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

```
public final class URL
```
extends Object (https://developer.android.com/reference/java/lang/Object.html) implements Serializable (https://developer.android.com/reference/java/io/Serializable.html)

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

Class **URL** represents a Uniform Resource Locator, a pointer to a "resource" on the World Wide Web. A resource can be something as simple as a file or a directory, or it can be a reference to a more complicated object, such as a query to a database or to a search engine. More information on the types of URLs and their formats can be found at: *Types of URL*
(http://web.archive.org/web/20051219043731/http://archive.ncsa.uiuc.edu/SDG/Software/Mosaic/Demo/url-primer.html)

In general, a URL can be broken into several parts. Consider the following example:

```
http://www.example.com/docs/resource1.html
```

The URL above indicates that the protocol to use is **`http`** (HyperText Transfer Protocol) and that the information resides on a host machine named **`www.example.com`**. The information on that host machine is named **`/docs/resource1.html`**. The exact meaning of this name on the host machine is both protocol dependent and host dependent. The information normally resides in a file, but it could be generated on the fly. This component of the URL is called the *path* component.

A URL can optionally specify a "port", which is the port number to which the TCP connection is made on the remote host machine. If the port is not specified, the default port for the protocol is used instead. For example, the default port for **`http`** is **`80`**. An alternative port could be specified as:

```
http://www.example.com:1080/docs/resource1.html
```

The syntax of **URL** is defined by *RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax* (http://www.ietf.org/rfc/rfc2396.txt), amended by *RFC 2732: Format for Literal IPv6 Addresses in URLs* (http://www.ietf.org/rfc/rfc2732.txt). The Literal IPv6 address format also supports scope_ids. The syntax and usage of scope_ids is described here (https://developer.android.com/reference/java/net/Inet6Address.html#scoped).

A URL may have appended to it a "fragment", also known as a "ref" or a "reference". The fragment is indicated by the sharp sign character "#" followed by more characters. For example,

```
http://java.sun.com/index.html#chapter1
```

This fragment is not technically part of the URL. Rather, it indicates that after the specified resource is retrieved, the application is specifically interested in that part of the document that has the tag **`chapter1`** attached to it. The meaning of a tag is resource specific.

An application can also specify a "relative URL", which contains only enough information to reach the resource relative to another URL. Relative URLs are frequently used within HTML pages. For example, if the contents of the URL:

```
http://java.sun.com/index.html
```

contained within it the relative URL:

```
FAQ.html
```

it would be a shorthand for:

```
http://java.sun.com/FAQ.html
```

The relative URL need not specify all the components of a URL. If the protocol, host name, or port number is missing, the value is inherited from the fully specified URL. The file component must be specified. The optional fragment is not inherited.

The URL class does not itself encode or decode any URL components according to the escaping mechanism defined in RFC2396. It is the responsibility of the caller to encode any

fields, which need to be escaped prior to calling URL, and also to decode any escaped fields, that are returned from URL. Furthermore, because URL has no knowledge of URL escaping, it does not recognise equivalence between the encoded or decoded form of the same URL. For example, the two URLs:

```
http://foo.com/hello world/ and http://foo.com/hello%20world
```

would be considered not equal to each other.

Note, the **URI** (https://developer.android.com/reference/java/net/URI.html) class does perform escaping of its component fields in certain circumstances. The recommended way to manage the encoding and decoding of URLs is to use **URI** (https://developer.android.com/reference/java/net/URI.html), and to convert between these two classes using **toURI()** (https://developer.android.com/reference/java/net/URL.html#toURI()) and **URI.toURL()** (https://developer.android.com/reference/java/net/URI.html#toURL()).

The **URLEncoder** (https://developer.android.com/reference/java/net/URLEncoder.html) and **URLDecoder** (https://developer.android.com/reference/java/net/URLDecoder.html) classes can also be used, but only for HTML form encoding, which is not the same as the encoding scheme defined in RFC2396.

Summary

Public constructors

URL (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))
(String (https://developer.android.com/reference/java/lang/String.html) **protocol, String** (https://developer.android.com/reference/java/lang/String.html) **host, int port, String** (https://developer.android.com/reference/java/lang/String.html) **file)**

Creates a **URL** object from the specified **protocol**, **host**, **port** number, and **file**.

