

000 00000000

0000 000 00000000 00 00000000 00 00000000 000 0000'0 0000000-00000 00000000000, 0000000000000 00000000 00 000000 00000000000, 000 000000000000, 000 000000000000 00000000. 00000'0 0 00000000000:

1. 0000 000000000000000:

```
0 export PATH=...: 0000 0000 0000 000 PATH 000000000000 000000000, 000000 000000 000 000000 000000 00 00000 000 000000000000 000000. 00 0000000000 000000000 000000000000 00000 000000 000 00000, 00000, 0000000000, 00000000, 000000000, 000 00000000 000000 0000. 00000 000000000 00000 0000000000 00000 000000 000000 000 00 000 0000000000 00000 000 0000000000.
```

2. 000000 0000000000000:

```
0 export GLOBAL_PROXY='127.0.0.1:7890': 00000 000000000 0 0000000000 GLOBAL_PROXY 000000000 0000 000000 00000000 000000000.
0 function start_proxy { ... }: 00000 0000000000 00000 0000 HTTP_PROXY,HTTPS_PROXY,http_proxy,https_proxy, 0000 ALL_PROXY 00000000000000 00000000000 00 0000 0000 00000000000 0000000. 00 00000 0000000000 00000 0000 00-00000000 0000 0000000000.
0 function start_proxy_without_prefix { ... }: 000000000 00 start_proxy, 0000 00000 0000 0000000 00000000000 000000000 0000 http:// 00000000.
0 function stop_proxy { ... }: 00000 0000000000 00000000 0000 000000 00000000000, 000000000000000 000000000000 0000 0000000. 00 00000 000000000 00000 0000 0000000000 0000 0000000000.
0 export NO_PROXY="localhost,127.0.0.1,.example.com,:1": 00000 00000000000 0 00000 00 000000 00000 00000000 00000000 00000 0000000.
```

3. 0000 0000000:

```
0 function start_git_proxy { ... }: 00000 0000000000 00000000000000 0000 00 0000 0000 00000000 0000000 0000 000000 000000000000000.
0 function stop_git_proxy { ... }: 00000 00000000000 000000000 0000 0000 000000 00000000000.
```

4. 00000000000 000000000000000:

```
0 eval "$(/opt/homebrew/bin/brew shellenv)": 00000 00000 0000000000000 00000000000 00000 0000 0000000 000000000-00000, 00000000000 0000 00 0000 00000000000 00000000000.
```

5. 000000000000000 0000000000:

```
0 alias gpa='python ~/bin/gitmessageai.py --api mistral': 00000 0000000000 00 0000000 gpa 00 0000 0 00000000 00000000 gitmessageai.py 000000 0000 0000000000 0000.
0 alias gca='python ~/bin/gitmessageai.py --no-push': 00000 0000000000 00 0000000 gca 00 0000 0000 00000 00000000 0000000000 0000000000.
```

```

alias gm='python ~/bin/gitmessageai.py --only-message': 通过 gitmessageai.py 脚本生成 git 提交消息
并保存在 .gitmessage 文件中。

alias gpam=/usr/local/bin/git-auto-commit: 通过 gpam 脚本自动生成 git-auto-commit
消息。

alias rougify=/Users/lzwjava/projects/rouge/bin/rougify: 通过 rougify 脚本自动生成
rougify 消息。

```

6. 设置环境变量:

```

export SSL_CERT_FILE=~/.bin/cacert.pem: 设置 SSL 证书文件路径。

```

7. 设置环境变量:

```

export HOMEBREW_NO_AUTO_UPDATE=1: 禁用 Homebrew 的自动更新。

```

8. 设置环境变量:

```

preexec() { ... }: 在每次执行命令前，先执行 preexec 函数中的命令。该函数用于在每次
执行命令前，先执行一些预处理操作，如设置环境变量、加载配置文件等。

local network_commands=( ... ): 设置本地网络命令的别名。

display_proxy() { ... }: 设置显示代理的函数。

```

9. 设置环境变量:

```

if [ -f '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc' ]; then . '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc'; fi: 设置 google-cloud-sdk 的补全脚本。

```

10. 设置环境变量:

```

export GOOGLE_APPLICATION_CREDENTIALS="/Users/lzwjava/bin/graphite-ally-445108-k3-035f0952219d.json": Sets the Google Application Credentials.
export DEEPSEEK_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. 设置环境变量:

```

conda activate base: Activates the base conda environment.

```

- 00000 000000 000000000000 0000 0000000000 00 00000 000 0000 00000000.
- 000 00000 000000000000000.
- 000000000000 0000 000000000.
- 00000000000 0000000 000 000000 00000.
- 000-000000000 00000 000000 000 0000000-0000000000 00000000.
- 000 0000 000 000000000000 000 00000000 00000000.
- 00000 0000000000000 00000000000.

```
export PATH=/opt/homebrew/Cellar/openjdk@17/17.0.13/libexec/openjdk.jdk/Contents/Home/bin:/opt/homebrew/opt/ruby/bin:/usr/local/bin:/usr/bin:/usr/sbin:/sbin:/opt/homebrew/bin:/opt/homebrew/sbin

# /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_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=pip3

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"
"glcloud"
# 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
done

```

```

        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

```