

ASWWU Web API Testing

Stephen Ermshar

CPTR 498

2020 June 02

Outline

1. Initial Proposal
2. Modified Proposal
3. Project Structure
4. Challenges
5. Project Outcome

Initial Proposal

- Replace Tornado with Django for handling HTTP requests.
- Replace SQLite with PostgreSQL.
- Replace SQLAlchemy with Django ORM.

Goals

- Tighter integration between technologies.
- Cleaner code, focusing more on API behavior.
- Standard, well-supported, "Google-able" solutions.

Modified Proposal

“ Add Automated API tests to support future progress on the initial proposal. ”

- Allows new developers to focus on learning one thing at a time.
- Sets expectations for how the API behaves, without prior knowledge.
- Can be independent of the server setup
 - useful regardless of future technologies ASWWU Web embraces
 - eases transitions to new technologies.

Project Structure

```
tests/  
├── aswwu  
│   ├── behaviors/  
│   │   ├── jobs/  
│   │   │   ├── jobs_data.py  
│   │   │   ├── jobs_requests.py  
│   │   │   ├── jobs_subtests.py  
│   │   │   └── test_jobs.py  
│   │   └── ...  
│   └── data/ (users.py, elections.py, ...)  
└── conftest.py, utils.py
```

Project Structure

1. Copy Databases into Temporary Directory `tmp`.
2. Clean Databases, empty non-Database contents of `tmp`.
3. Start Testing Server in a new Thread.
4. Run a single behavior test.
5. Stop the Testing Server and Thread.
6. Continue from (2).

Typical Structure of a Single Test

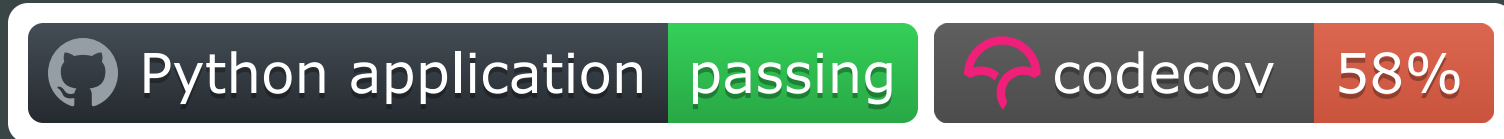
All server test interactions are through the API.

1. SAML requests generate required users for a behavior.
2. If necessary users profiles and roles are updated.
3. Users login with separate session objects if authentication is required for a behavior.
4. Users perform requests
5. Results are checked with assertions.
 - repeated actions are run in subtests.

Challenges

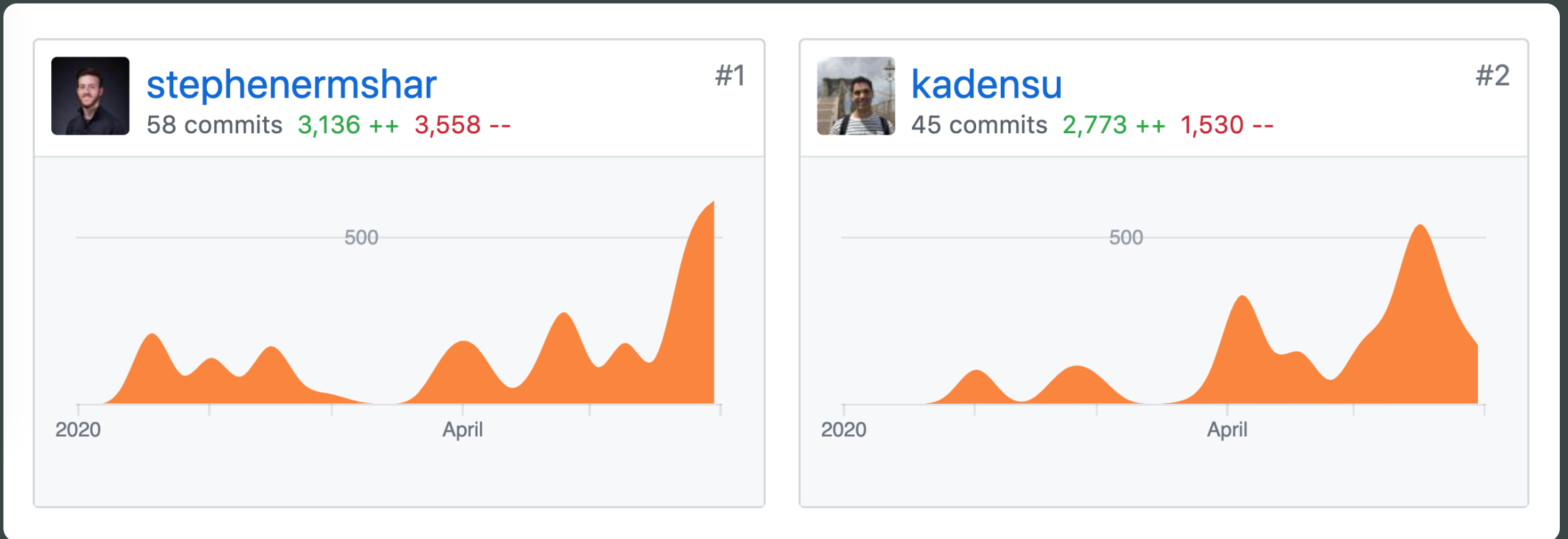
- test user authentication, SAML simulation.
- Performing time based requests.
- Handling file data.
- Balancing low code repetition with low complexity.
- Getting Tornado to run in a separate thread from pytest.
- Unicode support.
- Time management.

