's content type and length, modified dates and session cookies. The response body may be read from the stream returned by **`URLConnection.getInputStream()`**

(https://developer.android.com/reference/java/net/URLConnection.html#getInputStream()). If the response has no body, that method returns an empty stream.

5. Disconnect. Once the response body has been read, the `HttpURLConnection` should be closed by calling `disconnect()` (https://developer.android.com/reference/java/net/HttpURLConnection.html#disconnect()). Disconnecting releases the resources held by a connection so they may be closed or reused.

For example, to retrieve the webpage at `http://www.android.com/`:

```
URL url = new URL("http://www.android.com/");
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection()
try {
  InputStream in = new BufferedInputStream(urlConnection.getInputStream())
  readStream(in);
} finally {
  urlConnection.disconnect();
}
```

Secure Communication with HTTPS

Calling `URL.openConnection()` (https://developer.android.com/reference/java/net/URL.html#openConnection()) on a URL with the "https" scheme will return an `HttpsURLConnection`, which allows for overriding the default `HostnameVerifier` (https://developer.android.com/reference/javax/net/ssl/HostnameVerifier.html) and `SSLSocketFactory` (https://developer.android.com/reference/javax/net/ssl/SSLSocketFactory.html). An application-supplied `SSLSocketFactory` created from an `SSLContext` (https://developer.android.com/reference/javax/net/ssl/SSLContext.html) can provide a custom `X509TrustManager` (https://developer.android.com/reference/javax/net/ssl/X509TrustManager.html) for verifying certificate chains and a custom `X509KeyManager` (https://developer.android.com/reference/javax/net/ssl/X509KeyManager.html) for supplying client certificates. See `HttpsURLConnection` (https://developer.android.com/reference/javax/net/ssl/HttpsURLConnection.html) for more details.

Response Handling

`HttpURLConnection` will follow up to five HTTP redirects. It will follow redirects from one origin server to another. This implementation doesn't follow redirects from HTTPS to HTTP or vice versa.

If the HTTP response indicates that an error occurred, **URLConnection.getInputStream()** (https://developer.android.com/reference/java/net/URLConnection.html#getInputStream()) will throw an **IOException** (https://developer.android.com/reference/java/io/IOException.html). Use **getErrorStream()** (https://developer.android.com/reference/java/net/HttpURLConnection.html#getErrorStream()) to read the error response. The headers can be read in the normal way using **URLConnection.getHeaderFields()** (https://developer.android.com/reference/java/net/URLConnection.html#getHeaderFields()),

Posting Content

To upload data to a web server, configure the connection for output using **setDoOutput(true)** (https://developer.android.com/reference/java/net/URLConnection.html#setDoOutput(boolean)).

For best performance, you should call either **setFixedLengthStreamingMode(int)** (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFixedLengthStreamingMode(int)) when the body length is known in advance, or **setChunkedStreamingMode(int)** (https://developer.android.com/reference/java/net/HttpURLConnection.html#setChunkedStreamingMode(int)) when it is not. Otherwise **HttpURLConnection** will be forced to buffer the complete request body in memory before it is transmitted, wasting (and possibly exhausting) heap and increasing latency.

For example, to perform an upload:

```
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnectio
try {
  urlConnection.setDoOutput(true);
  urlConnection.setChunkedStreamingMode(0);

  OutputStream out = new BufferedOutputStream(urlConnection.getOutputStrea
  writeStream(out);

  InputStream in = new BufferedInputStream(urlConnection.getInputStream())
  readStream(in);
} finally {
  urlConnection.disconnect();
}
```

Performance

The input and output streams returned by this class are **not buffered**. Most callers should wrap the returned streams with `BufferedInputStream` (https://developer.android.com/reference/java/io/BufferedInputStream.html) or `BufferedOutputStream` (https://developer.android.com/reference/java/io/BufferedOutputStream.html). Callers that do only bulk reads or writes may omit buffering.

When transferring large amounts of data to or from a server, use streams to limit how much data is in memory at once. Unless you need the entire body to be in memory at once, process it as a stream (rather than storing the complete body as a single byte array or string).

To reduce latency, this class may reuse the same underlying `Socket` for multiple request/response pairs. As a result, HTTP connections may be held open longer than necessary. Calls to `disconnect()` (https://developer.android.com/reference/java/net/HttpURLConnection.html#disconnect()) may return the socket to a pool of connected sockets.

By default, this implementation of `HttpURLConnection` requests that servers use gzip compression and it automatically decompresses the data for callers of `URLConnection.getInputStream()` (https://developer.android.com/reference/java/net/URLConnection.html#getInputStream()). The Content-Encoding and Content-Length response headers are cleared in this case. Gzip compression can be disabled by setting the acceptable encodings in the request header:

```
urlConnection.setRequestProperty("Accept-Encoding", "identity");
```

Setting the Accept-Encoding request header explicitly disables automatic decompression and leaves the response headers intact; callers must handle decompression as needed, according to the Content-Encoding header of the response.

