Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

HttpsURLConnection

```
public abstract class HttpsURLConnection
extends HttpURLConnection (/reference/java/net/HttpURLConnection)
```

java.lang.Object (/reference/java/lang/Object)
  ↳ java.net.URLConnection (/reference/java/net/URLConnection)
    ↳ java.net.HttpURLConnection (/reference/java/net/HttpURLConnection)
      ↳ javax.net.ssl.HttpsURLConnection

`HttpsURLConnection` extends `HttpURLConnection` with support for https-specific features.

See http://www.w3.org/pub/WWW/Protocols/ (http://www.w3.org/pub/WWW/Protocols/) and RFC 2818 (http://www.ietf.org/) for more details on the https specification.

This class uses `HostnameVerifier` and `SSLSocketFactory`. There are default implementations defined for both classes. However, the implementations can be replaced on a per-class (static) or per-instance basis. All new `HttpsURLConnection`s instances will be assigned the "default" static values at instance creation, but they can be overriden by calling the appropriate per-instance set method(s) before `connect`ing.

Summary

Inherited constants

From class `java.net.HttpURLConnection` (/reference/java/net/HttpURLConnection)

`int`	`HTTP_ACCEPTED` (/reference/java/net/HttpURLConnection#HTTP_ACCEPTED) HTTP Status-Code 202: Accepted.

int	HTTP_BAD_GATEWAY (/reference/java/net/HttpURLConnection#HTTP_BAD_GATEWAY) HTTP Status-Code 502: Bad Gateway.
int	HTTP_BAD_METHOD (/reference/java/net/HttpURLConnection#HTTP_BAD_METHOD) HTTP Status-Code 405: Method Not Allowed.
int	HTTP_BAD_REQUEST (/reference/java/net/HttpURLConnection#HTTP_BAD_REQUEST) HTTP Status-Code 400: Bad Request.
int	HTTP_CLIENT_TIMEOUT (/reference/java/net/HttpURLConnection#HTTP_CLIENT_TIMEOUT) HTTP Status-Code 408: Request Time-Out.
int	HTTP_CONFLICT (/reference/java/net/HttpURLConnection#HTTP_CONFLICT) HTTP Status-Code 409: Conflict.
int	HTTP_CREATED (/reference/java/net/HttpURLConnection#HTTP_CREATED) HTTP Status-Code 201: Created.
int	HTTP_ENTITY_TOO_LARGE (/reference/java/net/HttpURLConnection#HTTP_ENTITY_TOO_LARGE) HTTP Status-Code 413: Request Entity Too Large.
int	HTTP_FORBIDDEN (/reference/java/net/HttpURLConnection#HTTP_FORBIDDEN) HTTP Status-Code 403: Forbidden.

int	HTTP_GATEWAY_TIMEOUT (/reference/java/net/HttpURLConnection#HTTP_GATEWAY_TIMEOUT) HTTP Status-Code 504: Gateway Timeout.
int	HTTP_GONE (/reference/java/net/HttpURLConnection#HTTP_GONE) HTTP Status-Code 410: Gone.
int	HTTP_INTERNAL_ERROR (/reference/java/net/HttpURLConnection#HTTP_INTERNAL_ERROR) HTTP Status-Code 500: Internal Server Error.
int	HTTP_LENGTH_REQUIRED (/reference/java/net/HttpURLConnection#HTTP_LENGTH_REQUIRED) HTTP Status-Code 411: Length Required.
int	HTTP_MOVED_PERM (/reference/java/net/HttpURLConnection#HTTP_MOVED_PERM) HTTP Status-Code 301: Moved Permanently.
int	HTTP_MOVED_TEMP (/reference/java/net/HttpURLConnection#HTTP_MOVED_TEMP) HTTP Status-Code 302: Temporary Redirect.
int	HTTP_MULT_CHOICE (/reference/java/net/HttpURLConnection#HTTP_MULT_CHOICE) HTTP Status-Code 300: Multiple Choices.
int	HTTP_NOT_ACCEPTABLE (/reference/java/net/HttpURLConnection#HTTP_NOT_ACCEPTABLE) HTTP Status-Code 406: Not Acceptable.
int	HTTP_NOT_AUTHORITATIVE (/reference/java/net/HttpURLConnection#HTTP_NOT_AUTHORITATIVE)