URL (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20java.lang.String))
(String (https://developer.android.com/reference/java/lang/String.html) **protocol, String** (https://developer.android.com/reference/java/lang/String.html) **host, String** (https://developer.android.com/reference/java/lang/String.html) **file)**

Creates a URL from the specified **protocol** name, **host** name, and **file** name.

URL

(https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String,%20java.net.URLStreamHandler))
(String (https://developer.android.com/reference/java/lang/String.html) **protocol, String** (https://developer.android.com/reference/java/lang/String.html) **host, int port, String** (https://developer.android.com/reference/java/lang/String.html) **file, URLStreamHandler** (https://developer.android.com/reference/java/net/URLStreamHandler.html) **handler)**

Creates a **URL** object from the specified **protocol**, **host**, **port** number, **file**, and **handler**.

URL (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String))**(String** (https://developer.android.com/reference/java/lang/String.html) **spec)**

Creates a **URL** object from the **String** representation.

URL
(https://developer.android.com/reference/java/net/URL.html#URL(java.net.URL,%20java.lang.String))
(URL (https://developer.android.com/reference/java/net/URL.html) **context, String** (https://developer.android.com/reference/java/lang/String.html) **spec)**

Creates a URL by parsing the given spec within a specified context.

URL
(https://developer.android.com/reference/java/net/URL.html#URL(java.net.URL,%20java.lang.String,%20java.net.URLStreamHandler))
(URL (https://developer.android.com/reference/java/net/URL.html) **context, String** (https://developer.android.com/reference/java/lang/String.html) **spec, URLStreamHandler** (https://developer.android.com/reference/java/net/URLStreamHandler.html) **handler)**

Creates a URL by parsing the given spec with the specified handler within a specified context.

Public methods

---

boolean

equals
(https://developer.android.com/reference/java/net/URL.html#equals(java.lang.Object))
(Object
(https://developer.android.com/reference/java/lang/Object.html)
obj)

Compares this URL for equality with another object.

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

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

Gets the authority part of this **URL**.

Object

getContent

(https://developer.android.com/reference/java/lang/Object.html)	(https://developer.android.com/reference/java/net/URL.html#getContent()) () Gets the contents of this URL.
Object (https://developer.android.com/reference/java/lang/Object.html)	getContent (https://developer.android.com/reference/java/net/URL.html#getContent(java.lang.Class[])) (Class[] (https://developer.android.com/reference/java/lang/Class.html) classes) Gets the contents of this URL.
int	getDefaultPort (https://developer.android.com/reference/java/net/URL.html#getDefaultPort()) () Gets the default port number of the protocol associated with this URL.
String (https://developer.android.com/reference/java/lang/String.html)	getFile (https://developer.android.com/reference/java/net/URL.html#getFile()) () Gets the file name of this URL.
String (https://developer.android.com/reference/java/lang/String.html)	getHost (https://developer.android.com/reference/java/net/URL.html#getHost()) () Gets the host name of this URL, if applicable.
String (https://developer.android.com/reference/java/lang/String.html)	getPath (https://developer.android.com/reference/java/net/URL.html#getPath()) () Gets the path part of this URL.
int	getPort (https://developer.android.com/reference/java/net/URL.html#getPort()) () Gets the port number of this URL.
String	getProtocol

(https://developer.android.com/reference/java/lang/String.html)	(https://developer.android.com/reference/java/net/URL.html#getProtocol()) () Gets the protocol name of this **URL**.
String (https://developer.android.com/reference/java/lang/String.html)	getQuery (https://developer.android.com/reference/java/net/URL.html#getQuery()) () Gets the query part of this **URL**.
String (https://developer.android.com/reference/java/lang/String.html)	getRef (https://developer.android.com/reference/java/net/URL.html#getRef()) () Gets the anchor (also known as the "reference") of this **URL**.
String (https://developer.android.com/reference/java/lang/String.html)	getUserInfo (https://developer.android.com/reference/java/net/URL.html#getUserInfo()) () Gets the userInfo part of this **URL**.
int	hashCode (https://developer.android.com/reference/java/net/URL.html#hashCode()) () Creates an integer suitable for hash table indexing.
URLConnection (https://developer.android.com/reference/java/net/URLConnection.html)	openConnection (https://developer.android.com/reference/java/net/URL.html#openConnection(java.net.Proxy)) (Proxy (https://developer.android.com/reference/java/net/Proxy.html) proxy) Same as openConnection() (https://developer.android.com/reference/java/net/URL.html#openConnection()) , except that the connection will be made through the specified proxy; Protocol handlers that do not support proxing will ignore the proxy parameter and make a normal connection.
URLConnection	openConnection (https://developer.android.com/reference/java/net/URL.html#openConnection())

