

Inhaltsverzeichnis

1Docker setup.....2

2Test JWT access.....6

1 Docker setup

- Clone from `git@github.com:GilbertHofstaetter/spring-boot-oauth2-resource-server-client-vaadin.git`
- Add **127.0.0.1 keycloak** to your hosts file

- **Linux/Max /etc/hosts**
- **Windows 10 c:\Windows\System32\Drivers\etc\hosts**

```
127.0.0.1      keycloak
```

- Change to directory to project root directory

```
spring-boot-oauth2-resource-server-client-vaadin %
```

- Run **docker-compose -f docker-compose-bitnami-custom.yml up**

```
spring-boot-oauth2-resource-server-client-vaadin % docker-compose -f docker-compose-bitnami-custom.yml up
```

- Postgres, keycloak and springboot-microservice are starting → keycloak imports Demo-Realm from file `./realms/Demo-Realm-realm.json`
- springboot-microservice fails on startup (Docker Desktop) → don't mind

Container Name	Image	Status	Ports	Restart	Created	Actions
spring-boot-oauth2-resource-server-client	-	Running (2/3)				
keycloak-1	bitnami/keycloak:latest	Running	8091:8091		19 seconds ago	
postgresql-1	bitnami/postgresql:latest	Running			20 seconds ago	
springboot-microservice-vaadin-1	springboot-microservice-vaadin:latest	Exited (137)	8080:8080			

