

NetworkInterface | API reference | Android Developers

Archived: 2026-04-06 00:37:27 UTC

NetworkInterface Stay organized with collections Save and categorize content based on your preferences.

```
public final class NetworkInterface
extends Object
```

This class represents a Network Interface made up of a name, and a list of IP addresses assigned to this interface. It is used to identify the local interface on which a multicast group is joined. Interfaces are normally known by names such as "le0".

Note that information about [NetworkInterface](#) s may be restricted. For example, non-system apps will only have access to information about [NetworkInterface](#) s that are associated with an [InetAddress](#) .

Summary

Public methods	
<code>boolean</code>	<pre><code>equals(Object obj)</code></pre> <p>Compares this object against the specified object.</p>
<code>static NetworkInterface</code>	<pre><code>getByIndex(int index)</code></pre> <p>Get a network interface given its index.</p>
<code>static NetworkInterface</code>	<pre><code>getByInetAddress(InetAddress addr)</code></pre> <p>Convenience method to search for a network interface that has the specified Internet Protocol (IP) address bound to it.</p>
<code>static NetworkInterface</code>	<pre><code>getByName(String name)</code></pre> <p>Searches for the network interface with the specified name.</p>
<code>String</code>	<pre><code>getDisplayname()</code></pre>

	Get the display name of this network interface.
byte[]	<p>getHardwareAddress()</p> <p>Returns the hardware address (usually MAC) of the interface if it has one and if it can be accessed given the current privileges.</p>
int	<p>getIndex()</p> <p>Returns the index of this network interface.</p>
Enumeration<InetAddress>	<p>getInetAddresses()</p> <p>Convenience method to return an Enumeration with all or a subset of the InetAddresses bound to this network interface.</p>
List<InterfaceAddress>	<p>getInterfaceAddresses()</p> <p>Get a List of all or a subset of the InterfaceAddresses of this network interface.</p>
int	<p>getMTU()</p> <p>Returns the Maximum Transmission Unit (MTU) of this interface.</p>
String	<p>getName()</p> <p>Get the name of this network interface.</p>
static Enumeration<NetworkInterface>	<p>getNetworkInterfaces()</p> <p>Returns all the interfaces on this machine.</p>
NetworkInterface	<p>getParent()</p> <p>Returns the parent NetworkInterface of this interface if this is a subinterface, or null if it is a physical (non virtual) interface or has no parent.</p>
Enumeration<NetworkInterface>	<p>getSubInterfaces()</p> <p>Get an Enumeration with all the subinterfaces (also known as virtual interfaces) attached to this network interface.</p>

<code>int</code>	<p>hashCode()</p> <p>Returns a hash code value for the object.</p>
<code>boolean</code>	<p>isLoopback()</p> <p>Returns whether a network interface is a loopback interface.</p>
<code>boolean</code>	<p>isPointToPoint()</p> <p>Returns whether a network interface is a point to point interface.</p>
<code>boolean</code>	<p>isUp()</p> <p>Returns whether a network interface is up and running.</p>
<code>boolean</code>	<p>isVirtual()</p> <p>Returns whether this interface is a virtual interface (also called subinterface).</p>
<code>boolean</code>	<p>supportsMulticast()</p> <p>Returns whether a network interface supports multicasting or not.</p>
String	<p>toString()</p> <p>Returns a string representation of the object.</p>

Inherited methods

From class [java.lang.Object](#)

Object	<p>clone()</p> <p>Creates and returns a copy of this object.</p>
<code>boolean</code>	<p>equals(Object obj)</p> <p>Indicates whether some other object is "equal to" this one.</p>
<code>void</code>	<p>finalize()</p>

	Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
<code>final</code> <code>Class<?></code>	<code>getClass()</code> Returns the runtime class of this <code>Object</code> .
<code>int</code>	<code>hashCode()</code> Returns a hash code value for the object.
<code>final void</code>	<code>notify()</code> Wakes up a single thread that is waiting on this object's monitor.
<code>final void</code>	<code>notifyAll()</code> Wakes up all threads that are waiting on this object's monitor.
<code>String</code>	<code>toString()</code> Returns a string representation of the object.
<code>final void</code>	<code>wait(long timeoutMillis, int nanos)</code> Causes the current thread to wait until it is awakened, typically by being <i>notified</i> or <i>interrupted</i> , or until a certain amount of real time has elapsed.
<code>final void</code>	<code>wait(long timeoutMillis)</code> Causes the current thread to wait until it is awakened, typically by being <i>notified</i> or <i>interrupted</i> , or until a certain amount of real time has elapsed.
<code>final void</code>	<code>wait()</code> Causes the current thread to wait until it is awakened, typically by being <i>notified</i> or <i>interrupted</i> .

