Implementacja mechanizmów autoryzacji

Laboratorium 5

Wymagania

1. Środowisko programistyczne umożliwiające przygotowanie aplikacji w frameworku Spring Boot. Alternatywnie dowolne środowisko programistyczne umożliwiające przygotowanie aplikacji implementującej interfejs REST API.
2. Narzędzie pełniące rolę klienta HTTP (np. cURL, Postman, Insomnia).

Przydatne źródła

1. [link] – RFC 6749 opisujące framework *OAuth2.0*.
2. [link] – RFC 7617 opisujące framework *Basic Authentication*.
3. [link] – opis wykorzystania przepływu w ramach frameworku *OAuth2.0 Client Credentials Grant*.

Wstęp

W trakcie trwania procesu projektowania aplikacji można skorzystać z wielu mechanizmów uwierzytelniania i autoryzacji. Decyzja o tym, który z nich wybrać zależy od szczegółów i wymagań projektu i może wynikać między innymi ze sposobu i kontekstu komunikacji. Jednym ze sposobów zapewnienia autoryzacji dostępu do interfejsów jest metoda wykorzystująca Basic Authentication, kolejna to autoryzacja w oparciu o framework Auth2.0, który wykorzystuje tokeny dostępu np. tokeny JWT. Na potrzeby poniższych zadań w przypadku frameworku OAuth2.0 zaleca się skorzystanie z przepływu OAuth2.0 Client Credentials Grant.

Zadanie 1

1. Na podstawie specyfikacji frameworku OAuth2.0 oraz metody autoryzacji opartej o Basic Authentication opisać wymienione mechanizmy.
2. Przedstawić wady i zalety Basic Authentication oraz OAuth2.0.
3. Zaproponować sytuacje, w której odpowiednie wydaje się wykorzystanie metody Basic Authentication oraz uzasadnić scenariusz, w którym warto skorzystać z OAuth2.0.

Zadanie 2

Korzystając z interfejsów przygotowanych w trakcie laboratorium 7 oraz z ich specyfikacji zaimplementować autoryzacje w oparciu o Basic Authentication. Autoryzacja powinna dotyczyć endpointów, które zostały opisane w specyfikacji jako te, do których dostęp zabezpieczony jest właśnie tą metodą.

1. Zaimplementować autoryzacje w oparciu o Basic Authentication. Dane logowania zgodnie ze wzorcem [numer_albumu]/123456. np. bm52123/123456.
2. Wykonać próbne zapytanie ze złymi danymi logowania i zaprezentować, że zostało ono odrzucone oraz odpowiedź systemu jest zgodna ze specyfikacją.
3. Wykonać poprawne zapytanie z prawidłowymi danymi logowania i zaprezentować, że zostało ono przetworzone oraz odpowiedź systemu jest zgodna ze specyfikacją.
4. Implementacje załączyć do sprawozdania w postaci kodu źródłowego. Dodatkowo zaprezentować przebieg opisanych kroków w sprawozdaniu z laboratorium.

Zadanie 3

Zaimplementować autoryzacje w oparciu o mechanizm OAuth2.0 Client Credentials Grant. Implementacja powinna wykorzystywać tokeny JWT generowane przez service autoryzacyjny dostępny pod wskazanym adresem URL: https://pba-auth-server.herokuapp.com. Do weryfikacji tokenów JWT należy wykorzystać certyfikat klucza publicznego znajdujący się poniżej.

