Dolphin Content新人Ramp-up

# 前言

为了方便新入职的员工/实习生 加入到 Dolphin Content项目的开发上来，期望能整理出一个 Ramp-up 的方案, 帮助其快速熟悉项目，并且方便融入到项目中。

# Ubuntu/Linux快速入门

## Ubuntu/Linux环境搭建(实体机/虚拟机)

1. 安装VMWare

软件地址： [\\share\ToolShare\Virtual\_PC\VMWare\VMware-workstation8.0](file:///\\share\ToolShare\Virtual_PC\VMWare\VMware-workstation8.0)

1. 安装Ubuntu
   1. 统一使用12.04版本的ubuntu系统
   2. http下载地址：<http://mirrors.yun-idc.com/ubuntu-releases//precise/ubuntu-12.04.3-desktop-amd64.iso>

本地共享：\\share\TeamShare\Content Team

1. 安装教程

<http://www.linuxidc.com/Linux/2012-07/64897.htm>

## Ubuntu相关知识及基础配置

### Vim

集成了Winmanager/NERDTree等插件以后就是一个很强大的编辑工具。链接：

<http://blog.csdn.net/bokee/article/details/6633193>

<http://www.cppblog.com/songhaifeng/archive/2011/03/20/142284.html>

还有Kun Li的git：<http://github.com/likun5053/vimrc/> 可以从这里获得插件，如果懒的话，可以直接用他的配置

* 一些常用的VIM命令

1、多行注释

1)、Ctrl+V 进入列模式，向下或向上移动光标，把需要注释的行的开头标记起来；然后按大写的I，再插入注释符(比如#)；再按Esc，就会全部注释。

2)、使用替换方式进行多行注释

:s/^/# # 用"#"注释当前行

:2,50s/^ /# # 在2~50行首添加"#"注释

:.,+3s/^/# # 用"#"注释当前行和当前行后面的三行

:%s/^/# # 用"#"注释所有行

2、文本替换

"s"代表替换，part1代表查找的内容，part2代表替换的内容，"%"代表所有行，"g"代表替换整行里所有的内容（如果不加"/g"则只替换每行第一个匹配part1的地方）。

:s/part1/part2 # 用part2替换当前行中第1个part1

:s/part1 /part2/g # 用part2替换当前行中所有的part1

:%s/part1/part2 # 用part2替换所有行中每行第1个part1

:%s/part1/part2/g # 用part2替换所有行中所有的part1

:2,50s/part1 /part2 # 用part2替换第2行到第50行中每行第1个part1

:2,50s/part1/part2/g # 用part2替换第2行到第50行中所有的part1

:.,+3s/part1/part2 # 用part2替换当前行以及当前行后面的三行中每行第1个part1

:.,+3s/part1/part2/g # 用part2替换当前行以及当前行后面的三行中所有的part1

3、字符映射

：ab 替换符 原始信息 for example: :ab mail wuxianglong098@163.com

4、 全局搜索

:vimgrep 搜索文本 搜索路径/\*\*

:cn[ext] 跳到下一个匹配文本

:cp[revious] 跳到上一个匹配文本

:cnf[ile] 跳到下一个文件

:cpf[ile] 跳到上一个文件

:cr[ewind] 跳到开始位置

:cla[st] 跳到结束位置

:col[der] iterate through historical quickfix lists

:cnew[er] iterate through historical quickfix lists

5、拷贝, 删除与粘贴

在vi 中y 表示拷贝, d 表示删除, p 表示粘贴. 其中拷贝与删除是与光标移动命令。结合的, 看几个例子就能够明白了.

yw 表示拷贝从当前光标到光标所在单词结尾的内容.

dw 表示删除从当前光标到光标所在单词结尾的内容.

y0 表示拷贝从当前光标到光标所在行首的内容.

d0 表示删除从当前光标到光标所在行首的内容.

y$ 表示拷贝从当前光标到光标所在行尾的内容.

d$ 表示删除从当前光标到光标所在行尾的内容.

yfa 表示拷贝从当前光标到光标后面的第一个a字符之间的内容.

dfa 表示删除从当前光标到光标后面的第一个a字符之间的内容.

yy 表示拷贝光标所在行.

dd 表示删除光标所在行.

D 表示删除从当前光标到光标所在行尾的内容.