(https://developer.android.com/reference/java/net/URLConnection.html)	**()** Returns a **URLConnection** (https://developer.android.com/reference/java/net/URLConnection.html) instance that represents a connection to the remote object referred to by the **URL**.
InputStream (https://developer.android.com/reference/java/io/InputStream.html)	**openStream** (https://developer.android.com/reference/java/net/URL.html#openStream()) **()** Opens a connection to this **URL** and returns an **InputStream** for reading from that connection.
boolean	**sameFile** (https://developer.android.com/reference/java/net/URL.html#sameFile(java.net.URL)) **(URL** (https://developer.android.com/reference/java/net/URL.html) **other)** Compares two URLs, excluding the fragment component.
static void	**setURLStreamHandlerFactory** (https://developer.android.com/reference/java/net/URL.html#setURLStreamHandlerFactory(java.net.URLStreamHandlerFactory)) **(URLStreamHandlerFactory** (https://developer.android.com/reference/java/net/URLStreamHandlerFactory.html) **fac)** Sets an application's **URLStreamHandlerFactory**.
String (https://developer.android.com/reference/java/lang/String.html)	**toExternalForm** (https://developer.android.com/reference/java/net/URL.html#toExternalForm()) **()** Constructs a string representation of this **URL**.
String (https://developer.android.com/reference/java/lang/String.html)	**toString** (https://developer.android.com/reference/java/net/URL.html#toString()) **()** Constructs a string representation of this **URL**.
URI (https://developer.android.com/reference/java/net/URI.html)	**toURI** (https://developer.android.com/reference/java/net/URL.html#toURI()))

()

Returns a **URI** (https://developer.android.com/reference/java/net/URI.html) equivalent to this URL.

Inherited methods

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://deve /reference/ja ml#clone()) ()

Creates and r this object.

boolean

equals (https://develc ference/java/la uals(java.lang. (**Object** (https://develc ference/java/la **obj**)

Indicates whet object is "equa

void

finalize (https://develc ference/java/la alize()) ()

Called by the g an object wher determines tha references to t

final Class (https://developer.android.com/reference/java/lang/Class.html)**<?>**

getClass (https://develc ference/java/la tClass()) ()

Returns the rur **Object**.

`int`	**hashCode** (https://develc ference/java/la shCode()) ( ) Returns a hash object.
`final void`	**notify** (https://develc ference/java/la tify()) ( ) Wakes up a sin waiting on this
`final void`	**notifyAll** (https://develc ference/java/la tifyAll()) ( ) Wakes up all th waiting on this
String (https://developer.android.com/reference/java/lang/String.html)	**toString** (https://develc ference/java/la String()) ( ) Returns a strin the object.
`final void`	**wait** (https://develc ference/java/la ait(long,%20int `(long milli` Causes the cur until another th **notify( )** (https://develc ference/java/la tify()) method or the

	(https://develc ference/java/la tifyAll()) method for this other thread in thread, or a cer time has elaps
`final void`	**`wait`** (https://develc ference/java/la ait(long)) **`(long milli`** Causes the cur until either ano the **`notify()`** (https://develc ference/java/la tify()) method or the (https://develc ference/java/la tifyAll()) method for this specified amou elapsed.
`final void`	**`wait`** (https://deve /reference/jav ml#wait()) **`()`** Causes the cu wait until ano invokes the **n** (https://deve /reference/jav ml#notify()) method or the (https://deve /reference/jav ml#notifyAll() method for th

Public constructors

URL

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

Creates a **URL** object from the specified **protocol**, **host**, **port** number, and **file**.

host can be expressed as a host name or a literal IP address. If IPv6 literal address is used, it should be enclosed in square brackets (' **[** ' and ' **]** '), as specified by RFC 2732 (http://www.ietf.org/rfc/rfc2732.txt); However, the literal IPv6 address format defined in *RFC 2373: IP Version 6 Addressing Architecture* (http://www.ietf.org/rfc/rfc2373.txt) is also accepted.

