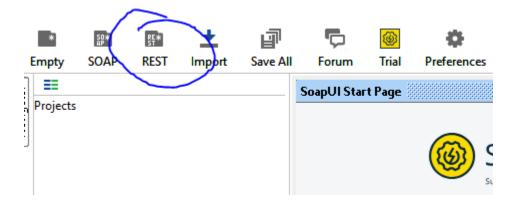
```
Install soapUI from https://www.soapui.org/downloads/soapui/
Install python dependencies
`pip install -r requirements.txt`

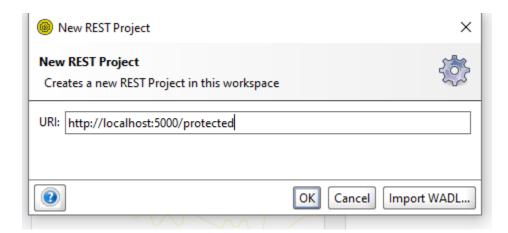
Run the flask server
`python3 api_service/app.py `
Generate jwts using the pyJWT library
`python jwt_generator/jwt_utils.py`

These tokens will be written a text file called 'jwt_tokens.txt' and be valid for an hour
```

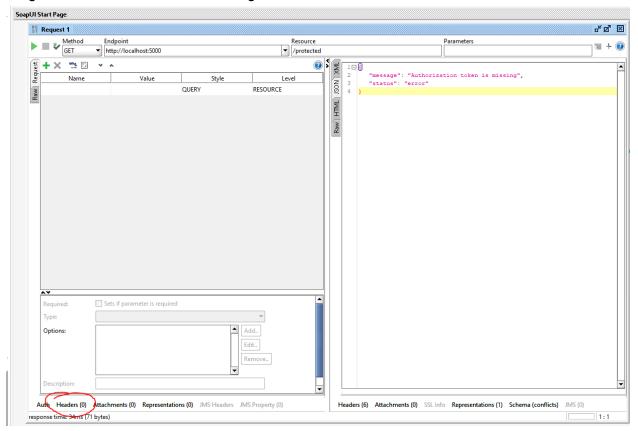
When you open soapUI click rest



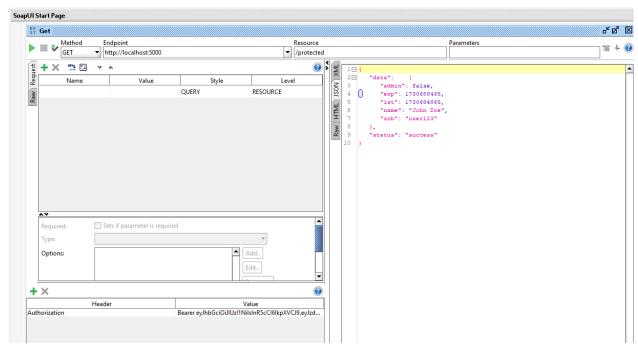
Then go ahead and enter our protected endpoint being served by our local app server.



For testing's sake, if you try to run a get request on this endpoint without adding a header you will get an "authorization token missing error."



If you add the authorization token, and send another get request we can see that pyJWTs created token was successfully authorized.



Steps: Ensure that no whitespace is in the header or values or you will get an incorrect token value.

If you are having trouble with this, take a look at this curl request

curl -H "Authorization: Bearer

eyJhbGciOiJIUzI1NiIsInR5cCl6lkpXVCJ9.eyJzdWliOiJ1c2VyMTlzliwibmFtZSl6lkpvaG4gRG9lliwiYWRtaW4iOmZhbHNlLCJpYXQiOjE3MzA2ODQ4NjUsImV4cCl6MTczMDY4ODQ2NX0.nBXE6Wyzn-NXUmHaYZaLajEtfLA1lG5tdpPrPWVnPGE" http://127.0.0.1:5000/protected

Everything highlighted in green is what is set in our value parameter, including all white space.

Next steps:

- Security Testing: Leverage SoapUI to conduct thorough security assessments, including testing for common vulnerabilities outlined by the OWASP Top 10, such as SQL injection, XSS, and improper authentication handling.
- Functional Testing: Validate that all API endpoints perform expected functions under various conditions and handle errors gracefully.
- Load and Performance Testing: Analyze how the application performs under stress, using an attempt to check if SoapUI can simulate high traffic and data load for a local server

- Mocking and Service Virtualization: Implement mocks for external services to test the API in isolation, ensuring functionality is preserved even when external dependencies are not available.

Currently to run endpoint tests you can go to test/security tests and run it.

This will run tests on any endpoints included in endpoints.json

As shown this currently is running a test on the protected endpoint in which we secured with pyJWT.