Certyfikat klucza publicznego w formacie X509:

```
-----BEGIN CERTIFICATE-----
MIIDQDCCAiigAwIBAgIEX8EtRzANBgkqhkiG9w0BAQsFADBiMQswCQYDVQQGEwJQ
TDELMAkGA1UECAwCWlMxETAPBgNVBAcMCFN6Y3plY2luMQswCQYDVQQKDAJXSTEM
MAoGA1UECwwDWlVUMRgwFgYDVQQDDA9QQkEgQVVUSCBTRVJWRVIwHhcNMjAxMTI3
MTY0NTU5WhcNMjExMTI3MTY0NTU5WjBiMQswCQYDVQQGEwJQTDELMAkGA1UECAwC
WlMxETAPBgNVBAcMCFN6Y3plY2luMQswCQYDVQQKDAJXSTEMMAoGA1UECwwDWlVU
MRgwFgYDVQQDDA9QQkEgQVVUSCBTRVJWRVIwggEiMA0GCSqGSIb3DQEBAQUAA4IB
DwAwggEKAoIBAQDEFcp+Uic4iKcGvZjSsQH1WQOn/5vNcwHRw+v3jAtSxXa5jzAj
SPYmiuYmZTYmU1aIiCckVU0HMWG85NPp55Evvb54odYKJnPYUoRyNNM+3XkF2Pvw
d7lYvPcHl7MK9kylgdszz41DXAKRC3cb9ku3FnvWGPrRXT9HFc/WW0VJxncgYXM2
kjWfDXV+hBPN47GaBi7SK6ohBdgFroilsFHZUpwpdr1rgzh7aMHoWKx+cRp7vTUq
GaMcw+jelTDNG2txJ6AFOa0QJBpbrrImJtexoSsvPhHSUSXKMCDy4PghkuueLbpX
XYeot6tVjeC5GblTaz1TYcEMpWiEP99NMnQzAgMBAAEwDQYJKoZIhvcNAQELBQAD
ggEBAC1Re3Fh6BmMuX+rdu3OWbX9WONw7xYTWaXDvGtg/qczTlp4DA6YlxpTCMLA
NnepHpk4O9b1ml2ukWzymq+YuT4XzBZU2RtHwtHqaal/KTHGYsVY9t8W6aUEArPd
rUeQ3bIzj19KZbRawlA9o6tWRDBDnF8fPAxNLz0YjWHAhZC5TgPbmgWcTOQ5ddrJ
5vrQWl9spRtWCuAXLz1dBgqujtBgTls5eU1nYWkH7Wy42TePWKIJDbIwQrb8wJWi
h/7BS2O0Skpa3T8Z3mryIfoaLZLrY9tn5sBXl3fILwce+Or6NDTV0toBb2gNTBJN
Nei+0jKD9yoAl8ffxN+o8x4uzYg=
-----END CERTIFICATE-----
```

1. Korzystając z klienta http (Postman/cURL/Insomnia) pozyskać z serwera autoryzacji token JWT. Dane uwierzytelniające do Basic Authentication przy wywołaniu metody ***/oauth/token*** zostaną przekazane podczas trwania laboratorium. Odczytać zawartość tokena dostępu (można skorzystać np. z https://jwt.io/) oraz potwierdzić, że załączony klucz publiczny pozwala na weryfikację jego integralności. Wynik weryfikacji załączyć w sprawozdaniu.
2. Zaimplementować metodę autoryzacji w oparciu o OAuth2.0 Client Credentials. Implementacja powinna obejmować dostęp do interfejsów, które zgodnie ze specyfikacją wymagają autoryzacji w oparciu o OAuth2.0.

3. Pozyskać token dostępu, a następnie wykorzystać go w poprawnym zapytaniu do utworzonej aplikacji. Zapytanie powinno skutkować poprawną odpowiedzią zgodnie ze specyfikacją interfejsu.
4. Pozyskać token dostępu, odczekać 30 sekund (czas ważności tokena JWT), a następnie wykorzystać go w poprawnym zapytaniu do utworzonej aplikacji. Zapytanie powinno skutkować błędem wynikającym z upłynięcia czasu ważności tokena JWT. Odpowiedź powinna być zgodna ze specyfikacją interfejsu.
5. Implementacje załączyć do sprawozdania w postaci kodu źródłowego. Dodatkowo zaprezentować przebieg opisanych kroków w sprawozdaniu z laboratorium.

Sprawozdania wraz z kodem źródłowym należy dostarczyć do pierwszego czwartku po zajęciach, do godziny 23:59:59.