

Attempt 1



2/18/2024

NEXT UP: Review Feedback

Attempt 1 Score:

N/A



Add Comment

Unlimited Attempts Allowed

1/15/2024 to 2/25/2024

Details

# Implementing a basic JWKS Server

## Objective

Develop a RESTful JWKS server that provides public keys with unique identifiers (kid) for verifying JSON Web Tokens (JWTs), implements key expiry for enhanced security, includes an authentication endpoint, and handles the issuance of JWTs with expired keys based on a query parameter.

Chooses an appropriate language and web server for the task.

Due to the simplicity of this assignment, I would prefer you complete it with an unfamiliar language... but as I have no way to verify it, it's not considered part of the rubric.


This project is for educational purposes. In a real-world scenario, you'd want to integrate with a proper authentication system and ensure security best practices.

## Background

1. [HTTP/web services](https://en.wikipedia.org/wiki/HTTP) <https://en.wikipedia.org/wiki/HTTP>
  - o Familiarize yourself with client/server HTTP services.
2. [REST](https://en.wikipedia.org/wiki/REST) <https://en.wikipedia.org/wiki/REST>
  - o Familiarize yourself with correct HTTP methods/headers/status codes for RESTful APIs.
3. [JOSE: JWT](https://en.wikipedia.org/wiki/JSON_Web_Token) [https://en.wikipedia.org/wiki/JSON\\_Web\\_Token](https://en.wikipedia.org/wiki/JSON_Web_Token) , [JWK \(and JWKS\):](https://datatracker.ietf.org/doc/rfc7517/) <https://datatracker.ietf.org/doc/rfc7517/>
  - o Familiarize yourself with the concepts of JWT, JWK.
  - o Understand the importance of key expiry, and `kid`.

## Requirements

1. **Key Generation**
  1. Implement RSA key pair generation.
  2. Associate a Key ID (`kid`) and expiry timestamp with each key.
2. **Web server with two handlers**
  1. Serve HTTP on port 8080
  2. A RESTful JWKS endpoint that serves the public keys in JWKS format.

1. Only serve keys that have not expired.
3. A `/auth` endpoint that returns an unexpired, signed JWT on a POST request.
  1. If the “expired” query parameter is present, issue a JWT signed with the expired key pair and the expired expiry.
3. **Documentation**
  1. Code should be organized.
  2. Code should be commented where needed.
  3. Code should be linted per your language/framework.
4. **Tests**
  1. Test suite for your given language/framework with tests for you.
  2. Test coverage should be over 80%.
5. **Blackbox testing**
  1. Ensure the **included test client**  (<https://github.com/jh125486/CSCE3550/releases>) functions against your server.
  2. The testing client will attempt a POST to `/auth` with no body. There is no need to check authentication for this project.
    1. **NOTE:** We are not actually testing user authentication, just mocking authentication and returning a valid JWT for this user

**Note:**

Using kid in JWKS is crucial for systems to identify which key to use for JWT verification. Ensure that the JWTs include the kid in their headers and that the JWKS server can serve the correct key when requested with a specific kid.


## Expected Outcome

At the end of the project, you should have a functional JWKS server with a RESTful API that can serve public keys with expiry and unique `kid` to verify JWTs.

The server should authenticate fake users requests, issue JWTs upon successful authentication, and handle the “expired” query parameter to issue JWTs signed with an expired key.

This project should take 1-12 hours, depending on your familiarity with your chosen language/framework, and web servers in general.

## Deliverables

- Provide a link to your GitHub repo containing your code.
  - Include in the repo a screenshot of the **test client**  (<https://github.com/jh125486/CSCE3550/releases>) running against your server.
  - Include in the repo a screenshot of your test suite (if present) showing the coverage percent.

As always with every screenshot, please include identifying information.

# Project 1

Criteria	Ratings		Pts
Valid JWT authentication <a href="#">view longer description</a>	15 pts Full Marks	0 pts No Marks	/ 15 pts
Expired JWT authentication <a href="#">view longer description</a>	5 pts Full Marks	0 pts No Marks	/ 5 pts
Proper HTTP Methods And Status Codes <a href="#">view longer description</a>	10 pts Full Marks	0 pts No Marks	/ 10 pts
Valid JWK Found In JWKS <a href="#">view longer description</a>	20 pts Full Marks	0 pts No Marks	/ 20 pts
Expired JWK Not Found In JWKS <a href="#">view longer description</a>	10 pts Full Marks	0 pts No Marks	/ 10 pts
Expired JWK is expired <a href="#">view longer description</a>	5 pts Full Marks	0 pts No Marks	/ 5 pts
Test suite is present <a href="#">view longer description</a>	15 pts Full Marks	0 pts No Marks	/ 15 pts
Test coverage <a href="#">view longer description</a>	5 pts Full Marks	0 pts No Marks	/ 5 pts
Documentation/Linting /Organization <a href="#">view longer description</a>	15 pts Full Marks	0 pts No Marks	/ 15 pts

Total Points: 0

<https://github.com/andym1125/jwt-assignment>

< Previous

(<https://unt.instructure.com/courses/99447/modules/items/6024606>)

New Attempt

Next >

(<https://unt.instructure.com/courses/99447/modules/items/6024608>)