**Using the REST API with Python**

Here is the list of available libraries for using REST API with Python:

1. [Python-Redmine](https://python-redmine.com/)
2. [PyRedmineWS](https://github.com/ianepperson/pyredminews)
3. [PyActiveResource](http://code.google.com/p/pyactiveresource/)
4. [PyRed](https://github.com/alourie/pyred)

**Python-Redmine:**

Python-Redmine is a library which supports 100% features of Redmine's REST API. It provides a simple but powerful Pythonic API inspired by a well-known Django ORM and is thoroughly tested. Example:

>>> **from** **redminelib** **import** **Redmine**

>>> redmine = Redmine('http://demo.redmine.org', username='foo', password='bar')

>>> project = redmine.project.get('vacation')

>>> project.id

30404

>>> project.identifier

'vacation'

>>> project.created\_on

datetime.datetime(2013, 12, 31, 13, 27, 47)

>>> project.issues

<redminelib.resultsets.ResourceSet **object** **with** Issue resources>

>>> project.issues[0]

<redminelib.resources.Issue #34441 "Vacation">

>>> **dir**(project.issues[0])

['assigned\_to', 'author', 'created\_on', 'description', 'done\_ratio',

'due\_date', 'estimated\_hours', 'id', 'priority', 'project', 'relations',

'start\_date', 'status', 'subject', 'time\_entries', 'tracker', 'updated\_on']

>>> project.issues[0].subject

'Vacation'

>>> project.issues[0].time\_entries

<redminelib.resultsets.ResourceSet **object** **with** TimeEntry resources>

**PyRedmineWS example:**

Suppose Eric fell ill and was out for several days. You need to crawl through the project called Parrot and move any due date for issues assigned to Eric out by two more weeks.

The dateutil library contains a handy method called relativedelta for calculating relative dates.

# Import the Redmine class

**from** **redmine** **import** **Redmine**

**from** **dateutil.relativedelta** **import** **relativedelta**

server = Redmine('http://my-server.com', username='Me', password='seakrit')

project = server.projects['parrot']

# Find Eric in the user data

**for** u **in** server.users:

**if** u.firstname == 'Eric' **and** u.lastname == 'Idle':

user = u

**break**

**else**:

**raise** **Exception**("Didn't find Eric Idle in the user dateabase")

# Extend issues in project assigned to user by two weeks

**for** issue **in** project.issues(assigned\_to\_id=user.id):

**if** issue.due\_date **is** **not** None:

issue.due\_date += relativedelta(weeks=+2)

issue.save('Giving Eric more time to complete - he was out ill')

**PyActiveResource example:**

# Importing pyactiveresource

**from** **pyactiveresource.activeresource** **import** **ActiveResource**

**class** **Issue**(ActiveResource):

\_site = 'http://redmine.foo.org'

\_user = 'username'

\_password = 'password'

# Get issues

issues = Issue.find()

# Get a specific issue, from its id

issue = Issue.find(1345)

# Issue attributes

# Updating an attribute

**PyRed example: TBD**[**¶**](http://www.redmine.org/projects/redmine/wiki/Rest_api_with_python#PyRed-example-TBD)