

说明

在项目组使用gitlab的同时，除了通过gitlab提供的管理功能进入系统操作外，还可以通过一些工具包和gitlab自身提供的钩子进行针对开发管理工作需要的定制化开发，本文整理了一些配置方法和工具包的使用，并且给出了每种使用的可应用场景，供项目组参考。

使用java-gitlab-api

基础设置

依赖引入

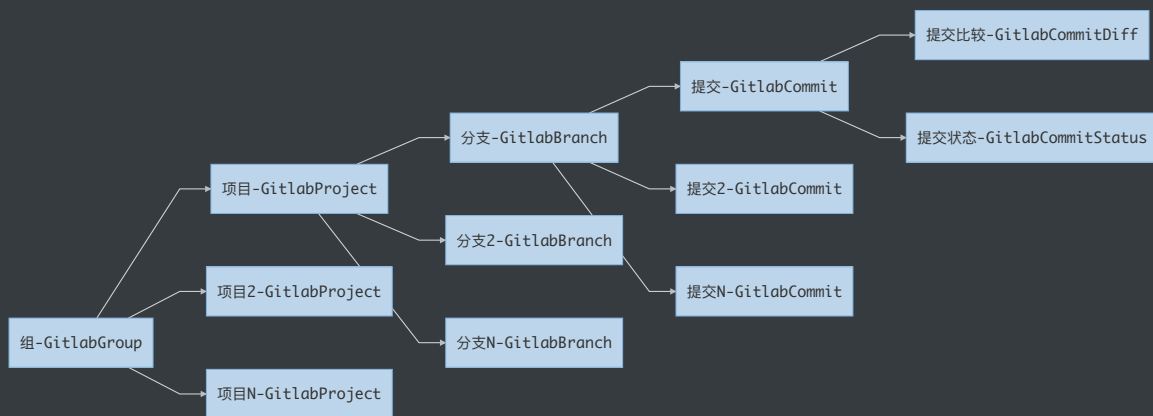
```
<dependency>
  <groupId>org.gitlab</groupId>
  <artifactId>java-gitlab-api</artifactId>
  <version>4.1.1</version>
</dependency>
```

连接GITLAB

```
public static GitlabAPI getGitlabApi(){
    if(gitlabAPI == null){
        gitlabAPI = GitlabAPI.connect(gitUrl,apiToken);
    }
    return gitlabAPI;
}
```

数据关系示意

- 项目关系



■ 用户关系



API使用

用户信息获取

用户信息获取使用java-gitlab-api 的getUsers()方法，返回List对象

```
List<GitlabUser> gitlabUserList = GitLabUtils.getGitlabApi().getUsers();
```

成员信息获取

成员分为组成员和项目成员

■ 组成员

组成员信息获取使用java-gitlab-api 的getGroupMembers(GitlabGroup group)方法，返回List对象

```
List<GitlabGroupMember> gitlabGroupMemberList =  
GitLabUtils.getGitlabApi().getGroupMembers(gitlabGroup);
```

- 项目成员

项目成员信息获取使用java-gitlab-api 的getProjectMembers(GitlabProject project)方法，返回List对象

```
List<GitlabProjectMember> gitlabProjectMemberList =  
GitLabUtils.getGitlabApi().getProjectMembers(gitlabProject);
```

组信息获取

组信息获取使用java-gitlab-api 的getGroups()方法，返回List对象

```
List<GitlabGroup> gitlabGroupList =  
GitLabUtils.getGitlabApi().getGroups();
```

项目信息获取

项目信息获取使用java-gitlab-api 的getProjects()方法，返回List对象

```
List<GitlabProject> gitlabProjectList =  
GitLabUtils.getGitlabApi().getProjects();
```

分支获取

分支信息获取使用java-gitlab-api 的getBranches(GitlabProject project)方法，返回List对象

```
List<GitlabProject> gitlabProjectList =  
GitLabUtils.getGitlabApi().getAllProjects();  
for(GitlabProject gitlabProject:gitlabProjectList){  
    List<GitlabBranch> gitlabBranchList =  
GitLabUtils.getGitlabApi().getBranches(gitlabProject);  
  
}
```

