

# Package ‘getip’

July 22, 2025

**Type** Package

**Title** 'IP' Address 'Lookup'

**Version** 0.1-4

**Description** A micro-package for getting your 'IP' address, either the local/internal or the public/external one. Currently only 'IPv4' addresses are supported.

**License** BSD 2-clause License + file LICENSE

**Depends** R (>= 3.0.0)

**ByteCompile** yes

**Maintainer** Drew Schmidt <wrathematics@gmail.com>

**RoxygenNote** 7.1.2

**NeedsCompilation** yes

**Author** Drew Schmidt [aut, cre],  
Wei-Chen Chen [aut]

**Repository** CRAN

**Date/Publication** 2023-12-10 21:10:02 UTC

## Contents

getip-package . . . . .	2
getip . . . . .	2
hostname2ip . . . . .	3
validip . . . . .	4
<b>Index</b>	<b>5</b>

---

`getip-package``getip`

---

**Description**

A micro-package for getting your IP address, either the local/internal or the public/external one. Currently only IPv4 addresses are supported.

**Author(s)**

Drew Schmidt

---

`getip``ip_external`

---

**Description**

Get your internal/local or external/public IP address. Currently only IPv4 addresses are supported.

**Usage**

```
getip(type = "local")
```

**Arguments**

type	One of "local" or "internal" for the local/internal IP, or one of "external" or "public" for the external/public IP.
------	--

**Details**

The internal/local IP lookup is done in-process, i.e., it does not call `system()` or the like.

For an external address, the function returns a lookup from one of several services, including Amazon AWS <http://checkip.amazonaws.com/>, httpbin <http://httpbin.org/ip>, ipify <https://www.ipify.org/>, and "My External IP address is ..." <http://myexternalip.com/>. You must be connected to the internet for this to work. Please note that pathological use could end up in your getting banned from these services, rendering the function useless. So don't call this function a billion times in a loop or something.

**Value**

Returns the requested IP address as a string.

**Examples**

```
library(getip)

# internal/local address
getip("local") # same as getip("internal")

# external/public
getip("public") # same as getip("external")
```

---

hostname2ip	<i>hostname2ip</i>
-------------	--------------------

---

**Description**

Converts a hostname (or vector of hostnames) to

**Usage**

```
hostname2ip(hosts)
```

**Arguments**

hosts                    A vector of hostnames as strings (or just one).

**Details**

Only IPv4 addresses will be returned. If only IPv6 addresses exist (and if that is the case, allow me to be the first to greet you, person from the distant future) then NA is returned for that hostname.

The function uses `getaddrinfo()` on \*NIX systems, and `gethostbyname()` on Windows.

**Value**

A list of vectors of IP's. Each IP is stored as a string.

**Examples**

```
library(getip)

hosts = c("www.google.com", "localhost", "www.yahoo.com")
hostname2ip(hosts)
```

---

`validip`*validip*

---

**Description**

Check if a string is a valid IP address. Currently only IPv4 addresses are supported.

**Usage**

```
validip(ip)
```

**Arguments**

`ip`                      A string or vector of strings.

**Details**

This function is not like a 'ping'. It merely checks if the string is correctly formatted and could theoretically be a valid IP address.

**Value**

Returns TRUE if the input is a potentially valid IP address, and FALSE otherwise.

**Examples**

```
library(getip)

validip(c("192.168.1.1", "700.168.1.1"))
```

# Index

\* **package**

    getip-package, [2](#)

getip, [2](#)

getip-package, [2](#)

hostname2ip, [3](#)

validip, [4](#)