6、快速移动光标

在vi中, 移动光标和编辑是两件事, 正因为区分开来, 所以可以很方便的进行光标定位和编辑. 因此能更快一点移动光标是很有用的.

w(e) 移动光标到下一个单词.

b 移动光标到上一个单词.

0 移动光标到本行最开头.

^ 移动光标到本行最开头的字符处.

$ 移动光标到本行结尾处.

H 移动光标到屏幕的首行.

M 移动光标到屏幕的中间一行.

L 移动光标到屏幕的尾行.

gg 移动光标到文档首行.

G 移动光标到文档尾行.

c-f (即ctrl 键与f 键一同按下) 本命令即page down.

c-b (即ctrl 键与b 键一同按下, 后同) 本命令即page up.

'' 此命令相当有用, 它移动光标到上一个标记处, 比如用gd, \* 等查找到某个单词后, 再输入此命令则回到上次停留的位置.

'. 此命令相当好使, 它移动光标到上一次的修改行.

### Eclipse

1. 官方下载地址：

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR1/eclipse-standard-kepler-SR1-linux-gtk-x86\_64.tar.gz

## 常用的命令/工具介绍

* crontab：运行定时任务。
* crul：文件获取工具，抓取网页内容很是方便。
* awk：经常用来处理日志，+ uniq + grep + sort + sed 处理日志超级给力。

http://www.ruanyifeng.com/blog/2012/01/a\_bash\_script\_of\_apache\_log\_analysis.html

* top, ps： 用来查看正在运行的命令（当跑任务奇慢无比的时候可以看看什么正在占用你的资源）
* 下载安装（linux里面的下载安装命令超级强大）：

1) wget + 空格 + 要下载文件的url路径

eg. wget http://www.linuxsense.org/xxxx/xxx.tar.gz

2) sudo apt-get install package(想要安装的插件)

3) pip 主要用来安装python插件

$ sudo pip install virtualenv（有了它就可以安装各种python插件了）

* 一些常用的其他命令，介绍了更改apt-get源的方法

见文件：

## 常见问题

# Python/Django快速入门

## Python开发环境搭建

Ubuntu系统默认安装了Python，所以不需要单独安装Python；但是项目开发过程中会需要安装一些基础的python包，主要包括：

* python2.7-dev (如果自己的python版本是2.6，则安装python2.6-dev)

sudo apt-get install python2.7-dev

* python-software-properties

sudo apt-get install python-software-properties

* python-pip (python的包管理工具)

sudo apt-get install python-pip

pip是安装Python Lib的最方便的方式，官方网站：<https://pypi.python.org/pypi/pip>；如果在安装的时候不知道Lib的具体名字和版本，可以到这个网站上搜索一下，pip一些常用的命令（以安装django为例）

sudo pip install django==1.4 # 安装lib，通过==制定版本号

sudo pip uninstall django # 卸载已安装的lib

sudo pip search django # 搜索名字里包含django字符的所有Python Lib

* pymongo (mongodb的python lib)

sudo pip install pymongo==2.0

* python-mysqldb (mysqldb的python lib)

sudo apt-get install python-mysqldb

另外，还有libxml2-dev，libjpeg8-dev等，这些依赖也可以在部署项目的时候使用-C参数自动安装，详见下面。

## Python常用库/文档/书籍/Demo Code

### 常用的LIB

* 正则表达式

这篇blog对正则介绍的很详细：<http://www.cnblogs.com/huxi/archive/2010/07/04/1771073.html>

* simplejson

安装：sudo pip install simplejson

用法：load(s) 🡪 将字符串或文件转化为json格式

dump(s) 🡪 将json格式内容转化为字符串

带s的函数是对字符串的处理，不带s的是对文件对象的处理

参考资料：<http://www.cnblogs.com/dwnblogs/archive/2012/11/01/2749299.html>

* urllib & urllib2 & requests

Python处理URL请求的模块，推荐使用requests，简单强大，网上有很多介绍这些lib的文章：

[Python: difference between urllib and urllib2](http://www.hacksparrow.com/python-difference-between-urllib-and-urllib2.html)

<http://stackoverflow.com/questions/2018026/should-i-use-urllib-or-urllib2-or-requests>

