

WROIS(实习报告)

学生姓名：李怡龙 指导老师：黄震春、李东

一、主要内容

- 1、开发基于 code-server 的多用户在线 Web IDE
- 2、针对超级机的远程操作
- 3、与 file manager、data manager 等交互及集成
- 4、用户管理、插件管理
- 5、远程编译、远程数据预览等

二、所用到的技术

前端：HTML、CSS、JavaScript、Vue.js、Bootstrap、Xterm.JS

后端：Golang

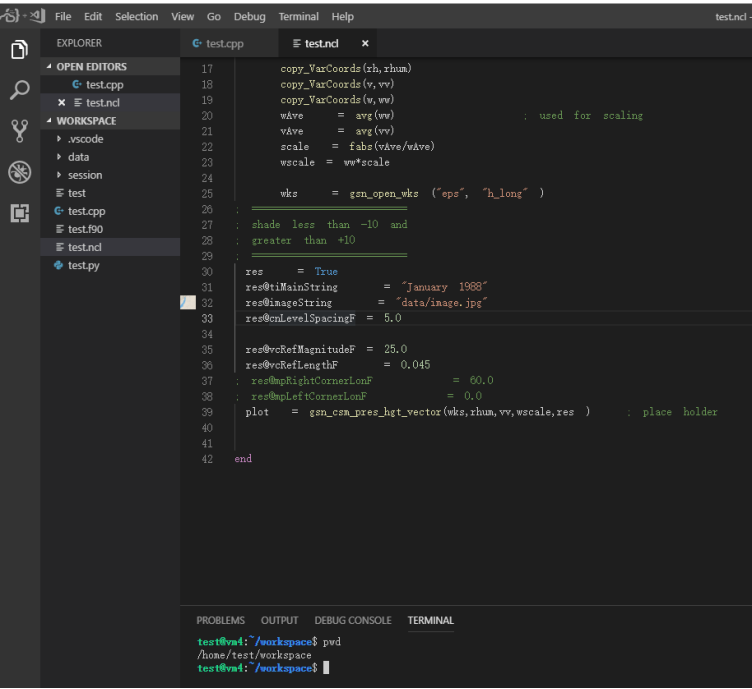
数据库：Sqlite、Redis

三、目前工作进度详细介绍

- 1、多用户的远程 Web IDE
目前已经完成多用户的开发，超时操作、资源回收、监控等操作。
登录页面：



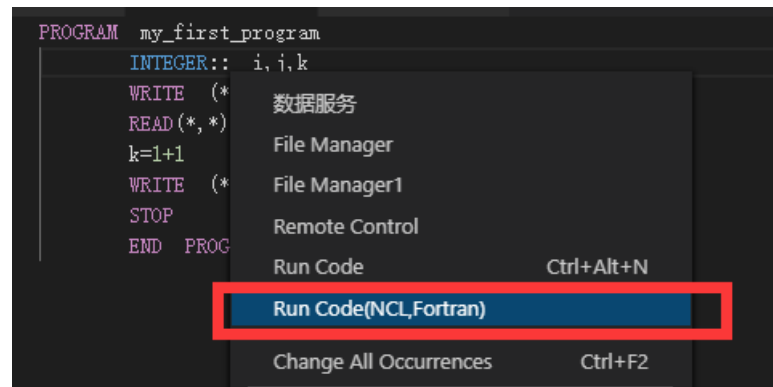
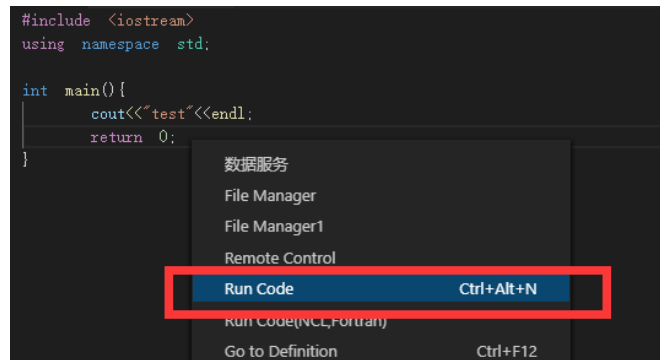
登录成功后页面:



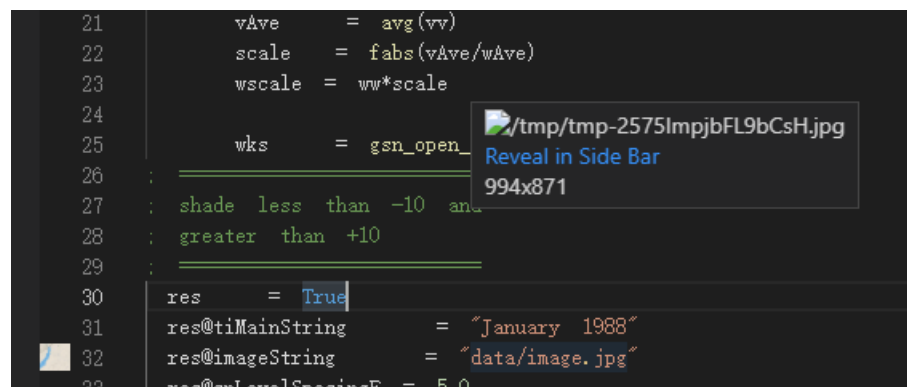
2、针对超级机的远程操作
目前正在进行远程执行插件的开发



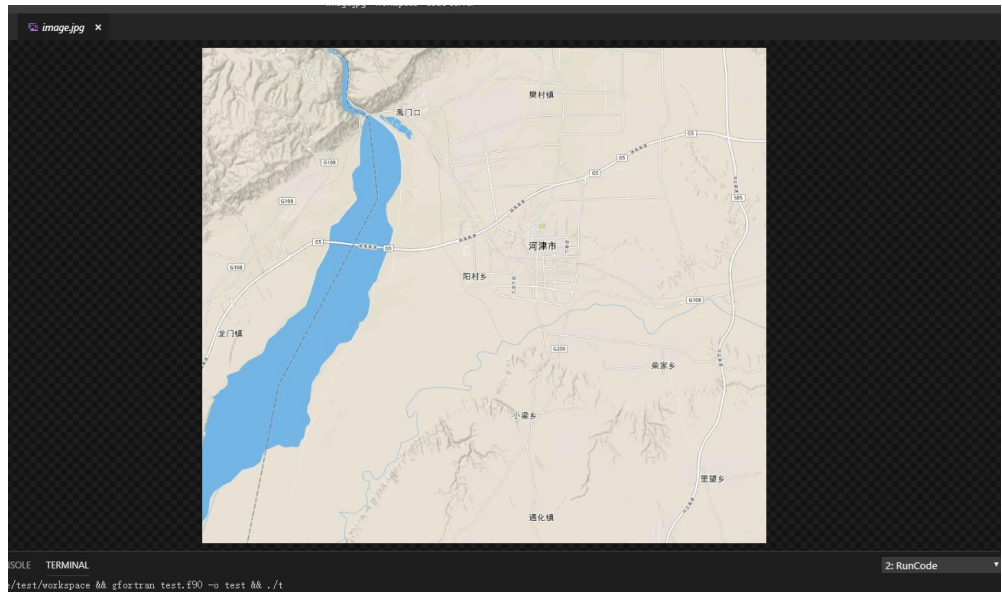
远程代码执行



资源预览(代码内嵌)



资源预览(打开)



- 3、与 file manager、data manager 等交互及集成
编写了一个万能插件(可以转跳至外部或直接启动在 VScode 中)
- 4、用户管理、插件管理
用户中心

WROIS Web IDE-用户中心

个人信息

你可以在这里修改个人信息和相关个性化配置

用户名(无法修改):

密码:

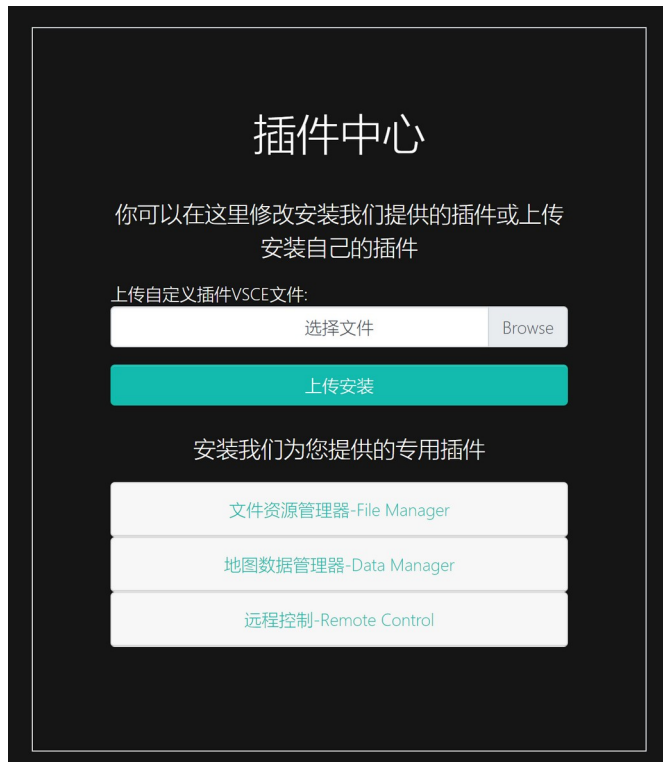
确认密码:

默认启动WorkSpace(默认为\$(HOME)/workspace):

IDE最大超时退出时间(留空恢复默认值300):

[进入插件中心](#)

插件中心



四、 每日工作简要

2019 年 7 月 19 日：

远程数据预览程序 KKFileView 的搭建与配置。

该服务是一个文件在线预览服务，基于 Java，引入该服务的重要原因是因为 code-server 原生支持的预览格式太少，常见的视频、document 都无法预览。

2019 年 7 月 22 日：

重构 code-server 的多用户机制

之前的多用户机制存在部分问题，所以经过团队内部的商讨，决定重构多用户，由胡浪负责编写相关 API，本人将其集成入主程序，并实现与全局的整合。

本次换用 sshpass 进行密码验证，并使用 sshpass 连接后启动 code-server 的形式进行权限限制。

2019 年 7 月 23 日：

超时检测机制

因为 code-server 在启动后会占用大量系统资源，所以当用户离开后就应该将其进程全部 kill，以释放系统资源。

初步采用的方法是设定一个超时时间，当用户长时间未操作时就将

其进程全部 kill

2019 年 7 月 24 日：

对胡浪部分 API 进行 golang 的重写

因为 HTTP 服务的特殊性，很多请求无法全部转发给 API 所在的 service，所以将启动 IDE 和监控 IDE 的服务使用 Golang 进行了重写，这样可以更快和更准确的启动和监控 IDE。

采用的原理与胡浪 API 一致，都是基于 sshpass 验证身份。正对超时检测，换用新方法：启动一个新线程监控 code-server，每次 http 请求进入后刷新 activeTime，当现在时间 - activeTime > 预设的 timeout 后，则立即杀死进程。

2019 年 7 月 25 日：

用户中心页面

根据老师的需求，添加一个用户中心，可以让用户自定义修改 Workspace、timeout 等数据。

主要思路就是前端 post 数据到后端，后端更新数据库及内存中的数据，然后重启 code-server



The screenshot shows a web interface for 'WROIS Web IDE-用户中心' (User Center). The main heading is '个人信息' (Personal Information). Below it, a subtitle says '你可以在这里修改个人信息和相关个性化配置' (You can modify personal information and related personalized configurations here). There are three input fields: 1. '用户名(无法修改):' (Username, cannot be modified) with a placeholder '{{username}}'. 2. '默认启动WorkSpace(默认为\$(HOME)/workspace,留空为不修改):' (Default workspace, default is \$(HOME)/workspace, leave empty for no change) with a placeholder '\$(HOME)/workspace'. 3. 'IDE最大超时退出时间(留空恢复默认值300):' (Maximum IDE timeout exit time, leave empty to restore default 300) with a placeholder '超时时间' (Timeout time). Below the fields are two buttons: '进入插件中心' (Enter Plugin Center) and '提交信息' (Submit Information). At the bottom, there are links for 'Github:code-server' and 'Github:InstantVSCode'.

2019 年 7 月 26 日：

插件中心

因为 code-server 中无法安装插件，所以提供一个在线的插件安装页面。

用户可以选择预置的插件进行安装，也可以选择自己上传插件安装。

插件安装的主要原理就是使用管道执行命令，调用 code-server 的 `install-extension` 命令进行安装



2019 年 7 月 29 日：

Remote Control 系统

Remote Control 主要用于帮助用户快速的将本地代码部署至远程计算机中，与 code-server 相结合，用户编写完成代码后可以立即将其移动到远程计算机进行执行并获得相关结果。

目前该系统暂定基于 SCP、NFS 等技术进行远程文件传输，前端基于 Xterm.JS 搭建在线 WebTerminal，数据传输完成后就立即进入交互式命令行，帮助用户快速执行运行、编译等操作。

五、未来工作

未来 7.30-8.2：

完成对 Remote Control 的全部开发及调试工作，并开发相关 VSCode 集成插件。

未来 8.5-8.9：

完成与 File Manager、Data Manager 的全部交互，并制作万能代码运行(Run Code)插件及万能远程代码运行插件(Remote Run Code)。

未来 8.12-8.15：

完成系统整体测试，推出第一个 Release 版本。

六、 总结

本次实习主要是为了帮助开发者快速和简单将自己的程序运行与算力更强的计算机，不仅可以在线编辑代码，也可以执行、编译代码等操作。也可以可视化选择 GIS 相关数据。同时拥有一个可视化文件在线管理器。

通过本次实习，我们了解了很多前后端开发知识。也学习到了很多程序内存优化、资源回收等知识。