

# golang http 客户端使用 http 代理 /socks 代理

作者: [solarhell](#)

原文链接: <https://ld246.com/article/1545305136994>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

网上流传的用法大都引用了 [golang.org/x/net/proxy](https://golang.org/x/net/proxy) (并且使用了已经 `deprecated` 的 `dialer`), 实际现在可以直接用这种更简单的写法。

```
package main

import (
    "fmt"
    "io/ioutil"
    "net/http"
    "net/url"
)

const (
    HttpProxy = "http://127.0.0.1:6152"
    SocksProxy = "socks5://127.0.0.1:6153"
)

func main() {
    proxy := func(_ *http.Request) (*url.URL, error) {
        return url.Parse(HttpProxy)
    }

    httpTransport := &http.Transport{
        Proxy: proxy,
    }

    httpClient := &http.Client{
        Transport: httpTransport,
    }

    req, err := http.NewRequest("GET", "https://api.ip.sb/ip", nil)
    if err != nil {
        // handle error
    }

    resp, err := httpClient.Do(req)
    defer resp.Body.Close()

    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        // handle error
    }

    fmt.Println(string(body))

    proxy = func(_ *http.Request) (*url.URL, error) {
        return url.Parse(SocksProxy)
    }

    httpTransport = &http.Transport{
        Proxy: proxy,
    }

    httpClient = &http.Client{
```

```
    Transport: httpTransport,
}

req, err = http.NewRequest("GET", "https://api.ip.sb/ip", nil)
if err != nil {
    // handle error
}

resp, err = httpClient.Do(req)
defer resp.Body.Close()

body, err = ioutil.ReadAll(resp.Body)
if err != nil {
    // handle error
}

fmt.Println(string(body))
}
```

可以验证：打印的响应中确实为代理IP。