

Code Run:

1. `git clone sugarlabs/sugar`
2. `cat README.md`
3. *Install Dependencies (such as, “`sudo apt-get install sugar-* -y`”)
4. `./autogen.sh`
5. `make`
6. `make install`
7. `python`
8. *Look for Unittest Documentation (<https://docs.python.org/2/library/unittest.html>)*
9. `>>> import unittest`
10. `>>> import PYTHON_FILE_FILENAME`
11. `>>> x = unittest.TestLoader().loadTestsFromTestCase(PYTHON_FILE_FILENAME.FILE_CLASSNAME)`
12. `>>> unittest.TextTestRunner(verbosity=2).run(x)`

test_activitieslist.py:

```
test_activity_list_palette (test_activitieslist.TestActivitiesList) ...
INFO:root:Try 1, name=Make favorite role_name=label
views/activitieslist.py:18: PyGIWarning: Gtk was imported without specifying a
version first. Use gi.require_version('Gtk', '3.0') before import to ensure
that the right version gets loaded.
    from gi.repository import Gtk
/usr/lib/python2.7/dist-packages/sugar3/graphics/icon.py:34: PyGIWarning: Rsvg
was imported without specifying a version first. Use gi.require_version('Rsvg',
'2.0') before import to ensure that the right version gets loaded.
    from gi.repository import Rsvg
/usr/lib/python2.7/dist-packages/sugar3/graphics/palettewindow.py:34:
PyGIWarning: SugarGestures was imported without specifying a version first. Use
gi.require_version('SugarGestures', '1.0') before import to ensure that the
right version gets loaded.
    from gi.repository import SugarGestures
/usr/local/lib/python2.7/dist-packages/jarabe/model/shell.py:21: PyGIWarning:
Wnck was imported without specifying a version first. Use
gi.require_version('Wnck', '3.0') before import to ensure that the right
version gets loaded.
    from gi.repository import Wnck
/usr/local/lib/python2.7/dist-packages/jarabe/model/shell.py:31: PyGIWarning:
SugarExt was imported without specifying a version first. Use
gi.require_version('SugarExt', '1.0') before import to ensure that the right
version gets loaded.
    from gi.repository import SugarExt
ERROR:root:ATTENTION: service_name property in the activity.info file is
deprecated, should be changed to bundle_id
ERROR:root:ATTENTION: service_name property in the activity.info file is
deprecated, should be changed to bundle_id
ERROR:root:ATTENTION: class property in the activity.info file is deprecated,
should be changed to exec
```

```

ERROR:root:ATTENTION: class property in the activity.info file is deprecated,
should be changed to exec
ERROR:root:ATTENTION: service_name property in the activity.info file is
deprecated, should be changed to bundle_id
ERROR:root:ATTENTION: class property in the activity.info file is deprecated,
should be changed to exec
Traceback (most recent call last):
  File "views/activitieslist.py", line 56, in <module>
    palette = ActivityListPalette(MockActivityInfo())
  File
"/usr/local/lib/python2.7/dist-packages/jarabe/desktop/activitieslist.py", line
666, in __init__
    registry = bundleregistry.get_registry()
  File "/usr/local/lib/python2.7/dist-packages/jarabe/model/bundleregistry.py",
line 716, in get_registry
    _instance = BundleRegistry()
  File "/usr/local/lib/python2.7/dist-packages/jarabe/model/bundleregistry.py",
line 116, in __init__
    self._load_hidden_activities()
  File "/usr/local/lib/python2.7/dist-packages/jarabe/model/bundleregistry.py",
line 215, in _load_hidden_activities
    with open(path) as file:
TypeError: coercing to Unicode: need string or buffer, NoneType found
INFO:root:Try 2, name=Make favorite role_name=label
INFO:root:Try 3, name=Make favorite role_name=label
INFO:root:Try 4, name=Make favorite role_name=label
INFO:root:Try 5, name=Make favorite role_name=label
INFO:root:Try 6, name=Make favorite role_name=label
INFO:root:Try 7, name=Make favorite role_name=label
INFO:root:Try 8, name=Make favorite role_name=label
INFO:root:Try 9, name=Make favorite role_name=label
INFO:root:Try 10, name=Make favorite role_name=label
INFO:root:Try 11, name=Make favorite role_name=label
INFO:root:Try 12, name=Make favorite role_name=label
^CERROR

