链滴

解决 solo 同步 GitHub 失败 bug

作者：jditlee

原文链接：https://ld246.com/article/1679483783227

来源网站：链滴

关于solo同步GitHub失败的问题：

[ERROR]-[2023-03-18 10:51:14]-[org.b3log.solo.util.GitHubs:87]: Get git tree of file [README.d] failed: {"message":"Not Found","documentation_url":"https://docs.github.com/rest/reference/git#get-a-tree"}

作为一个有那么一丢丢完美主义的人，是不能忍受这么好的功能无法使用的，然后就开始解决问题了：

第一步，当然是ask社区了，搜了一下，相关的应该就两个帖子：

一个提问贴：Solo github 备份失败 - 链滴 (ld246.com)

一个是留言贴：solo-blog 仓库同步功能回来了！ - 链滴 (ld246.com)

但是呢，都是提出了问题，没有解决问题，我也留言问了一下提问的大哥，看有没有解决，然而

D大可能精力都在思源上了，也没有来看这个bug

没办法了，只能抄起三年前的Java知识来读源码了，下载代码，导入idea，启动Server，失败！！！气呵成

然后就是各种百度，发现是环境原因，现在Java版本都用到11了嘛？？？

解决了环境问题，然后启动，debug，找到备份的代码，应该是export，嗯对了，打个断点：

找到ExportService.exportGitHub,读

```
public void exportGitHub() {
        try {
            if (Latkes.RuntimeMode.DEVELOPMENT == Latkes.getRuntimeMode()) {
                return;
            }

            final JSONObject preference = optionQueryService.getPreference();
            if (null == preference) {
                return;
            }

            String pat = preference.optString(Option.ID_C_GITHUB_PAT);
            if (StringUtils.isBlank(pat)) {
                return;
```

原文链接：解决 solo 同步 GitHub 失败 bug

```
        }

        LOGGER.log(Level.INFO, "Backup public articles to your GitHub repo [solo-blog]....");

        final JSONObject mds = exportHexoMDs();
        JdbcRepository.dispose();
        final List<JSONObject> posts = (List<JSONObject>) mds.opt("posts");

        final String tmpDir = System.getProperty("java.io.tmpdir");
        final String date = DateFormatUtils.format(new Date(), "yyyyMMddHHmmss");
        String localFilePath = tmpDir + File.separator + "solo-blog-repo-" + date;
        final File localFile = new File(localFilePath);

        final File postDir = new File(localFilePath + File.separator + "posts");
        exportHexoMd(posts, postDir.getPath());

        final File zipFile = ZipUtil.zip(localFile);
        byte[] zipData;
        try (final FileInputStream inputStream = new FileInputStream(zipFile)) {
            zipData = IOUtils.toByteArray(inputStream);
        }

        FileUtils.deleteQuietly(localFile);
        FileUtils.deleteQuietly(zipFile);

        final String clientTitle = preference.optString(Option.ID_C_BLOG_TITLE);
        final String clientSubtitle = preference.optString(Option.ID_C_BLOG_SUBTITLE);

        final JSONObject gitHubUser = GitHubs.getGitHubUser(pat);
        if (null == gitHubUser) {
            return;
        }

        final String loginName = gitHubUser.optString("login");
        final String repoName = "solo-blog";

        boolean ok = GitHubs.createOrUpdateGitHubRepo(pat, loginName, repoName, ":writi
g_hand: " + clientTitle + " - " + clientSubtitle, Latkes.getServePath());
        if (!ok) {
            return;
        }

        final String readme = genSoloBlogReadme(clientTitle, clientSubtitle, preference.optStr
ng(Option.ID_C_FAVICON_URL), loginName + "/" + repoName);
        JdbcRepository.dispose();
        ok = GitHubs.updateFile(pat, loginName, repoName, "README.md", readme.getBytes
StandardCharsets.UTF_8));
        if (ok) {
            ok = GitHubs.updateFile(pat, loginName, repoName, "backup.zip", zipData);
        }
        if (ok) {
            LOGGER.log(Level.INFO, "Exported public articles to your repo [solo-blog]");
        }
    } catch (final Exception e) {
```

```
            LOGGER.log(Level.ERROR, "Exports public articles to your repo failed: " + e.getMessag
());
        }
    }
```

