软件工程 - 第三阶段
ReadTogether 图书管理系统
项目管理文档

MarkAsRead 软工小组

December 14, 2013

Contents

Git9 使用情况报告

整体提交统计信息

从上次阶段结束的 11 月 16 日开始，MarkAsRead 小组在 Git9 上一共有 71 次 commits。

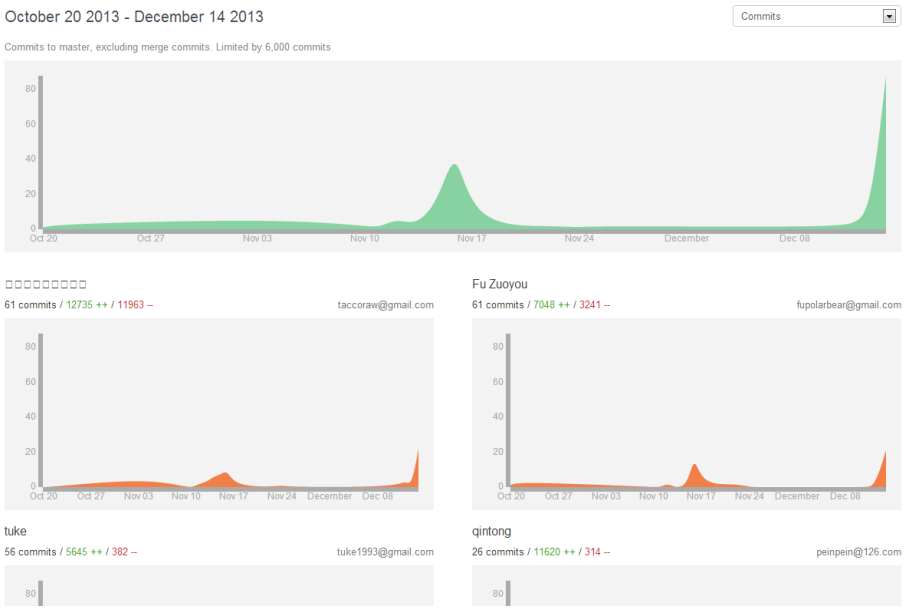

Figure 1: Commits 界面截图

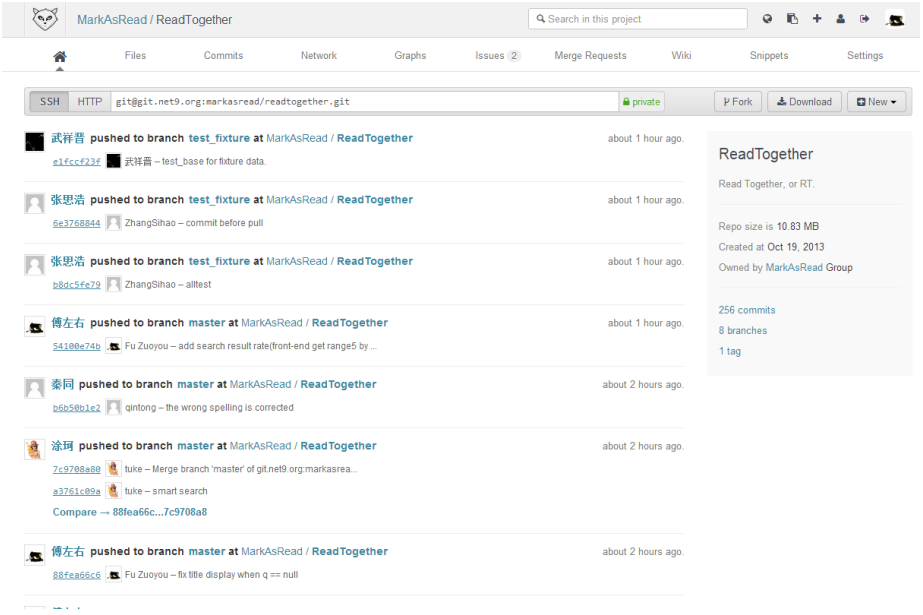

Figure 2: git9 主界面，已经有 256 个提交，可见开发活跃度非常高

Git 使用

通过前几个阶段的练习，我们对 git 的使用变的更加熟练。这一个阶段也是向原型系统添加各种功能。

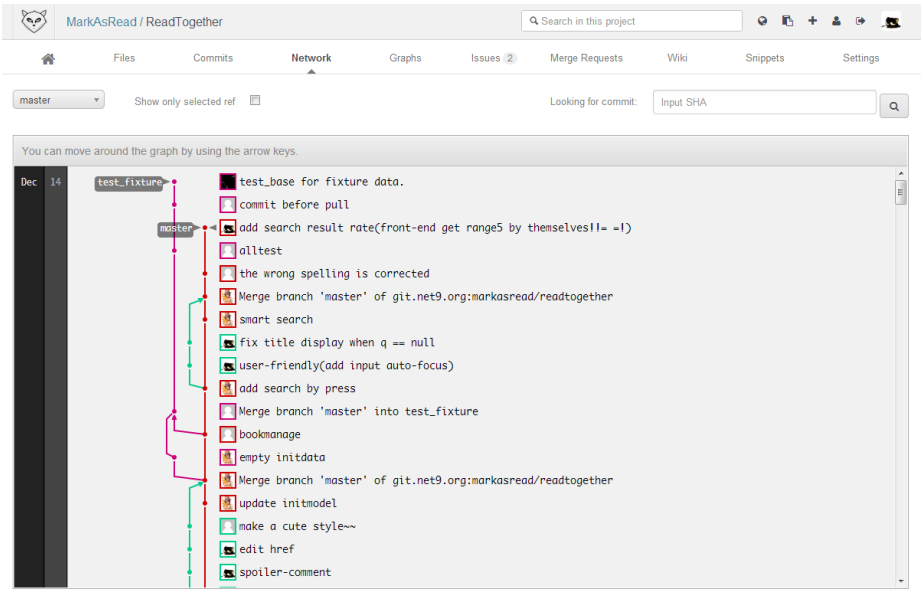

Figure 3: Git 网络图

IssueTracker 使用情况报告

整体 Issue 统计信息

从上次阶段结束的 11 月 16 日开始，MarkAsRead 小组一共发现并跟踪了 9 个 Issue。内容主要分布在代码风格、安全性、前后端衔接几个方面。

我们充分利用了 Issuetracker 便捷的邮件提示和论坛式的讨论功能。最后我们成功解决了大部分发现的 Issue，只剩下了 2 个 Issue 留待以后解决。