`URLConnection.getContentLength()` (https://developer.android.com/reference/java/net/URLConnection.html#getContentLength()) returns the number of bytes transmitted and cannot be used to predict how many bytes can be read from `URLConnection.getInputStream()` (https://developer.android.com/reference/java/net/URLConnection.html#getInputStream()) for compressed streams. Instead, read that stream until it is exhausted, i.e. when `InputStream.read()` (https://developer.android.com/reference/java/io/InputStream.html#read()) returns -1.

Handling Network Sign-On

Some Wi-Fi networks block Internet access until the user clicks through a sign-on page. Such sign-on pages are typically presented by using HTTP redirects. You can use **`URLConnection.getURL()`** (https://developer.android.com/reference/java/net/URLConnection.html#getURL()) to test if your connection has been unexpectedly redirected. This check is not valid until **after** the response headers have been received, which you can trigger by calling **`URLConnection.getHeaderFields()`** (https://developer.android.com/reference/java/net/URLConnection.html#getHeaderFields()) or **`URLConnection.getInputStream()`** (https://developer.android.com/reference/java/net/URLConnection.html#getInputStream()). For example, to check that a response was not redirected to an unexpected host:

```
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnectio
try {
  InputStream in = new BufferedInputStream(urlConnection.getInputStream())
  if (!url.getHost().equals(urlConnection.getURL().getHost())) {
    // we were redirected! Kick the user out to the browser to sign on?
  }
  ...
} finally {
  urlConnection.disconnect();
}
```

HTTP Authentication

`HttpURLConnection` supports HTTP basic authentication (http://www.ietf.org/rfc/rfc2617). Use **`Authenticator`** (https://developer.android.com/reference/java/net/Authenticator.html) to set the VM-wide authentication handler:

```
Authenticator.setDefault(new Authenticator() {
  protected PasswordAuthentication getPasswordAuthentication() {
    return new PasswordAuthentication(username, password.toCharArray());
  }
});
```

Unless paired with HTTPS, this is **not** a secure mechanism for user authentication. In particular, the username, password, request and response are all transmitted over the network without encryption.

Sessions with Cookies

To establish and maintain a potentially long-lived session between client and server, `HttpURLConnection` includes an extensible cookie manager. Enable VM-wide cookie management using `CookieHandler` (https://developer.android.com/reference/java/net/CookieHandler.html) and `CookieManager` (https://developer.android.com/reference/java/net/CookieManager.html):

```
CookieManager cookieManager = new CookieManager();
CookieHandler.setDefault(cookieManager);
```

By default, `CookieManager` accepts cookies from the origin server (http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html) only. Two other policies are included: `CookiePolicy.ACCEPT_ALL` (https://developer.android.com/reference/java/net/CookiePolicy.html#ACCEPT_ALL) and `CookiePolicy.ACCEPT_NONE` (https://developer.android.com/reference/java/net/CookiePolicy.html#ACCEPT_NONE). Implement `CookiePolicy` (https://developer.android.com/reference/java/net/CookiePolicy.html) to define a custom policy.

The default `CookieManager` keeps all accepted cookies in memory. It will forget these cookies when the VM exits. Implement `CookieStore` (https://developer.android.com/reference/java/net/CookieStore.html) to define a custom cookie store.

In addition to the cookies set by HTTP responses, you may set cookies programmatically. To be included in HTTP request headers, cookies must have the domain and path properties set.

By default, new instances of `HttpCookie` work only with servers that support RFC 2965 (http://www.ietf.org/rfc/rfc2965.txt) cookies. Many web servers support only the older specification, RFC 2109 (http://www.ietf.org/rfc/rfc2109.txt). For compatibility with the most web servers, set the cookie version to 0.

For example, to receive `www.twitter.com` in French:

```
HttpCookie cookie = new HttpCookie("lang", "fr");
cookie.setDomain("twitter.com");
cookie.setPath("/");
cookie.setVersion(0);
cookieManager.getCookieStore().add(new URI("http://twitter.com/"), cookie)
```

HTTP Methods

`HttpURLConnection` uses the `GET` method by default. It will use `POST` if `setDoOutput(true)` (https://developer.android.com/reference/java/net/URLConnection.html#setDoOutput(boolean)) has been called. Other HTTP methods (`OPTIONS`, `HEAD`, `PUT`, `DELETE` and `TRACE`) can be used with `setRequestMethod(String)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setRequestMethod(java.lang.String))
.

Proxies

By default, this class will connect directly to the origin server (http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html). It can also connect via an `HTTP` (https://developer.android.com/reference/java/net/Proxy.Type.html#HTTP) or `SOCKS` (https://developer.android.com/reference/java/net/Proxy.Type.html#SOCKS) proxy. To use a proxy, use `URL.openConnection(Proxy)` (https://developer.android.com/reference/java/net/URL.html#openConnection(java.net.Proxy)) when creating the connection.

IPv6 Support

This class includes transparent support for IPv6. For hosts with both IPv4 and IPv6 addresses, it will attempt to connect to each of a host's addresses until a connection is established.

Response Caching

Android 4.0 (Ice Cream Sandwich, API level 15) includes a response cache. See `android.net.http.HttpResponseCache` for instructions on enabling HTTP caching in your application.