报错是在GitHubs.updateFile里面，继续读：

```
public static boolean updateFile(final String pat, final String loginName, final String repoName
 final String filePath, final byte[] content) {
        final String fullRepoName = loginName + "/" + repoName;
        try {
            HttpResponse response = HttpRequest.get("https://api.github.com/repos/" + fullRep
Name + "/git/trees/master").header("Authorization", "token " + pat).
                    connectionTimeout(7000).timeout(60000).header("User-Agent", Solos.USER_AGE
T).send();
            int statusCode = response.statusCode();
            response.charset("UTF-8");
            String responseBody = response.bodyText();
            if (200 != statusCode && 409 != statusCode) {
                LOGGER.log(Level.ERROR, "Get git tree of file [" + filePath + "] failed: " + responseB
dy);
                return false;
            }

            final JSONObject body = new JSONObject().
                    put("message", ":memo: 更新博客").
                    put("content", Base64.getEncoder().encodeToString(content));
            if (200 == statusCode) {
                final JSONObject responseData = new JSONObject(responseBody);
                final JSONArray tree = responseData.optJSONArray("tree");
                for (int i = 0; i < tree.length(); i++) {
                    final JSONObject file = tree.optJSONObject(i);
                    if (StringUtils.equals(filePath, file.optString("path"))) {
                        body.put("sha", file.optString("sha"));
                        break;
                    }
                }
            }

            response = HttpRequest.put("https://api.github.com/repos/" + fullRepoName + "/con
ents/" + filePath).header("Authorization", "token " + pat).
                    connectionTimeout(7000).timeout(60000 * 2).header("User-Agent", Solos.USER_
GENT).bodyText(body.toString()).send();
            statusCode = response.statusCode();
            response.charset("UTF-8");
            responseBody = response.bodyText();
            if (200 != statusCode && 201 != statusCode) {
                LOGGER.log(Level.ERROR, "Updates repo [" + repoName + "] file [" + filePath + "] fa
led: " + responseBody);
                return false;
            }
            return true;
        } catch (final Exception e) {
            LOGGER.log(Level.ERROR, "Updates repo [" + repoName + "] file [" + filePath + "] fail
```

```
d: " + e.getMessage());
            return false;
        }
    }
```

嗯对，就是这报错了，断点打到87行，然后卧槽，日志报错里面不是已经写到这一行了嘛。。。略微点不熟练了。。。好吧，继续看是什么错：

状态statusCode返回不正常，就会打印一条error日志，也就是我们看到的那一条，看一下返回的response:

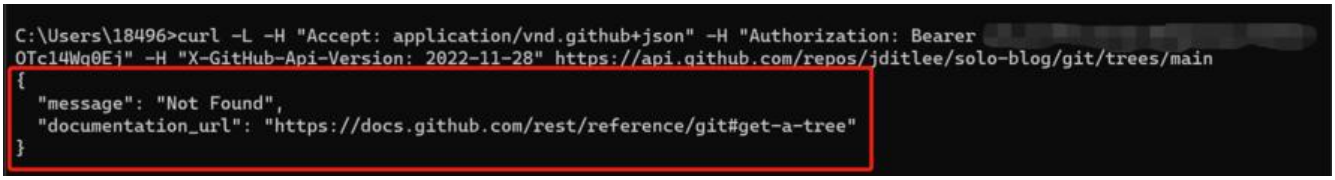
嗯，404，问题已经很明朗了，可能是github的api地址变了，访问body中的地址：

{"message":"Not Found","documentation_url":"https://docs.github.com/rest/reference/git#ge-a-tree"}

