
searchlight-ui Documentation

Release 9.0.0.0rc2.dev2

OpenStack Foundation

Sep 24, 2020

CONTENTS

1	Horizon Version Compatibility	1
2	Installation	2
3	Searchlight UI dashboard devstack plugin	3
4	Setup Local Dev Environment	4
5	Create and Install Local Package	6
6	Verifying JavaScript Features using npm	7
7	Contributing	8

HORIZON VERSION COMPATIBILITY

This project provides a Horizon plugin. The plugin relies on features developed in the Horizon framework, so the version of the plugin must be deployed with a compatible Horizon version. The below table provides an overview of the Horizon version compatibility matrix.

Searchlight UI Version	Horizon Version(s) Compatibility
2.x (Ocata)	11.x (Ocata)
1.x (Newton)	10.x (Newton)
0.1.0 (Mitaka)	9.x (Mitaka)

INSTALLATION

1. Package the searchlight_ui by running:

```
python setup.py sdist
```

This will create a python egg in the dist folder, which can be used to install on the horizon machine or within horizons python virtual environment.

2. Copy `_1001_project_search_panel.py` in `searchlight_ui/enabled` directory to `openstack_dashboard/local/enabled`.
3. (Optional) Copy the policy file into horizons policy files folder, and add this config POLICY_FILES:

```
'searchlight_ui': 'searchlight_ui',
```

4. Django has a compressor feature that performs many enhancements for the delivery of static files. If the compressor feature is enabled in your environment (`COMPRESS_OFFLINE = True`), run the following commands:

```
$ ./manage.py collectstatic  
$ ./manage.py compress
```

5. Finally restart your web server to enable searchlight-ui in your Horizon:

```
$ sudo service apache2 restart
```

SEARCHLIGHT UI DASHBOARD DEVSTACK PLUGIN

This directory contains the searchlight-ui devstack plugin.

To enable the plugin, add the following to your local.conf:

```
enable_plugin searchlight-ui <searchlight-ui GITURL> [GITREF]
```

where

<searchlight-ui GITURL> is the URL of a searchlight-ui repository [GITREF] is an optional git ref (branch/ref/tag). The default is master.

For example:

```
enable_plugin searchlight-ui https://opendev.org/openstack/searchlight-ui
```

This plugin also depends on the following services to be enabled: `horizon`, `searchlight-api`, and `searchlight-listener`.

- <https://opendev.org/openstack/searchlight/src/branch/master/devstack>

SETUP LOCAL DEV ENVIRONMENT

Assumptions:

- horizon is installed or cloned into <basedir>/horizon/
- searchlight-ui is installed or cloned into <basedir>/searchlight-ui/
- current working directory is <basedir>/horizon/

1. Install environment.:

```
#Locally cloned Horizon environment
# - environment that has migrated horizon to using tox (Ocata release)
.tox/runserver/bin/pip install -e ../searchlight-ui

# -environment that has not migrated to tox (pre-Ocata release)
./tools/with_venv.sh pip install -e ../searchlight-ui

#Devstack environment
pip install -e ../searchlight-ui
```

- #### 2. Copy <searchlight-ui>/searchlight_ui/enabled/_1001_project_search_panel.py to <horizon_dir>/openstack_dashboard/local/enabled/:

```
cp -rv ../searchlight-ui/searchlight_ui/enabled/_1001_project_search_
↪panel.py openstack_dashboard/local/enabled/
```

- #### 3. Set up the policy files. First copy the policy file <searchlight-ui>/searchlight_ui/conf/searchlight_policy.json into horizons policy files <horizon_dir>/openstack_dashboard/conf/ folder. Then copy <searchlight-ui>/local_settings.d/_1001_search_settings.py to <horizon_dir>/local/local_settings.d/:

```
cp ../searchlight-ui/searchlight_ui/conf/searchlight_policy.json ↪
↪openstack_dashboard/conf/

cp ../searchlight-ui/searchlight_ui/local_settings.d/_1001_search_
↪settings.py openstack_dashboard/local/local_settings.d/
```

- #### 4. (If offline compression is enabled - typical in production and devstack). Django has a compressor feature that performs many enhancements for the delivery of static files. It can be enable or disabled (COMPRESS_ENABLED). In addition, offline compression may be enabled or disabled (COMPRESS_OFFLINE = True). If offline compression is enabled in your environment, you must run the following commands the first time you install searchlight-ui and anytime you make changes to it.:

```
./manage.py collectstatic
./manage.py compress
```

5. Restart your horizon services.:

```
#Locally cloned Horizon environment (not under apache)
# - environment that has migrated horizon to using tox (Ocata release)
tox -e runserver 0.0.0.0:8005 (desired IP and port are optional)

# -environment that has not migrated to tox (pre-Ocata release)
./run_tests.sh --runserver 0.0.0.0:8005 (optionally set desired IP,
↳and port)

#Devstack
sudo service apache2 restart
```

CREATE AND INSTALL LOCAL PACKAGE

Change working directory to <basedir>/searchlight-ui/

1. Package the searchlight_ui by running:

```
python setup.py sdist
```

2. This will create a python egg in the dist folder, which can be used to install on the horizon machine or within horizons python virtual environment (prepend with ./tools/with_venv.sh):

```
pip install dist/searchlight-ui-0.0.0.tar.gz (use appropriate version)
```


VERIFYING JAVASCRIPT FEATURES USING NPM

The Node.js package manager *npm* can be used to verify JavaScript features in a development environment. The commands to do this include `npm run lint` and `npm run test`.

1. Install appropriate development packages for tox to run. In Ubuntu this may include, for example, `python-dev`. You can refer to the [Horizon documentations](#) for more details.
2. Run `npm install` to install the npm packages.

If those are successful, you should now be able to run `npm run lint` and `npm run test`.

When running the tests, if it appears to be using a non-current version of Horizon, you may want to clear your pip cache (`rm -rf <pip cache dir>/*`) and clear `.tox/venv` (`rm -rf .tox/*`). The pip cache is found at:

Linux and Unix: `~/.cache/pip` # respects the `XDGCACHEHOME` directory. OS X: `~/Library/Caches/pip` Windows: `<CSIDL_LOCAL_APPDATA>\pip\Cache`

CONTRIBUTING

If you would like to contribute to the development of OpenStack, you must follow the steps in documented at:

<https://docs.openstack.org/infra/manual/developers.html>

Once those steps have been completed, changes to OpenStack should be submitted for review via the Gerrit tool, following the workflow documented at:

<https://docs.openstack.org/infra/manual/developers.html#development-workflow>

Pull requests submitted through GitHub will be ignored.

Bugs should be filed on Storyboard, not GitHub:

https://storybook.openstack.org/#!/project_group/searchlight

Further documentation on feature request and bug report processes may be found here:

<https://docs.openstack.org/searchlight/latest/contributor/feature-requests-bugs.html>