

项目函数说明

src

`main.ts`: 负责插件初始化。

1. constructor: 初始化插件。

1. 确定local系统是否为win, 如果是, 保留powershell版本。

2. 支持devstar web的open with vscode链接 `/openProject`。

1. 如果用户没有登录, 则直接登录, 然后打开项目。

2. 如果同用户已经登录, 则直接打开项目。

3. 如果已登录某个用户, 将要登录不同的用户, 可以选择切换用户, 或者不切换登录用户, 直接打开。

3. 支持remote env打开项目的链接 `/openProjectSkippingLoginCheck`。

4. 加载侧边栏。

5. 注册全局变量。

6. 加载home页面。

`user.ts`: 负责与用户有关的操作

1. constructor

1. 加载当前global state中存储的token、username

2. 提取devstar域名, 用于本地ssh key的命名

2. login: 负责插件上的用户登录

1. 验证token、username是否正常

2. 检查本地是否有用户所属公钥, 没有则创建后上传

3. logout: 登出

4. isLogged: 是否登录

5. getUsernameFromLocal: 从global state中取出username

6. getUserTokenFromLocal: 从global state中取出token

7. setUsernameToLocal: 将username存储到global state中

8. setUserTokenToLocal: 将token存储到global state中

9. updateLocalUserPrivateKeyPath: 更新global state中用户所属ssh private key的路径

10. getLocalUserPrivateKeyPath: 获取global state中用户所属ssh private key的路径

11. getUserPrivateKeyPath: 获取用户所属ssh private key的路径

12. getUserPublicKeyPath: 获取用户所属ssh public key的路径

13. existUserPublicKey: 用户所属ssh public key是否存在

14. getUserPublicKey: 获取用户所属ssh public key

15. getUserPrivateKey: 获取用户所属ssh private key

16. createUserSSHKey: 创建用户所属ssh key (命名方式: id_rsa_username_domain)

`devstar-api.ts`: 负责调用devstar-api。

1. constructor函数: 获取devstar domain。
2. verifyToken函数: 验证token是否正确。
3. uploadUserPublicKey: 上传用户公钥。

`home.ts`: 负责显示home页面, 并与其通信。

1. constructor: 初始化context、user、remoteContainer、devstar domain
2. toggle
 1. 创建panel, 用于显示home页面。
 2. 与home页面通信 (get是指home get from vscode, set是指将set in vscode)
 1. getHomeConfig: 获取语言选项
 2. getUserToken: 获取token
 3. getUsername: 获取username
 4. firstOpenRemoteFolder: 第一次打开project
 5. openRemoteFolder: 直接打开project
 6. getDevstarDomain: 获取devstar的域名 (用于api)
 7. setUserToken: 将token存储到global state中
 8. setUsername: 将username存储到global state中
 9. getUserPublicKey: 获取当前用户的ssh公钥
 10. createUserPublicKey: 为当前用户创建属于他的ssh公钥
 11. getMachineName: 获取当前机器名称
 12. showInformationNotification、showWarningNotification、showErrorNotification: 调用vscode的notification显示info、warn、error信息
3. getWebviewContent: 嵌入home页面。
 1. 含有用于通信的函数

`remote-container.ts`: 负责与远程容器交互

1. constructor: 初始化user
2. firstOpenProject: 第一次打开远程项目
 1. local env: 调用firstConnect函数, 成功直接打开项目
 2. remote env: 打开local terminal, 根据插件初始化时存储的shell信息, 确定在terminal中执行的命令 (新window并执行local env的流程)
3. firstConnect: 第一次连接容器
 1. 检查公私钥是否存在, 如果不存在则创建后上传公钥。
 2. 通过ssh key连接容器, 并为容器安装vscode server, 安装devstar插件
 3. 上述步骤成功执行, 则将ssh信息存储在config文件中
4. storeProjectSSHInfo: 存储project的ssh连接信息
5. openRemoteFolder: 打开project (需要登录)
6. openProjectWithoutLogging: 无须登录, 直接打开project

`utils.ts`: 工具

1. getVsCodeCommitId: 获取当前机器上vscode版本的commit id, firstConnect中连接远程容器后为其安装vscode server时用到。
2. devstarDomain: 获取devstar domain
3. showErrorNotification: 如果出现错误, 为用户提供两个选择: 打开控制台查看错误、发issue报告错误
4. powershellVersion: 获取powershell的版本

