**Keycloak OIDC w/ OAuth2 Client**

**Keycloak, JWT기반 OAuth2 책임의 분산 (2/3) : 마이크로서비스와 OAuth2 구성요소인 Authorization Server/ Client/ Resource Server를 활용해 Single Sign-On 구현 모형을 실습한다. OAuth2에서 Client(Gateway)와 Resource Server(마이크로서비스)가 분리된 환경으로 Authorization Server로는 Keycloak을 활용한다.**

10분 이내 완료

[**INTO THE LAB**](https://labs.msaez.io/#/courses/cna-full/cna-full-aws/Keycloak-OAuth2-2)

**Instruction**

**Keycloak기반 OAuth2 - Client**

**Gateway OAuth2 Client 설정**

* **Gateway를 OAuth2 Client로 설정하는 랩으로 이전 랩에서 설정한 Keycloak 정보를 참조한다.**
* **Platform에서 작업이 원활히지 않을 경우, Local에서 수행한다.**

**Local 머신에 IDE(IntelliJ, VSCode)와 JDK 11 이상이 설치되어 있어야 한다.**

**Gateway OAuth2 Client 설정**

* **pom.xml을 통해 Gateway에 설정된 라이브러리(oauth2 client)를 확인한다.**
* **application.yml에 oauth2 client 설정을 주입한다.**
* **주석 부분을 해제하고 나의 keycloak 정보로 수정한다.**

**# security:**

**# oauth2:**

**# client:**

**# provider:**

**# my-keycloak-provider:**

**# issuer-uri: http://localhost:8080/realms/test\_realm**

**# registration:**

**# keycloak-test-client:**

**# provider: my-keycloak-provider**

**# client-id: test-client**

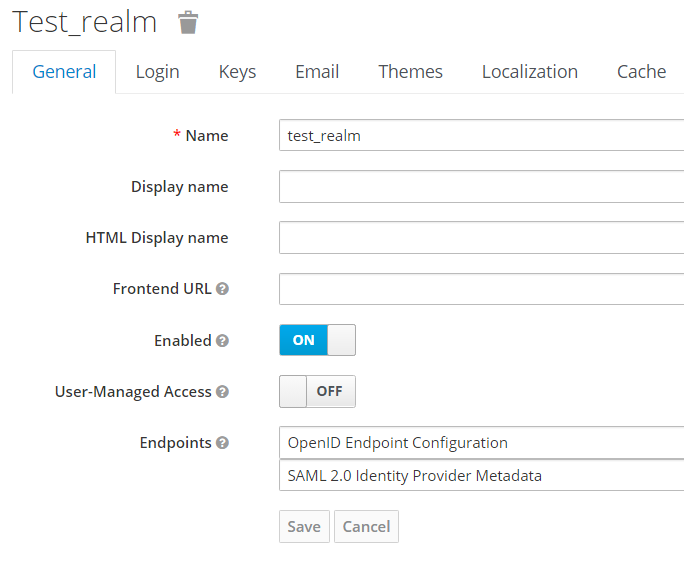
**# client-secret: HKFKYP7kb8OMldAgfvnk27FhRPOv8Y7H**

**# authorization-grant-type: authorization\_code**

**# redirect-uri: '{baseUrl}/login/oauth2/code/keycloak'**

**# scope: openid**

**20행과 같이 keycloak Endpoint 정보는 Realm에서 OIDC 링크를 통해 확인 가능하다.**

****

**Gateway Security Configuration**

* **SecurityConfig.java 파일을 열어 기본 설정을 확인한다.**
* **백엔드 마이크로서비스 단위의 화이트 리스트만 보이고, API 리소스에 대한 설정은 없어 간결하다.**

**Gateway에 Backend 라우팅**

* **application.yml을 다시 오픈한다.**
* **주문 마이크로서비스에 대한 라우팅 설정과 TokenRelay 필터를 적용해 준다.**

**# default-filters:**

**# - TokenRelay**

**# routes:**

**# - id: order**

**# uri: http://localhost:8081**

**# predicates:**

**# - Path=/orders/\*\*, /order/\*\***

* **application.yml을 저장한다.**
* **Gateway 서비스 Root("/")에 Static 페이지를 추가해 확인한다.**

**Gateway 서비스 기동**

* **Gateway를 기동한다. 이때 Authorization Server(Keycloak) 서버가 기동되고 있어야 한다.**

**cd gateway-with-keycloak-oauth2**

**mvn spring-boot:run**

**# Local mvn이 없을 경우, mvnw spring-boot:run**

**OAuth2 Client 확인**

* **브라우저를 열어 Gateway Root(**[**http://localhost:8088**](http://localhost:8088/)**)에 접속한다.**
* **Controller.java에 테스트용 Content("/")가 설정되어 있다.**
* **인증서버의 Login페이지가 출력되고, 등록한 사용자로 인증한다.**
* **아래와 같이 ACL이 적용된 콘텐츠가 출력된다.**

****

**CheckPoints**

1. 모든 요구사항을 만족하는가