OJ在线判题系统问题答疑

> 请利用好 `Ctrl + F` 来搜索需要的问题

1.前端

1.1 项目对应的版本及所需安装的软件

- nodejs 使用 v16 版本，大于 16.14，可以下载 nvm（Node.js 版本管理工具）来切换多个不同的 Node.js 版本。
- JDK 使用 1.8 版本
- Spring Boot 使用 2.x 版本
- 要下载 git(git 安装教程 <https://blog.csdn.net/weixin_47638941/article/details/120632890>)

推荐工具 —— 切换和管理 node 版本的工具 <https://github.com/nvm-sh/nvm> 。

> **注意：** 如果版本不对，一定会在启动、初始化出现问题🐶
> 看过来看过来(疯狂挥手.jpg)，鱼友一定要更改版本或下载对应软件，再进行项目哦(≧∇≦)/

1.2 为什么代码编辑器切换语言失败?

```
watch(
  () => props.language,
  () => {
    if (codeEditor.value) {
      monaco.editor.setModelLanguage(
        toRaw(codeEditor.value).getModel(),
        props.language
      );
    }
  }
);
```

1.3 为什么 openapi 报错，显示无法识别（或 openapi 启动失败、openapi不是内部或外部命令）？

1）安装 openapi-typescript-codegen，使用以下命令安装：

```
npm install openapi-typescript-codegen --save-dev
```

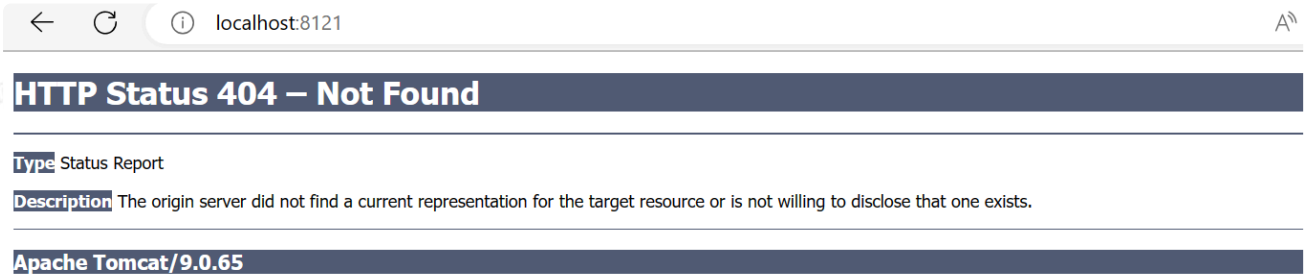
2）全局安装问题，使用以下命令安装：

```
npm install -g openapi-typescript-codegen

或

pnpm add -g openapi-typescript-codegen
```

1.4 为什么项目启动后访问 localhost:8121 显示404?



接口文档的地址是这个 http://localhost:8121/api/doc.html#/home。
<http://localhost:8121/api/doc.html#/home。>

1.5 为什么前端引入全局拦截器没有效果?

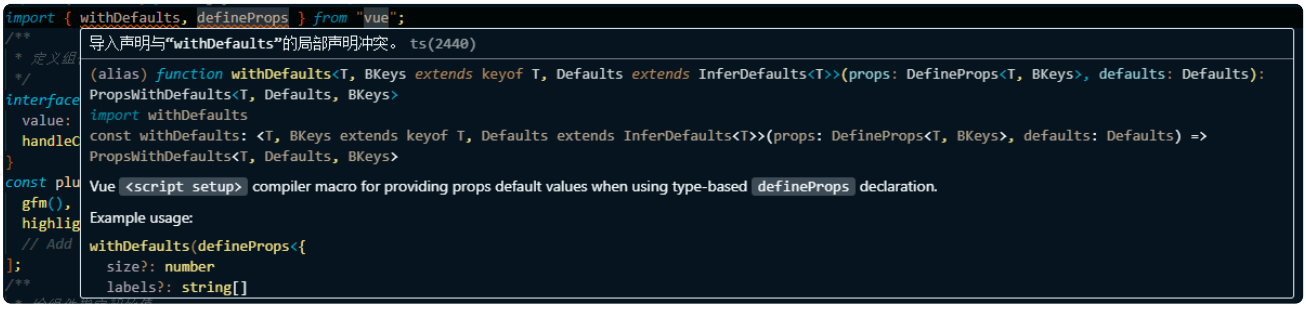
1）没安装成功

2）引入方式不正确

3）配置问题

4）浏览器缓存

1.6 为什么引入 withDefault 和 defineProps 显示：导入声明与“withDefaults”的局部声明冲突?



这是因为 'defineProps' 与编译器宏冲突，Vue 3 的 Script Setup 语法引入了 defineProps、defineEmits、defineExpose、withDefaults 的编译器宏。

