**User Stories: Implement OAuth scopes in APIv2**

**Rusncopes:**

* **OAuth Applications: osf.full\_read, osf.full\_write:** [**https://www.runscope.com/radar/3v895if0psf0/c4fe782e-a496-40c5-929c-7485e89ef5a6/overview**](https://www.runscope.com/radar/3v895if0psf0/c4fe782e-a496-40c5-929c-7485e89ef5a6/overview)
* **OAuth Files: osf.full\_read, osf.full\_write:** [**https://www.runscope.com/radar/3v895if0psf0/4dbda50d-e6fe-459f-a020-c7f6eba273dd/overview**](https://www.runscope.com/radar/3v895if0psf0/4dbda50d-e6fe-459f-a020-c7f6eba273dd/overview)
* **OAuth Nodes: osf.full\_read, osf.full\_write:** [**https://www.runscope.com/radar/3v895if0psf0/215624f3-ca0b-45ef-8c8c-902afa433d86/overview**](https://www.runscope.com/radar/3v895if0psf0/215624f3-ca0b-45ef-8c8c-902afa433d86/overview)
* **OAuth Users: osf.full\_read, osf.full\_write:** [**https://www.runscope.com/radar/3v895if0psf0/52e176e0-9b17-4f83-9035-56c50c1b5850/overview**](https://www.runscope.com/radar/3v895if0psf0/52e176e0-9b17-4f83-9035-56c50c1b5850/overview)

**Private Project for testing:** [**https://staging.osf.io/q7cux/**](https://staging.osf.io/q7cux/)

**Contributors:** API Test (read access), API W. Test (read-write access), Ultimate User (admin)

**Test 1** - *API views intended for internal usage only (such as `/applications/` and `/applications/<app\_id>`) are not accessible when making an oauth request (via token)*

* Generate scoped access tokens (one for osf.full\_read and one for osf.full\_write) for a user. Create a private project/node.
* With these tokens, attempt to access the node endpoints: /applications/` and `/applications/<app\_id>`. Make sure that access fails.

**Test 2** - *A user whose token has the read scope (only) can read data (including data private to their account), but not edit (write) data.*

Private project: <https://staging.osf.io/q7cux/>

* Use the read token you created in Test 1 for this test. The user should be a contributor on private project:<https://staging.osf.io/q7cux/>. Ping me to add the user to the project, although Ultimate User (id #y352a) is an admin on the project and can add anyone to the project.
* With the read token, make sure you **can** access/read node detail view (metadata) for the private project: <https://staging-api.osf.io/v2/nodes/q7cux/>
* With the read token, make sure you **cannot** write node detail view (metadata) for the private project: <https://staging-api.osf.io/v2/nodes/q7cux/>

**Test 3** - *A user whose token includes the write scope (only) can read AND write data. (write access implicitly grants read permissions) (includes data private to the user’s account).*

* Use the write token you created in Test 1 for this test. The user should be an **admin** contributor on private project:<https://staging.osf.io/s3jp6/> . Ping me to add the user to the project, although Ultimate User (id #y352a) is an admin on the project and can add anyone to the project.
* With the write token, make sure you can access node detail view for the private project: <https://staging.osf.io/s3jp6/>
* With the write token, make sure you can edit the private project: <https://staging.osf.io/s3jp6/>

**Test 4** - *It should not be possible to get an access token for a scope that doesn’t exist.*

* Make sure you cannot generate access token for fake scope: osf.qaisawesome. FYI: This is something that will probably only need to be tested once, and it may not be possible to do so automatically in runscope.

**Test 5** - *The user’s token should not grant access to data that belongs to another user’s account*

* Use the same read and write access tokens above for testing access to a private project (that the user with the tokens is not a contributor on): <https://staging.osf.io/fs9kp/>
* With the read token, make sure you **cannot** access/read node detail view for the private project: <https://staging-api.osf.io/v2/nodes/fs9kp/>
* With the write token, make sure you **cannot** access/read node detail view for the private project: <https://staging-api.osf.io/v2/nodes/fs9kp/> (should also not be able to edit this private project)

**Test 6 -** *APIv1 should not respond to OAuth2 tokens at all (only apiv2 is allowed for oauth). All such requests should be rejected.*

* Use the same read and write access tokens generated above to access user list on APIv1
* Make sure this fails.