Avoiding Bugs In Earlier Releases

Prior to Android 2.2 (Froyo), this class had some frustrating bugs. In particular, calling `close()` on a readable `InputStream` could poison the connection pool (http://code.google.com/p/android/issues/detail?id=2939). Work around this by disabling connection pooling:

```
  private void disableConnectionReuseIfNecessary() {
  // Work around pre-Froyo bugs in HTTP connection reuse.
  if (Integer.parseInt(Build.VERSION.SDK) < Build.VERSION_CODES.FROYO) {
    System.setProperty("http.keepAlive", "false");
  }
}
```

Each instance of `HttpURLConnection` may be used for one request/response pair. Instances of this class are not thread safe.

See also:

`disconnect()`
(https://developer.android.com/reference/java/net/HttpURLConnection.html#disconnect())

Summary

Constants

`int`	`HTTP_ACCEPTED` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_ACCEPTED) HTTP Status-Code 202: Accepted.
`int`	`HTTP_BAD_GATEWAY` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_BAD_GATEWAY) HTTP Status-Code 502: Bad Gateway.
`int`	`HTTP_BAD_METHOD` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_BAD_METHOD) HTTP Status-Code 405: Method Not Allowed.
`int`	`HTTP_BAD_REQUEST` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_BAD_REQUEST) HTTP Status-Code 400: Bad Request.

`int`	**HTTP_CLIENT_TIMEOUT** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_CLIENT_TIMEOUT) HTTP Status-Code 408: Request Time-Out.
`int`	**HTTP_CONFLICT** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_CONFLICT) HTTP Status-Code 409: Conflict.
`int`	**HTTP_CREATED** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_CREATED) HTTP Status-Code 201: Created.
`int`	**HTTP_ENTITY_TOO_LARGE** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_ENTITY_TOO_LARGE) HTTP Status-Code 413: Request Entity Too Large.
`int`	**HTTP_FORBIDDEN** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_FORBIDDEN) HTTP Status-Code 403: Forbidden.
`int`	**HTTP_GATEWAY_TIMEOUT** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_GATEWAY_TIMEOUT) HTTP Status-Code 504: Gateway Timeout.
`int`	**HTTP_GONE** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_GONE) HTTP Status-Code 410: Gone.
`int`	**HTTP_INTERNAL_ERROR** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_INTERNAL_ERROR) HTTP Status-Code 500: Internal Server Error.
`int`	**HTTP_LENGTH_REQUIRED** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_LENGTH_REQUIRED)

	HTTP Status-Code 411: Length Required.
`int`	**`HTTP_MOVED_PERM`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_MOVED_PERM) HTTP Status-Code 301: Moved Permanently.
`int`	**`HTTP_MOVED_TEMP`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_MOVED_TEMP) HTTP Status-Code 302: Temporary Redirect.
`int`	**`HTTP_MULT_CHOICE`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_MULT_CHOICE) HTTP Status-Code 300: Multiple Choices.
`int`	**`HTTP_NOT_ACCEPTABLE`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_NOT_ACCEPTABLE) HTTP Status-Code 406: Not Acceptable.
`int`	**`HTTP_NOT_AUTHORITATIVE`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_NOT_AUTHORITATIVE) HTTP Status-Code 203: Non-Authoritative Information.
`int`	**`HTTP_NOT_FOUND`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_NOT_FOUND) HTTP Status-Code 404: Not Found.
`int`	**`HTTP_NOT_IMPLEMENTED`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_NOT_IMPLEMENTED) HTTP Status-Code 501: Not Implemented.
`int`	**`HTTP_NOT_MODIFIED`** (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_NOT_MODIFIED) HTTP Status-Code 304: Not Modified.
`int`	**`HTTP_NO_CONTENT`**

	(https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_NO_CONTENT) HTTP Status-Code 204: No Content.
`int`	`HTTP_OK` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_OK) HTTP Status-Code 200: OK.
`int`	`HTTP_PARTIAL` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_PARTIAL) HTTP Status-Code 206: Partial Content.
`int`	`HTTP_PAYMENT_REQUIRED` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_PAYMENT_REQUIRED) HTTP Status-Code 402: Payment Required.
`int`	`HTTP_PRECON_FAILED` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_PRECON_FAILED) HTTP Status-Code 412: Precondition Failed.
`int`	`HTTP_PROXY_AUTH` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_PROXY_AUTH) HTTP Status-Code 407: Proxy Authentication Required.
`int`	`HTTP_REQ_TOO_LONG` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_REQ_TOO_LONG) HTTP Status-Code 414: Request-URI Too Large.
`int`	`HTTP_RESET` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_RESET) HTTP Status-Code 205: Reset Content.
`int`	`HTTP_SEE_OTHER` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_SEE_OTHER) HTTP Status-Code 303: See Other.

`int`	<u>`HTTP_SERVER_ERROR`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_SERVER_ERROR) *This constant was deprecated in API level 1. it is misplaced and shouldn't have existed.*
`int`	<u>`HTTP_UNAUTHORIZED`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_UNAUTHORIZED) HTTP Status-Code 401: Unauthorized.
`int`	<u>`HTTP_UNAVAILABLE`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_UNAVAILABLE) HTTP Status-Code 503: Service Unavailable.
`int`	<u>`HTTP_UNSUPPORTED_TYPE`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_UNSUPPORTED_TYPE) HTTP Status-Code 415: Unsupported Media Type.
`int`	<u>`HTTP_USE_PROXY`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_USE_PROXY) HTTP Status-Code 305: Use Proxy.
`int`	<u>`HTTP_VERSION`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#HTTP_VERSION) HTTP Status-Code 505: HTTP Version Not Supported.

