# 详细设计文档

南京大学软件学院

# 文档更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 描述 | 日期 |
| V0.1 | 蒙奕锟、陈欢 | 初稿 | 2015－03-03 |
|  |  |  |  |

# 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 |
|  |  |

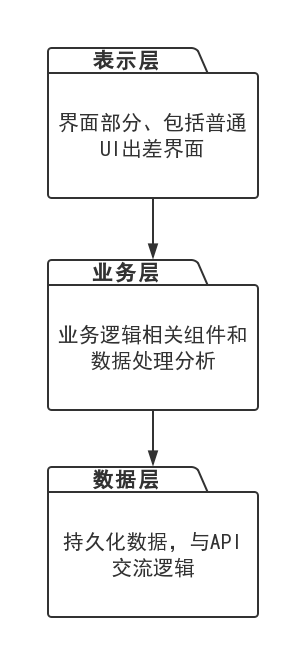
# 词汇表

|  |  |  |
| --- | --- | --- |
| 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 | itHub项目数据统计与分析系统 |  |

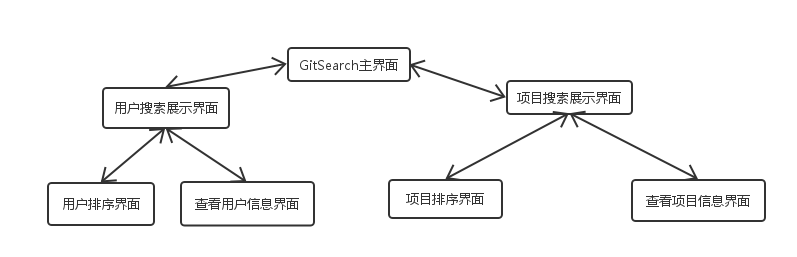
# 产品概述

## 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。

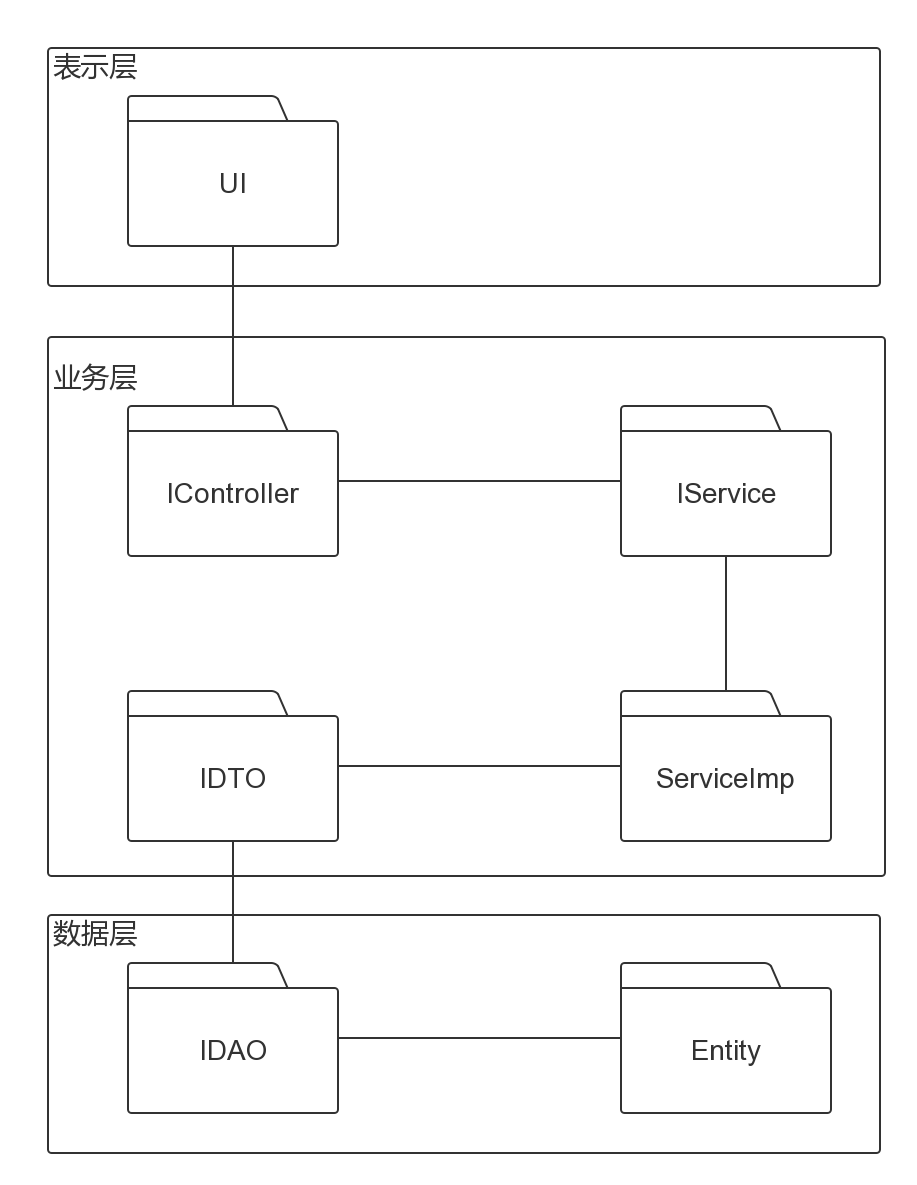
# 系统的分层架构



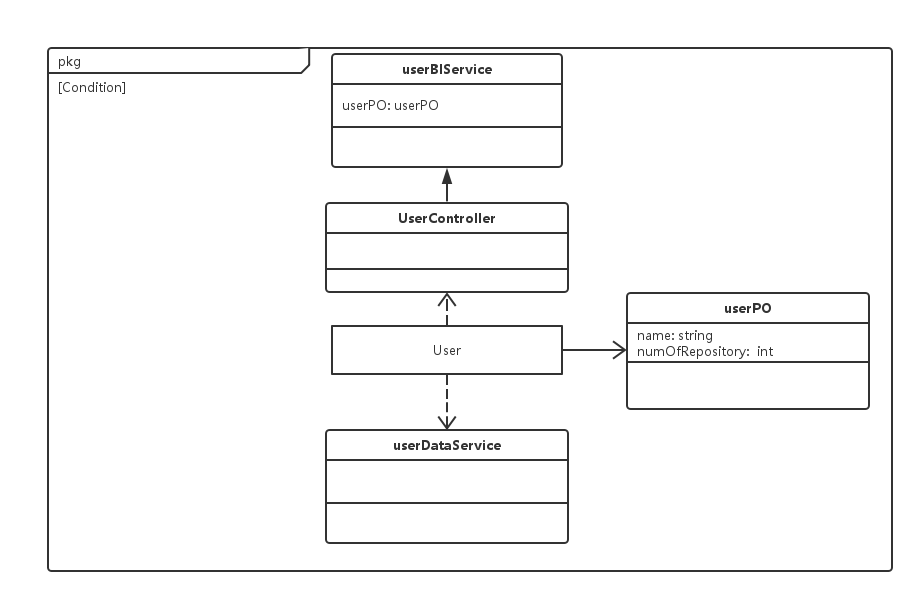