1. 检查 eslint-plugin-vue 的版本（是否为 v8.0.0 以上）

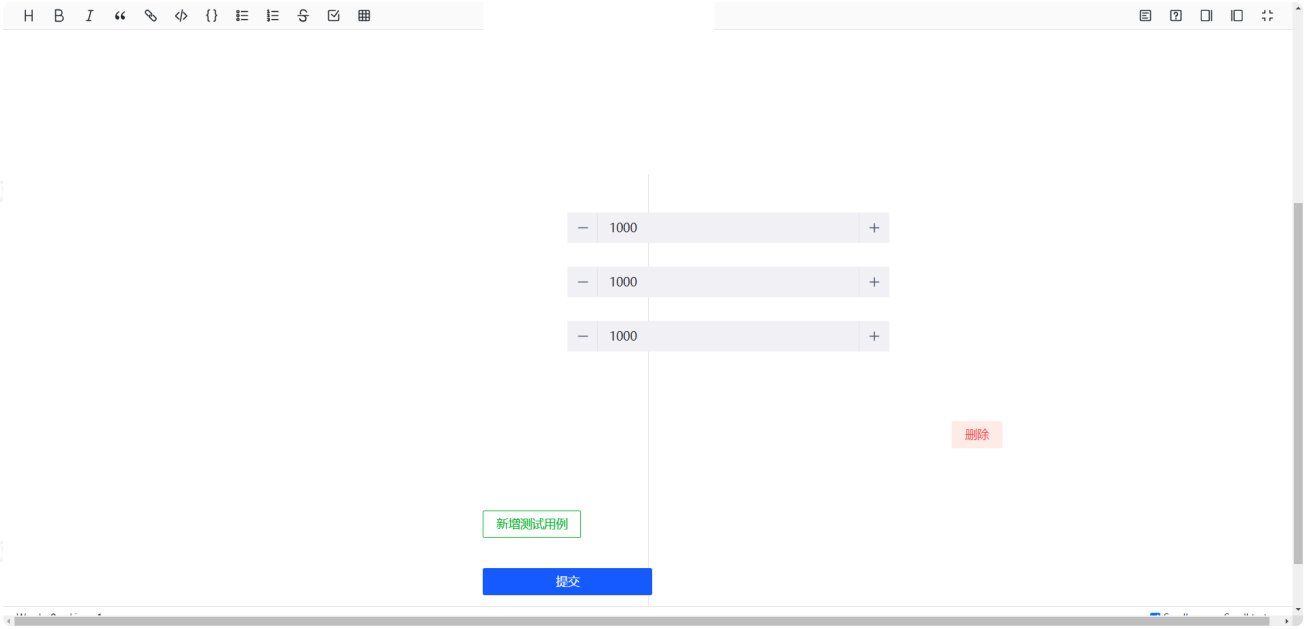
```
"devDependencies": {
  "@arco-design/web-vue": "^2.49.2",
  "@typescript-eslint/eslint-plugin": "^5.4.0",
  "@typescript-eslint/parser": "^5.4.0",
  "@vue/cli-plugin-babel": "~5.0.0",
  "@vue/cli-plugin-eslint": "~5.0.0",
  "@vue/cli-plugin-router": "~5.0.0",
  "@vue/cli-plugin-typescript": "~5.0.0",
  "@vue/cli-plugin-vuex": "~5.0.0",
  "@vue/cli-service": "~5.0.0",
  "@vue/eslint-config-typescript": "^9.1.0",
  "eslint": "^7.32.0",
  "eslint-config-prettier": "^8.3.0",
  "eslint-plugin-prettier": "^4.0.0",
  "eslint-plugin-vue": "^8.0.3",
  "openapi-typescript-codegen": "^0.25.0",
  "prettier": "^2.4.1",
  "typescript": "~4.5.5"
}
}
```

2. 打开 .eslintrc.js 文件：

```
env: {
    node: true,
    // The Follow config only works with eslint-plugin-vue v8.0.0+
    "vue/setup-compiler-macros": true,
  },
```

3. 删除导入直接用

1.7 为什么 md 编辑器全屏后有其他组件的出现?




```
:deep(.bytemd-fullscreen.bytemd) {
  z-index: 100;
}
```

项目-更新

1.8 为什么 const loginUser = store.state.user.loginUser 可以获取 store.state.user.loginUser 的值，但 store.state.user.loginUser 更新之后的值不能拿到?

```
router.beforeEach( guard: async (to : RouteLocationNormalized , from : RouteLocationNormalized , next : NavigationGuardNext ) : Promise<void>  => {
  const user = store.state.user;
  let currentUser = user.loginUser;
  console.log("登陆用户信息", currentUser);
  if (!currentUser || !currentUser.userRole) {
    await store.dispatch( type: "user/getLoginUser");
  }
  currentUser = user.loginUser;
  console.log("登陆用户信息", currentUser);
  const needAccess : string  = (to.meta?.access as string) ?? ACCESS_ENUM.NOT_LOGIN;
  if (needAccess !== ACCESS_ENUM.NOT_LOGIN) {
```

```
登陆用户信息 ▶ Proxy(Object) {userName: '请登录'}
登陆用户信息 ▶ Proxy(Object) {id: '1688197894597107714', userName: 'adam', userAvatar: null, userProfile: null, userRole: 'user', …}
```

1.9 为什么报错：Cannot access 'routes' before initialization ReferenceError: Cannot access 'routes' before initialization?

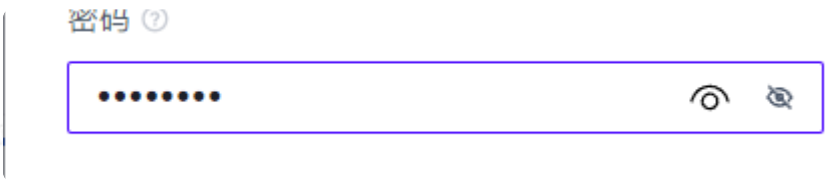
循环引用导致的报错，参考文章：https://zhuanlan.zhihu.com/p/589463077 <https://zhuanlan.zhihu.com/p/589463077>

```
import router from "@/router";

改成：

import { useRouter } from "vue-router";
const router = useRouter();
```

1.10 为什么登录时密码框有两个“眼睛”？



因为 ie 和 edge 浏览器自带的会出现这个问题。

```
<style>
      /*去除ie edge的密码框默认的快速清除钮（X图标）以及密码文字显示钮*/
      input[type="password"]::-ms-reveal{
        display: none;
      }
      input[type="password"]::-ms-clear{
        display: none;
      }
      input[type="password"]::-o-clear{
        display: none;
      }
</style>

```

1.11 为什么 vscode 的 prettier 格式化不生效?

