Java Assignment AuthServer

This is a simple authorization server that's intended for securing the APIs implemented by your candidate assignment solution.

The server is very simple and exposes only two endpoints:

- token endpoint: this endpoint is used to obtain the JWT-encoded access token to be used to access the secured APIs in your candidate assignment solution
- JWK set endpoint: this endpoint is used to obtain the public key (in JWK format) that should be used to verify the signature of the access token obtained from the token endpoint

Note that the key used for signing the access token is created automatically during server startup.

Running the Server

In the ZIP archive containing this document you will also find the server runtime (in `lib` directory) as well as OS-specific start scripts (in `bin` directory). Java 11 (at minimum) is required to run the server.

The server can be started by simply running the following command:

```
$ ./bin/java-assignment-authserver
```

By default, the server listens for HTTP traffic on port `8888`. If needed, the port can be changed using the `--server.port` argument.

For example, to run the server on port `8080` use the following command:

```
$ ./bin/java-assignment-authserver --server.port 8080
```

Usage

Obtaining the Access Token

To obtain the access token from the token endpoint, use the following request:

```
$ curl -s http://localhost:8888/token | jq
{
  "access_token":
"eyJraWQiOiJWMktjaXRKUHBkSXNnWkZxcVoyUGpyOUdROWRFOEFMQmhLZzdlZnczLUgwIiwiYWxnIjoiUlMyN
TYifQ.eyJzdWIiOiJhbGljZSIsImF1ZCI6Imh0dHBzOlwvXC9pbmZpbnVtLmNvbSIsImlzcyI6Imh0dHBzOlwv
XC9pbmZpbnVtLmNvbSIsImV4cCI6MTYxNDc4NDMyMywiaWF0IjoxNjE0NzgzNzIzLCJqdGkiOiIzT2c1N0lOdV
9vVDdyTGw3Qkk4OTZkYS1YeVVkSE04NkVmZ3VMZmxNNlUwIiwiY2xpZW50X2lkIjoiOS1haG9DSGp0akNuRGQ5
R2hHUk4xdExIU3FhQTE2SDlRckNzS1pPaEVwdyJ9.Lgm7iNX8buRJ191_kJIRB8XV4JeL9RiH7NdDv_jzVrCcR
6g0QadebXuydygr0M0HWMuko1J6c2l9CISv7buuk_MJuZZj3X-E7eeasTWKkXyFXed-
4ndv79hywUU4jfJermXAFhLn3pqUTpqyM5Jbib5aHxKCRq1y76oD2lCMMIHwkElMYMCWb8DnAzr5yei8JlYHa4
H47ndUzv1918cA8uWvYwFe2xztiYkSLrrmeygV0tSiUYOGqeNobk9wNQtC-4uxPJ7I0ncJxjTei7kk0AVaNW-
UjMnJj4azWH8w2C2FYpr7qJml-NKl_c_q6a81T4V4iTwuDW0GNWUDXEAMnw",
  "token_type": "Bearer",
  "expires_in": 600
}
```

As previously mentioned, the access token is JWT-encoded. To inspect its claims a service like jwt.io can be used.

The access token from previous example has the following header:

```
{
  "kid": "V2KcitJPpdIsgZFqqZ2Pjr9GQ9dE8ALBhKg7efw3-H0",
  "alg": "RS256"
}
```

While the payload is:

```
{
  "sub": "alice",
  "aud": "https://infinum.com",
  "iss": "https://infinum.com",
  "exp": 1614784323,
  "iat": 1614783723,
  "jti": "3Og57INu_oT7rLl7BI896da-XyUdHM86EfguLflM6U0",
  "client_id": "9-ahoCHjtjCnDd9GhGRN1tLHSqaA16H9QrCsKZOhEpw"
}
```