	HTTP Status-Code 203: Non-Authoritative Information.
int	HTTP_NOT_FOUND (/reference/java/net/HttpURLConnection#HTTP_NOT_FOUND) HTTP Status-Code 404: Not Found.
int	HTTP_NOT_IMPLEMENTED (/reference/java/net/HttpURLConnection#HTTP_NOT_IMPLEMENTED) HTTP Status-Code 501: Not Implemented.
int	HTTP_NOT_MODIFIED (/reference/java/net/HttpURLConnection#HTTP_NOT_MODIFIED) HTTP Status-Code 304: Not Modified.
int	HTTP_NO_CONTENT (/reference/java/net/HttpURLConnection#HTTP_NO_CONTENT) HTTP Status-Code 204: No Content.
int	HTTP_OK (/reference/java/net/HttpURLConnection#HTTP_OK) HTTP Status-Code 200: OK.
int	HTTP_PARTIAL (/reference/java/net/HttpURLConnection#HTTP_PARTIAL) HTTP Status-Code 206: Partial Content.
int	HTTP_PAYMENT_REQUIRED (/reference/java/net/HttpURLConnection#HTTP_PAYMENT_REQUIRED) HTTP Status-Code 402: Payment Required.
int	HTTP_PRECON_FAILED (/reference/java/net/HttpURLConnection#HTTP_PRECON_FAILED) HTTP Status-Code 412: Precondition Failed.

int	HTTP_PROXY_AUTH (/reference/java/net/HttpURLConnection#HTTP_PROXY_AUTH) HTTP Status-Code 407: Proxy Authentication Required.
int	HTTP_REQ_TOO_LONG (/reference/java/net/HttpURLConnection#HTTP_REQ_TOO_LONG) HTTP Status-Code 414: Request-URI Too Large.
int	HTTP_RESET (/reference/java/net/HttpURLConnection#HTTP_RESET) HTTP Status-Code 205: Reset Content.
int	HTTP_SEE_OTHER (/reference/java/net/HttpURLConnection#HTTP_SEE_OTHER) HTTP Status-Code 303: See Other.
int	HTTP_SERVER_ERROR (/reference/java/net/HttpURLConnection#HTTP_SERVER_ERROR) *This constant was deprecated in API level 15. it is misplaced and shouldn't have existed.*
int	HTTP_UNAUTHORIZED (/reference/java/net/HttpURLConnection#HTTP_UNAUTHORIZED) HTTP Status-Code 401: Unauthorized.
int	HTTP_UNAVAILABLE (/reference/java/net/HttpURLConnection#HTTP_UNAVAILABLE) HTTP Status-Code 503: Service Unavailable.
int	HTTP_UNSUPPORTED_TYPE (/reference/java/net/HttpURLConnection#HTTP_UNSUPPORTED_TYPE) HTTP Status-Code 415: Unsupported Media Type.

`int`	`HTTP_USE_PROXY` (/reference/java/net/HttpURLConnection#HTTP_USE_PROXY) HTTP Status-Code 305: Use Proxy.
`int`	`HTTP_VERSION` (/reference/java/net/HttpURLConnection#HTTP_VERSION) HTTP Status-Code 505: HTTP Version Not Supported.

Fields

`protected HostnameVerifier` (/reference/javax/net/ssl/HostnameVerifier)	`hostnameVerifier` (/reference/javax/net/ssl/HttpsURLConnection#hostnameVerif The **`hostnameVerifier`** for this object.

Inherited fields

From class `java.net.HttpURLConnection` (/reference/java/net/HttpURLConnection)

`protected int`	`chunkLength` (/reference/java/net/HttpURLConnection#chunkLengt The chunk-length when using chunked encoding strea for output.
`protected int`	`fixedContentLength` (/reference/java/net/HttpURLConnection#fixedConter The fixed content-length when using fixed-length stre
`protected long`	`fixedContentLengthLong` (/reference/java/net/HttpURLConnection#fixedConter The fixed content-length when using fixed-length stre

`protected boolean`	`instanceFollowRedirects` (/reference/java/net/HttpURLConnection#instanceFoll If `true`, the protocol will automatically follow redirects
`protected String` (/reference/java/lang/String)	`method` (/reference/java/net/HttpURLConnection#met The HTTP method (GET,POST,PUT,etc.).
`protected int`	`responseCode` (/reference/java/net/HttpURLConnection#responseCo An `int` representing the three digit HTTP Status-Code
`protected String` (/reference/java/lang/String)	`responseMessage` (/reference/java/net/HttpURLConnection#responseMe The HTTP response message.