没有识别到 .prettier.config.cjs 文件，新建文件 .prettierrc.js。

1.12 为什么使用 openapi 生成接口代码，调用 getLoginUser 后台自动拼接两个 /api?

因为 openapi 更新了生成规则，多生成了个 api 出来，去 openai.ts 修改全局 api 路径就可以了。

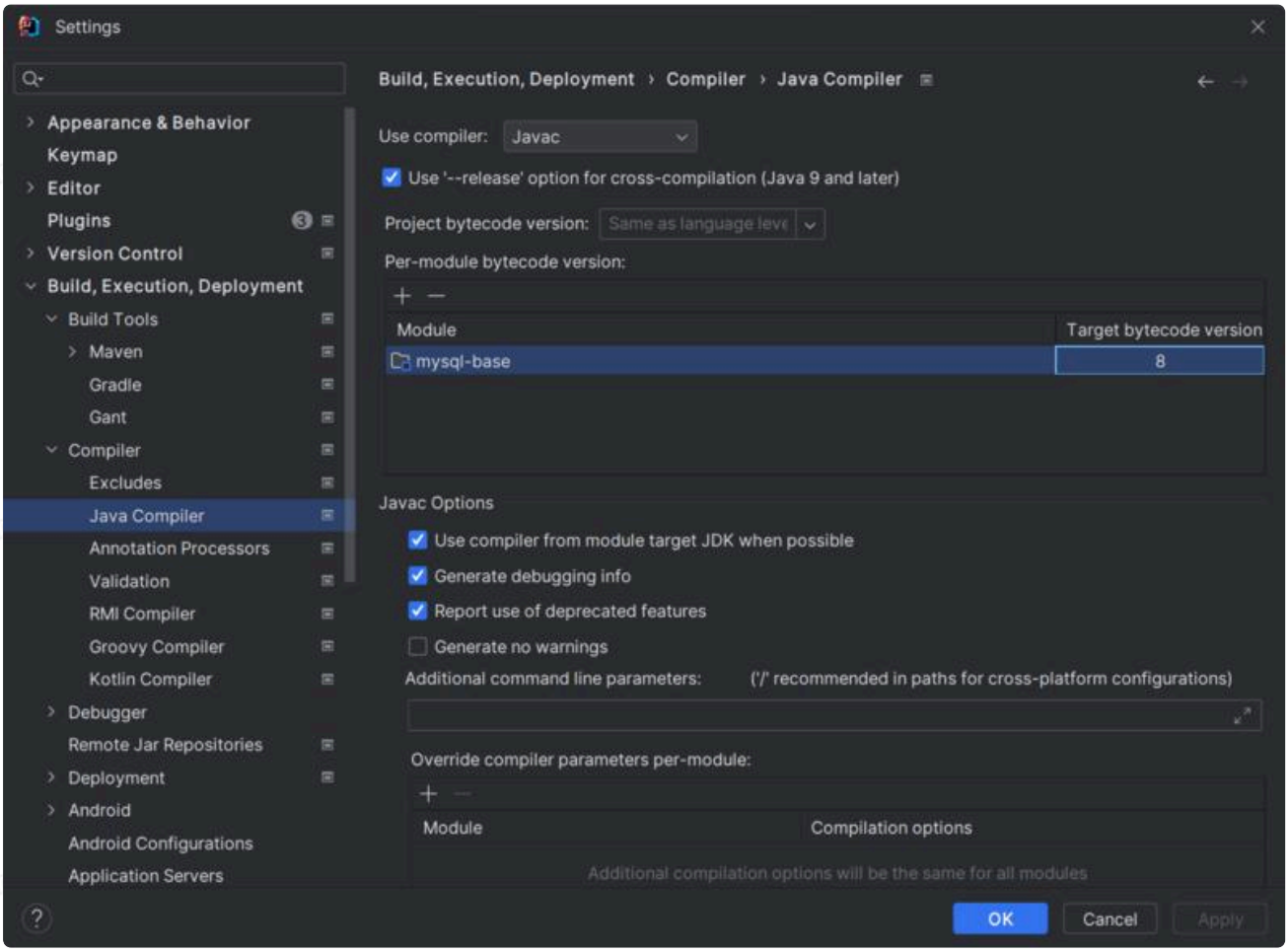
2.后端

2.1 为什么无法下载 knife4j 4.4.0 依赖?

