**GIT SEARCH**

**项目架构详细设计文档**



南京大学软件学院

吉特Fan 小组

目录

[GITSEARCH 1](#_Toc445503367)

[项目架构详细设计文档 1](#_Toc445503368)

[文档更新记录 3](#_Toc445503369)

[引言 3](#_Toc445503370)

[词汇表 4](#_Toc445503371)

[1. 产品概述 4](#_Toc445503372)

[2. 系统的分层架构 5](#_Toc445503373)

[3. 用户界面层的分解 7](#_Toc445503374)

[4. 系统的架构设计 8](#_Toc445503375)

[5. 系统的类图设计 9](#_Toc445503376)

[6. 系统顺序图 10](#_Toc445503377)

[7. 数据层的分解 11](#_Toc445503378)

[8. 接口规范设计 13](#_Toc445503379)

**文档更新记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 描述 | 日期 |
| V1.0 | 万兴 | 初稿 | 2015－03-15 |
| V1.1 | 万兴 | 终稿 | 2015-04-12 |

**引言**

|  |  |
| --- | --- |
| 内容和说明 | 项目 |
| 1.编写的目的（说明整个文档所有达到的目标） | 本文档提供GitSearch系统的软件架构概览，采用若干架构视图描述系统的不同方面，以便表示构造系统所需要的重要架构决策。 |
| 2.对象与范围（说明整个文档的内容范围和针对的读者对象） |  |
| 3.参考文献（说明文档中主要的所需引用） | 1.《软件需求规格说明书》，本团队；《软件架构文档模版》，Rational Software Corporation，2002；  2.骆斌。软件工程与计算（卷三）  3.丁二玉，刘钦。软件工程与计算（卷二）[M]机械工业出版社，2012：134—182 |
| 4.名词与术语(说明文档中常用的技术缩略和先相关词条) | GitSearch表示gitHub项目数据统计与分析系统的代名称 |

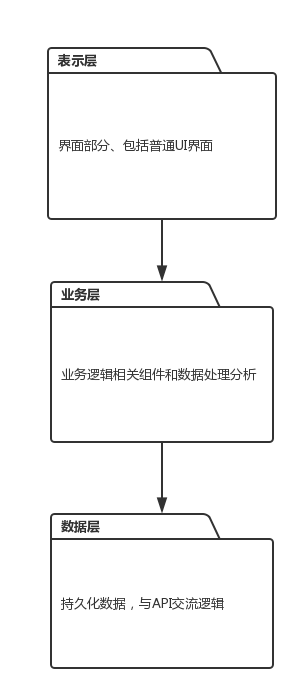
**词汇表**

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |
| GitSearch | GitHub项目数据统计与分析系统 |  |

