

□ □ □ □ □ □ □ □ □ □

1. □□□□ □□□□□□□□□□□□□□:

2. □□□□ □□□□□□□□:

```
▢ export GLOBAL_PROXY='127.0.0.1:7890': မြန်မာ လူတေသန၏ အကြောင်းအရာ၏ GLOBAL_PROXY အမြန်ဆုံး မြန်မာ လူတေသန  
အကြောင်းအရာ၏ အကြောင်းအရာ၏။  
▢ function start_proxy { ... }: မြန်မာ လူတေသန၏ အကြောင်းအရာ၏ HTTP_PROXY, HTTPS_PROXY, http_proxy, https_proxy,  
မြန်မာ ALL_PROXY အကြောင်းအရာ၏ အကြောင်းအရာ၏။ မြန်မာ လူတေသန၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏  
အကြောင်းအရာ၏။  
▢ function start_proxy_without_prefix { ... }: မြန်မာ၏ အကြောင်းအရာ၏ start_proxy, မြန်မာ၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏  
အကြောင်းအရာ၏။ http:// အကြောင်းအရာ၏။  
▢ function stop_proxy { ... }: မြန်မာ၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏။ မြန်မာ၏ အကြောင်းအရာ၏  
အကြောင်းအရာ၏။ မြန်မာ၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏။  
▢ export NO_PROXY="localhost,127.0.0.1,.example.com,:1": မြန်မာ၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏ အကြောင်းအရာ၏  
အကြောင်းအရာ၏။
```

3. □□□□□□□;

- ❑ `function start_git_proxy { ... }:` မြန်မာ စိတ်ဝင်းဆောင်ရွက်ခြင်း ဖြစ်ပါသည်။
- ❑ `function stop_git_proxy { ... }:` မြန်မာ စိတ်ဝင်းဆောင်ရွက်ခြင်း ဖြစ်ပါသည်။

4. □□□□□□□□ □□□□□□□□□□:

eval "\$(/opt/homebrew/bin/brew shellenv)": 本地に Homebrew のインストール場所が見つかりません。
Homebrew をインストールするには、以下の手順を実行してください。

5. □□□□□□□□□□□□□□□□;

□ alias gm='python ~/bin/gitmessageai.py --only-message': 定義するコマンドをgmとして登録する。引数を省略するとgit commitのメッセージを表示する。
□ alias gpam=/usr/local/bin/git-auto-commit: 定義するコマンドをgpamとして登録する。git commitの代わりにgit-auto-commitを実行する。
□ alias rougify=/Users/lzwjava/projects/rouge/bin/rougify: 定義するコマンドをrougifyとして登録する。rougifyを実行する。

6. SSL証明書の設定:

□ export SSL_CERT_FILE=~/bin/cacert.pem: SSL証明書をcacert.pemとして登録する。

7. Homebrewの自動更新:

□ export HOMEBREW_NO_AUTO_UPDATE=1: Homebrewの自動更新を無効にする。

8. ネットワークコマンドの登録:

□ pexec() { ... }: ネットワークコマンドをpexecとして登録する。引数を省略するとcurlを実行する。
□ local network_commands=(...): ネットワークコマンドを登録する。引数を省略するとcurlを実行する。
□ display_proxy() { ... }: ネットワークコマンドを登録する。

9. Google Cloud SDKの登録:

□ if [-f '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc']; then . '/Users/lzwjava/bin/google-cl...
fi: Google Cloud SDKのCompletionを登録する。

10. 環境変数の登録:

- `export GOOGLE_APPLICATION_CREDENTIALS="/Users/lzwjava/bin/graphite-ally-445108-k3-035f0952219d.json"`: Sets the Google Cloud Application Credentials.
- `export DEEPESEEK_API_KEY="xxx"`: Sets the DeepSeek API key.
- `export MISTRAL_API_KEY="xxx"`: Sets the Mistral API key.
- `export DYLD_LIBRARY_PATH=\$(brew --prefix curl)/lib`: Sets the dynamic library path for curl.
- `export SPEECH_ENDPOINT="https://ai-lzwjava-5596.cognitiveservices.azure.com/"`: Sets the Azure speech endpoint.
- `export DO_API_KEY="xxx"`: Sets the Digital Ocean API key.
- `export GEMINI_API_KEY="xxx"`: Sets the Gemini API key.

11. Conda環境の登録:

- `conda activate base`: Activates the base conda environment.

ମୁଁ କାହାରେତେ, କୋଣ କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା:

- କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା.
- କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା.
- କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା.
- କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା.
- କିମ୍ବା-କିମ୍ବା-କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା-କିମ୍ବା-କିମ୍ବା କିମ୍ବା.
- କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା.
- କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା.

କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା'ର କିମ୍ବା କିମ୍ବା.

```
export PATH=/opt/homebrew/Cellar/openjdk@17/17.0.13/libexec/openjdk.jdk/Contents/Home/bin:/opt/homebrew/opt/r
```

```
# /opt/homebrew/opt/openjdk/bin
```

```
# /opt/homebrew/Cellar/openjdk@17/17.0.13/libexec/openjdk.jdk/Contents/Home/bin
```

```
export GLOBAL_PROXY='127.0.0.1:7890'
```

```
# export GLOBAL_PROXY='http://192.168.1.1:7890'
```

```
function start_proxy {
```

```
    export HTTP_PROXY="http://$GLOBAL_PROXY"  
    export HTTPS_PROXY="http://$GLOBAL_PROXY"  
    export http_proxy="http://$GLOBAL_PROXY"  
    export https_proxy="http://$GLOBAL_PROXY"  
    export HTTP_PROXY_REQUEST_FULLURI=false  
    export HTTPS_PROXY_REQUEST_FULLURI=false  
    export ALL_PROXY=$http_proxy
```

```
}
```

```
function start_proxy_without_prefix {
```

```
    export http_proxy=$GLOBAL_PROXY  
    export HTTP_PROXY=$GLOBAL_PROXY  
    export https_proxy=$GLOBAL_PROXY  
    export HTTPS_PROXY=$GLOBAL_PROXY  
    export HTTP_PROXY_REQUEST_FULLURI=false  
    export HTTPS_PROXY_REQUEST_FULLURI=false  
    export ALL_PROXY=$http_proxy
```

```
}
```

```

function stop_proxy {
    export http_proxy=
    export HTTP_PROXY=
    export https_proxy=
    export HTTPS_PROXY=
    export HTTP_PROXY_REQUEST_FULLURI=true
    export HTTPS_PROXY_REQUEST_FULLURI=true
    export ALL_PROXY=
}

export NO_PROXY="localhost,127.0.0.1,.example.com,::1"

function start_git_proxy {
    git config --global http.proxy $GLOBAL_PROXY
    git config --global https.proxy $GLOBAL_PROXY
}

function stop_git_proxy {
    git config --global --unset http.proxy
    git config --global --unset https.proxy
}

eval "$( /opt/homebrew/bin/brew shellenv )"

start_proxy
# start_git_proxy

# alias python3=/opt/homebrew/bin/python3
# alias pip3=/opt/homebrew/bin/pip3
# alias pip=python3

alias gpa='python ~/bin/gitmessageai.py --api mistral'
alias gca='python ~/bin/gitmessageai.py --no-push'
alias gm='python ~/bin/gitmessageai.py --only-message'

alias gpam=/usr/local/bin/git-auto-commit

# bundle exec jekyll serve

```

```

export SSL_CERT_FILE=~/bin/cacert.pem

alias rougify=/Users/lzwjava/projects/rouge/bin/rougify

# git config --global core.editor "code --wait"
# git config --global -e

export HOMEBREW_NO_AUTO_UPDATE=1

# Function to check and display proxy settings before certain commands
# Function to check and display proxy settings before certain commands
preexec() {
    # Define network-dependent commands
    local network_commands=(
        "gpa"
        "git"
        "ssh"
        "scp"
        "sftp"
        "rsync"
        "curl"
        "wget"
        "apt"
        "yum"
        "dnf"
        "npm"
        "yarn"
        "pip"
        "pip3"
        "gem"
        "cargo"
        "docker"
        "kubectl"
        "ping"
        "traceroute"
        "netstat"
        "ss"
        "ip"
        "ifconfig"
    )
}
```

```

"dig"
"nslookup"
"nmap"
"telnet"
"ftp"
"nc"
"tcpdump"
"adb"
"bundle"
"brew"
"cpanm"
"bundle exec jekyll"
"make"
"python"
"glcoud"
# Add more commands as needed
)

# Extract the first word (command) from the command line
local cmd
cmd=$(echo "$1" | awk '{print $1}')

# Function to display proxy variables
display_proxy() {
    echo -e " **Proxy Settings Detected:**"

    [ -n "$HTTP_PROXY" ] && echo " - HTTP_PROXY: $HTTP_PROXY"
    [ -n "$HTTPS_PROXY" ] && echo " - HTTPS_PROXY: $HTTPS_PROXY"

    echo ""
}

# Check if the command is network-dependent
for network_cmd in "${network_commands[@]}"; do
    if [[ "$1" == "$network_cmd"* ]]; then
        if [ -n "$HTTP_PROXY" ] || [ -n "$http_proxy" ] || \
           [ -n "$HTTPS_PROXY" ] || [ -n "$https_proxy" ] || \
           [ -n "$ALL_PROXY" ] || [ -n "$all_proxy" ]; then
            display_proxy
    fi
fi
}

```

```
    fi
    break
fi
done
}

# The next line enables shell command completion for gcloud.
if [ -f '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc' ]; then . '/Users/lzwjava/bin/google-cloud-s

export GOOGLE_APPLICATION_CREDENTIALS="/Users/lzwjava/bin/graphite-ally-445108-k3-035f0952219d.json"

export DEEPSEEK_API_KEY="xxx"

export MISTRAL_API_KEY="xxx"

export DYLD_LIBRARY_PATH=$(brew --prefix curl)/lib

export SPEECH_ENDPOINT="https://ai-lzwjava-5596.cognitiveservices.azure.com/"

export DO_API_KEY="xxx"

export GEMINI_API_KEY="xxx"

conda activate base
```