```

```

=====
ERROR: test_activity_list_palette (test_activitieslist.TestActivitiesList)
-----
Traceback (most recent call last):
  File "test_activitieslist.py", line 27, in test_activity_list_palette
    self.assertIsNotNone(node)
  File "/usr/lib/python2.7/contextlib.py", line 35, in __exit__
    self.gen.throw(type, value, traceback)
  File "/usr/lib/python2.7/dist-packages/sugar3/test/unittest.py", line 60, in
run_view
    logging.debug(uitree.get_root().dump())

```

```
File "/usr/lib/python2.7/dist-packages/sugar3/test/uitree.py", line 72, in
dump
    self._crawl_accessible(self, 0, lines)
File "/usr/lib/python2.7/dist-packages/sugar3/test/uitree.py", line 171, in
_crawl_accessible
    self._crawl_accessible(child, depth + 1, lines)
File "/usr/lib/python2.7/dist-packages/sugar3/test/uitree.py", line 168, in
_crawl_accessible
    lines.append("  " * depth + str(node))
File "/usr/lib/python2.7/dist-packages/sugar3/test/uitree.py", line 138, in
__str__
    return "[%s | %s]" % (self.name, self.role_name)
File "/usr/lib/python2.7/dist-packages/sugar3/test/uitree.py", line 96, in
role_name
    return self._accessible.get_role_name()
Error: atspi_error: Did not receive a reply. Possible causes include: the
remote application did not send a reply, the message bus security policy
blocked the reply, the reply timeout expired, or the network connection was
broken. (1)
```

```
-----
Ran 1 test in 62.211s
```

```
FAILED (errors=1)
<unittest.runner.TextTestResult run=1 errors=1 failures=0>
```

test_backup.py:

```
test_backend_backup_restore (test_backup.TestBackup) ... ERROR:root:FOUND
BACKEND Local Device Backup
ok
test_get_backend (test_backup.TestBackup) ... ERROR:root:FOUND BACKEND Local
Device Backup
ok
test_need_stop_activities (test_backup.TestBackup) ... ERROR:root:FOUND BACKEND
Local Device Backup
ok
```

```
-----
Ran 3 tests in 0.043s
```

```
OK
<unittest.runner.TextTestResult run=3 errors=0 failures=0>
```

test_desktop_config.py:

```
test_set_views (test_desktop_config.TestDesktopConfig) ... ok
test_unset_views (test_desktop_config.TestDesktopConfig) ... ok
```

Ran 2 tests in 0.019s

OK

<unittest.runner.TextTestResult run=2 errors=0 failures=0>

test_downloader.py:

test_download (test_downloader.TestDownloader) ... 127.0.0.1 - - [30/Sep/2018
18:43:56] "GET /data/test.txt HTTP/1.1" 200 -

ok

test_download_to_temp (test_downloader.TestDownloader) ... 127.0.0.1 - -
[30/Sep/2018 18:43:57] "GET /data/test.txt HTTP/1.1" 200 -

ok

test_get_size (test_downloader.TestDownloader) ... 127.0.0.1 - - [30/Sep/2018
18:43:57] "HEAD /data/test.txt HTTP/1.1" 200 -

ok

Ran 3 tests in 1.534s

OK

<unittest.runner.TextTestResult run=3 errors=0 failures=0>

test_webservice.py Tests:

test_get_account (test_webservice.TestWebAccounts) ... ok

test_get_active_accounts (test_webservice.TestWebAccounts) ... ok

test_get_all_accounts (test_webservice.TestWebAccounts) ... ok

test_get_configured_accounts (test_webservice.TestWebAccounts) ... ok

test_get_description (test_webservice.TestWebAccounts) ... ok

test_get_service (test_webservice.TestWebAccounts) ... ok

test_get_webaccount_services (test_webservice.TestWebAccounts) ... ok

test_icon_theme (test_webservice.TestWebAccounts) ... ok

test_refresh_menu (test_webservice.TestWebAccounts) ... ok

test_share_menu (test_webservice.TestWebAccounts) ... ok

Ran 10 tests in 0.002s

OK

<unittest.runner.TextTestResult run=10 errors=0 failures=0>

Testing Process:

SugarLabs provided multiple testing files for us to use. While the files were there, there was not much explanation on how to use the tests. We were able to open the files and run them, but there was no feedback on what they were actually testing. From there we went to Python's unittest documentation to see how we can use these files. Based on

that we were able to get most files tested and got working output as shown above. Some of the files did give us a few errors.

Overall the project is very organized with some help on building the project as given, but there was some lacking information on testing what they've made. The developers were prepared for testing everything they needed to have tested, but only for themselves. It would have been more helpful for others looking in if there was a file that would automate the testing or at least more information on how to set up each individual test. We look forward to creating our own tests and exploring in more detail this project