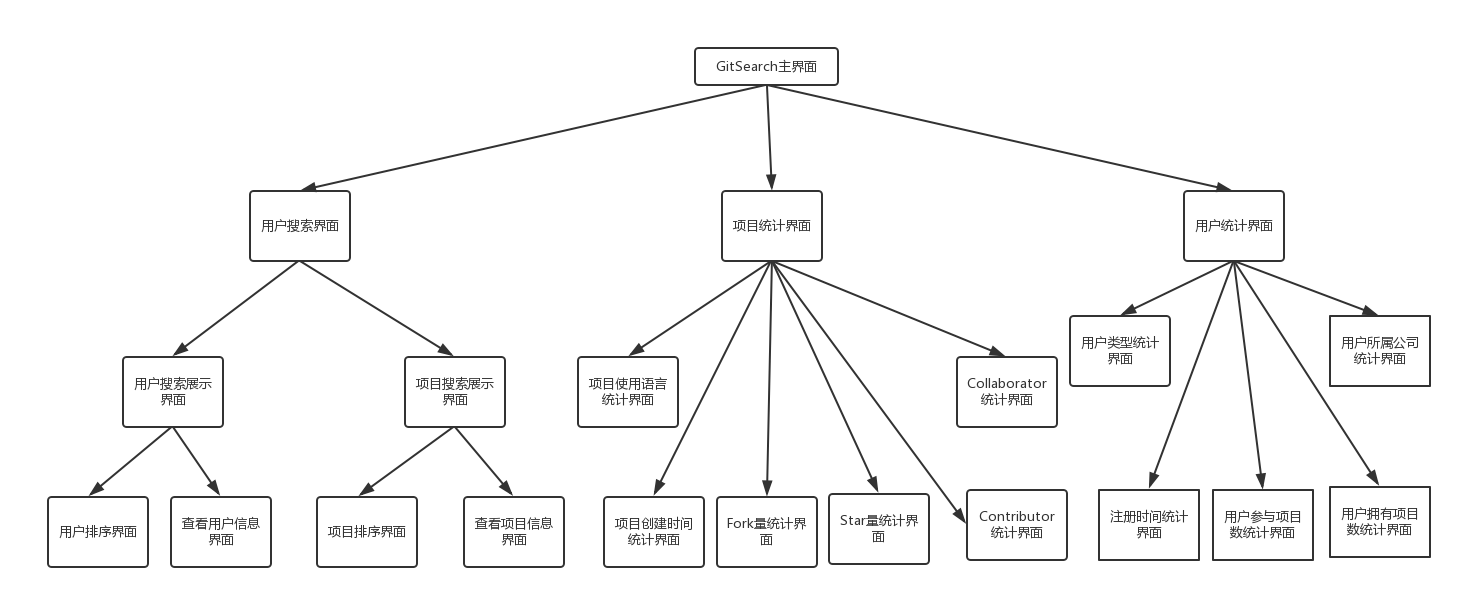
1. **产品概述**

GitSearch旨在通过对github网站项目和用户数据的分析展示，来帮助广大编程爱好者查找到自己感兴趣的项目，了解相关项目的一些基本信息。主要由Java语言开发，利用数据库相关功能配合简单和复杂的数理逻辑计算，为用户提供一个多角度全方位访问Github项目数据信息的窗口。良好的用户交互体验和准确全面的数据分析，使用户能找到自己感兴趣的项目。