From class `java.net.URLConnection` (/reference/java/net/URLConnection)

`protected boolean`	`allowUserInteraction` (/reference/java/net/URLConnection#allowUserInteraction) If `true`, this `URL` is being examined in a context in which it ma allow user interactions such as popping up an authentication
`protected boolean`	`connected` (/reference/java/net/URLConnection#connected) If `false`, this connection object has not created a communic the specified URL.
`protected boolean`	`doInput` (/reference/java/net/URLConnection#doInput) This variable is set by the `setDoInput` method.
`protected boolean`	`doOutput` (/reference/java/net/URLConnection#doOutput) This variable is set by the `setDoOutput` method.
`protected long`	`ifModifiedSince` (/reference/java/net/URLConnection#ifM

	Some protocols support skipping the fetching of the object u object has been modified more recently than a certain time.
`protected` URL (/reference/java/net/URL)	`url` (/reference/java/net/URLConnection#url) The URL represents the remote object on the World Wide Web connection is opened.
`protected boolean`	useCaches (/reference/java/net/URLConnection#useCaches If `true`, the protocol is allowed to use caching whenever it ca

Protected constructors

HttpsURLConnection (/reference/javax/net/ssl/HttpsURLConnection#HttpsURLConnection(java.net.URL))(URL (/reference/java/net/URL) `url`)

Creates an `HttpsURLConnection` using the URL specified.

Public methods

`abstract` String (/reference/java/lang/String)	getCipherSuite (/reference/javax/net/ssl/HttpsURLConnect Returns the cipher suite in use on this connection.
`static` HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier)	getDefaultHostnameVerifier (/reference/javax/net/ssl/HttpsURLConnection#getDefaultHos Gets the default `HostnameVerifier` that is inherited by new
`static` SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory)	getDefaultSSLSocketFactory (/reference/javax/net/ssl/HttpsURLConnection#getDefaultSSL Gets the default static `SSLSocketFactory` that is inherited b

HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier)	getHostnameVerifier (/reference/javax/net/ssl/HttpsURLC Gets the HostnameVerifier in place on this instance.
abstract Certificate[] (/reference/java/security/cert/Certificate)	getLocalCertificates (/reference/javax/net/ssl/HttpsURL Returns the certificate(s) that were sent to the server during h
Principal (/reference/java/security/Principal)	getLocalPrincipal (/reference/javax/net/ssl/HttpsURLCon Returns the principal that was sent to the server during hands
Principal (/reference/java/security/Principal)	getPeerPrincipal (/reference/javax/net/ssl/HttpsURLConn Returns the server's principal which was established as part o
SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory)	getSSLSocketFactory (/reference/javax/net/ssl/HttpsURLC Gets the SSL socket factory to be used when creating sockets
abstract Certificate[] (/reference/java/security/cert/Certificate)	getServerCertificates (/reference/javax/net/ssl/HttpsUR Returns the server's certificate chain which was established a
static void	setDefaultHostnameVerifier (/reference/javax/net/ssl/HttpsURLConnection#setDefaultHos (HostnameVerifier (/reference/javax/net/ssl/HostnameVeri Sets the default HostnameVerifier inherited by a new insta
static void	setDefaultSSLSocketFactory (/reference/javax/net/ssl/HttpsURLConnection#setDefaultSSL (SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFac Sets the default SSLSocketFactory inherited by new instan
void	setHostnameVerifier (/reference/javax/net/ssl/HttpsURLConnection#setHostnameV (HostnameVerifier (/reference/javax/net/ssl/HostnameVeri Sets the HostnameVerifier for this instance.