Fields

`protected int`	<u>`chunkLength`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#chunkLength) The chunk-length when using chunked encoding streaming mode for output.
`protected int`	<u>`fixedContentLength`</u> (https://developer.android.com/reference/java/net/HttpURLConnection.html#fixedContentLength)

	The fixed content-length when using fixed-length streaming mode.
`protected long`	[`fixedContentLengthLong`](https://developer.android.com/reference/java/net/HttpURLConnection.html#fixedContentLengthLong) (https://developer.android.com/reference/java/net/HttpURLConnection.html#fixedContentLengthLong) The fixed content-length when using fixed-length streaming mode.
`protected boolean`	[`instanceFollowRedirects`](https://developer.android.com/reference/java/net/HttpURLConnection.html#instanceFollowRedirects) (https://developer.android.com/reference/java/net/HttpURLConnection.html#instanceFollowRedirects) If `true`, the protocol will automatically follow redirects.
`protected` [`String`](https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/lang/String.html)	[`method`](https://developer.android.com/reference/java/net/HttpURLConnection.html#method) (https://developer.android.com/reference/java/net/HttpURLConnection.html#method) The HTTP method (GET,POST,PUT,etc.).
`protected int`	[`responseCode`](https://developer.android.com/reference/java/net/HttpURLConnection.html#responseCode) (https://developer.android.com/reference/java/net/HttpURLConnection.html#responseCode) An `int` representing the three digit HTTP Status-Code.
`protected` [`String`](https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/lang/String.html)	[`responseMessage`](https://developer.android.com/reference/java/net/HttpURLConnection.html#responseMessage) (https://developer.android.com/reference/java/net/HttpURLConnection.html#responseMessage) The HTTP response message.

Inherited fields

From class `java.net.URLConnection` (https://developer.android.com/reference/java/net/URLC

`protected boolean`	`allowUserInte` (https://develope /reference/java/n ction.html#allowU n) If `true`, this URL examined in a co it makes sense to interactions such up an authenticat
`protected boolean`	`connected`

	(https://developer. ference/java/net/U html#connected) If `false`, this conr has not created a communications li specified URL.
`protected boolean`	[`doInput`](https://developer.ference/java/net/U html#doInput) (https://developer. ference/java/net/U html#doInput) This variable is set `setDoInput` meth
`protected boolean`	`doOutput` (https://developer. ference/java/net/U html#doOutput) This variable is set `setDoOutput` me
`protected long`	`ifModifiedSinc` (https://developer. ference/java/net/U html#ifModifiedSir Some protocols su the fetching of the the object has beer more recently than
`protected` URL (https://developer.android.com/reference/java/net/URL.html)	`url` (https://developer. ference/java/net/U html#url) The URL represent object on the Worl which this connect
`protected boolean`	`useCaches` (https://develope /reference/java/n ction.html#useCa If `true`, the proto to use caching wl

Protected constructors

`HttpURLConnection` (https://developer.android.com/reference/java/net/HttpURLConnection.html#HttpURLConnection(java.net.URL))
`(URL` (https://developer.android.com/reference/java/net/URL.html) `u)`

Constructor for the HttpURLConnection.

Public methods

`abstract void`	`disconnect` (https://developer.android.com/reference/java/net/HttpURLConnection.html#disconnect()) `()` Indicates that other requests to the server are unlikely in the near future.
`InputStream` (https://developer.android.com/reference/java/io/InputStream.html)	`getErrorStream` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getErrorStream()) `()` Returns the error stream if the connection failed but the server sent useful data nonetheless.
`static boolean`	`getFollowRedirects` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getFollowRedirects()) `()` Returns a `boolean` indicating whether or not HTTP redirects (3xx) should be automatically followed.
`String` (https://developer.android.com/reference/java/lang/String.html)	`getHeaderField` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getHeaderField(int)) `(int n)` Returns the value for the **n**th header field.
`long`	`getHeaderFieldDate` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getHeaderFieldDate(java.lang.String,%20long)) `(String`

	(https://developer.android.com/reference/java/lang/String.html) name, long Default) Returns the value of the named field parsed as date.
String (https://developer.android.com/reference/java/lang/String.html)	getHeaderFieldKey (https://developer.android.com/reference/java/net/HttpURLConnection.html#getHeaderFieldKey(int)) (int n) Returns the key for the nth header field.
boolean	getInstanceFollowRedirects (https://developer.android.com/reference/java/net/HttpURLConnection.html#getInstanceFollowRedirects()) () Returns the value of this HttpURLConnection's instanceFollowRedirects field.
Permission (https://developer.android.com/reference/java/security/Permission.html)	getPermission (https://developer.android.com/reference/java/net/HttpURLConnection.html#getPermission()) () Returns a SocketPermission (https://developer.android.com/reference/java/net/SocketPermission.html) object representing the permission necessary to connect to the destination host and port.
String (https://developer.android.com/reference/java/lang/String.html)	getRequestMethod (https://developer.android.com/reference/java/net/HttpURLConnection.html#getRequestMethod()) () Get the request method.
int	getResponseCode (https://developer.android.com/reference/java/net/HttpURLConnection.html#getResponseCode()) () Gets the status code from an HTTP response message.
String (https://developer.android.com/reference/java/lang/String.html)	getResponseMessage (https://developer.android.com/reference/java/net/HttpURLConnection.html#getResponseMessage()) ()