阿里云远程仓库没有这个版本，版本号换成 4.3.0，或者 SpringBoot 版本号太低。

2.2 为什么启动项目报错：java: -source 1.5 中不支持 diamond 运算符（请使用 -source 7 或更高版本以启动 diamond 运算符）？

这里改成 8。



2.3 为什么使用 java 原生对用户操作进行限制，报错：access denied ("java.util.PropertyPermission" "user.dir" "read")?

```
public class DefaultSecurityManager extends SecurityManager{

  // 检查所有权限
  @Override
  public void checkPermission(Permission perm) {
    super.checkPermission(perm);
  }
}
```

这段不是放开所有权限吗？为什么我无法执行System.getProperty("user.dir");了？

在你提供的代码中，你创建了一个名为`DefaultSecurityManager`的自定义`SecurityManager`类，但在你的`checkPermission`方法中，你只是调用了`super.checkPermission(perm)`，这意味着它会委托给父类的`SecurityManager`来检查权限。这并不会完全禁用安全管理器，而是使用父类的默认实现来检查权限。

默认情况下，`SecurityManager`会限制一些敏感操作，例如访问系统属性（如`System.getProperty("user.dir")`）等。如果你想完全禁用`SecurityManager`以允许执行敏感操作，你需要自己实现`checkPermission`方法，将它设置为不执行任何权限检查。例如：

2.4 安装虚拟机，SSH 连接问题，ping 不通虚拟机 ip，只有桥接模式有网络，需要下 SSHserver 吗?

看看 22 号端口，开放了没，然后看看 openssh–server 下载了没，ubuntu 远程连接的时候不能使用 root 用户，默认情况下是不可以 root 的，但是可以修改配置然后设置个 root 密码就可以 root 登陆，我登陆云服务器都是直接 root。

2.5 为什么获取内存信息无法关闭?

想办法把它停掉就好了，比如定义一个变量来判断，执行完后更改变量的值。

2.6 为什么后端初始化后，接口文档没有接口，少了很多部分?

你的接口文档扫描的路径不对，看一下配置文件。

2.7 为什么后端的 JSON 对象还是 Long 类型，到前端就变成了 String 类型?

1．手动在前端把字符串转化为 number

2．把 JudgeConfig 类的类型改为 Integer，目测也用不到 Long（如果以 MS，KB，MB 等为单位的话）

2.8 为什么整合 Monaco-editor 报错 Static class blocks are not enabled（或 Cannot set properties of underfined(setting 'apex')）？

1. 打开Terminal / cmd终端，输入以下命令来安装必要的插件：

```
npm install --save-dev @babel/plugin-transform-class-static-block
```

2. 更新 Babel 配置：在 Babel 配置文件中（通常是.babelrc或babel.config.js），添加@babel/plugin-transform-class-static-block插件到plugins数组中。

```
module.exports = {
  presets: ['@vue/cli-plugin-babel/preset'],
  plugins: ['@babel/plugin-transform-class-static-block']
}
```

3. 重新启动项目，即可解决该问题。

2.9 为什么执行判题服务，打了断点不会进入，得等触发了异步才能调用 judgeservice?

首先，排查是否是 lazy 注解引发的。找了另外的类写了一个测试方法，这个 bean 不开启 lazy，发现可以正常调用。随后将 judge 方法的 lazy注 解注释掉，然后开启忽略循环依赖，测试发现问题依然存在，但变为仅有初次试验能成功调用异步判题服务，后续试验依旧按 F9 会跳过判题。怀疑是异步方法的问题，后根据：

浅谈java开启异步线程的几种方法(@Async,AsyncManager,线程池)_java 异步...
<https://blog.csdn.net/nhx900317/article/details/125478681>

将原异步方法注释，选择使用 springboot 的异步注解框架，@EnableAsync 和在dojudge方法上打上@Async注解，让异步线程sleep5秒，测试发现可以正常触发异步调用。

3.部署&上线

3.1 为什么 nacos 启动一直报错?

JDK 版本要使用 64 位的，和 window 系统一致。

3.2 为什么登录页面 404?

参考了用户中心解决跨域的配置，好吧不管用!

```
location ^~ /api/ {
    proxy_pass http://127.0.0.1:8080/api/;
    add_header 'Access-Control-Allow-Origin' "$http_origin";
    add_header 'Access-Control-Allow-Credentials' "true";
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, DELETE';
    add_header 'Access-Control-Allow-Headers' '*';
    if ($request_method = "OPTIONS") {
        add_header 'Access-Control-Allow-Credentials' "true";
        add_header 'Access-Control-Allow-Origin' "$http_origin";
        add_header 'Access-Control-Max-Age' 1728000;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, DELETE';
        add_header 'Access-Control-Allow-Headers' 'DNT,X-Mx-ReqToken,Keep-
Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-
Type,Authorization';
        add_header 'Content-Length' 0;
        add_header 'Content-Type' 'text/plain, charset=utf-8';
        return 204;
    }
 }
```

项目-更新

然后又找了球友猫十二懿的笔记https://www.yuque.com/kcsshier/zpovmy/mpld6wr5kte7wven?singleDoc#hqgCp配了跨域

```
# 跨域代理
location ^~ /api/{
  proxy_pass http://127.0.0.1:后端项目启动端口;
  proxy_set_header Host 127.0.0.1;
  proxy_set_header X-Real-IP $remote_addr;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_set_header REMOTE-HOST $remote_addr;

  add_header X-Cache $upstream_cache_status;

  # 添加以下两行
  proxy_set_header Origin 域名/IP;
  add_header 'Access-Control-Allow-Origin' '域名/IP';

  #Set Nginx Cache
  set $static_fileo6zfliSa 0;
  if ( $uri ~* "\.(gif|png|jpg|css|js|woff|woff2)$" )
  {
    set $static_fileo6zfliSa 1;
    expires 12h;
     }
  if ( $static_fileo6zfliSa = 0 )
  {
  add_header Cache-Control no-cache;
  }
}
location / {
  try_files $uri $uri/ /index.html;
}
```

