Web Experience Toolkit
Collaborative open source project led by the Government of Canada

Home → Documentation → Working behind a proxy

Working behind a proxy

The following steps can be used for those working behind proxies that require authentication. All these methods will store your password in plain text, so use caution when doing this.

Setting the proxy for NPM installs

1. Open a command prompt
2. Run the following commands replacing `USERNAME`, `PASSWORD`, `PROXY_ADDRESS`, and `PROXY_PORT` with your network's information:
    1. `npm config set proxy http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT`
    2. `npm config set https-proxy http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT`

Setting the proxy for Git

1. Open a command prompt
2. Run the following commands replacing `USERNAME`, `PASSWORD`, `PROXY_ADDRESS`, and `PROXY_PORT` with your network's information:
    1. `git config --global --add http.proxy http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT`
    2. `git config --global --add https.proxy http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT`
3. Run the following commands to prevent SSH problems:
    1. `git config --global url."https://".insteadOf git://`
    2. `git config --global url."https://github.com/".insteadOf git@github.com:` (*don't* remove the final colon)
4. If your computer is within a corporate network that uses personal network drives, ensure identical copies of your .gitconfig file exist on both the network drive and your local drive's user profile (e.g. `F:\` and `C:\Users\USERNAME`).

Setting the proxy for other tasks

Many plugins will also look for environmental variables of `HTTP_PROXY` and `HTTPS_PROXY` when they need to call out to the internet.

1. You can set these variables with the value `http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT` each session by calling:
    1. Windows Command Prompt: `set HTTP_PROXY=http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT`
    2. Bash Shell: `export HTTPS_PROXY=http://USERNAME:PASSWORD@PROXY_ADDRESS:PROXY_PORT`
2. To avoid doing this every time, you can add these commands to your console's "RC" file. If you are using the Git Bash Shell on Windows, you can add the export statements to the ".bashrc" file in the root of your user profile (`C:\Users\USERNAME`). That file is run every time you open a Git Bash session, so the environmental variables will be available from the command line. You may need to create that file with a text editor, since Windows doesn't allow creating files starting with a period in Windows Explorer.

Date modified: 2014-06-20