`void`	`setSSLSocketFactory` (/reference/javax/net/ssl/HttpsURLConnection#setSSLSocket (`SSLSocketFactory` (/reference/javax/net/ssl/SSLSocketFac Sets the `SSLSocketFactory` to be used when this instance c connections.

Inherited methods

From class `java.net.HttpURLConnection` (/reference/java/net/HttpURLConnection)

`abstract void`	`disconnect` (/reference/java/net/HttpURLConnectior Indicates that other requests to the server are unlikel
`InputStream` (/reference/java/io/InputStream)	`getErrorStream` (/reference/java/net/HttpURLConne Returns the error stream if the connection failed but t
`static boolean`	`getFollowRedirects` (/reference/java/net/HttpURL Returns a `boolean` indicating whether or not HTTP re
`String` (/reference/java/lang/String)	`getHeaderField` (/reference/java/net/HttpURLConne Returns the value for the **n**th header field.
`long`	`getHeaderFieldDate` (/reference/java/net/HttpURL Returns the value of the named field parsed as date.
`String` (/reference/java/lang/String)	`getHeaderFieldKey` (/reference/java/net/HttpURLC Returns the key for the **n**th header field.
`boolean`	`getInstanceFollowRedirects` (/reference/java/ne Returns the value of this `HttpURLConnection`'s `ins`

Permission (/reference/java/security/Permission)	getPermission (/reference/java/net/HttpURLConnec Returns a SocketPermission (/reference/java/net/S
String (/reference/java/lang/String)	getRequestMethod (/reference/java/net/HttpURLCo Get the request method.
int	getResponseCode (/reference/java/net/HttpURLCon Gets the status code from an HTTP response message
String (/reference/java/lang/String)	getResponseMessage (/reference/java/net/HttpURL Gets the HTTP response message, if any, returned alo
void	setChunkedStreamingMode (/reference/java/net/Ht This method is used to enable streaming of a HTTP re
void	setFixedLengthStreamingMode (/reference/java/n This method is used to enable streaming of a HTTP re
void	setFixedLengthStreamingMode (/reference/java/n This method is used to enable streaming of a HTTP re
static void	setFollowRedirects (/reference/java/net/HttpURL Sets whether HTTP redirects (requests with response
void	setInstanceFollowRedirects (/reference/java/ne Sets whether HTTP redirects (requests with response
void	setRequestMethod (/reference/java/net/HttpURLCo 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` (/reference/java/net/HttpURLConnectior Indicates if the connection is going through a proxy.

From class `java.net.URLConnection` (/reference/java/net/URLConnection)

`void`

`abstract void`

`boolean`

`int`

`Object` (/reference/java/lang/Object)

`Object` (/reference/java/lang/Object)

`String` (/reference/java/lang/String)

`int`

`long`

`String` (/reference/java/lang/String)

`long`

`static boolean`

`static String` (/reference/java/lang/String)

`boolean`

`boolean`

`boolean`

`long`

`static FileNameMap` (/reference/java/net/FileNameMap)

`String` (/reference/java/lang/String)

`String` (/reference/java/lang/String)

`long`

`int`

`String` (/reference/java/lang/String)

`long`

`Map` (/reference/java/util/Map)<`String` (/reference/java/lang/String), `List` (/reference/java/util/List)<S

`long`

`InputStream` (/reference/java/io/InputStream)

`long`

`OutputStream` (/reference/java/io/OutputStream)

`Permission` (/reference/java/security/Permission)

`int`

`Map` (/reference/java/util/Map)<`String` (/reference/java/lang/String), `List` (/reference/java/util/List)<S

`String` (/reference/java/lang/String)

URL (/reference/java/net/URL)

boolean

static String (/reference/java/lang/String)

static String (/reference/java/lang/String)

void

void

static void

static void

static void

void

void

void

static void

void

void

void

void	
String (/reference/java/lang/String)	

From class java.lang.Object (/reference/java/lang/Object)