页面打开正常了，直接登录，没效果，前端没有登录态，浏览器控制台也正常，可以看到后端也返回的用户数据，明明本地测试一切正常，题目数据也有，我百思不得其解，问别人我也不会问了，没报错啊。

然后我想到前端请求后端地址是localhost，此时我感觉没毛病，因为我觉得前后端都在同一台服务器上，是可以访问到的，关键是我页面是有题目数据的，访问前端页面是http://服务器ip，然后请求后端是http://localhost:8121，存在的跨域问题也解决的。到底为什么呢？当我快要抑郁的时候，发现，他奶奶的，我线上的前端调的是我电脑idea启动的后端的接口获取的数据，我还以为一直是我服务器上启动的后端服务。
这应该就是导致我登录后，前端没有登录态的原因吧？具体原因还没想明白，给为球友知道的话，评论区说一下。
以及为什么配了上面的跨域配置，就能正常打开登录页了呢？不会跳到nginx的404页面了
里面涉及到的原理以及nginx的跨域配置，没搞清楚的话，只能不断是。不断才坑了。
网上搜nginx的跨域配置真的太多花样的，球友们有没有一种通过的解决跨域的配置可以分享吗？

我以试一下的心态，我又配置了，线上前端通过服务器ip器范问后端。

```
const baseUrl =
  process.env.NODE_ENV === "development"
    ? "http://localhost:8121"
    : "http://服务器ip:8121";

OpenAPI.BASE = baseUrl;
console.log("当前环境：", process.env.NODE_ENV, "请求地址", baseUrl);
```

到这里所有问题解决了。 关机碎觉。。。。。。。。。。。。。

3.3 为什么 java 无法连接到 docker 守护进程的 unix?

为什么 java 无法连接到 docker 守护进程的 unix?

```
1   DockerClient dockerClient =DockerClientBuilder.getInstance("unix:///var/run/do
```

报错权限不足。

根据以下步骤：

1.检查 Docker Daemon 的权限：确保 Docker 守护进程的 Unix 套接字文件具有正确的权限。默认情况下，套接字文件的权限应该足够让docker组的成员访问它。

```
1   ls -l /var/run/docker.sock
```

```
1   sudo chmod 666 /var/run/docker.sock
```

然后再次运行 DockerDemo.java 文件。

DockerDemo.java

```java
package yuoj.yuojcodesandbox.docker;

import ...

public class DockerDemo {
    public static void main(String[] args) throws InterruptedException {
        // 获取Docker Client
        DockerClient dockerClient = DockerClientBuilder.getInstance( serverUrl: "unix:///var/run/docker.sock").build();
        String image = "nginx:latest";
        PullImageCmd pullImageCmd = dockerClient.pullImageCmd(image);
        PullImageResultCallback pullImageResultCallback = new PullImageResultCallback(){
            @Override
            public void onNext(PullResponseItem item) {
                System.out.println("下载镜像"+item.getStatus());
                super.onNext(item);
            }
        };
        pullImageCmd.exec(pullImageResultCallback).awaitCompletion();
        System.out.println("下载完成");
    }
}
```

就开始下载并拉取 nginx 镜像了：

```
17:08:50.439 [docker-java-stream--650138966] DEBUG com.github.dockerjava.api.command.PullImageResultCallback - Res
17:08:51.338 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "bd[\r][\n]"
17:08:51.338 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "{"status":"Downloadi
17:08:51.338 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "[\r][\n]"
下载镜像Downloading
17:08:51.339 [docker-java-stream--650138966] DEBUG com.github.dockerjava.api.command.PullImageResultCallback - Res
17:08:51.929 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "bd[\r][\n]"
17:08:51.929 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "{"status":"Downloadi
17:08:51.929 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "[\r][\n]"
下载镜像Downloading
17:08:51.930 [docker-java-stream--650138966] DEBUG com.github.dockerjava.api.command.PullImageResultCallback - Res
17:08:52.679 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "bd[\r][\n]"
17:08:52.679 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "{"status":"Downloadi
17:08:52.679 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "[\r][\n]"
下载镜像Downloading
17:08:52.679 [docker-java-stream--650138966] DEBUG com.github.dockerjava.api.command.PullImageResultCallback - Res
17:08:52.969 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "bd[\r][\n]"
17:08:52.969 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "{"status":"Downloadi
17:08:52.969 [docker-java-stream--650138966] DEBUG org.apache.http.wire - http-outgoing-0 << "[\r][\n]"
```

检查：

```
ubuntu@ubuntu-virtual-machine:~/桌面$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    593aee2afb64   9 hours ago    187MB
hello-world   latest    9c7a54a9a43c   5 months ago   13.3kB
```

注意点：关于Ubuntu版本

我这个电脑用的是ubuntu22.04版本感觉进行的比较顺利，之前使用18.04版本的时候，会特别卡壳，Idea的gateway上一直显示cpu超负荷运转，并且一直没反应。有的时候，Ubuntu的root权限也莫名其妙的获取不到，会导致任何操作都会受到限制，不知道什么原因。

以下是 gpt 的解答：

根据您提供的代码和错误信息，问题似乎是Docker Java库无法连接到Docker守护进程的Unix套接字。这通常是由于权限问题或配置问题引起的。以下是一些可能的解决方案：