### 书籍 & Demo Code

* Learn Python The Hard Way：<http://learnpythonthehardway.org/book/>
* Google Python风格指南：<http://zh-google-styleguide.readthedocs.org/en/latest/google-python-styleguide/>

## Django开发环境搭建

Dolphin Content使用Django + uWSGI + Nginx架构实现，搭建开发环境主要需要安装一下几个东西：

* Django

sudo pip install Django==1.4

* uWSGI

sudo pip install uwsgi

* Nginx

sudo add-apt-repository ppa:Nginx/stable

sudo apt-get update

sudo apt-get install nginx

* Admin Tools

sudo pip install django\_admin\_tools

* Admin Enhancer

sudo pip install django\_admin\_enhancer

## Django常用库/文档/书籍/Demo Code

### Demo Code

1、简明python教程：http://www.2cto.com/shouce/pythonjc/index.html（入门级材料）

2、比较pythonic的一些用法（ppt做的不错，一些用法可以让代码更简洁）：

http://www.slideshare.net/mpirnat/a-few-of-my-favorite-python-things?utm\_source=Python+Weekly+Newsletter&utm\_campaign=2d9cfcda12-Python\_Weekly\_Issue\_18\_January\_19\_2012&utm\_medium=email

3、Django入门：http://py3k.cn/

### Django常用命令

1、本地run django项目：sudo python manage.py runserver 0.0.0.0:8888

2、执行Django Command：sudo python manage.py (command 命令 + 参数)

3、同步数据库：sudo python manage.py syncdb

4、生成locale翻译文件：

django-admin.py makemessages -a

django-admin.py compilemessages --locale=zh\_CN

5、Django的命令行界面：sudo python manage.py shell （超级好用， 可以直接导入model，用python命令操作）

## 常见问题

# GIT快速入门

## Gerrit账户

Gerrit是管理GIT提交，review代码的WEB平台，具体的初始化步骤如下：

1. 进入http://codeserver.baina.com:8080网址
2. 点击右上角的sign in登陆，用户名和密码均为入职时已经分配好的outlook的用户名和密码，不必申请
3. 将用户名邮件发送给直属经理，主题为申请GIT权限，内容表明需要申请的GIT项目名称
4. 配置GIT和SSH的秘钥连接
   1. 进入ubuntu命令行，使用下面指令生成SSH Keys

ssh-keygen -t rsa -C "xian\*\*@gmail.com" -f ~/.ssh/csser-github

简单介绍下参数含义：

-t 指定密钥类型，默认即rsa ，可以省略

-C 设置注释文字，比如你的邮箱

-f 指定密钥文件存储文件名，会生成 csser-github 和 csser-github.pub 两个密钥文件

* 1. 回车后，遇到提示输入 yes 即可，剩下一路回车，密钥文件就在指定路径下生成了
  2. 将SSH公钥添加到 Github

登录 Github 帐号，找到帐号设置 -> SSH Keys

点击 Add New SSH Key

将本地生成的公钥文件（csser-github.pub）中的文字全选复制到 key 栏，点击 add key 保存。

PS：可以参考文档进行操作

## GIT常用命令及流程

1、git status：查看对那些文件进行了修改，或者当前的提交状态

2、git diff：查看对文件做了哪些修改（也可以安装gitk/gitg，直接用图形化界面查看做的修改，很方便哦）

3、git add . (后面有个点，stage当前所有add/modified的文件，不包括delete的文件)

4、git add –A：stage本地GIT项目中所有修改过的文件，包括add/modified/delete的文件

5、git commit (会打开一个页面，用来说明此次做了哪些修改)

6、git pull --rebase (用来从线上拉下最新的代码， 注意，当你的修改与最新代码有冲突的时候会在这里做提示confilct，你就有进行第7步，当然如果你没有冲突，可以直接进行第9步，提交代码)

7、解决冲突， 打开有冲突的文件，搜索<<<定位冲突代码， 会出现这个问题，是因为你和其他同事同时修改了这部分代码，需要与同事沟通，相关代码的功能，删除多余代码及<<< >>>等字符

8、 冲突解决之后， git status 会发现又会列出所有的修改文件，需要git add . 添加文件，然后 git rebase –continue

9、代码没有问题了，开始push代码了，git push origin HEAD:refs/for/master

PS： 你提交的代码此时还没有merge到develop分支上，要邀请同事code review, 打分， 有两分就可以merge了。在merge之前呢，如果发现一些问题，还可以修改代码对你的提交打补丁，即patch，修改完之后一切安装上述步骤来处理，只是第5步的命令改成 git commit –amend

