gateway 서비스에서 리소스서버 설정

- ServerHttpSecurity 생성시, oauth2ResourceServer() 리소스 서버역할을 부여하고 .jwt() 를 선언해 jwt 형식의 Authorization을 지정

```
@Bean
SecurityWebFilterChain springSecurityFilterChain(ServerHttpSecurity http) throws Exception {

    http
            .cors().and()
            .csrf().disable()
            .authorizeExchange()
            .pathMatchers("/products/**","/goods/**","/oauth/**","/login/**").permitAll()   //이 주소에 대해서는 인증 필요 없음
            .anyExchange().authenticated()
            .and()
            .oauth2ResourceServer() //나머지에 대해서는 인증을 요구하는 설정
            .jwt() //jwt 형식의 Authorization을 지정
            ;

    return http.build();
}
```

- 인증/인가를 위한 Url은 JWK(Json Web Key)로 정의해 application.yaml에 선언

```
spring:
  profiles: default
  security:
    oauth2:
      resourceserver:
        jwt:
          jwk-set-uri: http://localhost:8090/.well-known/jwks.json  #인증/인가를 위한 Url
```

order 서비스를 바로 접근

```
root@labs--317592847:/home/project/oauth2# http localhost:8081/orders
HTTP/1.1 200
Content-Type: application/hal+json;charset=UTF-8
Date: Sat, 12 Mar 2022 14:39:25 GMT
Transfer-Encoding: chunked

{
    "_embedded": {
        "orders": []
    },
    "_links": {
        "profile": {
            "href": "http://localhost:8081/profile/orders"
        },
        "self": {
            "href": "http://localhost:8081/orders{?page,size,sort}",
            "templated": true
        }
    },
    "page": {
        "number": 0,
        "size": 20,
        "totalElements": 0,
        "totalPages": 0
    }
}
```

접근 성공

gateway 서비스를 통하여 접근

```
root@labs--317592847:/home/project/oauth2# http localhost:8088/orders
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Expires: 0
Pragma: no-cache
Referrer-Policy: no-referrer
WWW-Authenticate: Bearer
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1 ; mode=block
content-length: 0
```

(401 Unauthorized)

OAuth 인증서버 구동하기

- **Oauth 서버 구동 후, 토큰을 요청하는 API(/oauth/token) 를 호출하여 토큰 가져오기**

```
http --form POST localhost:8088/oauth/token "Authorization: Basic dWVuZ2luZS1jbGllbnQ6dWVuZ2luZS1zZWNyZXQ=“ grant_type=password
username=1@uengine.org password=1
```

Base64 값으로, 인증서버에 등록된 Gateway의 인코딩된 CLIENT_ID:CLIENT_SECRET 정보

Gateway(8088) application.yaml 파일 일부

```
- id: oauth
  uri: http://localhost:8090
  predicates:
    - Path=/oauth/**
```