- Restart springboot-microservice-vaadin-1

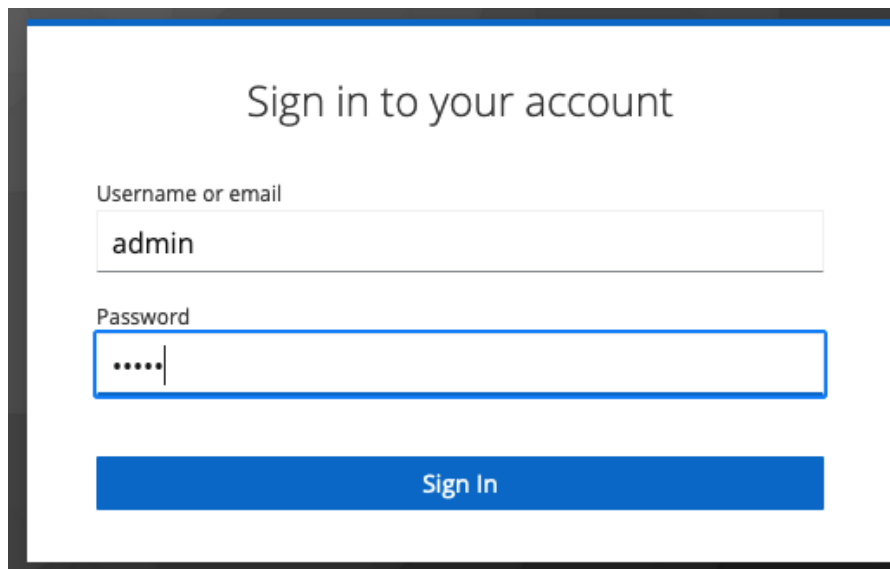
```

:: Spring Boot ::
               (v2.7.5)

2022-11-14 07:07:43.914 INFO 16 --- [main] .O.Oauth2ResourceServerClientApplication : Starting Oauth2ResourceServerClientApplication v1.0-SNAPSHOT using Java 17.0.5 on sff1940a4104 with PID 16 [/app/target/spring-boot-oauth2-resource-server-client-1.0-SNAPSHOT-jar started by root in /app/target]
2022-11-14 07:07:43.924 INFO 16 --- [main] .O.Oauth2ResourceServerClientApplication : No active profile set, falling back to 1 default profile: "default"
2022-11-14 07:07:45.878 INFO 16 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-11-14 07:07:45.980 INFO 16 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 84 ms. Found 2 JPA repository interfaces.
2022-11-14 07:07:47.052 INFO 16 --- [main] o.s.w.e.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-11-14 07:07:47.074 INFO 16 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-11-14 07:07:47.075 INFO 16 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.68]
2022-11-14 07:07:47.218 INFO 16 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-11-14 07:07:47.218 INFO 16 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3119 ms
2022-11-14 07:07:47.546 INFO 16 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-11-14 07:07:47.851 INFO 16 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-11-14 07:07:47.925 INFO 16 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000024: Processing PersistenceUnitInfo [name: default]
2022-11-14 07:07:48.008 INFO 16 --- [main] org.hibernate.Version : HHH0000412: Hibernate ORM core version 5.6.12.Final
2022-11-14 07:07:48.561 INFO 16 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2022-11-14 07:07:48.759 INFO 16 --- [main] org.hibernate.dialect.Dialect : HHH0000400: Using dialect: org.hibernate.dialect.H2Dialect
2022-11-14 07:07:49.612 INFO 16 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-11-14 07:07:49.626 INFO 16 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized Jpa EntityManagerFactory for persistence unit 'default'
2022-11-14 07:07:52.340 WARN 16 --- [main] javaBaseConfigurationsJpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2022-11-14 07:07:53.475 INFO 16 --- [main] o.s.w.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEncodeUrlFilter#64920cd2, org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter#59a7943d, org.springframework.security.web.context.SecurityContextPersistenceFilter#fd53053, org.springframework.security.web.header.HeaderWriterFilter#7c17a57, org.springframework.security.web.filter.CharacterEncodingFilter#439ac8ad, org.springframework.security.web.csrf.CsrfFilter#4601203a, org.springframework.security.web.authentication.logout.LogoutFilter#5c723f2d, org.springframework.security.oauth2.client.web.OAuth2AuthorizationRequestRedirectFilter#4ida3ae, org.springframework.security.oauth2.client.web.OAuth2LoginAuthenticationFilter#5dc0ff7d, org.springframework.security.web.authentication.ui.DefaultLogoutPageGeneratingFilter#527fcbe, org.springframework.security.web.authentication.ui.DefaultLogoutPageGeneratingFilter#3dbde9e, org.springframework.security.oauth2.server.resource.web.BearerTokenAuthenticationFilter#8da5fb0, org.springframework.security.web.savedrequest.RequestCacheAwareFilter#70b1a44, org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter#7132a9dc, org.springframework.security.web.authentication.AnonymousAuthenticationFilter#6751e3a, org.springframework.security.web.session.SessionManagementFilter#e1ea8a64, org.springframework.security.web.access.ExceptionTranslationFilter#60b34931, org.springframework.security.web.access.intercept.FilterSecurityInterceptor#fd9893c]
2022-11-14 07:07:54.338 INFO 16 --- [main] o.s.w.e.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-11-14 07:07:54.371 INFO 16 --- [main] .O.Oauth2ResourceServerClientApplication : Started Oauth2ResourceServerClientApplication in 11.42 seconds (JVM runn
ng for 12.252)

```

- springboot-microservice-vaadin is up and running
- keycloak has imported Demo-Realm-realm.json at startup → Realm, springboot-microservice, scope read, Userroles (app-user, app-admin, admin, user) are now present
- Create 3 Users through keycloak admin console → localhost:8091
- User: admin, Pwd: admin



Sign in to your account

Username or email

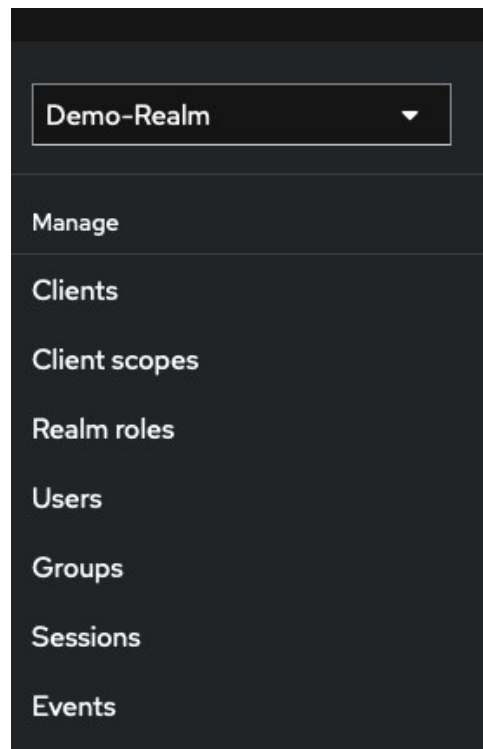
admin

Password

.....