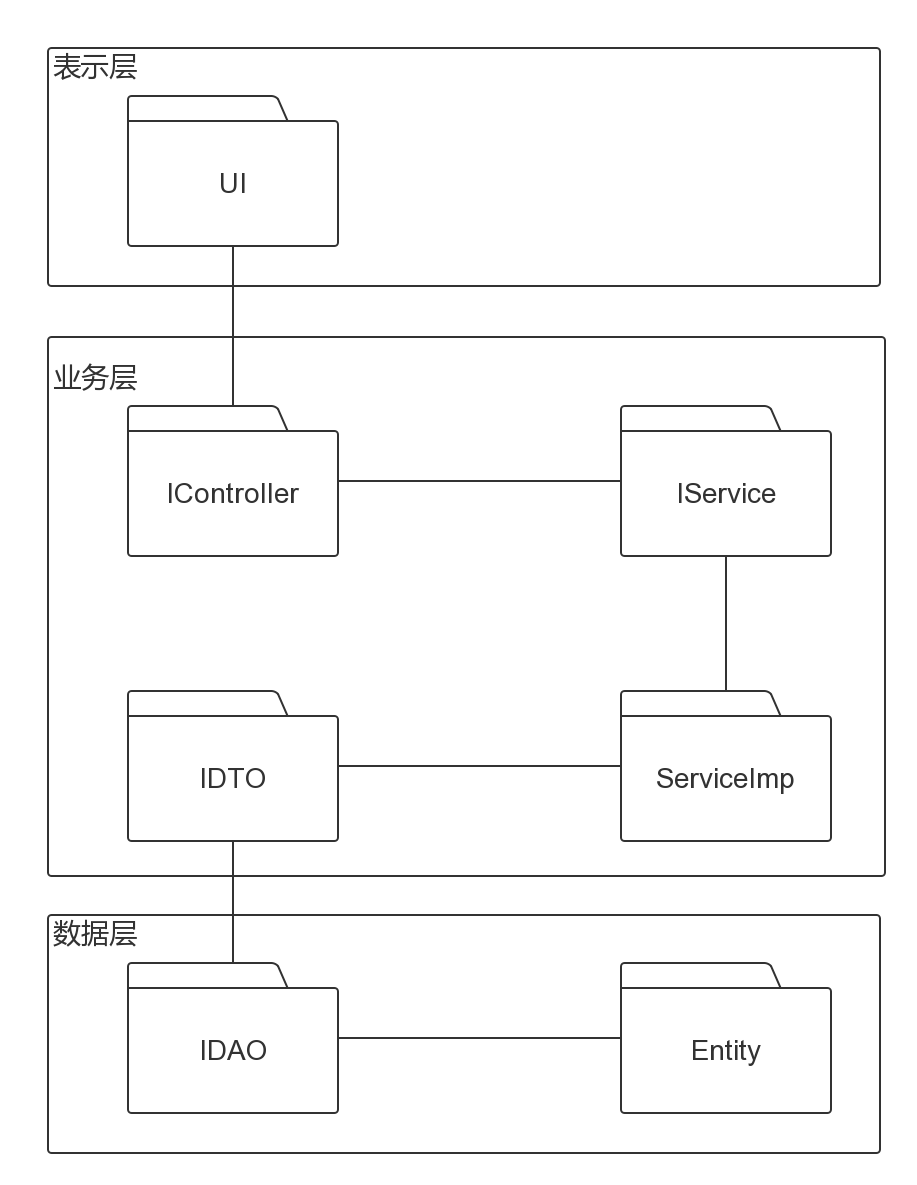
1. **系统的分层架构**

****

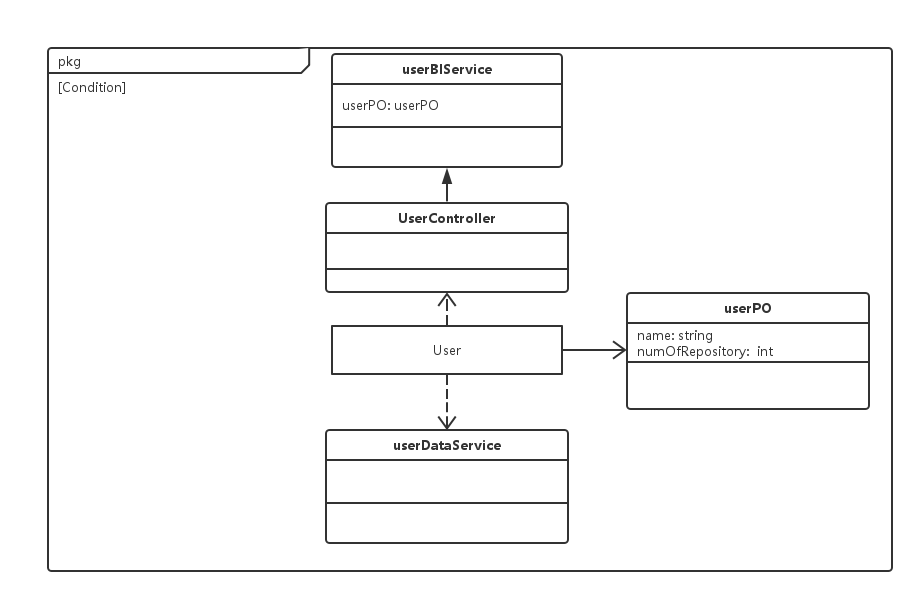
1. **用户界面层的分解**

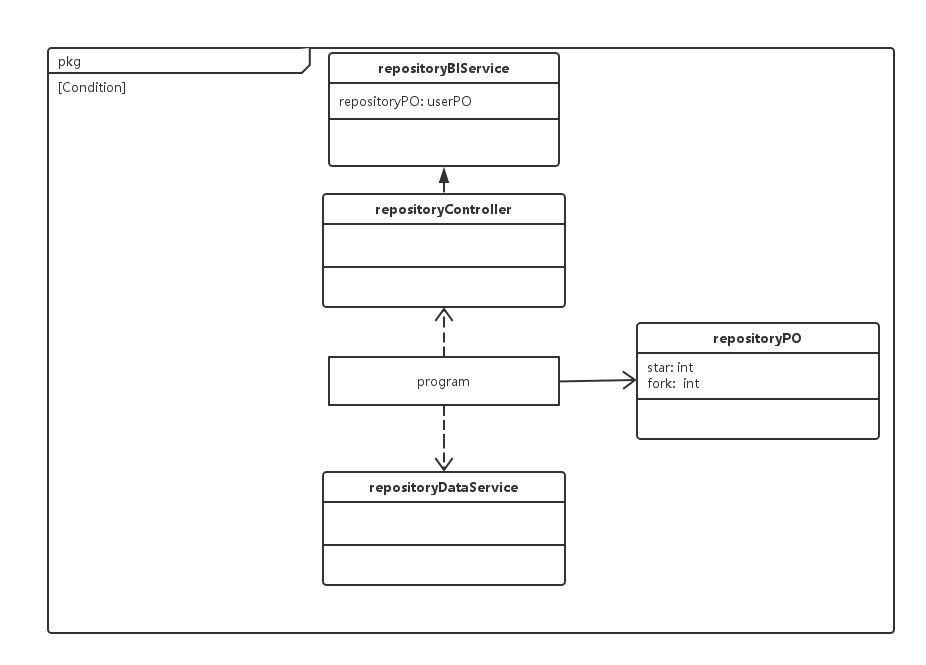


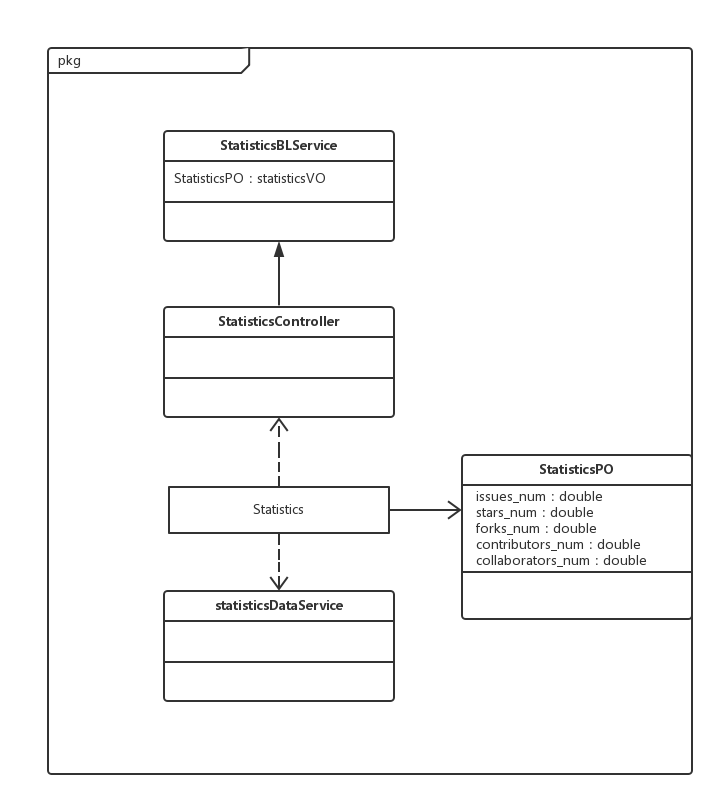
1. **系统的架构设计**



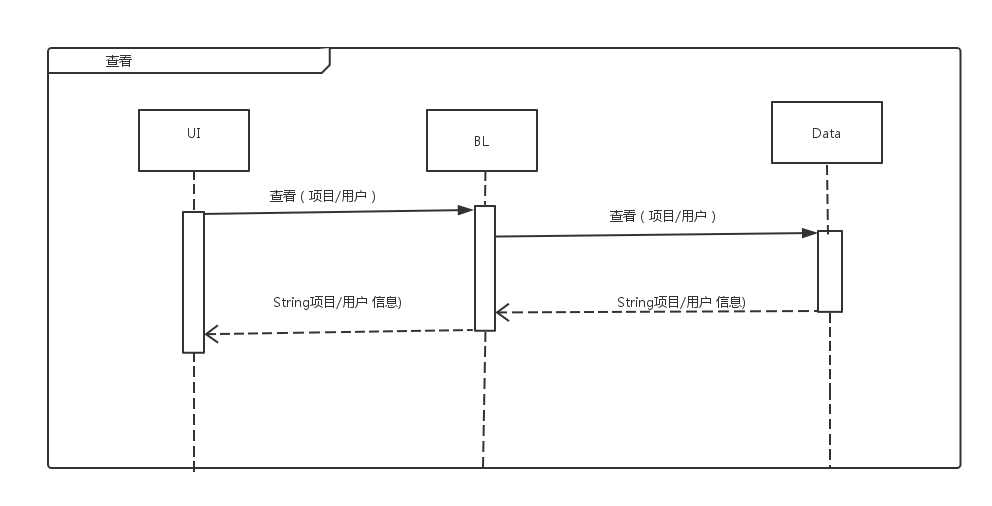
1. **系统的类图设计**

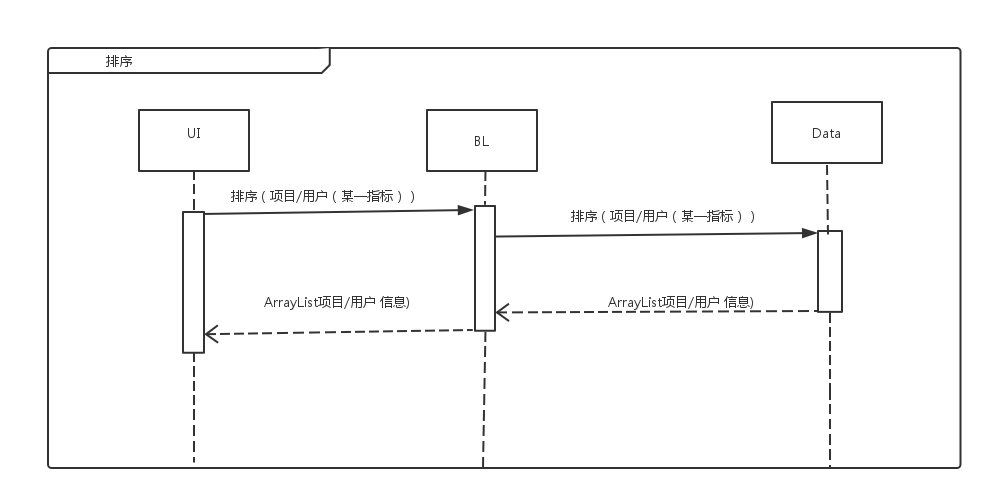


