Python的Jira模块用于将JIRA的REST API进一步封装, 从而我们不需要直接用Python的requests模块进行直接请求，相当于进一步等封装。提高更好的效率。

1.

创建Jira实例：

options = {

'server': 'https://jira.atlassian.com'

}

jira = JIRA (options) # 方式1，传入option字典，指定传入的地址，还可以指定进行认证的方式等。

Jira = JIRA ('https://jira.atlassian.com')# 方式2，指定Jira服务器的地址

2.

Jira模块支持多种认证方式。

采用基于Cookie的方式, auth\_jira = JIRA(auth=('username', 'password'))

采用基本的基于HTTP的方式 auth\_jira = JIRA(basic\_auth=('username', 'password'))

采用OAuth方式

3.

Jira模块将Jira中的issue, comment, projects, watchers都作为对象进行处理。

--- Issue对象

issue = jira.issue('JRA-1330') # 通过Jira模块获取一个issue对象

summary = issue.fields.summary # 获取issue中的一个域名

issue = jira.issue('JRA-1330', fields='summary,comment') # 只获取issue的其中几个域

jira.assign\_issue(issue, 'newassignee') # 将issue重新指派给其他人

issue.update(summary='new summary', description='A new summary was added') # 通过关键字参数更新域的值

issue.update(fields={'summary': 'new summary', 'description': 'A new summary was added'}) # 通过传入字典地键值对更新issue域的值

issue\_dict = {

'project': {'id': 123},

'summary': 'New issue from jira-python',

'description': 'Look into this one',

'issuetype': {'name': 'Bug'},

}

new\_issue = jira.create\_issue(fields=issue\_dict) # 通过字典传入域的值，生成一个issue, 也可以传入字典列表，一次性生成多个issue.

issue\_in\_project =jira.search\_issues('project=PROJ and assignee != currentUser()') # 采用Jira Searching Language（JQL）来进行搜索

---- Comment对象

comments\_a = issue.fields.comment.comments # 通过issue获取comment对象

comments\_b = jira.comments(issue) # 通过issue获取comment对象

comment = jira.add\_comment (issue, 'new comment') # 为issue加入新的comment

---- Project对象

jra = jira.project('JRA') # 获取jira的project对象

print(jra.name) # 获取project对象的名字

---- Watchers对象

watcher = jira.watchers(issue) # 获取issue的watcher

jira.add\_watcher(issue, 'username') # 为issue加入watcher

jira.remove\_watcher(issue, 'username') # 为issue移除watcher

Attachment使用：

我们可以在Jira当中上传Attachment, Python的Jira模块也提供支持。

# upload file from `/some/path/attachment.txt`

jira.add\_attachment(issue=issue, attachment='/some/path/attachment.txt') # 提供文件路径上传文件

# read and upload a file (note binary mode for opening, it's important):

with open('/some/path/attachment.txt', 'rb') as f:

jira.add\_attachment(issue=issue, attachment=f)

# attach file from memory (you can skip IO operations). In this case you MUST provide `filename`.

import StringIO

attachment = StringIO.StringIO()

attachment.write(data)

jira.add\_attachment(issue=issue, attachment=attachment, filename='content.txt')

4.

Jira Shell可以创建一个属于Jira Python模块的shell环境，在其中去试用各种语句。Jira Shell将连同pip安装jira模块的时候，一起依赖安装。并且jira shell建立在ipython上，所以需要安装ipyhon.

jirashell -s <http://jira.atlassian.com> # 进入jira shell并且连接给定的jira服务器地址。