

# HTTPClient-Transport

gclient.Clientthttp.Clientthttp.Clienttgclient.ClientTransport

## Unix Socket

Unix SocketTransport

### Content Menu

- [Unix Socket](#)
- 

```
func (*Guardian) ConvertContainerPathToHostPath(
    ctx context.Context, namespace, podName, containerName,
    containerPath string,
) (string, error) {
    var (
        client = g.Client()
        url     = "http://localhost/api/v1/pod/path"
        req     = webservice.HostPathInfoReq{
            Namespace:    namespace,
            PodName:      podName,
            ContainerName: containerName,
            ContainerPath: containerPath,
        }
        res *webservice.HostPathInfoRes
    )
    client.Transport = &http.Transport{
        DialContext: func(ctx context.Context, network, addr
string) (net.Conn, error) {
            return net.Dial("unix", serviceSocketPath)
        },
    }
    err := client.ContentJson().GetVar(ctx, url, req).Scan(&res)
    if err != nil {
        return "", gerror.Wrapf(
            err,
            `request guardian failed for url: %s, req: %s`,
            url, gjson.MustEncodeString(req),
        )
    }
    if res == nil {
        return "", gerror.Newf(
            `nil response from guardian request url: %s, req: %
s`,
            url, gjson.MustEncodeString(req),
        )
    }
    return res.HostPath, nil
}
```

```

func ExampleNew_MultiConn_Recommend() {
    var (
        ctx    = gctx.New()
        client = g.Client()
    )

    // controls the maximum idle(keep-alive) connections to keep per-
host
    client.Transport.(*http.Transport).MaxIdleConnsPerHost = 5

    for i := 0; i < 5; i++ {
        go func() {
            if r, err := client.Get(ctx, "http://127.0.0.1:8999
/var/json"); err != nil {
                panic(err)
            } else {
                fmt.Println(r.ReadAllString())
                r.Close()
            }
        }()
    }

    time.Sleep(time.Second * 1)

    // Output:
    //{"id":1,"name":"john"}
    //{"id":1,"name":"john"}
    //{"id":1,"name":"john"}
    //{"id":1,"name":"john"}
    //{"id":1,"name":"john"}
}

```