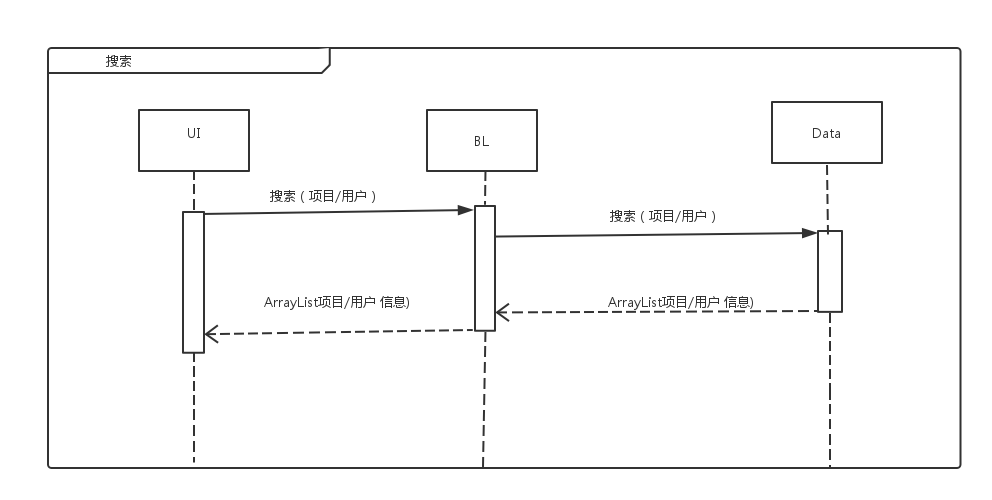


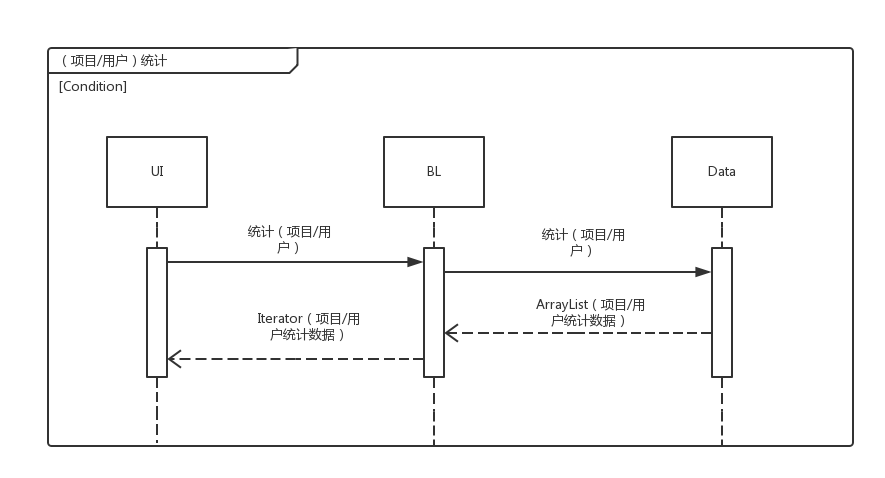


1. **系统顺序图**









1. **数据层的分解**

数据层主要给业务逻辑层提供数据服务，包括对于持久化数据的增、删、改、查。业务逻辑需要的服务DataService接口提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的具体描述如图：

接口组件设计结构如下表格：

|  |  |  |  |
| --- | --- | --- | --- |
| 接口ID | 连接组件 | 接口信息 | |
| I1 | 连接UI与IController | 语法 | Return(Response) Interface(Request) |
| 前置条件 | 用户的输入正确 |
| 后置条件 | 处理控制组件处理请求并且响应 |
| 不变量 | 用户请求信息 |
| I2 | 连接IContraller与IService | 语法 | Return(result) Interface() |
| 前置条件 | 无 |
| 后置条件 | 对应的IService执行对应的业务逻辑 |
| 不变量 | 无 |
| I3 | 连接IService与IDTO | 语法 | Return(dataSet) Interface(command) |
| 前置条件 | 无 |
| 后置条件 | 对应的IDTO组件调用特定IDAO类获取数据层数据，并返回数据集 无 |
| 不变量 | 无 |
| I4 | 连接IDAO与Entity | 语法 | Return(data) Interface(criteria) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | IDAO中的类将Entity对象写入数据库或从数据库中返回Entity对象 |
| 不变量 | 无 |

