

ipRestriction

August 18, 2026

ipRestriction

Kind: Function

Source: `src/middleware/ip-restriction/index.ts`

Part of: `Middleware`

IP Restriction Middleware for Hono.

`ipRestriction` creates Hono middleware that checks a request's client IP address against configured allow and deny lists. Attach it to an application or route to reject requests from disallowed addresses before route handlers run.

Signature

```
function ipRestriction(getIP: GetIPAddr, { denyList = [], allowList = [] }: IPRestrictionRules,
  onError: (remote: { addr: string; type: AddressType }, c: Context) => Response)
```

Parameters

Name	Type
<code>getIP</code>	<code>GetIPAddr</code>
<code>{ denyList = [], allowList = [] }</code>	<code>IPRestrictionRules</code>
<code>onError</code>	<code>`( remote: { addr: string; type: AddressType }, c: Context ) => Response`</code>

Returns: `MiddlewareHandler`

Diagram



```
Middleware --> IP[Read client IP]
IP --> Lists[Check allowList and denyList]
Lists --> |Allowed| Handler[Route handler]
Lists --> |Denied| Forbidden[Forbidden response]
```

Usage

```
import { Hono } from 'hono'
import { ipRestriction } from 'hono/ip-restriction'

const app = new Hono()

app.use(
  '*',
  ipRestriction({
    allowList: ['192.168.0.2', '10.0.0.0/24'],
    denyList: ['192.168.0.10'],
  })
)

app.get('/', (c) => c.text('Access granted'))

export default app
```

AI Coding Instructions

- Register `ipRestriction` with `app.use()` before routes that require IP checks.
- Keep IP addresses and CIDR ranges in `allowList` and `denyList` aligned with the deployment network.
- Test requests from both allowed and denied addresses when changing list rules.
- Account for proxies and platform networking when determining the client IP visible to Hono.