Public methods

equals

```
public boolean equals (Object obj)
```

Compares this object against the specified object. The result is `true` if and only if the argument is not `null` and it represents the same `NetworkInterface` as this object.

Two instances of `NetworkInterface` represent the same `NetworkInterface` if both name and address are the same for both.

Parameters	
<code>obj</code>	<code>Object</code> : the object to compare against.
Returns	
<code>boolean</code>	<code>true</code> if the objects are the same; <code>false</code> otherwise.

getByIndex

```
public static NetworkInterface getByIndex (int index)
```

Get a network interface given its index.

Parameters	
<code>index</code>	<code>int</code> : an integer, the index of the interface
Returns	
NetworkInterface	the <code>NetworkInterface</code> obtained from its index, or <code>null</code> if an interface with the specified index does not exist or can't be accessed .
Throws	
IllegalArgumentException	if index has a negative value
SocketException	if an I/O error occurs.

getByInetAddress

```
public static NetworkInterface getByInetAddress (InetAddress addr)
```

Convenience method to search for a network interface that has the specified Internet Protocol (IP) address bound to it.

If the specified IP address is bound to multiple network interfaces it is not defined which network interface is returned.

Parameters	
<code>addr</code>	<code>InetAddress</code> : The <code>InetAddress</code> to search with.
Returns	
NetworkInterface	A <code>NetworkInterface</code> or <code>null</code> if there is no network interface with the specified IP address.
Throws	
NullPointerException	If the specified address is <code>null</code> .
SocketException	If an I/O error occurs.

getByName

```
public static NetworkInterface getByName (String name)
```

Searches for the network interface with the specified name.

Parameters	
<code>name</code>	<code>String</code> : The name of the network interface.
Returns	
NetworkInterface	A <code>NetworkInterface</code> with the specified name, or <code>null</code> if the network interface with the specified name does not exist or can't be accessed .
Throws	
NullPointerException	If the specified name is <code>null</code> .
SocketException	If an I/O error occurs.

getDisplayName

```
public String getDisplayName ()
```

Get the display name of this network interface. A display name is a human readable String describing the network device.

Returns[String](#)

a non-empty string representing the display name of this network interface, or null if no display name is available.

getHardwareAddress

```
public byte[] getHardwareAddress ()
```

Returns the hardware address (usually MAC) of the interface if it has one and if it can be accessed given the current privileges. If a security manager is set, then the caller must have the permission

[NetPermission](#) ("getNetworkInformation").

Returns[byte\[\]](#)

a byte array containing the address, or `null` if the address doesn't exist, is not accessible or a security manager is set and the caller does not have the permission `NetPermission("getNetworkInformation")`. For example, this method will generally return `null` when called by non-system apps (or `02:00:00:00:00:00` for apps having `targetSdkVersion < android.os.Build.VERSION_CODES.R`).

Throws[SocketException](#)

if an I/O error occurs.

getIndex

```
public int getIndex ()
```

Returns the index of this network interface. The index is an integer greater or equal to zero, or `-1` for unknown. This is a system specific value and interfaces with the same name can have different indexes on different machines.

Returns[int](#)

the index of this network interface or `-1` if the index is unknown

getInetAddresses

```
public Enumeration<InetAddress> getInetAddresses ()
```

Convenience method to return an Enumeration with all or a subset of the InetAddresses bound to this network interface.

If there is a security manager, its `checkConnect` method is called for each `InetAddress`. Only `InetAddresses` where the `checkConnect` doesn't throw a `SecurityException` will be returned in the Enumeration. However, if the caller has the [NetPermission](#) ("getNetworkInformation") permission, then all `InetAddresses` are returned.

Returns	
Enumeration<InetAddress>	an Enumeration object with all or a subset of the <code>InetAddresses</code> bound to this network interface

getInterfaceAddresses

```
public List<InterfaceAddress> getInterfaceAddresses ()
```

Get a List of all or a subset of the `InterfaceAddresses` of this network interface.