提交信息获取

提交信息获取可以使用全量获取和条件获取两种方式

- 全量获取

全量获取使用java-gitlab-api 的 `getAllCommits(Serializable projectId, String branchOrTag)`方法，返回List对象

```
List<GitlabCommit> gitlabCommitList =  
GitLabUtils.getGitlabApi().getAllCommits(gitlabProject.getId(),  
gitlabBranch.getName());
```

- 条件获取（按照提交时间）

条件获取可以自己传入条件，再调用对应的ap（支持的条件请查询gitlab的api列表），本例只介绍按照提交时间查询的方法，处理部分的代码直接取自java-gitlab-api

```
private List<GitlabCommit> getCommitByCommiteDate(Serializable projectId,  
String branchOrTag,String calDate) throws IOException {  
    Query query = new Query();  
    if (branchOrTag != null) {  
        query.append("ref_name", branchOrTag);  
    }  
    if(calDate!=null){  
        DateTime dateTime = new DateTime(calDate);  
        query.append("since",dateTime.toString());  
    }  
    String tailUrl = GitlabProject.URL + "/" +  
sanitizeId(projectId,"projectId") +  
        "/repository" + GitlabCommit.URL + query;  
    GitlabCommit[] commits =  
GitLabUtils.getGitlabApi().retrieve().to(tailUrl, GitlabCommit[].class);  
    return Arrays.asList(commits);  
}
```

```
//调用代码  
List<GitlabCommit> gitlabCommitList =  
getCommitByCommiteDate(gitlabProject.getId(),  
gitlabBranch.getName(),calculateDate);
```

应用示例

用于统计开发人员代码提交量

<div><div><</div><div>当日工时</div></div>		<div><div>🔄</div><div>🔍</div><div>☰</div></div>	
工时汇总		工时明细	提交汇总
提交明细			
日期		2020-04-29	
总任务数		1	
总打卡数		1	
总耗时(时)		8小时	
任务名(JGIT 应用测试)		▼	
日期		2020-04-29	
累计提交次数		2	
累计新增文件数		1	
累计修改文件数		8	
累计移除文件数		0	
提交内容(microservice-id-generator: no message)		^	
备注		no message	
新增文件数		1	
修改文件数		2	
移除文件数			
提交内容(microservice-ewechat-chart: 项目初始化)		▼	

使用jgit

基础设置

依赖引入

```
<dependency>
  <groupId>org.eclipse.jgit</groupId>
  <artifactId>org.eclipse.jgit</artifactId>
  <version>5.6.1.202002131546-r</version>
</dependency>
<dependency>
  <groupId>org.eclipse.jgit</groupId>
  <artifactId>org.eclipse.jgit.http.server</artifactId>
  <version>5.6.1.202002131546-r</version>
</dependency>
```

连接GITLAB

```
CredentialsProvider credentialsProvider = new
UsernamePasswordCredentialsProvider(userName, password);

Git git = Git.cloneRepository().setURI(remoteUrl)
                              .setCredentialsProvider(credentialsProvider)

                              .setBranch(gitlabProjectTable.getDefaultBranch())
                              .setDirectory(file)
                              .call();
```

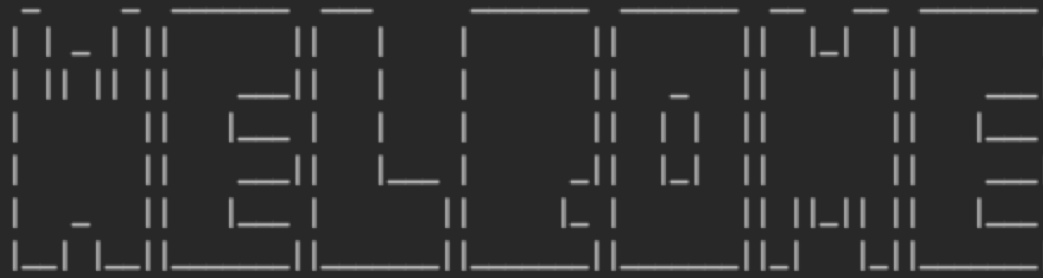
API使用

clone项目

```
git.pull().setCredentialsProvider(credentialsProvider)
          .setRemoteBranchName(gitlabProjectTable.getDefaultBranch())
          .call();
```