Specifying a **port** number of **-1** indicates that the URL should use the default port for the protocol.

If this is the first URL object being created with the specified protocol, a *stream protocol handler* object, an instance of class **URLStreamHandler**, is created for that protocol:

1. If the application has previously set up an instance of **URLStreamHandlerFactory** as the stream handler factory, then the **createURLStreamHandler** method of that instance is called with the protocol string as an argument to create the stream protocol handler.
2. If no **URLStreamHandlerFactory** has yet been set up, or if the factory's **createURLStreamHandler** method returns **null**, then the constructor finds the value of the system property:

   ```
   java.protocol.handler.pkgs
   ```

   If the value of that system property is not **null**, it is interpreted as a list of packages separated by a vertical slash character '|'. The constructor tries to load the class named:

   ```
   <package>.<protocol>.Handler
   ```

where <*package*> is replaced by the name of the package and <*protocol*> is replaced by the name of the protocol. If this class does not exist, or if the class exists but it is not a subclass of **`URLStreamHandler`**, then the next package in the list is tried.

3. If the previous step fails to find a protocol handler, then the constructor tries to load from a system default package.

```
<system default package>.<protocol>.Handler
```

If this class does not exist, or if the class exists but it is not a subclass of **`URLStreamHandler`**, then a **`MalformedURLException`** is thrown.

Protocol handlers for the following protocols are guaranteed to exist on the search path :-

```
http, https, file, and jar
```

Protocol handlers for additional protocols may also be available.

No validation of the inputs is performed by this constructor.

Parameters

`protocol`	**`String`**: the name of the protocol to use.
`host`	**`String`**: the name of the host.
`port`	**`int`**: the port number on the host.
`file`	**`String`**: the file on the host

Throws

`MalformedURLException` (https://developer.android.com/reference/java/net/MalformedURLException.html)	if an unknown protocol is specified.

See also:

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

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

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

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

URL

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

Creates a URL from the specified **`protocol`** name, **`host`** name, and **`file`** name. The default port for the specified protocol is used.

This method is equivalent to calling the four-argument constructor with the arguments being **`protocol`**, **`host`**, **`-1`**, and **`file`**. No validation of the inputs is performed by this constructor.

Parameters

`protocol`	**`String`**: the name of the protocol to use.
`host`	**`String`**: the name of the host.
`file`	**`String`**: the file on the host.

Throws

`MalformedURLException` (https://developer.android.com/reference/java/net/MalformedURLException.html)	if an unknown protocol is specified.

See also:

URL(java.lang.String, java.lang.String, int, java.lang.String)
(https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

URL

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

Creates a `URL` object from the specified `protocol`, `host`, `port` number, `file`, and `handler`. Specifying a `port` number of `-1` indicates that the URL should use the default port for the protocol. Specifying a `handler` of `null` indicates that the URL should use a default stream handler for the protocol, as outlined for: java.net.URL#URL(java.lang.String, java.lang.String, int, java.lang.String)

If the handler is not null and there is a security manager, the security manager's `checkPermission` method is called with a `NetPermission("specifyStreamHandler")` permission. This may result in a SecurityException. No validation of the inputs is performed by this constructor.

Parameters

`protocol`	`String`: the name of the protocol to use.
`host`	`String`: the name of the host.
`port`	`int`: the port number on the host.
`file`	`String`: the file on the host
`handler`	`URLStreamHandler`: the stream handler for the URL.

Throws

`MalformedURLException`	if an unknown protocol is specified.

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

SecurityException (https://developer.android.com/reference/java/lang/SecurityException.html)	if a security manager exists and its `checkPermission` method doesn't allow specifying a stream handler explicitly.

See also:

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

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

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

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

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

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

URL

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

Creates a `URL` object from the `String` representation.

This constructor is equivalent to a call to the two-argument constructor with a `null` first argument.

Parameters

`spec`	`String`: the `String` to parse as a URL.

Throws

`MalformedURLException` if no protocol is specified, or an unknown protocol is found, or `spec` is `null`.