## GIT参考书籍

1、Qi Chen分享的GIT图解，讲的很清楚，有各种高级用法：

http://nettedfish.sinaapp.com/blog/2013/08/05/deep-into-git-with-diagrams/

2、GIT入门教程：https://www.atlassian.com/git

## 常见问题

### GIT commit之后push到server的时候提示没有commit id

**解决方法：**需要将一个commit-msg脚本文件放到DolphinContent-Server/.git/hooks目录下面，文件：

# Dolphin Content项目/开发团队介绍

## 项目介绍

Dolphin Content是一个包括小说、视频和游戏的内容平台，该平台包含爬虫、数据管理控制台、内容API和搜索API，其中API Service主要被用于海豚浏览器炫风版客户端和WEB端页面：

* 爬虫：爬取小说/视频的源数据，小说主要来自百度；视频有优酷、爱奇艺、乐视等内容源
* 控制台：运营小说/视频的源数据，以及榜单、卡片等客户端/WEB端数据
* 内容API：提供客户端/WEB端小说（视频）榜单、卡片、详情等数据的API
* 搜索API：提供根据用户输入关键词搜索小说（视频）的API，Java实现
* 客户端：在浏览器（海豚炫风版）中以卡片的形式展现，如下图：



* WEB端：基于小说（视频）数据实现的WEB页面，包括榜单、专题等，地址如下：

小说： <http://v5kanshu.com/novel.html>

视频： <http://55yule.com/video.html>

## 开发团队

Dolphin Content开发团队人员如下：

|  |  |  |
| --- | --- | --- |
| 模块 | 开发人员 | 备注 |
| 爬虫 | Kun Li/Liuxi Wu/Qingpei Guo | 初期由Yu Xia负责，现在主要是以上三位负责开发 |
| 控制台 | Qi Chen/Xiaoxue Wang/Pingyuan Chen | 初期是Qi和Xiaoxue主要负责，Pingyuan近期加入 |
| 内容API | Kun Li/Qi Chen | 目前很少改动 |
| 搜索API | Wenyuan Wu | 暂时停止开发 |
| 客户端 | Bole Wang/Jie Li | 客户端暂时不再开发 |
| WEB端 | Pingyuan Chen/Bo Wang | 分别负责小说/视频的WEB端页面 |

# Dolphin Content项目上手

## 获取源代码

使用以下指令下载源代码：

爬虫：git clone ssh://username@codeserver:29418/Coolpad.git

控制台 & API：git clone ssh://username@codeserver:29418/DolphinContentServer.git

其中：username为入职时邮箱的用户名，即@bainainfo.com之前的内容

## 本地环境配置

所有的项目部署都是使用ServerDeployEngine工具，所以需要先下载该GIT项目的代码，然后执行安装脚本：

git clone ssh://username@codeserver:29418/ServerDeployEngine.git

cd ServerDeployEngine/

sudo python setup.py install

sudo ./install.sh

ServerDeployEngine使用文档（使用outlook打开）：

### 爬虫

1、安装Eclipse IDE、MongoDB

$ Sudo apt-get install openjdk-7-jdk

$ sudo apt-get install eclipse

$sudo apt-get mongo

2、部署CoolPad到本地环境并安装依赖

* 打开Terminal在Coolpad路径下执行命令：ddep test –C –r contentservice，加上-C参数会自动安装项目环境依赖；如果成功部署，整个爬虫项目至此已经部署成功并且正在运行。
* 如果要停止爬虫service：

在Coolpad目录下：

$ cd contentserivce/scripts/

$ sudo ./stop.sh

3、调式单个爬虫（以bdzy.py为例）

* 在eclipse中安装PyDev插件

具体步骤：http://blog.sina.com.cn/s/blog\_3fe961ae01013iko.html

* 安装成功后，在eclipse中导入项目，打开bdzy.py(contentservice/contentservice/crawlerimpl

/video/bdzy.py)，按F11启动单个爬虫调试

* 单个爬虫入口为if \_\_name\_\_ ==”\_\_main\_\_”