Sign In

- Open realm „**Demo-Realm**“



- Create 3 Users
 - employee1 → Assign Role app-user, set Email verified to on, set credential switch „Temporary“ to off
 - employee2 → Assign Role app-admin, set Email verified to on, set credential switch „Temporary“ to off
 - employee3 → Assign Role app-user, app-admin, set Email verified to on, set credential switch „Temporary“ to off

Demo-Realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Users > Create user

Create user

Username *

employee2

Email

Email verified ?

On

First name

Last name

Enabled ?

On

Users > User details

employee3

Details	Attributes	Credentials	Role mapping	Groups	Consents	
<div><div>Q Search by name</div><div>→</div></div>		<div><input checked="" type="checkbox"/> Hide inherited roles</div>	<div>Assign role</div>	<div>Unassign</div>		
<input type="checkbox"/>	Name	Inherited	Description			
<input type="checkbox"/>	app-user	False	—			
<input type="checkbox"/>	default-roles-demo-realm	False	`\${role_default-role:			
<input type="checkbox"/>	app-admin	False	—			

Users > User details

employee3

Set password for employee3

Password *

.....

👁

Password confirmation *

.....

👁

Temporary ?

☐ Off

Save

Cancel

- Open <http://localhost:8080/swagger-ui.html>
- Sign in with employee3

localhost:8080/swagger-ui/index.html

Swagger powered by SMARTBEAR /api/api-docs Explore

springboot-microservice ^{v1} OAS3

/api/api-docs

Demo project for keycloak integration as an authorisation server in a spring boot resource server and client with rbac and scopes authorisation based on spring security.

Servers

http://localhost:8080 - Generated server url Authorize

index-controller

GET	/userOAuth2	🔒
GET	/userJwt	🔒
GET	/logout	🔒
GET	/loggedout	🔒

- Try out
 - <http://localhost:8080/api/customer/list>
 - <http://localhost:8080/api/employee/list>
 - <http://localhost:8080/userJwt>
 - <http://localhost:8080/userOAuth2>
 - <http://localhost:8080/logout>

2 Test JWT access

- With Docker the Keycloak host changed from localhost to „keycloak“
- Use <http://keycloak:8091/realms/Demo-Realm/protocol/openid-connect/token> to request an access token from Keycloak. A request with host localhost:8091 delivers an invalid token

POST

http://keycloak:8091/realms/Demo-Realm/protocol/openid-connect/token

Send request







Headers >

Basic auth >

Request body ▾

Type

URLEncoded form data ▾

client_id	springboot-microservice	
grant_type	password	
client_secret	L7CYws3HULR0IAsA4aSwssvZZrS9iz10	
username	employee3	
password	*your user password*	
scope	profile roles email read	

[+Add parameter](#)

[illegible]

- Request <http://localhost:8080/api/company/list> from resource server

GET

http://localhost:8080/api/company/list

Send request

Headers

Authorization

u0VCU0KIB1aI3Ffm1HbzoK2_W932F31TogxGpKRfWwGnjXm-C0g

+ Add header

Basic auth

Response (0.051s) - http://localhost:8080/api/company/list

200

Headers >

```
{
  "id": 4,
  "name": "Company",
  "employees": [
    {
      "id": 2,
      "firstName": "Max1",
      "lastName": "Mustermann1",
      "email": "m1.m1@company.com",
      "birthDate": "2022-11-15T09:39:39.359+00:00"
    },
    {
      "id": 3,
      "firstName": "Max2",
      "lastName": "Mustermann2",
      "email": "m2.m2@company.com",
      "birthDate": "2022-11-15T09:39:39.362+00:00"
    },
    {
      "id": 1,
      "firstName": "Max0",
      "lastName": "Mustermann0",
      "email": "m0.m0@company.com",
      "birthDate": "2022-11-15T09:39:39.237+00:00"
    }
  ]
}
```