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

See also:

`URL(java.net.URL, java.lang.String)`

(https://developer.android.com/reference/java/net/URL.html#URL(java.net.URL,%20java.lang.String))

URL

added in API level 1

(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Creates a URL by parsing the given spec within a specified context. The new URL is created from the given context URL and the spec argument as described in RFC2396 "Uniform Resource Identifiers : Generic * Syntax" :

```
<scheme>://<authority><path>?<query>#<fragment>
```

The reference is parsed into the scheme, authority, path, query and fragment parts. If the path component is empty and the scheme, authority, and query components are undefined, then the new URL is a reference to the current document. Otherwise, the fragment and query parts present in the spec are used in the new URL.

If the scheme component is defined in the given spec and does not match the scheme of the context, then the new URL is created as an absolute URL based on the spec alone. Otherwise the scheme component is inherited from the context URL.

If the authority component is present in the spec then the spec is treated as absolute and the spec authority and path will replace the context authority and path. If the authority component is absent in the spec then the authority of the new URL will be inherited from the context.

If the spec's path component begins with a slash character "/" then the path is treated as absolute and the spec path replaces the context path.

Otherwise, the path is treated as a relative path and is appended to the context path, as described in RFC2396. Also, in this case, the path is canonicalized through the removal of

directory changes made by occurrences of ".." and ".".

For a more detailed description of URL parsing, refer to RFC2396.

Parameters

`context`	`URL`: the context in which to parse the specification.
`spec`	`String`: the `String` to parse as a URL.

Throws

`MalformedURLException` (https://developer.android.com/reference/java/net/MalformedURLException.html)	if no protocol is specified, or an unknown protocol is found, or `spec` is `null`.

See also:

`URL(java.lang.String, java.lang.String, int, java.lang.String)` (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

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

`URLStreamHandler.parseURL(java.net.URL, java.lang.String, int, int)` (https://developer.android.com/reference/java/net/URLStreamHandler.html#parseURL(java.net.URL,%20java.lang.String,%20int,%20int))

URL

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

Creates a URL by parsing the given spec with the specified handler within a specified context. If the handler is null, the parsing occurs as with the two argument constructor.

Parameters

`context`	`URL`: the context in which to parse the specification.

`spec`	`String`: the `String` to parse as a URL.
`handler`	`URLStreamHandler`: the stream handler for the URL.

Throws

`MalformedURLException` (https://developer.android.com/reference/java/net/MalformedURLException.html)	if no protocol is specified, or an unknown protocol is found, or `spec` is `null`.
`SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html)	if a security manager exists and its `checkPermission` method doesn't allow specifying a stream handler.

See also:

`URL(java.lang.String, java.lang.String, int, java.lang.String)` (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

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

`URLStreamHandler.parseURL(java.net.URL, java.lang.String, int, int)` (https://developer.android.com/reference/java/net/URLStreamHandler.html#parseURL(java.net.URL,%20java.lang.String,%20int,%20int))

Public methods

equals

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

Compares this URL for equality with another object.

If the given object is not a URL then this method immediately returns `false`.

Two URL objects are equal if they have the same protocol, reference equivalent hosts, have the same port number on the host, and the same file and fragment of the file.

Returns true if this URL equals `o`. URLs are equal if they have the same protocol, host, port, file, and reference.

Network I/O Warning

Some implementations of URL.equals() resolve host names over the network. This is problematic:

- **The network may be slow.** Many classes, including core collections like `Map` (https://developer.android.com/reference/java/util/Map.html) and `Set` (https://developer.android.com/reference/java/util/Set.html) expect that `equals` and `hashCode` will return quickly. By violating this assumption, this method posed potential performance problems.
- **Equal IP addresses do not imply equal content.** Virtual hosting permits unrelated sites to share an IP address. This method could report two otherwise unrelated URLs to be equal because they're hosted on the same server.
- **The network may not be available.** Two URLs could be equal when a network is available and unequal otherwise.
- **The network may change.** The IP address for a given host name varies by network and over time. This is problematic for mobile devices. Two URLs could be equal on some networks and unequal on others.

This problem is fixed in Android 4.0 (Ice Cream Sandwich). In that release, URLs are only equal if their host names are equal (ignoring case).