	Gets the HTTP response message, if any, returned along with the response code from a server.
`void`	`setChunkedStreamingMode` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setChunkedStreamingMode(int)) `(int chunklen)` This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is **not** known in advance.
`void`	`setFixedLengthStreamingMode` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFixedLengthStreamingMode(int)) `(int contentLength)` This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is known in advance.
`void`	`setFixedLengthStreamingMode` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFixedLengthStreamingMode(long)) `(long contentLength)` This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is known in advance.
`static void`	`setFollowRedirects` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFollowRedirects(boolean)) `(boolean set)` Sets whether HTTP redirects (requests with response code 3xx) should be automatically followed by this class.
`void`	`setInstanceFollowRedirects` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setInstanceFollowRedirects(boolean)) `(boolean followRedirects)` Sets whether HTTP redirects (requests with response code 3xx) should be automatically followed by this `HttpURLConnection` instance.
`void`	`setRequestMethod` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setRequestMethod(java.lang.String)) `(String`

(https://developer.android.com/reference/java/lang/String.html)
`method)`

Set the method for the URL request, one of:

- GET
- POST
- HEAD
- OPTIONS
- PUT
- DELETE
- TRACE

are legal, subject to protocol restrictions.

`abstract boolean`

`usingProxy`
(https://developer.android.com/reference/java/net/HttpURLConnection.html#usingProxy())
`()`

Indicates if the connection is going through a proxy.

Inherited methods

From class **`java.net.URLConnection`** (https://developer.android.com/reference/java/net/URLC

`void`

`abstract void`

`boolean`
`int`
`Object` (https://developer.android.com/reference/java/lang/Object.html)
`Object` (https://developer.android.com/reference/java/lang/Object.html)

`String` (https://developer.android.com/reference/java/lang/String.html)

`int`

`long`

`String` (https://developer.android.com/reference/java/lang/String.html)

`long`

`static boolean`

`static` [`String`](https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/lang/String.html)

`boolean`

`boolean`

`boolean`

`long`

`static` `FileNameMap` (https://developer.android.com/reference/java/net/FileNameMap.html)

`String` (https://developer.android.com/reference/java/lang/String.html)

`String` (https://developer.android.com/reference/java/lang/String.html)

`long`

`int`

`String` (https://developer.android.com/reference/java/lang/String.html)

`long`

`Map` (https://developer.android.com/reference/java/util/Map.html)<`String` (https://developer.andr
`List` (https://developer.android.com/reference/java/util/List.html)<`String` (https://developer.and

`long`
`InputStream` (https://developer.android.com/reference/java/io/InputStream.html)
`long`
`OutputStream` (https://developer.android.com/reference/java/io/OutputStream.html)
`Permission` (https://developer.android.com/reference/java/security/Permission.html)
`int`

Map (https://developer.android.com/reference/java/util/Map.html)<String (https://developer.andr
List (https://developer.android.com/reference/java/util/List.html)<String (https://developer.and

String (https://developer.android.com/reference/java/lang/String.html)

URL (https://developer.android.com/reference/java/net/URL.html)

boolean

`static` `String` (https://developer.android.com/reference/java/lang/String.html)

`static` `String` (https://developer.android.com/reference/java/lang/String.html)

`void`

`void`

```
static void
```

```
static void
```

```
static void
```

```
void
```

```
void
```

```
void
```

```
static void
```

`void`

`void`

`void`

`void`

`String` (https://developer.android.com/reference/java/lang/String.html)	

From class `java.lang.Object` (https://developer.android.com/reference/java/lang/Object.html)

`Object` (https://developer.android.com/reference/java/lang/Object.html)	`clone` (https Creates and r
`boolean`	`equals` (https://develo (`Object` (http Indicates whet
`void`	`finalize` (htt Called by the g no more refere
`final Class` (https://developer.android.com/reference/java/lang/Class.html)`<?>`	`getClass` (htt Returns the rur
`int`	`hashCode` (htt Returns a hash
`final void`	`notify` (https Wakes up a sir
`final void`	`notifyAll` (h Wakes up all th
`String` (https://developer.android.com/reference/java/lang/String.html)	`toString` (htt Returns a strin
`final void`	`wait` (https://c `millis, int`

	Causes the cur (https://develc **notifyAll()** method for this real time has e
`final void`	**wait** (https://c **millis)** Causes the cur (https://develc **notifyAll()** method for this
`final void`	**wait** (https:/ Causes the cu (https://deve **notifyAll(** method for th

Constants

HTTP_ACCEPTED

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 202: Accepted.

Constant Value: 202 (0x000000ca)

HTTP_BAD_GATEWAY

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 502: Bad Gateway.