Project Outcome



- Tests written for all major projects, except *Pages* which will likely be replaced.
- Deterministic Build Environment added with Pipenv.
- Tests do not rely on server code, and run entirely over the API. (with the exception of one use of SQLAlchemy at startup)
- Removed around 700 lines of dead code. ([#181](#), [#182](#))
- Next: Python 3 and Docker.

Stats



LOC contributed between 2020-01-01 and 2020-06-02.

Stats

```
# https://stackoverflow.com/questions/4589731/git-blame-commit-statistics
% git ls-tree -r -z --name-only HEAD -- ./(tests|scripts|*.py) \
    -- ./src/aswwu/application.py \
    | xargs -0 -n1 git blame --since=2020-01-01 --line-porcelain HEAD \
    | grep "^author " | sort | uniq -c | sort -nr

1296 author Stephen Ermshar
1198 author kadensu
```

LOC in working directory contributed by author since 2020-01-01, excluding most `src/` code and auxiliary files (readme, pipfile, etc.).

Stats

```
collected 47 items
```

```
tests/aswwu/behaviors/auth/test_auth.py ... [ 6%]  
tests/aswwu/behaviors/elections/ballot/test_ballot.py .... [ 14%]  
tests/aswwu/behaviors/elections/candidate/test_candidate.py ..... [ 25%]  
tests/aswwu/behaviors/elections/election/test_election.py ..... [ 38%]  
tests/aswwu/behaviors/elections/position/test_position.py .... [ 46%]  
tests/aswwu/behaviors/elections/vote/test_vote.py ..... [ 57%]  
tests/aswwu/behaviors/jobs/test_jobs.py ..... [ 80%]  
tests/aswwu/behaviors/mask/test_mask.py ..... [100%]
```

PyTest Output from the most recent merge. Each . is a test, percentage refers to the percent of test functions run, not coverage.

Stats

file	coverage
base_handlers.py	81.05%
route_handlers/elections.py	91.34%
route_handlers/forms.py	60.90%
route_handlers/mask.py	67.74%
route_handlers/pages.py	10.70%

Resources

https://github.com/aswwu-web/python_server

#138 +42 -16	#149 +65 -94	#165 +59 -25	#176 +536 -3
#140 +1,659 -224	#151 +1,659 -224	#167 +382 -329	#178 +701 -195
#141 +22 -0	#152 +382 -848	#169 +92 -23	#181 +19 -652
#142 +1,373 -75	#163 +651 -85	#173 +421 -331	#182 +0 -45
#145 +141 -154	#164 +110 -64	#174 +271 -110	

Thanks to *Kaden Sukachevin* for writing tests in the Elections Project, and *Professor Carman* for weekly guidance and suggestions.