2024-06

2024-06-week3

☑读取客户端.ssh/*.pub key，然后传入服务器，存储到~/.ssh/authorized_keys

2024-06-week4

☑完成firstconnect

☑测试webview call vscode

2024-07

2024-07-week1

☑js转ts

☑测试嵌入插件中的网页，能不能调用起acquireVscodeApi

☑实现网页调用remote-ssh打开远程文件夹

2024-07-week2

☑自己实现一个fetch函数，嵌入网页内容，测试vscode中打开能否正常调用vscode接口

☑将remote-container的相关功能从main.ts中分离出来

功能：firstConnect，openRemoteFolder

☑对接前端

first open project时，先完成第一次链接（并上传publickey），之后在vscode中自动打开远程文件夹；

open project时，自动打开远程文件夹；

~~上传publickey之后请求修改本机密码~~

2024-08~2025-03

package.json

- view
    - quick access
- contribute
    - 设置
        - devstar domain
    - 命令
        - open home page
    - URL
        - open project

Main

设计

1. 全局只维护一个User类，保证状态一致。

中英双语支持

- [x] 支持中英双语

Home页面语言设置有三处

- [x] 原home页面处的组件会根据vscode中的语言设置 `ctx.Locale.Tr`，并显示对应的语言
- [x] 后增加的静态组件，如Login提示、登录窗口等
- [x] 后增加的动态组件，如每个项目容器对应的button
- [ ] 通知vscode的信息内容

VSCode语言设置

- [x] package.json控制显示的内容，需要两个文件：package.nls.json、package.nls.zh-CN.json。（插件名称、描述、命令名称等）
- [x] l10n/bundle.l10n.zh-CN.json，负责插件内部语言，如侧边栏中的选项名称，tong等。

只配置显示语言，error报错语言用英文

Home页面：将devstar的页面嵌入vscode中，并能正常使用

- [x] 实现devstar、webview和vscode的通信，数据传输
- [x] 将在vscode中使用的功能（登录/项目相关）集中到单个网页中

vscode webview web content

- [x] 将index页面合并回webview的web content中。
- [x] devstar home page url由package.json中的字段配置。

index页面 (webview)

（截至2024.11.08）将这部分内容放到index页面中开发（方便测试）。

- [x] 重构作为home页面与vscode通信中转的功能，将其简化并抽象（abstract and simplify the communication between home page and vscode)
    - index页面作为单纯的消息中转站，不再对某个具体的action做出额外操作，只按规定的格式为home和vscode传递信息（home -> vscode -> home）

接口定义

提供给home页面的接口，postMessage的格式

```
{
    "target": "vscode",
    "action": ${action},
    "data": ${data}
}
```

和vscode通信的接口

communicateVSCode

```
{
    "command": ${action},
    "data": ${data}
}
```

返回给home页面的内容，格式

```
{
    "action": ${action}
    "data": ${data}
}
```

action和home发送过来的action名称保持一致；

data就是vscode发送过来的data，原封不动；

devstar home页面（以及*配套的vscode插件功能*）

暂时（截至2024.11.08之前）将这部分内容放在本地（home.html）开发，方便测试。页面中和vscode通信的功能完成后，移至devstar后端项目中继续开发。

home页面与vscode通信

- 通过webview作为中间部分，周转信息
- [x] 单向通信（从home页面到vscode），不要返回值。
- [x] 重构biCommunication2Webview为communicateVSCodeByWebview
    - 通信格式参考index页面提供的接口

读取配置

- [x] 从package.json中读取api的域名
    - [x] 把所有api的域名集中为一个全局变量

信息通知

- [x] 统一通知的样式和调用。（通知的显示统一交给vscode）
- [x] 错误信息需要反馈给用户
    - 登录；创建repo；创建devcontainer；删除devcontainer

登录

- [x] 存储user name和user token

上传【新建】的公钥这个操作只在登录成功之后执行一次。

- [x] 登录后刷新和登录状态有关的页面模块。
- [x] 检查是否有名为 `id_rsa_${username}` 的密钥，如果存在，则视为服务端已经存储对应的公钥，不用操作；如果不存在，则新建密钥对，上传公钥
    - 公私钥名称一致
    - [x] （请求vscode执行）usename需要和userToken一样，存储到vscode global state中，生命周期和userToken一致。

        - 作为home页面的全局变量
        - 存/取和userToken保持一致
    - [x] （请求vscode执行）获取属于用户的公钥
    - [x] （请求vscode执行） 如果没有指定的公钥，则新建公钥
    - [x] 如果是新建的公钥，再次请求获取该公钥，然后上传到服务器
        - 公钥名称为 `${username}-${machineName}`
        - [x] (请求vscode执行)获取machine name