1. **检查Docker Daemon的运行状态**：首先确保Docker守护进程正在运行。您可以使用以下命令检查它的状态：

```bash
sudo systemctl status docker
```

如果它没有运行，可以使用以下命令启动它：

```bash
sudo systemctl start docker
```

2. **检查用户权限**：确保您的用户有权限访问Docker套接字。通常，要访问Docker套接字，用户需要属于`docker`组。您可以将用户添加到`docker`组中：

```bash
sudo usermod -aG docker your_username
```

替换`your_username`为您的用户名。然后注销并重新登录，以使更改生效。

3. **Docker套接字路径**：确保Docker套接字的路径在代码中正确设置。通常情况下，Docker套接字路径是`unix:///var/run/docker.sock`。在代码中，您可以这样设置套接字路径：

```java
DockerClient dockerClient = DockerClientBuilder.getInstance("unix:///var/ru
```

```
DockerClient dockerClient =
DockerClientBuilder.getInstance("unix:///var/run/docker.sock").build();
```

4. **检查Docker Daemon的配置**：如果您自定义了Docker守护进程的配置，请确保在Docker配置中正确配置了Unix套接字的监听。查看Docker配置文件`/etc/docker/daemon.json`，确保套接字的配置与代码中一致。
5. **重新编译代码并运行**：在应用程序中应用上述更改后，重新编译代码并再次运行。

如果仍然遇到问题，可以尝试在代码中捕获异常并记录更多信息，以便更好地诊断问题。可能需要查看Docker守护进程的详细日志以获取更多信息，以及检查网络和防火墙规则是否会影响Docker连接。

项目-更新

权限问题：

您的应用程序仍然遇到权限问题，无法连接到Docker守护进程的Unix套接字。请按以下步骤进行诊断和解决问题：

1. **检查Docker Daemon的权限**：确保Docker守护进程的Unix套接字文件具有正确的权限。默认情况下，套接字文件的权限应该足够让`docker`组的成员访问它。您可以使用以下命令检查权限：

```bash
ls -l /var/run/docker.sock
```

如果套接字文件的权限不正确，您可以使用以下命令更改它：

```bash
sudo chmod 666 /var/run/docker.sock
```

请注意，更改套接字文件的权限可能会导致安全问题，因此请根据您的环境谨慎操作。

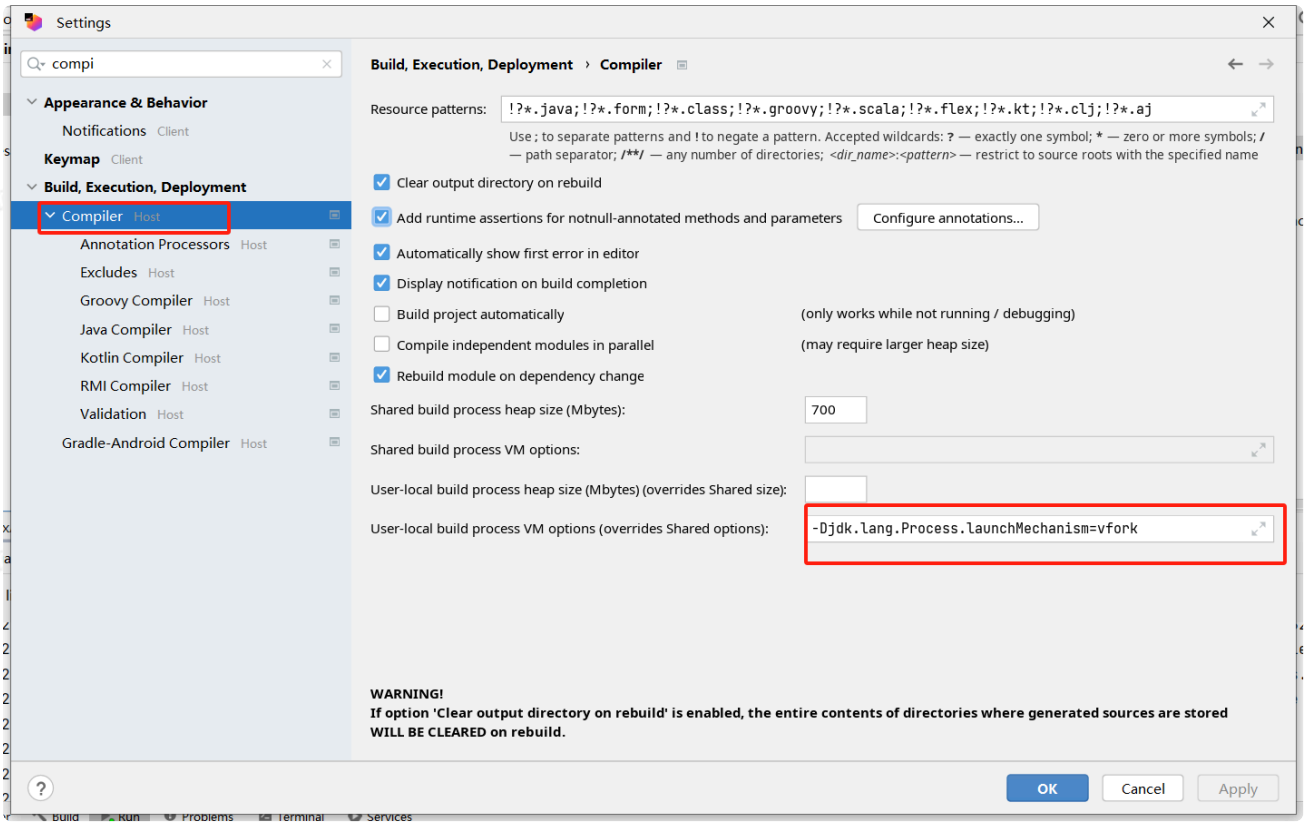
2. **检查SELinux或AppArmor**：如果您的系统使用了SELinux或AppArmor等安全模块，它们可能会限制应用程序访问Docker套接字。您需要根据您的系统的安全策略配置允许应用程序访问Docker。
3. **使用`sudo`**：在开发和测试阶段，您可以尝试使用`sudo`来运行您的Java应用程序。这将赋予您的应用程序更高的权限，以便访问Docker套接字。不过在生产环境中，通常不建议以`sudo`权限运行应用程序，因为这可能会引入安全风险。
4. **检查其他安全设置**：检查您的系统上的其他安全设置，例如防火墙规则和容器安全策略，以确保它们不会阻止应用程序与Docker通信。