If there is a security manager, its `checkConnect` method is called with the `InetAddress` for each `InterfaceAddress`. Only `InterfaceAddresses` where the `checkConnect` doesn't throw a `SecurityException` will be returned in the List.

Returns	
List<InterfaceAddress>	a <code>List</code> object with all or a subset of the <code>InterfaceAddress</code> s of this network interface

getMTU

```
public int getMTU ()
```

Returns the Maximum Transmission Unit (MTU) of this interface.

Returns	
<code>int</code>	the value of the MTU for that interface.
Throws	
SocketException	if an I/O error occurs.

getName

```
public String getName ()
```

Get the name of this network interface.

Returns[String](#)

the name of this network interface

getNetworkInterfaces

```
public static Enumeration<NetworkInterface> getNetworkInterfaces ()
```

Returns all the interfaces on this machine. The [Enumeration](#) contains at least one element, possibly representing a loopback interface that only supports communication between entities on this machine. NOTE: can use `getNetworkInterfaces()+getInetAddresses()` to obtain all IP addresses for this node

For non-system apps, this method will only return information for [NetworkInterface](#) s associated with an [InetAddress](#) .

ANDROID NOTE: On Android versions before S (API level 31), this method may throw a `NullPointerException` if called in an environment where there is a virtual interface without a parent interface present.

Returns[Enumeration<NetworkInterface>](#)an Enumeration of NetworkInterfaces found on this machine that [are accessible](#).**Throws**[SocketException](#)

if an I/O error occurs.

getParent

```
public NetworkInterface getParent ()
```

Returns the parent NetworkInterface of this interface if this is a subinterface, or `null` if it is a physical (non virtual) interface or has no parent.

Returns[NetworkInterface](#)The [NetworkInterface](#) this interface is attached to.**getSubInterfaces**

```
public Enumeration<NetworkInterface> getSubInterfaces ()
```

Get an Enumeration with all the subinterfaces (also known as virtual interfaces) attached to this network interface.

For instance eth0:1 will be a subinterface to eth0.

Returns	
Enumeration<NetworkInterface>	an Enumeration object with all of the subinterfaces of this network interface

hashCode

```
public int hashCode ()
```

Returns a hash code value for the object. This method is supported for the benefit of hash tables such as those provided by [HashMap](#) .

The general contract of `hashCode` is:

- Whenever it is invoked on the same object more than once during an execution of a Java application, the `hashCode` method must consistently return the same integer, provided no information used in `equals` comparisons on the object is modified. This integer need not remain consistent from one execution of an application to another execution of the same application.
- If two objects are equal according to the `equals` method, then calling the `hashCode` method on each of the two objects must produce the same integer result.
- It is *not* required that if two objects are unequal according to the `equals` method, then calling the `hashCode` method on each of the two objects must produce distinct integer results. However, the programmer should be aware that producing distinct integer results for unequal objects may improve the performance of hash tables.

Returns	
<code>int</code>	a hash code value for this object.

isLoopback

```
public boolean isLoopback ()
```

Returns whether a network interface is a loopback interface.

Returns	
<code>boolean</code>	<code>true</code> if the interface is a loopback interface.
Throws	
SocketException	if an I/O error occurs.

isPointToPoint

```
public boolean isPointToPoint ()
```

Returns whether a network interface is a point to point interface. A typical point to point interface would be a PPP connection through a modem.

Returns	
boolean	true if the interface is a point to point interface.
Throws	
SocketException	if an I/O error occurs.

isUp

```
public boolean isUp ()
```

Returns whether a network interface is up and running.

Returns	
boolean	true if the interface is up and running.
Throws	
SocketException	if an I/O error occurs.

isVirtual

```
public boolean isVirtual ()
```

Returns whether this interface is a virtual interface (also called subinterface). Virtual interfaces are, on some systems, interfaces created as a child of a physical interface and given different settings (like address or MTU). Usually the name of the interface will be the name of the parent followed by a colon (:) and a number identifying the child since there can be several virtual interfaces attached to a single physical interface.

Returns	
boolean	true if this interface is a virtual interface.

supportsMulticast

```
public boolean supportsMulticast ()
```

Returns whether a network interface supports multicasting or not.

Returns	
boolean	true if the interface supports Multicasting.
Throws	
SocketException	if an I/O error occurs.

toString

```
public String toString ()
```

Returns a string representation of the object.

Returns	
String	a string representation of the object.

Source: <https://developer.android.com/reference/java/net/NetworkInterface.html>