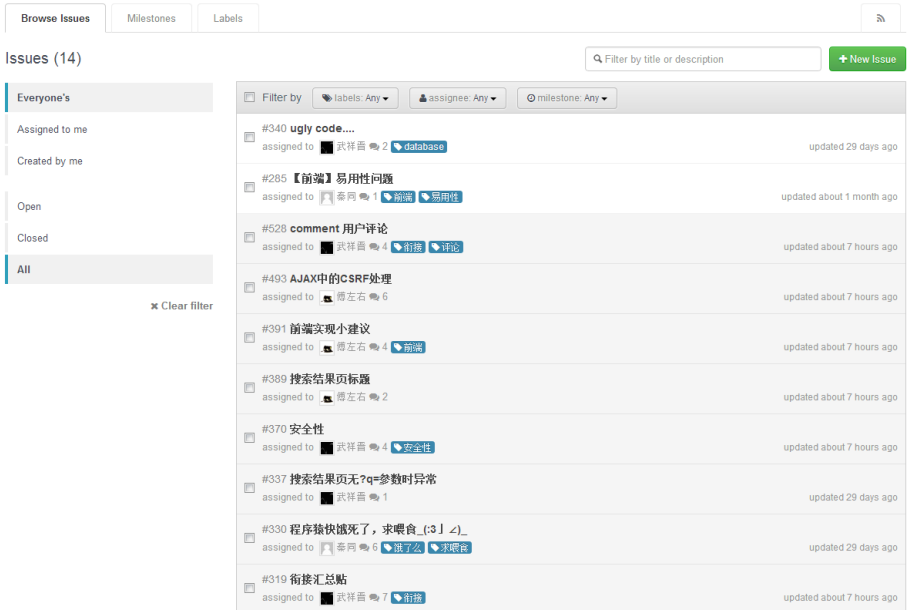

Figure 4: Issuetracker 使用情况截图

其他辅助工具的使用

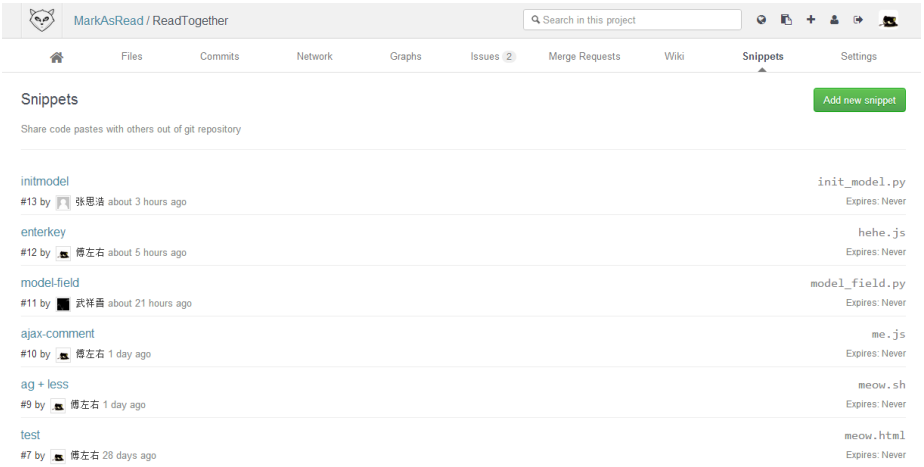

Figure 5: Git9 代码交流，在这里我们可以分享好的代码实现

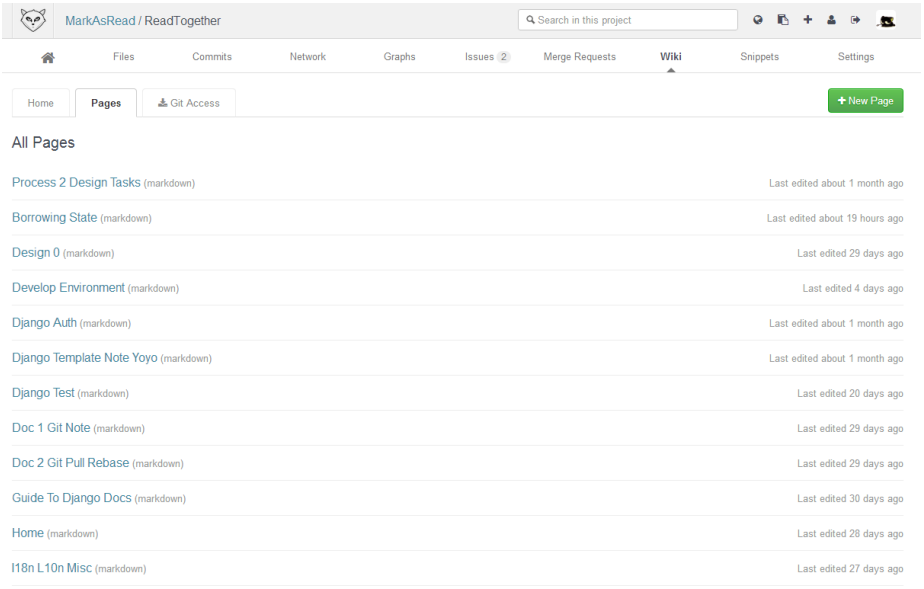

Figure 6: wiki 极大的帮助我们分享工具的使用、交流编码经验和进行内部设计约定

项目分工

姓名	主要职责
武祥晋	组长、前后端衔接、测试
涂珂	数据库设计与管理
傅左右	前端设计、项目管理文档
秦同	前端设计、测试
李响	测试文档
张思浩	关键测试

项目进度安排

- 2013-11-17 至 2013-11-25

  进行初次讨论。由组长武祥晋同学研究 Django1.6 新版本特性、迁移方案以及了解相关的测试工具。相关文档发布至 Git9 上的 wiki 供组内同学充分参考。

- 2013-11-25 至 2013-12-10

  负责测试的同学充分了解测试工具的使用和测试目的。

- 2013-11-11 至 2013-12-14

  进行所有测试。对系统的各个模块进行充分细致的测试。

阶段总结

在这一个阶段里，我们又圆满的完成了任务。首先我们大大的完善了我们的图书管理系统，向原型系统中补充和添加了需求和设计文档相应的要求。

然后我们使用了 Django Unittest、Selenium、Coverage、Pylot 诸多工具完成了对我们图书管理系统的单元测试、集成测试、覆盖率测试、负载测试。我们的图书馆有着良好的性能和稳定性。

通过软件测试，我们对软件工程的测试流程有了进一步的认识，也对我们编写的图书管理系统的功能完善和稳定性有了更多的信心。

[附录] 11-18 Django 研究记录

计划包括以下内容：字符编码、时区、显示格式

名词定义

- Internationalization: 缩写 i18n，程序中为 l10n 所做的准备。
- localization: 缩写 l10n，包括翻译和格式化。
- Translation: 文字翻译。Django 中的 `USE_I18N` 设置。
- Formatting: 日期时间、数字等的格式化。Django 中的 `USE_L10N` 设置。
- locale: 语言或语言 + 地区，语言小写国家大写，如 *zh_CN* 、*en_US* 。
- HTTP Accept-Language code: *zh-cn* 。（我看 Chrome 是 *zh-CN* ）
- message file: 即 `.po` 文件。
- translation string: 待翻译的字符串。
- format file: 定义了特定 locale 数据格式的 Python 模块（module)。
- naive datetime: 无时区信息。
- aware datetime: 有时区信息，时间为该时区时间。