Object (/reference/java/lang/Object)	clone (/reference/java/lang/Object#clone())() Creates and returns a copy of this object.
boolean	equals (/reference/java/lang/Object#equals(java.lang.Ob Indicates whether some other object is "equal to" this one
void	finalize (/reference/java/lang/Object#finalize())() Called by the garbage collector on an object when garbag
final Class (/reference/java/lang/Class)<?>	getClass (/reference/java/lang/Object#getClass())() Returns the runtime class of this Object.
int	hashCode (/reference/java/lang/Object#hashCode())() Returns a hash code value for the object.
final void	notify (/reference/java/lang/Object#notify())() Wakes up a single thread that is waiting on this object's m
final void	notifyAll (/reference/java/lang/Object#notifyAll())() Wakes up all threads that are waiting on this object's mon

String (/reference/java/lang/String)	toString (/reference/java/lang/Object#toString())() Returns a string representation of the object.
final void	wait (/reference/java/lang/Object#wait(long,%20int))(lo Causes the current thread to wait until it is awakened, typ
final void	wait (/reference/java/lang/Object#wait(long))(long tir Causes the current thread to wait until it is awakened, typ
final void	wait (/reference/java/lang/Object#wait())() Causes the current thread to wait until it is awakened, typ

Fields

hostnameVerifier

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
protected HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier) hostnameVerifier
```

The `hostnameVerifier` for this object.

Protected constructors

HttpsURLConnection

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
protected HttpsURLConnection (URL (/reference/java/net/URL) url)
```

Creates an `HttpsURLConnection` using the URL specified.

Parameters

`url`	**`URL`**: the URL

Public methods

getCipherSuite

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public abstract String (/reference/java/lang/String) getCipherSuite ()
```

Returns the cipher suite in use on this connection.

Returns

`String` (/reference/java/lang/String)	the cipher suite

Throws

`IllegalStateException` (/reference/java/lang/IllegalStateException)	if this method is called before the connection has been established.

getDefaultHostnameVerifier

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier) getDefaultHostna
```

Gets the default `HostnameVerifier` that is inherited by new instances of this class.

Returns

HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier)	the default host name verifier

See also:

`setDefaultHostnameVerifier(HostnameVerifier)` (/reference/javax/net/ssl/HttpsURLConnection#setDefaultHostnameVerifier(javax.net.ssl.HostnameVerifier))

getDefaultSSLSocketFactory

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory) getDefaultSSLSo
```

Gets the default static `SSLSocketFactory` that is inherited by new instances of this class.

The socket factories are used when creating sockets for secure https URL connections.

Returns

SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory)	the default `SSLSocketFactory`

See also:

`setDefaultSSLSocketFactory(SSLSocketFactory)` (/reference/javax/net/ssl/HttpsURLConnection#setDefaultSSLSocketFactory(javax.net.ssl.SSLSocketFactory))

getHostnameVerifier Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier) getHostnameVerifier ()
```

Gets the `HostnameVerifier` in place on this instance.

Returns

HostnameVerifier (/reference/javax/net/ssl/HostnameVerifier)	the host name verifier

See also:

`setHostnameVerifier(HostnameVerifier)` (/reference/javax/net/ssl/HttpsURLConnection#setHostnameVerifier(javax.net.ssl.HostnameVerifier))

`setDefaultHostnameVerifier(HostnameVerifier)` (/reference/javax/net/ssl/HttpsURLConnection#setDefaultHostnameVerifier(javax.net.ssl.HostnameVerifier))

getLocalCertificates Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public abstract Certificate[] (/reference/java/security/cert/Certificate) getLocalCertificate
```

Returns the certificate(s) that were sent to the server during handshaking.

Note: This method is useful only when using certificate-based cipher suites.

When multiple certificates are available for use in a handshake, the implementation chooses what it considers the "best" certificate chain available, and transmits that to the other side. This method allows the caller to know which certificate chain was actually sent.

Returns

Certificate[] (/reference/java/security/cert/Certificate)	an ordered array of certificates, with the client's own certificate first followed by any certificate authorities. If no certificates were sent, then null is returned.

Throws

IllegalStateException (/reference/java/lang/IllegalStateException)	if this method is called before the connection has been established.

See also:

`getLocalPrincipal()` (/reference/javax/net/ssl/HttpsURLConnection#getLocalPrincipal())

getLocalPrincipal

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public Principal (/reference/java/security/Principal) getLocalPrincipal ()
```

Returns the principal that was sent to the server during handshaking.