Constant Value: 502 (0x000001f6)

HTTP_BAD_METHOD

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 405: Method Not Allowed.

Constant Value: 405 (0x00000195)

HTTP_BAD_REQUEST

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 400: Bad Request.

Constant Value: 400 (0x00000190)

HTTP_CLIENT_TIMEOUT

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 408: Request Time-Out.

Constant Value: 408 (0x00000198)

HTTP_CONFLICT

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 409: Conflict.

Constant Value: 409 (0x00000199)

HTTP_CREATED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 201: Created.

Constant Value: 201 (0x000000c9)

HTTP_ENTITY_TOO_LARGE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 413: Request Entity Too Large.

Constant Value: 413 (0x0000019d)

HTTP_FORBIDDEN

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 403: Forbidden.

Constant Value: 403 (0x00000193)

HTTP_GATEWAY_TIMEOUT

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 504: Gateway Timeout.

Constant Value: 504 (0x000001f8)

HTTP_GONE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 410: Gone.

Constant Value: 410 (0x0000019a)

HTTP_INTERNAL_ERROR

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 500: Internal Server Error.

Constant Value: 500 (0x000001f4)

HTTP_LENGTH_REQUIRED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 411: Length Required.

Constant Value: 411 (0x0000019b)

HTTP_MOVED_PERM

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 301: Moved Permanently.

Constant Value: 301 (0x0000012d)

HTTP_MOVED_TEMP

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 302: Temporary Redirect.

Constant Value: 302 (0x0000012e)

HTTP_MULT_CHOICE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 300: Multiple Choices.

Constant Value: 300 (0x0000012c)

HTTP_NOT_ACCEPTABLE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 406: Not Acceptable.

Constant Value: 406 (0x00000196)

HTTP_NOT_AUTHORITATIVE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 203: Non-Authoritative Information.

Constant Value: 203 (0x000000cb)

HTTP_NOT_FOUND

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 404: Not Found.

Constant Value: 404 (0x00000194)

HTTP_NOT_IMPLEMENTED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 501: Not Implemented.

Constant Value: 501 (0x000001f5)

HTTP_NOT_MODIFIED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 304: Not Modified.

Constant Value: 304 (0x00000130)

HTTP_NO_CONTENT

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 204: No Content.

Constant Value: 204 (0x000000cc)

HTTP_OK

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 200: OK.

Constant Value: 200 (0x000000c8)

HTTP_PARTIAL

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 206: Partial Content.

Constant Value: 206 (0x000000ce)

HTTP_PAYMENT_REQUIRED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 402: Payment Required.

Constant Value: 402 (0x00000192)

HTTP_PRECON_FAILED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 412: Precondition Failed.

Constant Value: 412 (0x0000019c)

HTTP_PROXY_AUTH

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 407: Proxy Authentication Required.

Constant Value: 407 (0x00000197)

HTTP_REQ_TOO_LONG

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 414: Request-URI Too Large.

Constant Value: 414 (0x0000019e)

HTTP_RESET

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 205: Reset Content.

Constant Value: 205 (0x000000cd)

HTTP_SEE_OTHER

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 303: See Other.

Constant Value: 303 (0x0000012f)

HTTP_SERVER_ERROR

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

This constant was deprecated in API level 1.
it is misplaced and shouldn't have existed.

HTTP Status-Code 500: Internal Server Error.

Constant Value: 500 (0x000001f4)

HTTP_UNAUTHORIZED

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 401: Unauthorized.

Constant Value: 401 (0x00000191)

HTTP_UNAVAILABLE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 503: Service Unavailable.

Constant Value: 503 (0x000001f7)

HTTP_UNSUPPORTED_TYPE

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 415: Unsupported Media Type.

Constant Value: 415 (0x0000019f)

HTTP_USE_PROXY

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 305: Use Proxy.

Constant Value: 305 (0x00000131)

HTTP_VERSION

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

HTTP Status-Code 505: HTTP Version Not Supported.

Constant Value: 505 (0x000001f9)

Fields

chunkLength

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

The chunk-length when using chunked encoding streaming mode for output. A value of -1 means chunked encoding is disabled for output.

fixedContentLength

`protected int fixedContentLength`

added in API level 1
 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

The fixed content-length when using fixed-length streaming mode. A value of `-1` means fixed-length streaming mode is disabled for output.

NOTE: `fixedContentLengthLong`
 (https://developer.android.com/reference/java/net/HttpURLConnection.html#fixedContentLengthLong) is recommended instead of this field, as it allows larger content lengths to be set.

fixedContentLengthLong

added in API level 19
 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

The fixed content-length when using fixed-length streaming mode. A value of `-1` means fixed-length streaming mode is disabled for output.

instanceFollowRedirects

added in API level 1
 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

If `true`, the protocol will automatically follow redirects. If `false`, the protocol will not automatically follow redirects.

This field is set by the `setInstanceFollowRedirects` method. Its value is returned by the `getInstanceFollowRedirects` method.

Its default value is based on the value of the static followRedirects at HttpURLConnection construction time.

