



```
alias rougify=/Users/lzwjava/projects/rouge/bin/rougify: سكريبت لتشغيل rougify الاختصار هذا ينشئ  
rougify.
```

## 6. شهادة:

```
export SSL_CERT_FILE=~/.bin/cacert.pem: مخصص. شهادة ملف إلى المسار هذا يعين
```

## 7. التلقائي التحديث:

```
export HOMEBREW_NO_AUTO_UPDATE=1: التلقائي التحديثات هذا يعطل.
```

## 8. التنفيد: قبل الوكيل فحص

```
preexec() { ... } الـ شبكة. على المدة الـ أوامر قائم في الأمر كان إذا مما تتحقق أمر. كل قبل الدالة هذه تنفيذي تم: { ... }  
الوكيل. إعدادات يعرض إنه وكي، متغيرات أي تعيّن تم وإذا كذلك، الأمر كان إذا  
local network_commands=( ... ) الـ شبكة. على المدة تتعبّر التي الـ أوامر المصروفة هذه تحدد: ( ... )  
display_proxy() { ... } الحالية. الوكيل إعدادات الدالة هذه تعرض: { ... }
```

## 9. أوامر إكمال:

```
if [ -f '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc' ]; then . '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc'; fi: أوامر إكمال تمكين على السطر هذا يعمل
```

## 10. والشهادات: مفاتيح

```
`export GOOGLE_APPLICATION_CREDENTIALS="/Users/lzwjava/bin/graphite-ally-445108-k3-035f0952219d.json":  
`export DEEPSEEK_API_KEY="xxx": API DeepSeek.  
`export MISTRAL_API_KEY="xxx": API Mistral.  
`export DYLD_LIBRARY_PATH=$(brew --prefix curl)/lib: curl.  
`export SPEECH_ENDPOINT="https://ai-lzwjava-5596.cognitiveservices.azure.com/": Azure.  
`export DO_API_KEY="xxx": API Digital Ocean.  
`export GEMINI_API_KEY="xxx": API Gemini.
```

## 11. بيئية:

```
`conda activate base`: Conda .
```

ذلك: في بما للمطور، شاملًا إعدادًا هذا تّعريف ملف يوفر باختصار،

```
الوكيل. وإيقاف لبدء دوال مع للوكيل سهولة إدارة  
وكيل تكوين.  
مع تكامل.
```

☐ للراحة. أوامر اختصارات

☐ الشبكية. على الممتدة للأوامر للتنفيذ قبل الوكيل فحوص

☐ المختلطة. للخدمات والشهادات ☐☐☐ مفاتيح

☐ .☐☐☐☐☐ بيئية تنشيط.

أسهل. المخطلة التطوير وخدمات أدوات وإدارة وجعل المستخدم عمل سير العمل بسيط المرف هذا تصميم تم

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

```
# /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=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
        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
```