Note that, by default, the issued access token is valid for 10 minutes and has no scope attached to it. If an access token with a specific scope is required, it can be obtained by specifying the `scope` parameter in the request to the token endpoint:

```
$ curl -s http://localhost:8888/token?scope=admin | jq
{
  "access_token":
"eyJraWQiOiJWMktjaXRKUHBkSXNnWkZxcVoyUGpyOUdROWRFOEFMQmhLZzdlZnczLUgwIiwiYWxnIjoiUlMyN
TYifQ.eyJzdWIiOiJib2IiLCJhdWQiOiJodHRwczpcL1wvaW5maW51bS5jb20iLCJzY29wZSI6ImFkbWluIiwi
aXNzIjoiaHR0cHM6XC9cL2luZmludW0uY29tIiwiZXhwIjoxNjE0Nzg0NDEyLCJpYXQiOjE2MTQ3ODM4MTIsIm
p0aSI6IjBWWGlnYUpFSkZNakttd2l4cHF4UkV3cXhEd2hTejlMTHhGcW9HcEY0Q3ciLCJjbGllbnRfaWQiOiJO
UHl6YnhLV3hVM3Jpbk5tX2ZiaGQyMl85U2hLQklhMDVyT0wwb3VPa1Y0In0.Ids5SEBfHCYdhN2SXBql3I7k0v
A8G9Wl-
HZLzY2KU9cSmRvxxA4oVhE4673l4wvdV7rnlf884rr6bwlbb0Vd1cR0IzAaH55wl6Ao4yZmZD2XjUDRwOh80Nh
QAQR99WChh8ZjU1xKhvEWa2ut2kAme3mpozEgMgQtkA-v4BvDrKLDq8E93B8B5-
fShnRvCzdL4wqAiARW2Xn5v2xANUzbK5khqxbNrGn99I3lTQEW2RTkp2558F50Xyz3-1gOr_Do_tmGbFyL4I-
V4J_rth6mPGtspby-rflON1VIrQmgE-s8Kx-x9-Cue29reSCik9fxAL42N5ZNdicJofgJSqjfHw",
  "scope": "admin",
  "token_type": "Bearer",
  "expires_in": 600
}
```

In this case the access token has the following payload:

```
{
  "sub": "bob",
  "aud": "https://infinum.com",
  "scope": "admin",
  "iss": "https://infinum.com",
  "exp": 1614784412,
  "iat": 1614783812,
  "jti": "0VXigaJEJFMjKmwixpqxREwqxDwhSz9LLxFqoGpF4Cw",
  "client_id": "NPyzbxKWxU3rinNm_fbhd22_9ShKBIa05rOL0ouOkV4"
}
```

Verifying the Access Token Signature

JWT signature should be verified using the appropriate public key. The key currently used for signing access tokens is available on the JWK set endpoint:

```
$ curl -s http://localhost:8888/jwks.json | jq
{
  "keys": [
    {
      "kty": "RSA",
      "e": "AQAB",
      "use": "sig",
      "kid": "V2KcitJPpdIsgZFqqZ2Pjr9GQ9dE8ALBhKg7efw3-H0",
      "alg": "RS256",
      "n":
"ii4XucxegBxk3FV037bLeYQiUtW8KC5PJUxgx0RulGvwKxyOvca31ZEBnBtz33oz5rmfSC62v5WEAgARG5gNv
bejPTUy0tjrnspLVmyaGPWrmfPLsZiLHTlwfuWoWy0zuEecPlZT9RGZDViJ3hRs9EEHyUuZ2jJw3puvGdFZzge
lxxxv4SBmxPkC8k2vD35EYfuMSUrQfLfptG05df9uWpig6GSmIOATL1jk3mnuoyrnkv1CQyK44npt_ZQgAP_Oz
LIf50VVtM8mMiIObee4rQI6nP2ZUgQviwdZ309ymI_VDDlCNbXUPwHHeClOSwNqHQnYxB3f70ODMbOLhxp70Q"
    }
  ]
}
```