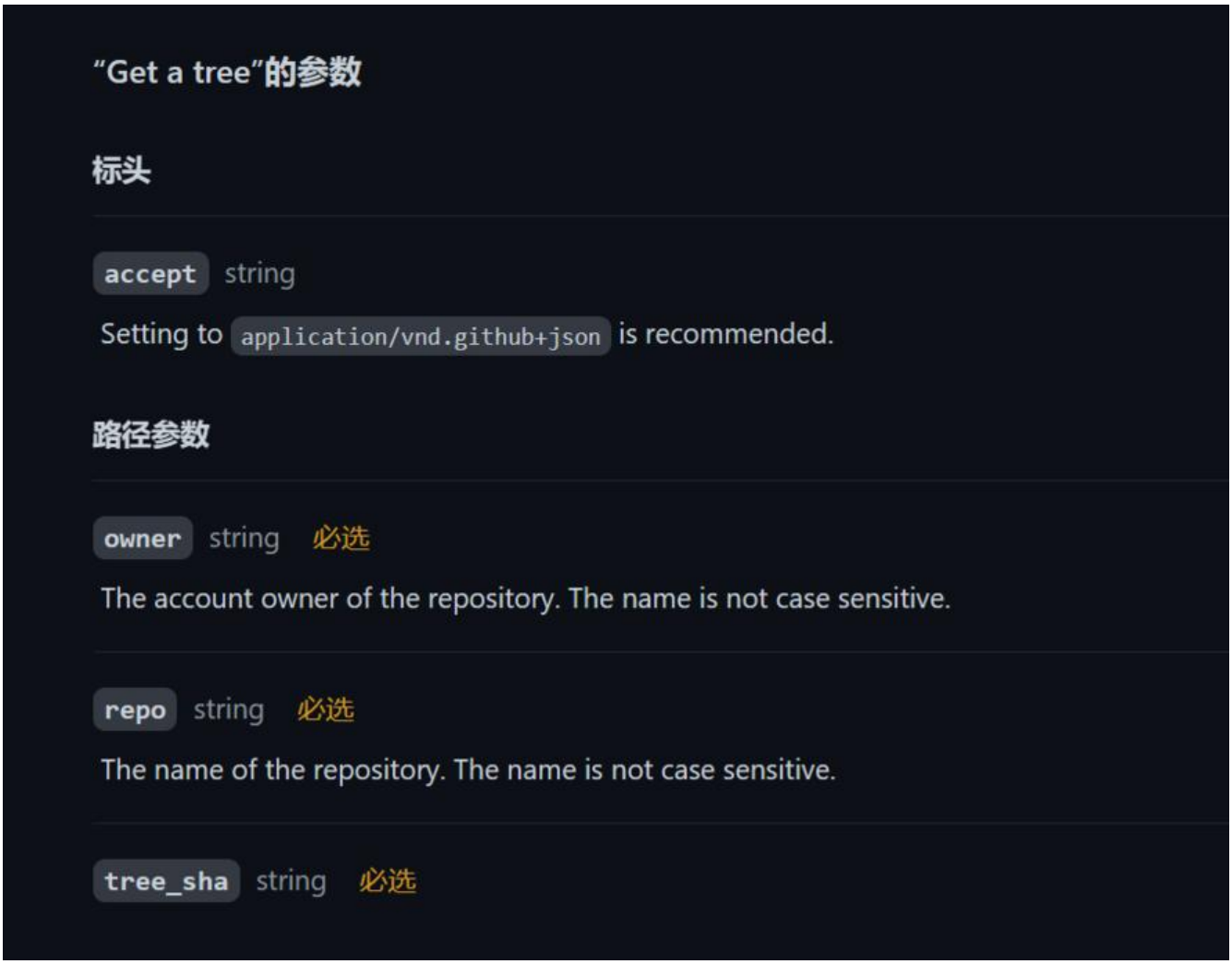
api好像没问题，自己组装curl访问一下

curl -L -H "Accept: application/vnd.github+json" -H "Authorization: Bearer 你的token" -H "X-GtHub-Api-Version: 2022-11-28" https://api.github.com/repos/jditlee/solo-blog/git/trees/master

还是一样的错误



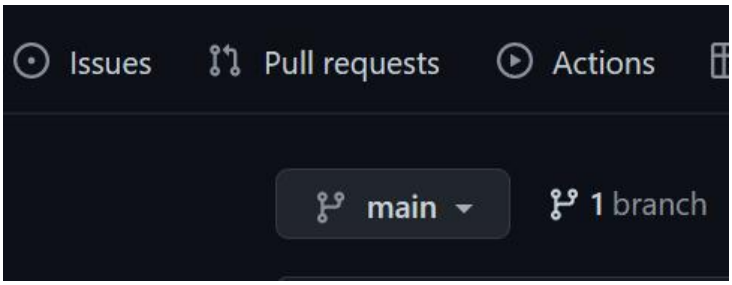
那看一下参数：



owner：仓库所有者用户名

repo：仓库名

tree_sha：文档没说，但应该是分支，前面两个都没问题，那问题就出在分支上了，看一下代码拼接master分支应该没问题呀，为了严谨，再看一下GitHub仓库分支：



？？？怎么会是main

从 2020 年 10 月 1 日开始，GitHub 上的所有新库都将用中性词「main」命名，取代原来的「maste」，因为后者是一个容易让人联想到奴隶制的术语。

就这，好吧，问题找到了，也就知道怎么解决了：

方法一：修改分支名称为master

方法二：修改代码里面的分支为main

当然是方法一好搞一点了，改代码还要重新部署

然后呢，我为什么要写这么多废话，因为我解决这个bug浪费了很多时间，得多写一点记录一下，哈哈

告辞