Zsh Profile

This zsh profile is designed to enhance the user's command-line experience, particularly focusing on proxy management, git integration, and convenience aliases. Here's a breakdown:

1. **PATH Configuration:**

   - `export PATH=...`: This line sets the `PATH` environment variable, which tells the shell where to look for executable files. It includes various directories like those for Java, Ruby, Homebrew, Python, Flutter, and Google Cloud SDK. This ensures that commands from these tools can be run directly from the terminal.

2. **Proxy Management:**

   - `export GLOBAL_PROXY='127.0.0.1:7890'`: This defines a variable `GLOBAL_PROXY` holding the proxy server address.
   - `function start_proxy { ... }`: This function sets the `HTTP_PROXY`, `HTTPS_PROXY`, `http_proxy`, `https_proxy`, and `ALL_PROXY` environment variables to use the specified proxy. It also disables full URI requests for proxies.
   - `function start_proxy_without_prefix { ... }`: Similar to `start_proxy`, but sets the proxy variables without the `http://` prefix.
   - `function stop_proxy { ... }`: This function unsets the proxy variables, effectively disabling the proxy. It also enables full URI requests for proxies.
   - `export NO_PROXY="localhost,127.0.0.1,.example.com,::1"`: This specifies a list of hosts that should bypass the proxy.

3. **Git Proxy:**

   - `function start_git_proxy { ... }`: This function configures git to use the global proxy for HTTP and HTTPS connections.
   - `function stop_git_proxy { ... }`: This function unsets the git proxy settings.

4. **Homebrew Integration:**

   - `eval "$(/opt/homebrew/bin/brew shellenv)"`: This line integrates Homebrew into the shell environment, allowing you to use Homebrew commands.

5. **Convenience Aliases:**

   - `alias gpa='python ~/bin/gitmessageai.py --api mistral'`: This creates an alias `gpa` to run a python script `gitmessageai.py` with the mistral API.
   - `alias gca='python ~/bin/gitmessageai.py --no-push'`: This creates an alias `gca` to run the same script without pushing changes.

- `alias gm='python ~/bin/gitmessageai.py --only-message'`: This creates an alias `gm` to run the same script and only print the commit message.
- `alias gpam=/usr/local/bin/git-auto-commit`: This creates an alias `gpam` to run the `git-auto-commit` script.
- `alias rougify=/Users/lzwjava/projects/rouge/bin/rougify`: This creates an alias `rougify` to run the `rougify` script.

6. SSL Certificate:

- `export SSL_CERT_FILE=~/bin/cacert.pem`: This sets the path to a custom SSL certificate file.

7. Homebrew Auto-Update:

- `export HOMEBREW_NO_AUTO_UPDATE=1`: This disables Homebrew's automatic updates.

8. Pre-Execution Proxy Check:

- `preexec() { ... }`: This function is executed before every command. It checks if the command is in a list of network-dependent commands. If it is, and if any proxy variables are set, it displays the proxy settings.
- `local network_commands=( ... )`: This array lists commands that are considered network-dependent.
- `display_proxy() { ... }`: This function displays the current proxy settings.

9. Google Cloud SDK Completion:

- `if [ -f '/Users/lzwjava/bin/google-cloud-sdk/completion.zsh.inc' ]; then . '/Users/lzwjava/bin/google-clo fi`: This line enables shell command completion for gcloud.

10. API Keys and Credentials:

- `export GOOGLE_APPLICATION_CREDENTIALS="/Users/lzwjava/bin/graphite-ally-445108-k3-035f0952219d.json"`: Sets
- `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 endpo
- `export DO_API_KEY="xxx"`: Sets the Digital Ocean API key.
- `export GEMINI_API_KEY="xxx"`: Sets the Gemini API key.

11. Conda Environment:

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

In summary, this zsh profile provides a comprehensive setup for a developer, including:

- Easy proxy management with functions to start and stop proxies.
- Git proxy configuration.
- Integration with Homebrew.
- Convenient aliases for common tasks.
- Pre-execution proxy checks for network-dependent commands.
- API keys and credentials for various services.
- Conda environment activation.

This profile is designed to streamline the user's workflow and make it easier to manage various development tools and services.

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

# /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 {
```

```bash
    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"
```

```bash
    "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
        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-sc

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
```