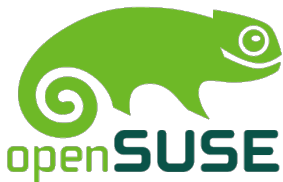


# How to write OBS plugins – oSC14

Extend the functionality of OBS



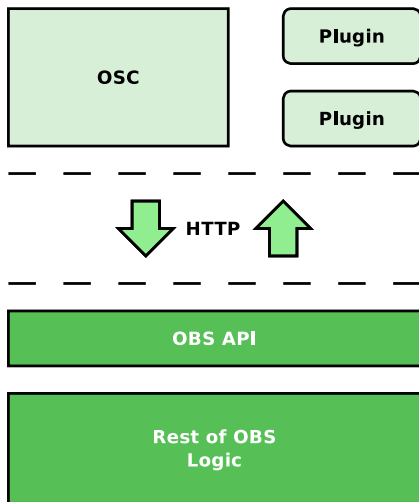
Alberto Planas  
<aplanas@suse.de>  
openSUSE Team

April 23, 2014

# Introduction

# OBS Architecture Overview

---



## OSC and OBS Relation

---

- OSC is an interface to OBS
- Can be extended using plugins
- Use HTTP request to the API, using a REST-like protocol
- Results are usually serialized in XML
- And is written in Python!

# A Basic Example

## First Example. Print Raw Request

---

```
from osc import cmdln
from osc.core import http_GET
from osc.core import makeurl

@cmdln.alias('vr')
def do_viewrequest(self, subcmd, opts, *args):
    """${cmd_name}: view the raw content of a request

    Usage:
        ${cmd_name} SR ...
            View the raw content of a request
    ${cmd_option_list}
    """
    apiurl = self.get_api_url()
    for sr in args:
        url = makeurl(apiurl, ['request', str(sr)])
        print http_GET(url).read()
```

# Plugging Deployment

---

```
# Copy or create a link
ln -sr example1.py ~/.osc-plugins

# Check that the new command is there
osc --help

# Check that docstring is OK
osc vr --help
```

# OSC Plugins

---

An OSC plugin consist in:

- An entry point (`do_XXX`) as a unbounded method
- A function decorator to set parameters and alias
- A docstring to describe the command, with examples
- A way to create request to OBS API (GET and POST)
- We can use high level functions from `osc.core`
- A way to parse XML results



# Plugin Entry Point

---

- The entry point is an unbounded method for the class `osc.Osc`
- This allows you to access some functions via `self`
- You decorate the function (method) with `@cmdln` to add parameters:

```
@cmdln.alias('vr')
@cmdln.option('-p', '--project', action='append', help=
```

Use the same syntax that Python `optparse` library

- The docstring will appear when *osc command -help*

## Low Level Functions

---

In *osc.core* you can find some low level functions to access OBS API.

- *makeurl()* – Build the URL needed to access a service
- *http\_[GET/POST/PUT/DELETE]()* – To create a HTTP request to the URL
- *ET.parse()* – Use cElementTree to parse the XML

```
url = makeurl(apiurl, [<PATH>], query=<QUERY_DIC>)
try:
    root = ET.parse(http_GET(url)).getroot()
except urllib2.HTTPError, e:
    print('ERROR in URL %s [%s]' % (url, e))
```

## Examples of URLs. Change review status

---

```
# Use POST to change the review status of a request
query = {
    'cmd': 'changereviewstate',
    'newstate': 'declined'
    'by_user': 'the_boos',
}
url = makeurl(opts.apiurl, ['request', sr_id], query=query)
try:
    root = ET.parse(http_POST(url, data='Sorry')).getroot()
    code = root.attrib['code']
except urllib2.HTTPError, e:
    print('ERROR in URL %s [%s]' % (url, e))
```

## Examples of URLs. User information

---

```
# Get user information
url = makeurl(apiurl, ['person', 'aplanas'])
try:
    root = ET.parse(http_GET(url)).getroot()
    print root.find("realname").text, root.find("email").text
except urllib2.HTTPError, e:
    print('ERROR in URL %s [%s]' % (url, e))
```

## High Level Functions

---

Dig into *osc.core* to find more high level functions

```
# Print user 's email
from osc.core import get_user_data
get_user_data(apiurl, u, 'email')
```

```
# Search packages
from osc.core import search
xpath = "@name='%s' or @name='%s'" % (pkg1, pkg2)
result = search(apiurl, package=xpath)
```

## I Want to Learn More

---

- An introduction about OSC plugins  
[http://en.opensuse.org/openSUSE:OSC\\_plugins](http://en.opensuse.org/openSUSE:OSC_plugins)
- OSC Collab. An excellent plugin from the GNOME team  
[https://en.opensuse.org/openSUSE:Osc\\_Collab](https://en.opensuse.org/openSUSE:Osc_Collab)  
<https://github.com/openSUSE/osc-plugin-collab>
- The Repo Checker plugging. Now with Staging  
<https://github.com/openSUSE/osc-plugin-factory>

End Note

# SUSE is hiring

---

## Join the Geeko!

Learn more about SUSE's openings, globally:

1. Talk to our colleagues at the booth
2. Check out our careers page [www.suse.com/careers](http://www.suse.com/careers)
3. Contact our recruiting team at [jobs@suse.com](mailto:jobs@suse.com)





# Thanks

---

Thank you for your attention.