See also:

`setInstanceFollowRedirects(boolean)`
 (https://developer.android.com/reference/java/net/HttpURLConnection.html#setInstanceFollowRedirects(boolean))

`getInstanceFollowRedirects()`
 (https://developer.android.com/reference/java/net/HttpURLConnection.html#getInstanceFollowRedirects())

`setFollowRedirects(boolean)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFollowRedirects(boolean))

method

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

The HTTP method (GET,POST,PUT,etc.).

responseCode

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

An `int` representing the three digit HTTP Status-Code.

- 1xx: Informational
- 2xx: Success
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

responseMessage

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

The HTTP response message.

Protected constructors

HttpURLConnection

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Constructor for the HttpURLConnection.

Parameters

u	`URL`: the URL

Public methods

disconnect

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Indicates that other requests to the server are unlikely in the near future. Calling disconnect() should not imply that this HttpURLConnection instance can be reused for other requests.

getErrorStream

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns the error stream if the connection failed but the server sent useful data nonetheless. The typical example is when an HTTP server responds with a 404, which will cause a FileNotFoundException to be thrown in connect, but the server sent an HTML help page with suggestions as to what to do.

This method will not cause a connection to be initiated. If the connection was not connected, or if the server did not have an error while connecting or if the server had an error but no error data was sent, this method will return null. This is the default.

Returns

`InputStream` (https://developer.android.com/reference/java/io/InputStream.html)	an error stream if any, null if there have been no errors, the connection is not connected or the server sent no useful data.

getFollowRedirects

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns a `boolean` indicating whether or not HTTP redirects (3xx) should be automatically followed.

Returns

`boolean`	`true` if HTTP redirects should be automatically followed, `false` if not.

See also:

`setFollowRedirects(boolean)`
(https://developer.android.com/reference/java/net/HttpURLConnection.html#setFollowRedirects(boolean))

getHeaderField

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns the value for the `n`th header field. Some implementations may treat the `0`th header field as special, i.e. as the status line returned by the HTTP server.

This method can be used in conjunction with the `getHeaderFieldKey` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getHeaderFieldKey(int)) method to iterate through all the headers in the message.

Parameters

n	`int`: an index, where `n>=0`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the value of the `n`th header field, or `null` if the value does not exist.

See also:

getHeaderFieldKey(int)
(https://developer.android.com/reference/java/net/HttpURLConnection.html#getHeaderFieldKey(int))

getHeaderFieldDate

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns the value of the named field parsed as date. The result is the number of milliseconds since January 1, 1970 GMT represented by the named field.

This form of `getHeaderField` exists because some connection types (e.g., `http-ng`) have pre-parsed headers. Classes for that connection type can override this method and short-circuit the parsing.

Parameters

`name`	`String`: the name of the header field.
`Default`	`long`: a default value.

Returns

`long`	the value of the field, parsed as a date. The value of the `Default` argument is returned if the field is missing or malformed.

getHeaderFieldKey

added in API level 1
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns the key for the `n`th header field. Some implementations may treat the `0`th header field as special, i.e. as the status line returned by the HTTP server. In this case, getHeaderField(0)
(https://developer.android.com/reference/java/net/HttpURLConnection.html#getHeaderField(int))
returns the status line, but `getHeaderFieldKey(0)` returns null.

Parameters

`n`	`int`: an index, where `n >=0`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the key for the `n`th header field, or `null` if the key does not exist.

getInstanceFollowRedirects

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns the value of this `HttpURLConnection`'s `instanceFollowRedirects` field.

Returns

`boolean`	the value of this `HttpURLConnection`'s `instanceFollowRedirects` field.

See also:

`instanceFollowRedirects` (https://developer.android.com/reference/java/net/HttpURLConnection.html#instanceFollowRedirects)

`setInstanceFollowRedirects(boolean)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setInstanceFollowRedirects(boolean))

getPermission

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Returns a `SocketPermission` (https://developer.android.com/reference/java/net/SocketPermission.html) object representing the permission necessary to connect to the destination host and port.

Returns

`Permission` (https://developer.android.com/reference/java/security/Permission.html)	a `SocketPermission` object representing the permission necessary to connect to the destination host and port.

Throws

`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an error occurs while computing the permission.

getRequestMethod

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Get the request method.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the HTTP request method

See also:

`setRequestMethod(java.lang.String)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setRequestMethod(java.lang.String))

getResponseCode

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Gets the status code from an HTTP response message. For example, in the case of the following status lines:

```
HTTP/1.0 200 OK
HTTP/1.0 401 Unauthorized
```

It will return 200 and 401 respectively. Returns -1 if no code can be discerned from the response (i.e., the response is not valid HTTP).

Returns

`int`	the HTTP Status-Code, or -1

Throws

`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an error occurred connecting to the server.

getResponseMessage

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Gets the HTTP response message, if any, returned along with the response code from a server. From responses like:

```
HTTP/1.0 200 OK
HTTP/1.0 404 Not Found
```

Extracts the Strings "OK" and "Not Found" respectively. Returns null if none could be discerned from the responses (the result was not valid HTTP).

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the HTTP response message, or `null`

Throws

`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an error occurred connecting to the server.

setChunkedStreamingMode

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is **not** known in advance. In this mode, chunked transfer encoding is used to send the request body. Note, not all HTTP servers support this mode.