Parameters

`obj`	`Object`: the URL to compare against.

Returns

`boolean`	`true` if the objects are the same; `false` otherwise.

getAuthority

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

Gets the authority part of this `URL`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the authority part of this `URL`

getContent

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

Gets the contents of this URL. This method is a shorthand for:

```
openConnection().getContent()
```

Returns

`Object` (https://developer.android.com/reference/java/lang/Object.html)	the contents of this URL.

Throws

`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an I/O exception occurs.

See also:

`URLConnection.getContent()` (https://developer.android.com/reference/java/net/URLConnection.html#getContent())

getContent

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

Gets the contents of this URL. This method is a shorthand for:

```
openConnection().getContent(Class[])
```

Parameters

`classes`	`Class`: an array of Java types

Returns

`Object` (https://developer.android.com/reference/java/lang/Object.html)	the content object of this URL that is the first match of the types specified in the classes array. null if none of the requested types are supported.

Throws

`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an I/O exception occurs.

See also:

`URLConnection.getContent(Class[])`
(https://developer.android.com/reference/java/net/URLConnection.html#getContent(java.lang.Class[]))

getDefaultPort

added in API level 9
(https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Gets the default port number of the protocol associated with this `URL`. If the URL scheme or the URLStreamHandler for the URL do not define a default port number, then -1 is returned.

Returns

`int`	the port number

getFile

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

Gets the file name of this `URL`. The returned file portion will be the same as `getPath()`, plus the concatenation of the value of `getQuery()`, if any. If there is no query portion, this method and `getPath()` will return identical results.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the file name of this `URL`, or an empty string if one does not exist

getHost

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

Gets the host name of this `URL`, if applicable. The format of the host conforms to RFC 2732, i.e. for a literal IPv6 address, this method will return the IPv6 address enclosed in square brackets (`'['` and `']'`).

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the host name of this `URL`.

getPath

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

Gets the path part of this `URL`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the path part of this `URL`, or an empty string if one does not exist

getPort

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

Gets the port number of this `URL`.

Returns

`int`	the port number, or -1 if the port is not set

getProtocol

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

Gets the protocol name of this `URL`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the protocol of this `URL`.

getQuery

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

Gets the query part of this `URL`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the query part of this `URL`, or `null` if one does not exist

getRef

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

Gets the anchor (also known as the "reference") of this `URL`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the anchor (also known as the "reference") of this `URL`, or `null` if one does not exist

getUserInfo

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

Gets the userInfo part of this `URL`.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	the userInfo part of this `URL`, or `null` if one does not exist

hashCode

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

Creates an integer suitable for hash table indexing.

The hash code is based upon all the URL components relevant for URL comparison. As such, this operation is a blocking operation.

Returns

`int`	a hash code for this `URL`.

openConnection

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

Same as `openConnection()`
(https://developer.android.com/reference/java/net/URL.html#openConnection()), except that the connection will be made through the specified proxy; Protocol handlers that do not support proxing will ignore the proxy parameter and make a normal connection. Invoking this method preempts the system's default ProxySelector settings.

Parameters

`proxy`	`Proxy`: the Proxy through which this connection will be made. If direct connection is desired, Proxy.NO_PROXY should be specified.

Returns

`URLConnection` (https://developer.android.com/reference/java/net/URLConnection.html)	a `URLConnection` to the URL.

Throws

`IOException` (https://developer.android.com/reference/java/io/IOException.html)	if an I/O exception occurs.
`SecurityException` (https://developer.android.com/reference/java/lang/SecurityException.html)	if a security manager is present and the caller doesn't have permission to connect to the proxy.
`IllegalArgumentException` (https://developer.android.com/reference/java/lang/IllegalArgumentException.html)	will be thrown if proxy is null, or proxy has the wrong type
`UnsupportedOperationException` (https://developer.android.com/reference/java/lang/UnsupportedOperationException.html)	if the subclass that implements the protocol handler doesn't support this method.

See also:

`URL(java.lang.String, java.lang.String, int, java.lang.String)` (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

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

`URLStreamHandler.openConnection(java.net.URL, java.net.Proxy)` (https://developer.android.com/reference/java/net/URLStreamHandler.html#openConnection(java.net.URL,%20java.net.Proxy))

openConnection

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

Returns a `URLConnection` (https://developer.android.com/reference/java/net/URLConnection.html) instance that represents a connection to the remote object referred to by the `URL`.