3.4 为什么 nacos 服务自动关闭?

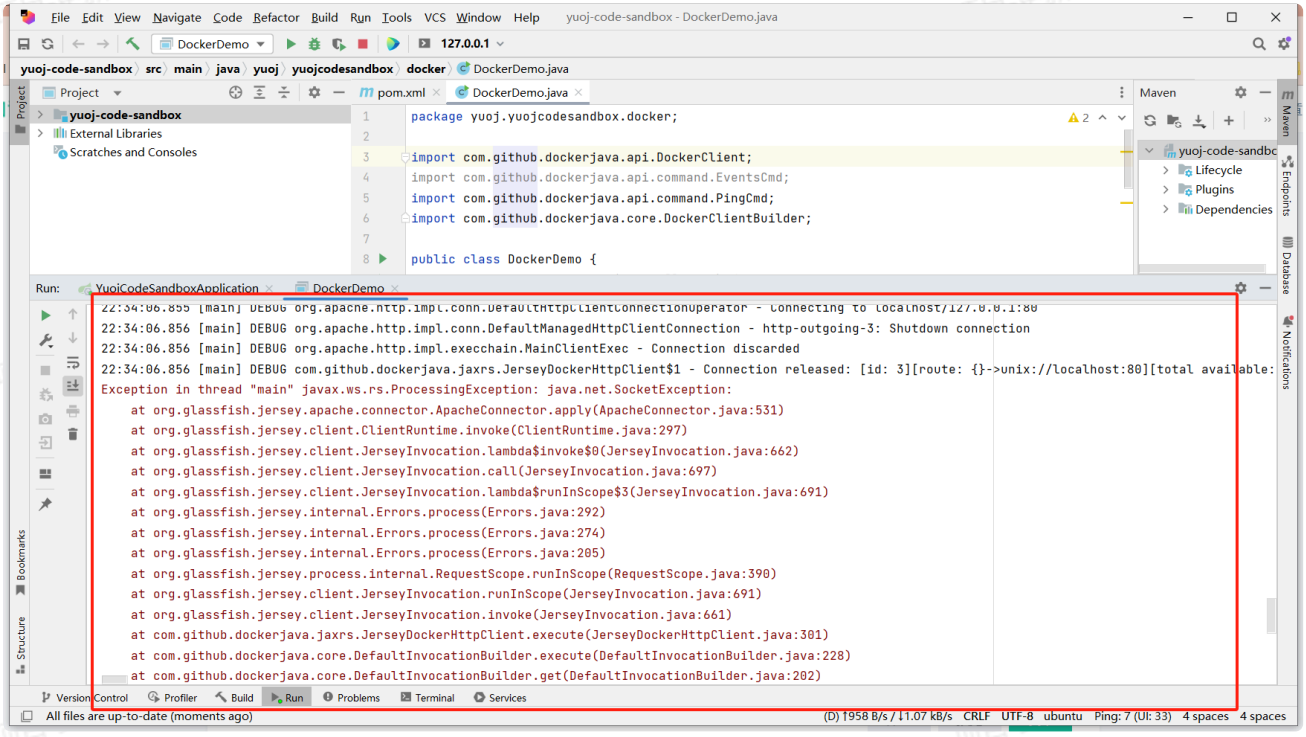
java 堆内存太大，调整队的参数即可。

3.5 为什么使用 idea 的remote development 进行远程开发，运行 linux 虚拟机的程序报错：java: Cannot run program "/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java" (indirectory "/home/ubuntu/.cache/JetBrains/RemoteDev-IU/_usr_yuoj-code-sandbox/compile-server"): error=0, Failed to exec spawn helper: pid: 8347, exitvalue: 1?

点击左上角选项： settings -> Compiler ->找到 User-local build process VM options(overrides Sharedoptions)



输入命令 `-Djdk.lang.Process.launchMechanism=vfork` 即可。



3.6 为什么远程文件同步失败：Failed to create folder'/usr/yuoj-code-sandbox'. Could not create folder "sftp://192.168.129.128/usr/yuoj-code-sandbox". (Permissiondenied)?

文件操作权限不足，以管理员身份指定同步的目标文件权限为 777 即可。

项目-更新

3.7 配置 SSH，和鱼总的 idea 版本不一样，为什么找不到 remote host?