위 명령어 실행 결과 (토큰 확인)

```
root@labs--317592847:/home/project/oauth2#  http --form POST localhost:8088/oauth/token "Authorization: Basic dWVuZ2luZS1jbGllbnQ6dWVuZ2luZS1zZWNyZXQ=" grant_type=password username=1@uengine.org password=1
HTTP/1.1 200 OK
Cache-Control: no-store
Content-Type: application/json;charset=UTF-8
Date: Sat, 12 Mar 2022 15:08:44 GMT
Pragma: no-cache
Referrer-Policy: no-referrer
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
transfer-encoding: chunked

{
    "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZGRyZXNzIjoi7ISc7Jq47IucIiwidXNlcl9uYW1lIjoiMUB1ZW5naW5lLm9yZyIsInNjb3BlIjpbInJlYWQiLCJ3cml0ZSIsInRydXN0Il0sIm5pY2tuYW1lIjoi7Jyg7JeU7KeEIiwiY29tcGFueSI6IlVlbmdpbmUiLCJleHAiOjE2NDcxODQxMjMsImF1dGhvcml0aWVzIjpbIlVTRVJfQURNSU4iXSwianRpIjoiS29vZEFXWmRuUTF1RHo0bmlHQnZhZXBPZ0Z3PSIsImNsaWVudF9pZCI6InVlbmdpbmUtY2xpZW50In0.HmD_-bG__FPRu0gvkQyBhWWUqmzWhBUtag71EfJmwR8uRzSOcVYdSTPgeRGyqeMsTQ16jTXmxsILCMjFomnXEn85lcH0qSjXHOR-rst-ATrzqBdiuGcBT9HV34maf-VyJPVgaNnoJiiw3bDcIyJlEe-RPmE3UVj4uCoKnA73RlWfpU7KXdSOe_97ZYebZaAM7VQbwhOmAZ--p_5CdR4Te7uD3WF8WIXuGnCgqWX6u2NKAt6Oa30V7NB8CRGZh7bgRvq6Df3LaivaSIZsD2ObiwZIzBH9ktCknCi10tg2OUuE1UvP3Qmto-sjXoWJuIYGto-3WlqawtJXr7lwim5Dag",
    "address": "서울시",
    "company": "Uengine",
    "expires_in": 86399,
    "jti": "KoodAWZdnQ1uDz4niGBvaepOgFw=",
    "nickname": "유엔진",
    "refresh_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZGRyZXNzIjoi7ISc7Jq47IucIiwidXNlcl9uYW1lIjoiMUB1ZW5naW5lLm9yZyIsInNjb3BlIjpbInJlYWQiLCJ3cml0ZSIsInRydXN0Il0sImF0aSI6Iktvb2RBV1pkblExdUR6NG5pR0J2YWVwT2dGdz0iLCJuaWNrbmFtZSI6IuycoOyXlOynhCIsImNvbXBhbnkiOiJVZW5naW5lIiwiZXhwIjoxNjQ3MTE5MzIzLCJhdXRob3JpdGllcyI6WyJVU0VSX0FETUlOIl0sImp0aSI6IjBhS1ROZWNLRUlGSjFUYlVlSjZ2MEVtang2MD0iLCJjbGllbnRfaWQiOiJ1ZW5naW5lLWNsaWVudCJ9.gkko2MMVZnGV_7Ql-lpE15oFahvRFV07-ttUWsfOlUrvyzW2zml8VpIF1pggrQXPMOwcO-2xJYl0fVlkOE8d_Ba37pQl84M7z_lPBscp-simw7ktZAjDpHA-zv3CLdb2qIop2p1RPhtyTOk6ZSAJdKB_wSF-N4K_FmVUq1HuDfVOpdIjV2Ah1_53KT1ITGgwUY3vnu-KH5fb2bVCOO-9m6yVNYOfOKuKn265F1Dcyv04LVjbk4I4XftL5mh6K5JCxeFsbGdZEc_nWEkkqvSKp8NzJmdoyESEAVtxJrGq2wcIvOHWWxi2ikmwHe9mhbYX7AC1fUzAlV_NZATDpuAtcw",
    "scope": "read write trust",
    "token_type": "bearer"
}
```