应用示例

对于gitlab里面组里面子项目过多的，可以实现组clone项目，一次性将组里面的子项目clone到本地，并且可以结合spring-shell 实现命令行操作



```
git shell:>help
```

```
AVAILABLE COMMANDS
```

Built-In Commands

```
clear: Clear the shell screen.
```

```
exit, quit: Exit the shell.
```

```
help: Display help about available commands.
```

```
script: Read and execute commands from a file.
```

```
stacktrace: Display the full stacktrace of the last error.
```

Git Command

```
groupclone: 组clone
```

```
projectinfoclean: 用户既往信息清除
```

```
pullbystatus: 批量拉取
```

```
git shell:>
```

使用gitlab webhook

基础设置

系统钩子配置

登录gitlab控制台->管理区域->系统钩子，点击”增加系统钩子“按钮，配置钩子的地址、安全令牌和触发器

消息接受和解析

报文分析

分析gitlab推送的报文内容

以下为转换报文示例

Request headers:

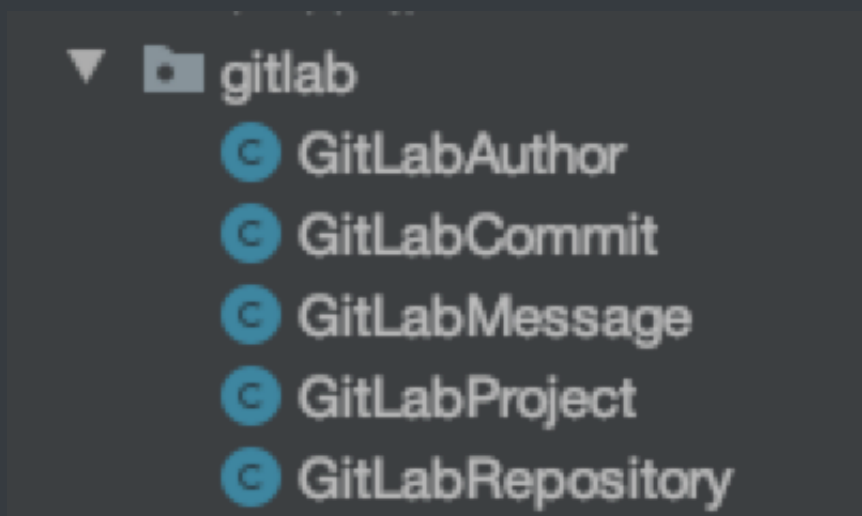
```
Content-Type: application/json
X-Gitlab-Event: System Hook
X-Gitlab-Token: [REDACTED]
```

Request body:

```
{
  "event_name": "repository_update",
  "user_id": 11,
  "user_name": "[REDACTED]",
  "user_email": "[REDACTED]",
  "user_avatar": "http://www.gravatar.com/avatar/b3d87534ee90eae7af5f47d401d0970c?s=80&d=identicon",
  "project_id": 99,
  "project": {
    "name": "[REDACTED]",
    "description": "",
    "web_url": "[REDACTED]",
    "avatar_url": null,
    "git_ssh_url": "git@[REDACTED]:[REDACTED]",
    "git_http_url": "http://[REDACTED]:[REDACTED]",
    "namespace": "microservice-product",
    "visibility_level": 0,
    "path_with_namespace": "[REDACTED]",
    "default_branch": "master",
    "ci_config_path": null,
    "homepage": "[REDACTED]",
    "url": "[REDACTED]",
    "ssh_url": "[REDACTED]",
    "http_url": "[REDACTED]"
  },
  "changes": [
    {
      "before": "28789df60790249bba1f327ef8570f092dc4db77",
      "after": "d64fce04b3a3c39ffc8ff9ff5c02ea1c52085b4b",
      "ref": "refs/heads/develop"
    }
  ],
  "refs": [
    "refs/heads/develop"
  ]
}
```

建立转换对象

根据分析出来的报文，建立转换对象，方便操作



转换逻辑开发

开发自己的代码转换逻辑

应用示例

可以结合企业微信，推送开发人员提交代码的信息