1. **接口规范设计**

Repository\_BL的接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| Repository\_BL.getRepositories | 语法 | Iterator<RepositoryVO> getRepository (); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回所有项目详情列表 |
| Repository\_BL. getRepositoriesNames | 语法 | Iterator<String> getRepositoriesNames(); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回项目名称列表 |
| Repository\_BL. languagesOf Repository | 语法 | Map<String,Integer> languagesOfProgram(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回项目的语言使用情况 |
| Repository\_BL. checkRepository | 语法 | RepositoryVO checkRepository (String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的信息 |
| Repository\_BL.screen | 语法 | Iterator<RepositoryVO> screen(ScreenVO vo)； |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回多个项目列表 |
| Repository\_BL.Search | 语法 | Iterator<RepositoryVO> Search(String name) |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回多个项目列表 |
| Repository\_BL.sort | 语法 | Iterator<RepositoryVO> sort |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回多个项目列表 |
| Repository\_BL. getContributors | 语法 | Iterator<UserVO> getContributors(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回项目所有参与者的信息 |
| Repository\_BL. getStar | 语法 | StaStrVO getStar()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Star量信息 |
| Repository\_BL. getScores | 语法 | StatisticsVO getScores(RepositoryVO vo)； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Score量信息 |
| Repository\_BL. getForks | 语法 | StaStrVO getForks()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Fork量信息 |
| Repository\_BL. getRepoCreated | 语法 | StaStrVO getRepoCreated()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回项目创建信息 |
| Repository\_BL. getLanguage | 语法 | StaStrVO getLanguage()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Language信息 |
| Repository\_BL. getUserCreated | 语法 | StaStrVO getUserCreated()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回UserCreated信息 |
| Repository\_BL. getUserType | 语法 | StaStrVO getUserType()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回UserType信息 |
| Repository\_BL. getUserHas | 语法 | StaIntVO getUserHas()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回拥有项目数信息 |
| Repository\_BL. getUserRelated | 语法 | StaIntVO getUserRelated()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回参与项目数信息 |
| Repository\_BL. getCompany | 语法 | StaStrVO getCompany()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Company信息 |
| Repository\_BL. getHasNum | 语法 | Integer getHasNum(String name)； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回创建项目数信息 |
| Repository\_BL. getCollaborator | 语法 | StaStrVO getCollaborator()； |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Collaborator信息 |
| Repository\_BL. getContributor | 语法 | StaStrVO getContributor(); |
| 前置条件 | 已创建一个Repository\_BL对象 |
| 后置条件 | 返回Contributor信息 |
| Repository\_BL. NamesOfContributors | 语法 | Iterator<String> NamesOfContributors(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回所有项目参与者的用户名 |
| Repository\_BL. getBranches | 语法 | Iterator<ProgramVO> getBranches(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回项目的所有版本 |
| Repository\_BL. NamesOfBranches | 语法 | Iterator<String> NamesOfBranches(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回项目的所有版本的名字 |
| Repository\_BL. checkBranch | 语法 | ProgramVO checkBranch(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 查看某一个版本的信息 |
| Repository\_BL. BranchInfo | 语法 | String BranchInfo(String userName, String reponame, BranchInfo info); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个版本的某项信息 |
| Repository\_BL. getForks | 语法 | Iterator<ProgramVO> getForks(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回一个项目的所有Fork项目 |
| Repository\_BL. NamesOfForks | 语法 | Iterator<String> NamesOfForks(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的所有fork项目的全称 |
| Repository\_BL. checkFork | 语法 | ProgramVO checkFork(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个fork项目的信息 |
| Repository\_BL. ForkInfo | 语法 | String ForkInfo(String userName, String reponame, ForkInfo info); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个fork项目的某项信息 |
| Repository\_BL. getPulls | 语法 | Iterator<String> getPulls(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Pull Request内容（前50个） |
| Repository\_BL. PullInfo | 语法 | String PullInfo(String userName, String reponame, String num, PullInfo info); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个Pull Request的某项信息 |
| Repository\_BL. getPull | 语法 | String getPull**(**String userName**,** String reponame**,int** numbers**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个Pull Request的某项信息 |
| Repository\_BL. getPullInfo | 语法 | String getPullInfo**(**String userName**,** String reponame**,int** number**,**Item item**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Issues内容,一页50个 |
| Repository\_BL. getIssues | 语法 | Iterator**<**String**>** getIssues**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Issue编号,一页50个 |