<access_token Decode 한 값>

: Header, Payload, Signature로 파싱됨

Encoded PASTE A TOKEN HERE


```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey
JhZGRyZXNzIjoi7ISc7Jq47IucIiwidXNlcl9uY
W1lIjoiMUB1ZW5naW5lLm9yZyIsInNjb3BlIjpb
InJlYWQiLCJ3cml0ZSIsInRydXN0Il0sIm5pY2t
uYW1lIjoi7Jyg7JeU7KeEIiwiY29tcGFueSI6Il
VlbmdpbmUiLCJleHAiOjE2NDcxODQxMjMsImF1d
Ghvcml0aWVzIjpbIlVTRVJfQURNSU4iXSwianRp
IjoiS29vZEFXWmRuUTF1RHo0bmlHQnZhZXBPZ0Z
3PSIsImNsaWVudF9pZCI6InVlbmdpbmUtY2xpZW
50In0.HmD_-
bG__FPRu0gvkQyBhWWUqmzWhBUtag71EfJmwR8u
RzSOcVYdSTPgeRGyqeMsTQ16jTXmxsILCMjFomn
XEn85lcH0qSjXHOR-rst-
ATrzqBdiuGcBT9HV34maf-
VyJPVgaNnoJiiw3bDcIyJlEe-
RPmE3UVj4uCoKnA73RlWfpU7KXdSOe_97ZYebZa
AM7VQbwhOmAZ--
p_5CdR4Te7uD3WF8WIXuGnCgqWX6u2NKAt6Oa30
V7NB8CRGZh7bgRvq6Df3LaivaSIZsD2ObiwZIzB
H9ktCknCi10tg2OUuE1UvP3Qmto-
sjXoWJuIYGto-3WlqawtJXr7lwim5Dag
```


Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "alg": "RS256",
  "typ": "JWT"
}
```


PAYLOAD: DATA


```
{
  "address": "서울시",
  "user_name": "1@uengine.org",
  "scope": [
    "read",
    "write",
    "trust"
  ],
  "nickname": "유엔진",
  "company": "Uengine",
  "exp": 1647184123,
  "authorities": [
    "USER_ADMIN"
  ],
  "jti": "KoodAWZdnQ1uDz4niGBvaepOgFw=",
  "client_id": "uengine-client"
}
```


VERIFY SIGNATURE


```
RSASHA256(
```

토큰을 이용하여 게이트웨이를 통하여 주문 서비스를 조회
: 200 OK 정상 조회

```
root@labs--317592847:/home/project/oauth2# http localhost:8088/orders "Authorization: Bearer $access_token"
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Content-Type: application/hal+json;charset=UTF-8
Date: Sat, 12 Mar 2022 15:25:35 GMT
Expires: 0
Pragma: no-cache
Referrer-Policy: no-referrer
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1 ; mode=block
transfer-encoding: chunked

{
    "_embedded": {
        "orders": []
    },
    "_links": {
        "profile": {
            "href": "http://localhost:8081/profile/orders"
        },
        "self": {
            "href": "http://localhost:8081/orders{?page,size,sort}",
            "templated": true
        }
    },
    "page": {
        "number": 0,
        "size": 20,
        "totalElements": 0,
        "totalPages": 0
    }
}
```

<토큰 유효성 확인>

200 OK

```
root@labs--317592847:/home/project/oauth2# http --form POST localhost:8088/oauth/check_token token=$access_token
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Content-Type: application/json
Date: Sat, 12 Mar 2022 15:24:54 GMT
Expires: 0
Pragma: no-cache
Referrer-Policy: no-referrer
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
transfer-encoding: chunked

{
    "active": true,
    "address": "서울시",
    "authorities": [
        "USER_ADMIN"
    ],
    "client_id": "uengine-client",
    "company": "Uengine",
    "exp": 1647185002,
    "jti": "PvbSTQm/zRgOFbulAXDVz9YIYSs=",
    "nickname": "유엔진",
    "scope": [
        "read",
        "write",
        "trust"
    ],
    "user_name": "1@uengine.org"
}
```

<토큰 변경하여 유효성 확인>

400 Bad Request

```
root@labs--317592847:/home/project/oauth2# http --form POST localhost:8088/oauth/check_token token=$access_token
HTTP/1.1 400 Bad Request
Cache-Control: no-store
Content-Type: application/json
Date: Sat, 12 Mar 2022 15:24:26 GMT
Pragma: no-cache
Referrer-Policy: no-referrer
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
transfer-encoding: chunked

{
    "error": "invalid_token",
    "error_description": "Invalid access token"
}
```