When output streaming is enabled, authentication and redirection cannot be handled automatically. A HttpRetryException will be thrown when reading the response if authentication or redirection are required. This exception can be queried for the details of the error.

This method must be called before the URLConnection is connected.

Parameters

`chunklen`	`int`: The number of bytes to write in each chunk. If chunklen is less than or equal to zero, a default value will be used.

Throws

`IllegalStateException` (https://developer.android.com/reference/java/lang/IllegalStateException.html)	if URLConnection is already connected or if a different streaming mode is already enabled.

See also:

`setFixedLengthStreamingMode(int)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFixedLengthStreamingMode(int))

setFixedLengthStreamingMode

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is known in advance.

An exception will be thrown if the application attempts to write more data than the indicated content-length, or if the application closes the OutputStream before writing the indicated amount.

When output streaming is enabled, authentication and redirection cannot be handled automatically. A HttpRetryException will be thrown when reading the response if authentication or redirection are required. This exception can be queried for the details of the error.

This method must be called before the URLConnection is connected.

NOTE: `setFixedLengthStreamingMode(long)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFixedLengthStreamingMode(long)) is recommended instead of this method as it allows larger content lengths to be set.

Parameters

`contentLength`	`int`: The number of bytes which will be written to the OutputStream.

Throws

`IllegalStateException` (https://developer.android.com/reference/java/lang/IllegalStateException.html)	if URLConnection is already connected or if a different streaming mode is already enabled.
`IllegalArgumentException` (https://developer.android.com/reference/java/lang/IllegalArgumentException.html)	if a content length less than zero is specified.

See also:

`setChunkedStreamingMode(int)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setChunkedStreamingMode(int))

setFixedLengthStreamingMode

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

This method is used to enable streaming of a HTTP request body without internal buffering, when the content length is known in advance.

An exception will be thrown if the application attempts to write more data than the indicated content-length, or if the application closes the OutputStream before writing the indicated amount.

When output streaming is enabled, authentication and redirection cannot be handled automatically. A HttpRetryException (https://developer.android.com/reference/java/net/HttpRetryException.html) will be thrown when reading the response if authentication or redirection are required. This exception can be queried for the details of the error.

This method must be called before the URLConnection is connected.

The content length set by invoking this method takes precedence over any value set by `setFixedLengthStreamingMode(int)` (https://developer.android.com/reference/java/net/HttpURLConnection.html#setFixedLengthStreamingMode(int))
.

Parameters

`contentLength`	`long`: The number of bytes which will be written to the OutputStream.

Throws

`IllegalStateException` (https://developer.android.com/reference/java/lang/IllegalStateException.html)	if URLConnection is already connected or if a different streaming mode is already enabled.
`IllegalArgumentException` (https://developer.android.com/reference/java/lang/IllegalArgumentException.html)	if a content length less than zero is specified.

setFollowRedirects

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Sets whether HTTP redirects (requests with response code 3xx) should be automatically followed by this class. True by default. Applets cannot change this variable.

If there is a security manager, this method first calls the security manager's `checkSetFactory` method to ensure the operation is allowed. This could result in a SecurityException.

Parameters

`set`	`boolean`: a `boolean` indicating whether or not to follow HTTP redirects.

Throws

`SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html)	if a security manager exists and its `checkSetFactory` method doesn't allow the operation.

See also:

`SecurityManager.checkSetFactory()` (https://developer.android.com/reference/java/lang/SecurityManager.html#checkSetFactory())

`getFollowRedirects()` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getFollowRedirects())

setInstanceFollowRedirects

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Sets whether HTTP redirects (requests with response code 3xx) should be automatically followed by this `HttpURLConnection` instance.

The default value comes from followRedirects, which defaults to true.

Parameters

`followRedirects`	`boolean`: a `boolean` indicating whether or not to follow HTTP redirects.

See also:

`instanceFollowRedirects`
 (https://developer.android.com/reference/java/net/HttpURLConnection.html#instanceFollowRedirects)

`getInstanceFollowRedirects()`
 (https://developer.android.com/reference/java/net/HttpURLConnection.html#getInstanceFollowRedirects())

setRequestMethod

added in API level 1
 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Set the method for the URL request, one of:

- GET
- POST
- HEAD
- OPTIONS
- PUT
- DELETE
- TRACE

are legal, subject to protocol restrictions. The default method is GET.

Parameters

`method`	`String`: the HTTP method

Throws

`ProtocolException`	if the method cannot be reset or if the requested method isn't valid for HTTP.

(https://developer.android.com/reference/java/net/ProtocolException.html)	
`SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html)	if a security manager is set and the method is "TRACE", but the "allowHttpTrace" NetPermission is not granted.

See also:

`getRequestMethod()` (https://developer.android.com/reference/java/net/HttpURLConnection.html#getRequestMethod())

usingProxy

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Indicates if the connection is going through a proxy.

Returns

`boolean`	a boolean indicating if the connection is using a proxy.

Last updated June 6, 2018.

Twitter
Follow @AndroidDev on Twitter

Google+
Follow Android Developers on Google+

YouTube
Check out Android Developers on YouTube