Note: Subclasses should override this method. If not overridden, it will default to returning the X500Principal of the end-entity certificate that was sent to the server for certificate-based ciphersuites or, return null for non-certificate based ciphersuites, such as Kerberos.

Returns

Principal (/reference/java/security/Principal)	the principal sent to the server. Returns an X500Principal of the end-entity certificate for X509-based cipher suites, and KerberosPrincipal for Kerberos cipher suites. If no principal was sent, then null is returned.

Throws

IllegalStateException (/reference/java/lang/IllegalStateException)	if this method is called before the connection has been established.

See also:

`getLocalCertificates()` (/reference/javax/net/ssl/HttpsURLConnection#getLocalCertificates())

`getPeerPrincipal()` (/reference/javax/net/ssl/HttpsURLConnection#getPeerPrincipal())

getPeerPrincipal

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public Principal (/reference/java/security/Principal) getPeerPrincipal ()
```

Returns the server's principal which was established as part of defining the session.

Note: Subclasses should override this method. If not overridden, it will default to returning the X500Principal of the server's end-entity certificate for certificate-based ciphersuites, or throw an SSLPeerUnverifiedException for non-certificate based ciphersuites, such as Kerberos.

Returns

Principal (/reference/java/security/Principal)	the server's principal. Returns an X500Principal of the end-entity certitiicate for X509-based cipher suites, and KerberosPrincipal for Kerberos cipher suites.

Throws

SSLPeerUnverifiedException	if the peer was not verified

(/reference/javax/net/ssl/SSLPeerUnverifiedException)

IllegalStateException (/reference/java/lang/IllegalStateException)	if this method is called before the connection has been established.

See also:

`getServerCertificates()` (/reference/javax/net/ssl/HttpsURLConnection#getServerCertificates())

`getLocalPrincipal()` (/reference/javax/net/ssl/HttpsURLConnection#getLocalPrincipal())

getSSLSocketFactory Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory) getSSLSocketFactory ()
```

Gets the SSL socket factory to be used when creating sockets for secure https URL connections.

Returns

SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory)	the `SSLSocketFactory`

See also:

`setSSLSocketFactory(SSLSocketFactory)` (/reference/javax/net/ssl/HttpsURLConnection#setSSLSocketFactory(javax.net.ssl.SSLSocketFactory))

getServerCertificates Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public abstract Certificate[] (/reference/java/security/cert/Certificate) getServerCertificat
```

Returns the server's certificate chain which was established as part of defining the session.

Note: This method can be used only when using certificate-based cipher suites; using it with non-certificate-based cipher suites, such as Kerberos, will throw an SSLPeerUnverifiedException.

Returns

`Certificate[]` (/reference/java/security/cert/Certificate)	an ordered array of server certificates, with the peer's own certificate first followed by any certificate authorities.

Throws

`SSLPeerUnverifiedException` (/reference/javax/net/ssl/SSLPeerUnverifiedException)	if the peer is not verified.
`IllegalStateException` (/reference/java/lang/IllegalStateException)	if this method is called before the connection has been established.

See also:

`getPeerPrincipal()` (/reference/javax/net/ssl/HttpsURLConnection#getPeerPrincipal())

setDefaultHostnameVerifier

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static void setDefaultHostnameVerifier (HostnameVerifier (/reference/javax/net/s
```

Sets the default `HostnameVerifier` inherited by a new instance of this class.

Developers are *strongly* discouraged from changing the default `HostnameVerifier` as `getDefaultHostnameVerifier()`

(/reference/javax/net/ssl/HttpsURLConnection#getDefaultHostnameVerifier()) is used by several classes for hostname verification on Android.

User	Effect
Android's default TrustManager (/reference/javax/net/ssl/TrustManager), as used with Android's default SSLContext (/reference/javax/net/ssl/SSLContext), SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFactory) and SSLSocket (/reference/javax/net/ssl/SSLSocket) implementations.	The HostnameVerifier is used to verify the peer's SSLParameters.setEndpointIdentification. Instances use the *current* default HostnameVerifi
SSLCertificateSocketFactory (/reference/android/net/SSLCertificateSocketFactory)	The current default HostnameVerifier is used fro SSLCertificateSocketFactory (/reference/and details; for example SSLCertificateSocketFact (/reference/android/net/SSLCertificateSocketFactor
Android's default HttpsURLConnection (/reference/javax/net/ssl/HttpsURLConnection) implementation.	The HostnameVerifier is used after a successful TLS session server. Instances use the default Hostr unless overridden with setHostnameVerifier(ja (/reference/javax/net/ssl/HttpsURLConnection#setH . Android's HttpsURLConnection relies on the Hos verification step.