**Test 7** *- Verify that files can be read and edited directly off of waterbutler.*

**Private Project (Is a contributor) for testing:** [**https://staging.osf.io/q7cux/**](https://staging.osf.io/q7cux/)

**Contributors:** API Test (read access), API W. Test (read-write access), Ultimate User (admin)

* Use the same read token generated above to test this.
* With the read token, make sure you can access/read files for the private project: <https://staging-api.osf.io/v2/nodes/q7cux/files/>
* With the read token, make sure you can download files for the private project: <https://staging-api.osf.io/v2/nodes/q7cux/files/>
* With the write token, make sure you can **create, rename, upload, delete** files for the private project for which the user with the token IS a contributor: <https://staging.osf.io/s3jp6/>

**Public Project (Is a contributor) for testing:** <https://staging.osf.io/w4c6f/>

**Contributors:** API Test (read access), API W. Test (read-write access), Ultimate User (admin)

* Use the read token you created in Test 1 for this test. The user should be a contributor on public project:<https://staging.osf.io/w4c6f/>. Ping me to add the user to the project, although Ultimate User (id #y352a) is an admin on the project and can add anyone to the project.
* With the same read token generated for the user in Test 1, make sure you can access/read files for the public project: <https://staging-api.osf.io/v2/nodes/w4c6f/files/>
* With the read token, make sure you can download files for the public project: <https://staging-api.osf.io/v2/nodes/w4c6f/files/>

**Private Project (Is not a contributor) for testing:** <https://staging.osf.io/fs9kp/>

* Use the same read token generated in Test 1 for this test. The user should not be a contributor on this private project: <https://staging.osf.io/fs9kp/>
* With the read token, make sure you cannot access/read files for the private project: [https://staging-api.osf.io/v2/nodes/](https://staging-api.osf.io/v2/nodes/q7cux/files/)[fs9kp](https://staging.osf.io/fs9kp/)[/files/](https://staging-api.osf.io/v2/nodes/q7cux/files/)
* With the read token, make sure you cannot download files for the private project: [https://staging-api.osf.io/v2/nodes/](https://staging-api.osf.io/v2/nodes/q7cux/files/)[fs9kp](https://staging.osf.io/fs9kp/)[/files/](https://staging-api.osf.io/v2/nodes/q7cux/files/)
* With the write token, make sure you cannot read or download files for the private project for which the user with the token is a contributor: [https://staging-api.osf.io/v2/nodes/](https://staging-api.osf.io/v2/nodes/q7cux/files/)[fs9kp](https://staging.osf.io/fs9kp/)[/files/](https://staging-api.osf.io/v2/nodes/q7cux/files/)

**Public Project (Is a not contributor) for testing:** <https://staging.osf.io/fwy82/>

* Use the read token you created in Test 1 for this test. The user should not be a contributor on this public project: <https://staging.osf.io/fwy82/>.
* With the same read token generated for the user in Test 1, make sure you can access/read files for the public project: [https://staging-api.osf.io/v2/nodes/](https://staging-api.osf.io/v2/nodes/w4c6f/files/)[fwy82](https://staging.osf.io/fwy82/)[/files/](https://staging-api.osf.io/v2/nodes/w4c6f/files/)
* With the read token, make sure you can download files for the public project: [https://staging-api.osf.io/v2/nodes/](https://staging-api.osf.io/v2/nodes/w4c6f/files/)[fwy82](https://staging.osf.io/fwy82/)[/files/](https://staging-api.osf.io/v2/nodes/w4c6f/files/)
* With the write token, make sure you can read and download files for the public project for which you are not a contributor: <https://staging-api.osf.io/v2/nodes/w4c6f/files/>

**Test 8** - Make sure you cannot generate tokens for a deactivated user or an unconfirmed user.