



链滴

Windows 下 git 设置代理服务器

作者: [greatSean](#)

原文链接: <https://ld246.com/article/1545012913793>

来源网站: [链滴](#)

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

<h2 id="Windows下git设置代理服务器">Windows 下 git 设置代理服务器</h2>

<p>SVN 中，使用 TortoiseSVN 来进行版本控制时，设置代理非常简单，只需要在设置里面添加代的信息即可。而 <code>git</code> 在 GUI(v0.17.GITGUI)中却无法找到类似的设置，只能求助 <code>git bash</code> 来设置。</p>

<p>Git 支持四种协议 1，而除本地传输外，还有：git://，基于 HTTP 协议，这些协议又被分为哑议 (HTTP 协议) 和智能传输协议。对于这些协议，要使用代理的设置也有些差异：</p>

使用 <code>git协议</code> 时，设置代理需要配置 <code>core.gitproxy</code>

使用 <code>HTTP协议</code> 时，设置代理需要配置 <code>http.proxy</code>

而是用 <code>ssh协议</code> 时，代理需要配置 ssh 的 <code>ProxyCommand</code> 参数

<p>由于个人需求仅仅是 HTTP 的代理 (相对来说，HTTP 有比较好的通适性，Windows 配置 git/ss 比较棘手)，设置的时候，只需要针对单个设置 <code>http.proxy</code> 即可，在需要使用代的项目下面使用 <code>git bash</code> 如下命令进行设置——你的 Uri 和 port 可能和我的不，你懂的。：</p>

<p>git config http.proxy http://127.0.0.1 # 也可以是 uri:port 形</p>

<p>这个是不需要鉴权的代理设置，如果需要鉴权，可能需要添加用户名密码信息：</p>

<p>git config http.proxy http://username:password@127.0.0.1:8088</p>

<p>如果 git 的所有项目都需要启用代理，那么可以直接启用全局设置：</p>

<p>git config --global http.proxy http://127.0.0.1</p>

<p>为了确认是否已经设置成功，可以使用 <code>--get</code> 来获取：</p>

<p>git config --get --global http.proxy</p>

<p>这样可以看到你设置在 global 的 <code>http.proxy</code> 值。</p>

<p>需要修改的时候，再次按照上面的方法设置即可，git 默认会覆盖原有的配置值。</p>

<p>当我们的网络出现变更时，可能需要删除掉原有的代理配置，此时需要使用 <code>--unset</code> 来进行配置：</p>

<p>git config --global --unset http.proxy</p>

<p>在命令之后，指定位置的设置值将会被清空，你可以再次使用 <code>--get</code> 来查看具的设置情况。</p>

<p>如果使用了 HTTPS，肯呢个会碰到 HTTPS 证书错误的情况，比如提示：<code>SSL certificate problem</code>。。。，此时，可以尝试将 <code>sslVerify</code> 设置为 <code>>false</code> </p>

<p>git config --global http.sslVerify false</p>

<p>恩，到此，可以试试 git 来获取/更改项目了，此时，项目应该是使用代理来进行通讯的。</p>

<h4 id="后记-">后记：</h4>

<p>如果非必要，一般不使用 <code>--global</code> 的方式来设置代理，毕竟代理有的时候访一些项目比直接访问还慢，特别是当代理在国外，项目源在国内的时候，按需使用才是王道。</p>

<p>不要多次使用不同的参数来设置代理，一般使用文中两种方式酌情选用即可，<code>--global</code>，<code>--system</code>，<code>--local</code> 各级设置后，可能会给自己带来必要的麻烦。git 默认是先到 git Repository 的配置文件中查找配置文件，如果没有才会到 <code>--global</code> 设置的文件中查找，因此，单个项目文件中的设置会覆盖 <code>--global</code> 的设置。</p>

<p>使用 <code>--global</code> 来配置的信息保存在当前用户的根目录下的 <code>.config</code> 文件中，而仓库中的配置保存在项目仓库的根目录下的 <code>.git/config</code> 文件中。</p><p>如果是 Linux 的用户，再使用 <code>git/ssh协议</code> 时，根据网上的说法，需要使用 connect</p><p>工具来做代理的转换。—— 本人 Linux 下的暂时没有配置 git 使用代理，暂时无法验证，摘录法如下：</p><p>GIT 协议配置：安装完毕 <code>connect</code> 之后，你可以在特定的目录中建立一个 <code>socks5_proxy_wrapper</code> (或其他的文件名亦可)，然后文件内容改为：</p><pre>#!/bin/sh
connect -S 127.0.0.1:8088 "\$@"</pre><p>而后即可以配置 git 了，设置 <code>gitproxy</code> ：</p><pre>git config core.gitproxy /path/to/socks5_proxy_wrapper</pre><h2 id="路径要改">路径要改</h2><p>抑或 <code>export GIT_PROXY_COMMAND</code> ：</p><pre>export GIT_PROXY_COMMAND="/path/to/socks5_proxy_wrapper"</pre><p>SSH 协议配置，同样需要建立一个文件，假设命名为： <code>socks5_proxy_ssh</code> ，文件内容为：</p><pre>#!/bin/sh
ssh -o ProxyCommand="/path/to/socks5_proxy_wrapper %h %p" "\$@"</pre><p>配置 git 使用该文件：</p><pre>export GIT_SSH="/path/to/socks5_proxy_ssh"</pre><p>亦可配置 <code>~/.ssh/config</code> 中的 <code>ProxyCommand</code> 。</p><p>HTTP 协议配置，这里直接使用 <code>http.proxy</code> 设置成 <code>socks5</code> 即可：</p><pre>git config http.proxy </pre><p>如果要全部使用 HTTP 代理，可以将上面的 <code>socks5_proxy_wrapper</code> 文件设为：</p><pre>#!/bin/sh
connect -H 192.168.1.100:8080 "\$@"</pre><p>其他保持不变即可。而 <code>http.proxy</code> 设置同本文设置即可。</p>