### 控制台

环境配置文件是在DolphinContent-Server/conf中，已经有local, test两个环境，local 文件配置的是172.16.255.246的测试机， test文件配置的是172.16.255.245的测试机

控制台部署如下 (-C参数会自动安装项目依赖)：

ddep local dolphinop\_admin\_service

ddep test dolphinop\_admin\_service

### Service API

类似于控制台，部署命令：

ddep local dolphinop\_service

ddep test dolphinop\_ service

### 搜索API

1. 安装依赖
   1. sudo apt-get install openjdk-6-jre
   2. sudo apt-get install jetty

sudo vi /etc/default/jetty

NO\_START=0

JETTY\_HOST=0.0.0.0

1. 部署项目
   1. cd DolphinContent-Server/

./dolphinop-search/make\_package.sh

ddep env -r dolphinop-search

# Dolphin Content项目开发流程/发布流程

## 开发流程

### 团队协作

每个release的团队协作开发流程如下：



### 分支管理

1. develop分支
   1. 正在开发的分支，代码不稳定
   2. 提前开发下一版本
   3. 周一至周三在此分支上是代码不稳定的开发时间
   4. DEV在此分支上进行自测
2. release分支
   1. develop完成一个版本的开发之后会将代码合并到release分支
   2. 该分支是Tester测试和DEV进行bug fix的分支，时间一般是周四和周五
   3. 部署上线使用release分支的代码
3. master分支
   1. 只读分支，不做修改，只会从其他分支合并代码
   2. release上线或hotfix之后，需要合并代码到master；确认没有问题之后，打tag
4. hotfix分支
   1. 主要用来做线上hotfix
   2. 做hotfix的时候，需要先从master同步代码，然后在此分支上进行bug fix
   3. hotfix完成之后需要将代码合并到master分支

开发过程中GIT代码流程图：



## 发布流程

Release时间定在周一，如有问题不能发布，则顺延一到两天，具体的流程如下：

1. 上午11点之前，将测试通过，确认可以发布的代码合并到master分支；
2. 12点之前，发出release note邮件，说明本次发布内容，负责人以及出现问题时的回滚方法，建议参考：
3. 15点的时候执行上线部署操作
4. 各个负责人线上检查，确保上线的功能点正常
5. 如果线上发现问题，则从master分支迁一份代码到hotfix分支，在hotfix分支修改之后重新部署上线；之后再将代码合并到master分支
6. 确认没有问题之后，在master分支上打tag

# Dolphin Content实习生基本要求/考核指标/管理

## 基本要求

* 根据Ramp-up文档独立搭建开发环境，熟悉GIT和项目开发工具
* 熟练掌握基于Python & Django的开发
* 每天发送Daily　Report汇报当天的工作进展以及遇到的一些问题
* 与PM讨论确认自己负责的功能，Code之前需要完成Design Doc，并和相关人员进行Review
* 了解并遵守公司的开发流程和代码规范（Python的PEP8）

## 考核指标

1. 优秀标准
   1. 出勤率占实习期的98％以上（最多请假2天）
   2. 实习期间参与Dolphin Content Server端的开发，能够独立完成PM提出的关于控制台的需求，代码质量达到上线标准
   3. 能够独立完成参与的开发工作的Dev Design Doc，并在review会议上回答大家的提问，改善系统的设计
   4. 能够plan自己一周的工作项，并且能够按时完成，每天按时Daily Report汇报工作进展，及时提出开发中碰到的问题，保证计划的工作按时上线，对自己开发的模块负责维护
   5. 能够熟练的运用python语言进行开发，熟练的使用Django框架
2. 一般标准
   1. 出勤率刚好达到公司的要求
   2. 实习期间参与Dolphin Content Server端的开发，能够完成PM提出的关于控制台的简单需求，代码质量达到上线标准
   3. 能够在mentor的帮助下完成Dev Design Doc，并能够按照plan完成所分配的开发任务
   4. 每天按时Daily Report汇报工作进展，及时提出开发中碰到的问题
   5. 能够熟练的运用python语言进行开发

## 管理

* mentor每两周需要和实习生进行一次One One Meeting
* mentor需要关注实习生的Daily Report，对其遇到的问题及时进行解答
* 实习生的mentor需要对其写的Design Doc和提交的代码进行review，并提出改进意见