A new instance of URLConnection (https://developer.android.com/reference/java/net/URLConnection.html) is created every time when invoking the URLStreamHandler.openConnection(URL) (https://developer.android.com/reference/java/net/URLStreamHandler.html#openConnection(java.net.URL)) method of the protocol handler for this URL.

It should be noted that a URLConnection instance does not establish the actual network connection on creation. This will happen only when calling URLConnection.connect() (https://developer.android.com/reference/java/net/URLConnection.html#connect()).

If for the URL's protocol (such as HTTP or JAR), there exists a public, specialized URLConnection subclass belonging to one of the following packages or one of their subpackages: java.lang, java.io, java.util, java.net, the connection returned will be of that subclass. For example, for HTTP an HttpURLConnection will be returned, and for JAR a JarURLConnection will be returned.

Returns

`URLConnection` (https://developer.android.com/reference/java/net/URLConnection.html)	a `URLConnection` (https://developer.android.com/reference/java/net/URLConnection.html) linking to the URL.

Throws

IOException (https://developer.android.com/reference/java/io/IOException.html)	if an I/O exception occurs.

See also:

URL(java.lang.String, java.lang.String, int, java.lang.String) (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

openStream

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

Opens a connection to this `URL` and returns an `InputStream` for reading from that connection. This method is a shorthand for:

```
openConnection().getInputStream()
```

Returns

InputStream (https://developer.android.com/reference/java/io/InputStream.html)	an input stream for reading from the URL connection.

Throws

IOException (https://developer.android.com/reference/java/io/IOException.html)	if an I/O exception occurs.

See also:

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

URLConnection.getInputStream() (https://developer.android.com/reference/java/net/URLConnection.html#getInputStream())

sameFile

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

Compares two URLs, excluding the fragment component.

Returns `true` if this `URL` and the `other` argument are equal without taking the fragment component into consideration.

Parameters

`other`	`URL`: the `URL` to compare against.

Returns

`boolean`	`true` if they reference the same remote object; `false` otherwise.

setURLStreamHandlerFactory

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

Sets an application's `URLStreamHandlerFactory`. This method can be called at most once in a given Java Virtual Machine.

The `URLStreamHandlerFactory` instance is used to construct a stream protocol handler from a protocol name.

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

`fac`	`URLStreamHandlerFactory`: the desired factory.

Throws

`Error`	if the application has already set a factory.

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

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:

URL(java.lang.String, java.lang.String, int, java.lang.String) (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

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

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

toExternalForm

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

Constructs a string representation of this `URL`. The string is created by calling the `toExternalForm` method of the stream protocol handler for this object.

Returns

String (https://developer.android.com/reference/java/lang/String.html)	a string representation of this object.

See also:

URL(java.lang.String, java.lang.String, int, java.lang.String) (https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

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

toString

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

Constructs a string representation of this `URL`. The string is created by calling the `toExternalForm` method of the stream protocol handler for this object.

Returns

`String` (https://developer.android.com/reference/java/lang/String.html)	a string representation of this object.

See also:

`URL(java.lang.String, java.lang.String, int, java.lang.String)`
(https://developer.android.com/reference/java/net/URL.html#URL(java.lang.String,%20java.lang.String,%20int,%20java.lang.String))

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

toURI

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

Returns a `URI` (https://developer.android.com/reference/java/net/URI.html) equivalent to this URL. This method functions in the same way as `new URI (this.toString())`.

Note, any URL instance that complies with RFC 2396 can be converted to a URI. However, some URLs that are not strictly in compliance can not be converted to a URI.

Returns

`URI` (https://developer.android.com/reference/java/net/URI.html)	a URI instance equivalent to this URL.

Throws

`URISyntaxException`	if this URL is not formatted strictly according to to RFC2396 and

(https://developer.android.com/recannot be converted to a URI.
ference/java/net/URISyntaxExcept
ion.html)

Last updated June 6, 2018.

Twitter
Follow @AndroidDev on Twitter

Google+
Follow Android Developers on Google+

YouTube
Check out Android Developers on YouTube