# 用户界面层的分

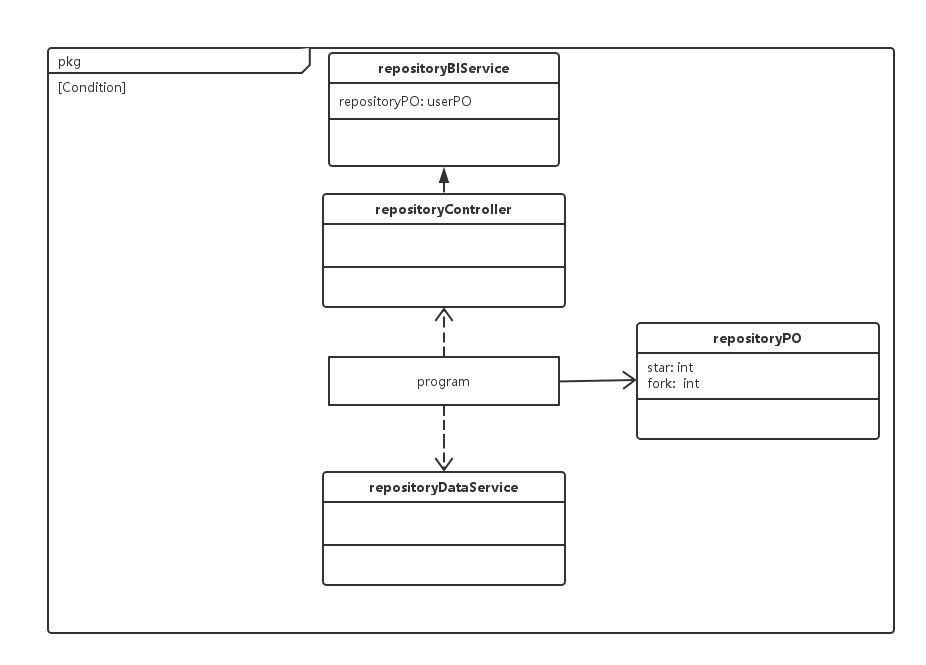


# 系统的架构设计

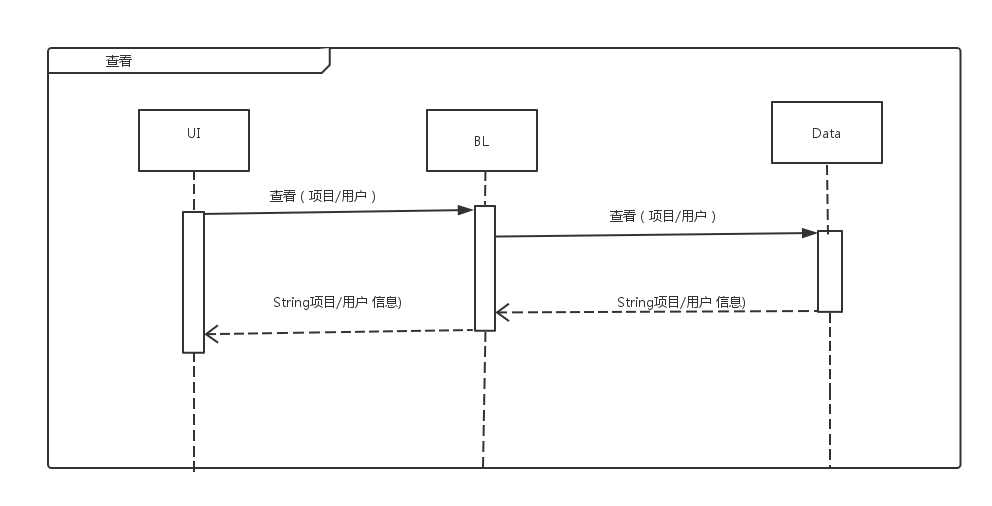


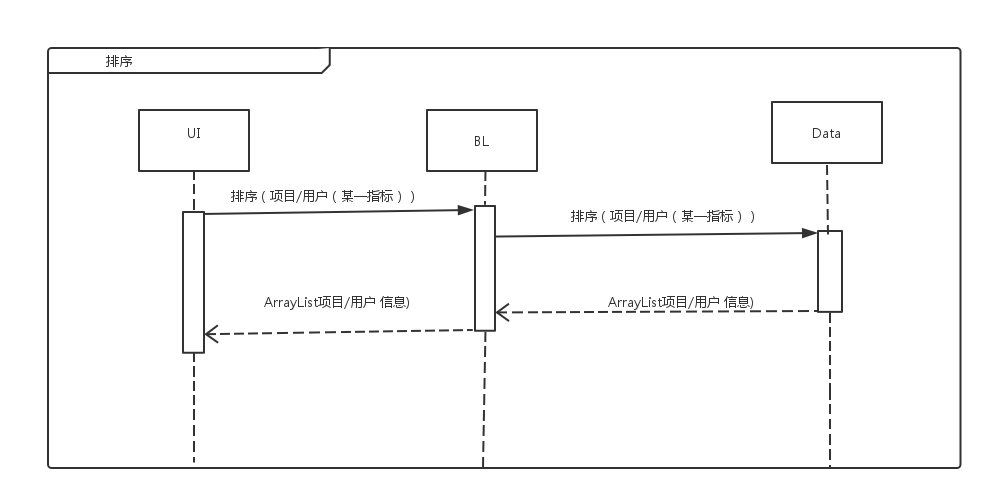
# 系统的类图设计

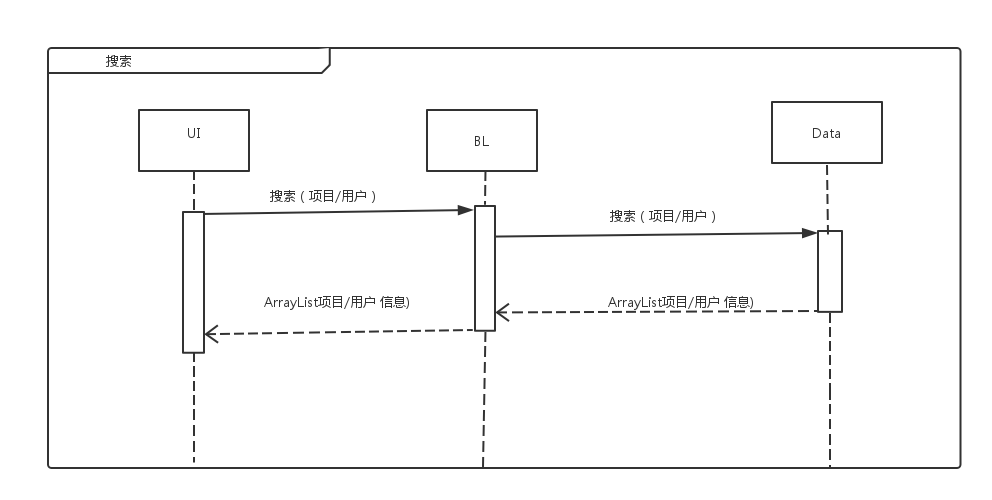




# 系统顺序图







# 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 |

# 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User\_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User\_bl领域对象 |
| 后置条件 | 返回项目的语言使用情况 |
| Repository\_BL. RepositoryInfo | 语法 | String RepositoryInfo(String userName, String reponame, RepositoryInfo info); |
| 前置条件 | 已创建一个User\_bl领域对象 |
| 后置条件 | 返回单个项目的信息 |
| 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. 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 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. | 语法 | Iterator**<**String**>** getUserInfo**(**User user**);** |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回某个user的详细信息 |
| User\_BL. | 语法 | String getUserInfo**(**User user**,**UserInfo info**);** |
| 前置条件 | 已创建一个User\_BL.领域对象 |
| 后置条件 | 返回某个user的某项信息 |
| User\_BL. | 语法 | Iterator**<**String**>** getstargazers**(**User user**,**String userName**,** String reponame **,int** pageNum**);** |
| 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. | 语法 | Iterator**<**String**>** getstargazerNames**(**User user**,**String userName**,** String reponame**,int** pageNum**);** |
| 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. | 语法 | Iterator**<**String**>** getstargazerNames**(**User user**,**String userName**,** String reponame**,int** pageNum**);** |
| 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. | 语法 | Iterator**<**String**>** getSubscriberNames**(**User user**,**String userName**,** String reponame **,int** pageNum**);** |
| 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登录名,一页50个 |