主要表现就是他不让你把本机的文件同步到 linux 虚拟机里。

原因是文件夹的操作权限不足，只需要在以管理员的身份指定同步的目标文件夹的权限为 777 即可：

```
sudo su chmod-R 777/usr/yuoj-code-sandbox（你的文件夹目录）
```

3.8 为什么设置了允许携带 cookie，看网页还是没有呢?

如果是 localhost 就两边都用 localhost，浏览器也用 localhost 访问，如果是 ip 就都用 ip。

3.9 为什么用 Nginx 部署项目后，使用 ip 能正常访问，用域名就访问不了?

1．将 openAPI.ts 里的地址改为 http://域名。

package.json | OpenAPI.ts

```
export const OpenAPI: OpenAPIConfig = {
    BASE: 'http://zsp2023.top',
    VERSION: '1.0',
    WITH_CREDENTIALS: true,
    CREDENTIALS: 'include',
    TOKEN: undefined,
    USERNAME: undefined,
    PASSWORD: undefined,
    HEADERS: undefined,
    ENCODE_PATH: undefined,
```

2．在 ngin x配置 location，让 nginx 代理所有 ip:80/api 的请求，然后转发到ip:8101

```
server {
    listen       80;
    server_name  [IP];
    # 指定前端项目所在的位置
    location / {
        root   /usr/share/nginx/html/dist/;
        index  index.html;
        try_files $uri $uri/ /index.html;
    }
    location /api/ {
        proxy_pass http://[IP]:8101;
    }
}
```

这样一来，前端调用后端时，就会先访问 http://zsp2023.top（openAPI.ts <http://zsp2023.top（openAPI.ts> 里设置的地址），由于访问的是 80 端口且请求带有 /api，就会由 nginx 反向代

理到 8101 端口。

3.10 为什么拉取源码部署，登录后显示未登录?

把前端 generated/core/OpenAPI.ts下OpenAPIConfig 里的 WITH_CREDENTIALS: true, 改为 true。

3.11 为什么部署之后页面排版异常、乱套了?

问题原因：nginx代理没有配置默认的请求数据流。

nginx配置中加上:

include mime.types;

default_type application/octet-stream;

3.12 为什么 jetbrains gateway 明明之前还打得开，后面就一直闪退?

先升级到 idea2023.3.6 的版本，然后魔法上网。

4.提问

4.1 项目会不会很吃配置?

2 核 4 G 即可，如果虚拟机实在带不起来，那就花不到 100 到腾讯 / 阿里 / 华为 / 百度云买台学生用服务器吧，其他操作都是一样的。

问题来源帖参考：

4.2 做题通过率要怎么写?

通过率只需要用总通过数 / 总提交数即可。

4.3 项目的开发环境有哪些呢?

文档：https://www.code-nav.cn/post/1797431968853835777 <https://www.code-nav.cn/post/1797431968853835777>

4.4 是学完微服务再去做这个项目练手？还是通过这个去了解微服务呢?

可以先学习一边微服务，然后实操该项目来强化学习。

4.6 这个项目可以只学后端（或前端）部分吗?

可以的。

4.7 前端为什么会自动登录？自动登录的账号密码从哪里来？做题做到一半 session 过期了，还怎么做题?

自动登录想做就做，没啥原因，因为其他网站也有自动登录，喜欢就做；浏览器的 cookie 和 session 能够记住账号和密码；怕 session 过期就加长 session 的到期时间，跟自动登录没有必然的关系。

4.8 前端怎么实现根据选择的不同语言初始化不同的的代码?

前端 / 后端 / 数据库里存储不同语言对应的初始化代码内容，然后根据用户的选择加载即可。

4.9 登录之后，为什么浏览器重启，会话 cookie 还在?

当用户在浏览器中登录后，后端会将登录状态信息存储在会话中，并将会话标识存储在浏览器的 cookie 中。当用户关闭浏览器并重新打开时，浏览器会发送之前保存的 cookie 到后端，从而恢复用户的会话状态，实现自动登录的效果。

实际上，浏览器会将 cookie 存储在本地，并在每次请求时自动附加到请求头中。这样，后端就可以根据 cookie 的值来识别用户的身份。通过在浏览器中设置 cookie 的过期时间，可以控制登录状态的有效期。

另外，为了确保安全性，通常会使用一些额外的技术，例如使用 HTTPS 进行加密传输，将 cookie 标记为 HTTP Only，以防止 JavaScript 访问等。这些措施可以提高用户数据的安全性。

总之，通过将用户登录信息存储在 session 中，并将会话标识存储在浏览器的 cookie 中，可以实现浏览器重启时的自动登录功能。这是一种常见的用户身份认证和会话管理的方式。

4.10 用 2核4G 的服务器部署项目，还是内存不够，怎么调参数?

通过调整JVM参数来控制内存使用。可以尝试增加堆内存（-Xmx）和初始堆内存（-Xms）的设置值，确保服务器可以分配足够的内存给你的应用程序。

```
java -Xms512m -Xmx1024m -jar your-application.jar
```

这将设置初始堆大小为 512MB，最大堆大小为 1024MB。根据服务器实际情况决定。

4.11 判题系统中每道题目的资源限制是不同的，是每道题目都对应生成一个容器吗?

最好互不影响，每个容器只执行一个题目，用容器池机制复用容器，资源限制是最大值。接着往下看教程，讲过怎么判断题目是否超出资源。

url=https%3A%2F%2Fwww.yuque.com%2Fu37765561%2Fak85bt%2F332f92018f4ba872058aa7fb2