登出

- [x] user name和user token设置为空
- [x] 刷新和登录状态有关的页面模块。

显示repolist

- [x] 重构，不再用table组织，用div加上flex-list组织，按钮根据容器的状态变化。
- [x] 简单显示固定数量的repolist

新增repo

- [x] 基本完成新增repo的功能
- [x] 复用devstar web新建仓库的样式代码
- [[js获取表单中的数据]]
- [x] 使用mock数据，先将样式与web端保持一致。
- [x] 真实数据从api获取。
    - [x] 把所需数据的json格式整理出来，发issue。
    - [ ] 获取template的api

创建devContainer

- [x] 创建devContainer
    - [x] 重构：请求创建容器
- [x] 重构：创建容器后等待容器创建完成，此时创建容器的按钮为灰色（不可选中）；轮询容器创建状态，直到容器完全准备好，才能消失；然后唤起open project与delete container。

打开devContainer

- [x] (请求vsocde执行) 添加等待容器启动的提醒
- [x] 获取已经启动容器的连接信息 (api：通过repoId打开devContainer，如果容器正常启动，返回容器的ssh连接信息)
- [x] 取消手动拼接：~~进入容器的默认路径设为/data/workspace（clone项目的默认存储路径）~~

删除开发容器

- [x] 如果有容器，再显示delete

侧边栏

Home

clean

与远程容器交互（Remote-Container类）

第一次打开容器

区分local和remote环境

- local：firstConnect成功之后，打开项目
- remote：打开新的local窗口，然后执行local环境下第一次打开容器的操作。

第一次连接容器

- 重构：第一次连接容器，通过密钥对连接
    - ~~重载firstConnect，firstConnect接口中password为可选。~~ 仅存在密钥登录方式
    - [x] 如果本地不存在密钥对，新生成一对，存储到本地。
        - [x] 重构：新生成的密钥对，名称为id_rsa_${username}
        - [x] public key存储public key fingerprint
        - [x] 私钥[读/写]的权限限制在当前用户（600）
        - [x] 私钥的passphrase设置为空字符串
        - [x] 密钥对的type为pkcs1
- 存储project的ssh连接信息
    - [x] 传入参数：**host**、hostname、port、username
    - [x] 检查ssh config文件中是否已有当前容器的连接信息，没有再添加
    - [x] ~~将hostname与port拼接为host，方便分辨容器和打开容器时选择容器~~

打开容器

- [x] host的实际名称为 `${host}-${port}`

User类

负责所有与用户相关的操作

remote环境支持

- [x] 支持get/update local user private key path（通过global state中存取）

local环境支持

- [x] 存/取username和usertoken
- [x] 获取public/private key的路径
    - 必须要处于登录状态（vscode gloabl state中user name和user token不为空）才能查询到密钥对的位置，否则拿到的路径为空
    - 密钥对名称为id_rsa_${username}
- [x] 检查user public/private key是否存在
- [x] 获取user public/private key
    - [x] 去除public key中的 `\r` 和 `\n`
- [x] 添加新的user ssh key

公共方法

- [x] 将和user ssh有关的方法集中到User类中
- [x] 获取vscode的commit id

2025-04

- [x] 在已经remote wsl或remote ssh的环境下，正常触发devstar插件进入devcontainer
- [ ] 【调研发现】
    1. vscode server与local端能够双向同步状态，如global state、display language。
    2. vscode 远程环境独有命令：`workbench.action.terminal.newLocal`

2025-05

- [x] feat
  成功SSH登录devcontainer后，左下角显示SSH:dev-PROJECT_NAME
    - [x] open with vscode链接添加host=repoName参数
    - [x] home页面：打开devcontainer的函数，参数包括project的连接信息与repoName
    - [x] devstar-vscode通信接口：打开project的接口新增host参数（repoName）
    - [x] 重构firstOpenProject、firstConnect，新增host参数（repoName）
- [x] chore
  重建虚拟机环境
- [x] feat
  local/remote环境支持open with vscode链接
    - [x] firstOpenProject、firstConnect区分local/remote环境下的操作
    - [x] 存储host信息区分local/remote环境下的操作
    - [x] open with vscode区分local/remote环境下的操作
- [x] 优化open project的启动速度
    - [x] 调研为什么会慢（网速问题）
- [x] 新建容器完成之后，再显示open/delete project的按钮。

2025-06

- [x] bug
  vsocde-home：创建新仓库后，需要自动创建devcontainer.json，否则无法创建devcontainer
    - [x] 通过api创建容器时，如果没有devcontainer.json，则创建，再继续
- [x] feat
  devstar：api新增容器状态字段，用于确认容器是否可连接
- [x] bug
  ~~vscode-home：避免重复打开一个项目~~
    - [x] plugin/openRemoteFolder：local环境下，打开目标项目之前，把项目名称（host）存储到global host中，供remote环境查用。
    - [x] plugin/home通信接口：给home提供当前打开项目的名称（如果打开）
    - [x] home：全局变量：被打开的目标项目名称
    - [x] home/repolist，被打开项目不显示按钮

- [x] devstar/open with vsocde：提供目标项目名称
- [x] plugin/open with vscode功能检测即将打开的项目是否已经打开，已经打开则提醒用户