相关 Django 设置

USE_I18N、LANGUAGE_CODE、LANGUAGE_COOKIE_NAME、LANGUAGES、LOCALE_PATHS；USE_L10N、USE_THOUSAND_SEPARATOR、FORMAT_MODULE_PATH USE_TZ、TIME_ZONE

`TEMPLATE_CONTEXT_PROCESSORS` - `django.core.context_processors.i18n` - `django.core.context_processors.tz`

`MIDDLEWARE_CLASSES` - `django.middleware.locale.LocaleMiddleware`

时区

Python 的 `datetiime` 模块包含以下类：- `date` - `datetime` - `time` - `timedelta` - `tzinfo` Django 的 `django.utils.timezone` 模块包含以下函数：- `now` - `is_aware` - `is_naive` - `make_aware` - `make_naive`

参考资料

- Django 1.5 topics/i18n/timezones
- pytz
- datetime
- MySQL 5.5 服务器时区支持

[附录] 11-25 测试工具研究记录

testing

主要是在 `tests.py` 中定义 `django.test.TestCase` 的子类进行 unit test，可以不用 doctest。

testing/overview

使用 `from django.utils import unittest` 而不是 `import unittest` 。一般直接使用 `from django.test import TestCase` 而不要使用 `unittest.TestCase` ，除非你对此很了解。即 `unittest` 要用 Django 的不用 Python 自带的，`TestCase` 要用 test 包的不用 unittest 包的。

使用 `python manage.py test` 运行测试，一次 Ctrl-C 安全退出，第二次强制退出。具体指令格式另行整理。运行测试时无需用 `python manage.py runserver` 启动一个服务器。

It is a bad idea to have such import-time database queries in your code anyway - rewrite your code so that it doesn't do this.

test client

`django.test.client.Client`

用于检查正确的模版是否被渲染，以及是否从 view 得到正确的变量。不能取代 Selenium 。后者用于模拟浏览器检查 HTML 是否正确以及 JavaScript 功能是否正确。后者的使用需要配合 `LiveServerTestCase` 。

`django.test` 包中的 `django.utils.unittest.TestCase` 的子类

- `SimpleTestCase`
- `TransactionTestCase`
- `TestCase`
- `LiveServerTestCase`

特性

`TestCase` 的子类可直接使用 `self.client` 而无需自己构造 `Client` 。

fixture 加载。

自定义测试用的 urlconf。相当于测试中的 URL Reverse。

临时自定义设置。

各种 Assertion。

testing/advanced

The request factory

独立测试 view 函数用的。

自定义测试框架

与 coverage.py 集成

在开启 coverage 的情况下运行测试：`coverage run --source='.' manage.py test myapp`

显示报告：`coverage report` 或 `coverage html`

Other Tools

- cucumber
- Lettuce
- splinter
- Guide with Django

Selenium -> Capybara -> Cucumber Capybara = Pycabara Rspec = should-dsl, PySpec

SpecLoud uses nose with beautiful syntax and highlight with green & red ludibrio provides mocks & stubs

Gherkin syntax: Pyccuracy, lettuce, freshen, behave pycuke is dead, use freshen instead; freshen is dead, uses nose, complex! behave > lettuce: pythonic, use test database, but poor with django

[附录] 11-30 python 学习小记

启用 Python 2.7 的警告 `-Wd`

Python 3 兼容？`-3` 警告开关

`__future__` 特性

列表综合，元组 pack/unpack

字符串格式化

[附录] 12-13 会议记录

以下 url 对应的行为解释：

- borrow: 对于一本在架上的书，管理员登记其被借走（copy_id, myuser_id）
- return: 对于一本借出的书，管理员收到了用户的归还，处于 ``整理中'' (同时通知预约的用户)
- queue_next: 被通知的预约用户向管理员取走了书
- readify: 整理中的书被摆到架上
- disappear: 被借出的书已经向管理员赔偿，被标记为永久丢失

新书上架、图书永久下架先不考虑。

方便管理员的用户查询、拷贝查询、图书查询 AJAX GET 接口稍后讨论。