| Repository\_BL. getIssueNum | 语法 | Iterator**<**Integer**>** getIssueNum**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的某个issue |
| Repository\_BL. getIssue | 语法 | String getIssue**(**String userName**,** String reponame**,int** issueNum**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个Issue的某项信息 |
| Repository\_BL. getIssueInfo | 语法 | String getIssueInfo**(**String userName**,** String reponame**,int** num**,**IssueInfo info**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Commits内容,一页50个 |
| Repository\_BL. getCommits | 语法 | Iterator**<**String**>** getCommits**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的ommit编号,一页50个 |
| Repository\_BL. getForks | 语法 | Iterator<ProgramVO> getForks(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回一个项目的所有Fork项目 |
| Repository\_BL. NamesOfForks | 语法 | Iterator<String> NamesOfForks(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的所有fork项目的全称 |
| Repository\_BL. checkFork | 语法 | ProgramVO checkFork(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个fork项目的信息 |
| Repository\_BL. ForkInfo | 语法 | String ForkInfo(String userName, String reponame, ForkInfo info); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个fork项目的某项信息 |
| Repository\_BL. getPulls | 语法 | Iterator<String> getPulls(String userName, String reponame); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Pull Request内容（前50个） |
| Repository\_BL. PullInfo | 语法 | String PullInfo(String userName, String reponame, String num, PullInfo info); |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个Pull Request的某项信息 |
| Repository\_BL. Getpull | 语法 | String getPull**(**String userName**,** String reponame**,int** numbers**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个Pull Request的某项信息 |
| Repository\_BL. getPullInfo | 语法 | String getPullInfo**(**String userName**,** String reponame**,int** number**,**Item item**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Issues内容,一页50个 |
| Repository\_BL. getIssues | 语法 | Iterator**<**String**>** getIssues**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Issue编号,一页50个 |
| Repository\_BL. getIssueNum | 语法 | Iterator**<**Integer**>** getIssueNum**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的某个issue |
| Repository\_BL. getIssue | 语法 | String getIssue**(**String userName**,** String reponame**,int** issueNum**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个Issue的某项信息 |
| Repository\_BL. getIssueInfo | 语法 | String getIssueInfo**(**String userName**,** String reponame**,int** num**,**IssueInfo info**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的Commits内容,一页50个 |
| Repository\_BL.getCommits | 语法 | Iterator**<**String**>** getCommits**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的ommit编号,一页50个 |
| Repository\_BL.getCommitNum | 语法 | Iterator**<**Integer**>** getCommitNum**(**String userName**,** String reponame**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的某个commit |
| Repository\_BL.getCommit | 语法 | String getCommit**(**String userName**,** String reponame**,int** sha**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个commit的某项信息 |
| Repository\_BL. getCommitInfo | 语法 | String getCommitInfo**(**String userName**,** String reponame**,int** sha**,**CommitInfo info**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回单个项目的某个issue的评论 |
| Repository\_BL.getIssueComm-ent | 语法 | String getIssueComment**(**String userName**,** String reponame**,int** num**);** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个commit的评论 |
| Repository\_BL. getCommitComment | 语法 | String getCommitComment**(**String userName**,** String reponame**,int** num**)** |
| 前置条件 | 已创建一个Repository\_BL领域对象 |
| 后置条件 | 返回某个commit的评论 |

User\_BL的接口规范：

|  |  |  |
| --- | --- | --- |
| User\_BL. checkUser | 语法 | UserVO checkUser(String name); |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回某个user所有信息 |
| User\_BL. UserInfo | 语法 | String UserInfo(String user, UserInfo info); |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回user某个信息 |
| User\_BL. getStargazers | 语法 | Iterator<String> getStargazers(String userName, String reponame); |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回单个项目的点赞者,一页50个 |
| User\_BL. getStargazerNames | 语法 | Iterator<String> getStargazerNames(String userName, String reponame; |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回单个项目的点赞者登录名,一页50个 |
| User\_BL. getSubscribers | 语法 | Iterator<UserVO> getSubscribers(String userName, String reponame); |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 |  |
| User\_BL. NamesOfSubscriber | 语法 | Iterator<String> NamesOfSubscriber(String userName, String reponame); |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回单个项目的订阅者,一页50个 |
| User\_BL. sortUser | 语法 | Iterator<UserVO> sortUser(User\_Sort sort); |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回符合某项信息的用户，一页50个 |
| User\_BL. screenTime | 语法 | Iterator<UserVO> screenTime(String time) |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回符合某项信息的用户，一页50个 |