If this method is not called, the default HostnameVerifier will check the hostname according to RFC 2818.

Parameters

v	HostnameVerifier: the default host name verifier

Throws

IllegalArgumentException (/reference/java/lang/IllegalArgumentException)	if the `HostnameVerifier` parameter is null.
SecurityException (/reference/java/lang/SecurityException)	if a security manager exists and its `checkPermission` method does not allow `SSLPermission("setHostnameVerifier")`

See also:

`getDefaultHostnameVerifier()` (/reference/javax/net/ssl/HttpsURLConnection#getDefaultHostnameVerifier())

setDefaultSSLSocketFactory

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public static void setDefaultSSLSocketFactory (SSLSocketFactory (/reference/javax/net/s
```

Sets the default `SSLSocketFactory` inherited by new instances of this class.

The socket factories are used when creating sockets for secure https URL connections.

Parameters

`sf`	`SSLSocketFactory`: the default SSL socket factory

Throws

IllegalArgumentException (/reference/java/lang/IllegalArgumentException)	if the SSLSocketFactory parameter is null.

SecurityException (/reference/java/lang/SecurityException)	if a security manager exists and its `checkSetFactory` method does not allow a socket factory to be specified.

See also:

`getDefaultSSLSocketFactory()` (/reference/javax/net/ssl/HttpsURLConnection#getDefaultSSLSocketFactory())

setHostnameVerifier

Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void setHostnameVerifier (HostnameVerifier (/reference/javax/net/ssl/HostnameVerifi
```

Sets the `HostnameVerifier` for this instance.

New instances of this class inherit the default static hostname verifier set by `setDefaultHostnameVerifier` (/reference/javax/net/ssl/HttpsURLConnection#setDefaultHostnameVerifier(javax.net.ssl.HostnameVerifier)). Calls to this method replace this object's `HostnameVerifier`.

Android's `HttpsURLConnection` relies on the `HostnameVerifier` for the *entire* hostname verification step.

Parameters

v	`HostnameVerifier`: the host name verifier

Throws

IllegalArgumentException (/reference/java/lang/IllegalArgumentException)	if the `HostnameVerifier` parameter is null.

See also:

`getHostnameVerifier()` (/reference/javax/net/ssl/HttpsURLConnection#getHostnameVerifier())

`setDefaultHostnameVerifier(HostnameVerifier)` (/reference/javax/net/ssl/HttpsURLConnection#setDefaultHostnameVerifier(javax.net.ssl.HostnameVerifier))

setSSLSocketFactory Added in API level 1 (/guide/topics/manifest/uses-sdk-element#ApiLevels)

```
public void setSSLSocketFactory (SSLSocketFactory (/reference/javax/net/ssl/SSLSocketFact
```

Sets the `SSLSocketFactory` to be used when this instance creates sockets for secure https URL connections.

New instances of this class inherit the default static `SSLSocketFactory` set by `setDefaultSSLSocketFactory` (/reference/javax/net/ssl/HttpsURLConnection#setDefaultSSLSocketFactory(javax.net.ssl.SSLSocketFactory)). Calls to this method replace this object's `SSLSocketFactory`.

Parameters

`sf`	`SSLSocketFactory`: the SSL socket factory

Throws

`IllegalArgumentException` (/reference/java/lang/IllegalArgumentException)	if the `SSLSocketFactory` parameter is null.
`SecurityException` (/reference/java/lang/SecurityException)	if a security manager exists and its `checkSetFactory` method does not allow a socket factory to be specified.

See also:

`getSSLSocketFactory()` (/reference/javax/net/ssl/HttpsURLConnection#getSSLSocketFactory())

Last updated 2023-02-08 UTC.