

Diseño y Construcción de Microservicios

LAB Microservicios y Spring Security Obteniendo Tokens

Objetivos

- Mostrar al participante el procedimiento para la obtención de tokens con Spring Security

Procedimiento

1. Crea las contraseñas encriptadas para los usuarios. En el proyecto **springboot-servicio-oauth** edita la clase `SpringbootServicioOauthApplication.java`.

```
package com.cognos.app.oauth;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
import org.springframework.cloud.openfeign.EnableFeignClients;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

@EnableFeignClients
@EnableEurekaClient
@SpringBootApplication
public class SpringbootServicioOauthApplication implements CommandLineRunner{

    @Autowired
    private BCryptPasswordEncoder passwordEncode;

    public static void main(String[] args) {
        SpringApplication.run(SpringbootServicioOauthApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        String password = "12345";

        for (int i=0; i<4; i++) {
            String passwordBCrypt = passwordEncode.encode(password);
            System.out.println(passwordBCrypt);
        }
    }
}
```

2. Inicia los servicios servicio-eureka-server, servicio-usuarios y servicio-oauth. En la consola se mostrara las cuatro encriptaciones de la contraseña "12345".

```
springboot-servicio-oauth - SpringbootServicioOauthApplication [Spring Boot App] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (17 may. 2020 17:15:35)
2020-05-17 17:15:53.400 INFO 726172 --- [ restartedMain] c.n.discovery.InstanceInfoReplicator : InstanceInfoReplicator or
2020-05-17 17:15:53.406 INFO 726172 --- [ restartedMain] com.netflix.discovery.DiscoveryClient : Discovery Client initiali
2020-05-17 17:15:53.407 INFO 726172 --- [ restartedMain] o.s.c.n.e.s.EurekaServiceRegistry : Registering application s
2020-05-17 17:15:53.408 INFO 726172 --- [ restartedMain] com.netflix.discovery.DiscoveryClient : Saw local status change e
2020-05-17 17:15:53.412 INFO 726172 --- [nfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_SERVICIO-
2020-05-17 17:15:53.447 INFO 726172 --- [nfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_SERVICIO-
2020-05-17 17:15:53.901 INFO 726172 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s)
2020-05-17 17:15:53.903 INFO 726172 --- [ restartedMain] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9100
2020-05-17 17:15:54.562 INFO 726172 --- [ restartedMain] c.a.o.SpringbootServicioOauthApplication : Started SpringbootServi
$2a$10$PR4mN139rOdbdyGQJ4kCbeJzQBR05h2XIMatgFkKyaImTxGum9Niu
$2a$10$qsru5gVeP7Ic1rPdEes23.76Yn568pv7KRCgJUN.81I4V4M6Y03HK
$2a$10$bt5eUHzE3q0i0YMcD0opCe9RgowzUjSmZaiv8N5Kkd0My.01zrkK
$2a$10$Pz6Y9iyc0lonuiwDXVbC6eVd01CcbT7VEK9B0JGnMEF2PRcD5j0Vy
```

3. Reemplaza la contraseña "12345" por la contraseña encriptada en el archivo import.sql del servicio servicio-usuarios.

```
"import.sql"
1 insert into usuarios (username,password,enabled,nombre,apellido,email) values('carlos','$2a$10$PR4mN139rOdbdyGQJ4kCbeJzQBR05h2XIMatgFkKyaImTxGum9Niu',1,
2 insert into usuarios (username,password,enabled,nombre,apellido,email) values('franklin','$2a$10$qsru5gVeP7Ic1rPdEes23.76Yn568pv7KRCgJUN.81I4V4M6Y03HK',1,
4 insert into roles (nombre) values('ROLE_USER');
5 insert into roles (nombre) values('ROLE_ADMIN');
6
7 insert into usuarios_roles (usuario_id, role_id) values (1,1);
8 insert into usuarios_roles (usuario_id, role_id) values (2,2);
9 insert into usuarios_roles (usuario_id, role_id) values (2,1);

springboot-servicio-oauth - SpringbootServicioOauthApplication [Spring Boot App] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (17 may. 2020 17:15:35)
2020-05-17 17:15:53.400 INFO 726172 --- [ restartedMain] c.n.discovery.InstanceInfoReplicator : InstanceInfoReplicator onDemand update allowed rate per
2020-05-17 17:15:53.406 INFO 726172 --- [ restartedMain] com.netflix.discovery.DiscoveryClient : Discovery Client initialized at timestamp 1589753753404
2020-05-17 17:15:53.407 INFO 726172 --- [ restartedMain] o.s.c.n.e.s.EurekaServiceRegistry : Registering application SERVICIO-OAUTH with eureka with
2020-05-17 17:15:53.408 INFO 726172 --- [ restartedMain] com.netflix.discovery.DiscoveryClient : Saw local status change event StatusChangeEvent [timesta
2020-05-17 17:15:53.412 INFO 726172 --- [nfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_SERVICIO-OAUTH/host.docker.internal:serv
2020-05-17 17:15:53.447 INFO 726172 --- [nfoReplicator-0] com.netflix.discovery.DiscoveryClient : DiscoveryClient_SERVICIO-OAUTH/host.docker.internal:serv
2020-05-17 17:15:53.901 INFO 726172 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9100 (http) with context path
2020-05-17 17:15:53.903 INFO 726172 --- [ restartedMain] s.c.n.e.s.EurekaAutoServiceRegistration : Updating port to 9100
2020-05-17 17:15:54.562 INFO 726172 --- [ restartedMain] c.a.o.SpringbootServicioOauthApplication : Started SpringbootServicioOauthApplication in 13.464 sec
$2a$10$PR4mN139rOdbdyGQJ4kCbeJzQBR05h2XIMatgFkKyaImTxGum9Niu
$2a$10$qsru5gVeP7Ic1rPdEes23.76Yn568pv7KRCgJUN.81I4V4M6Y03HK
$2a$10$bt5eUHzE3q0i0YMcD0opCe9RgowzUjSmZaiv8N5Kkd0My.01zrkK
$2a$10$Pz6Y9iyc0lonuiwDXVbC6eVd01CcbT7VEK9B0JGnMEF2PRcD5j0Vy
2020-05-17 18:26:15.103 INFO 726172 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
```

4. Reinicia el servicio servicio-usuarios. Verifica que la contraseña cifrada se haya registrado en la base de datos.

```
ernate.SQL : alter table usuarios_roles add constraint UKqjasp7473pnu9y4jxhnds8r2 unique (usuario_id, role_id)
ernate.SQL : alter table usuarios_roles add constraint FKihom0uklpkpfppfipxpyf7b74 foreign key (role_id) references roles
ernate.SQL : alter table usuarios_roles add constraint FKqcxu02bqixpr7c7jy9dmhwee foreign key (usuario_id) references usuarios
chema.internal.SchemaCreatorImpl : HH0000476: Executing Import script 'ScriptSourceInputFromUrl[file:/C:/springCloud/workspace/springboot-servicio-usuarios/targ
ernate.SQL : insert into usuarios (username,password,enabled,nombre,apellido,email) values('carlos','$2a$10$PR4mN139rOdbdyGQJ4kCbeJzQBR05h
ernate.SQL : insert into usuarios (username,password,enabled,nombre,apellido,email) values('franklin','$2a$10$qsru5gVeP7Ic1rPdEes23.76Yn56
ernate.SQL : insert into roles (nombre) values('ROLE_USER')
ernate.SQL : insert into roles (nombre) values('ROLE_ADMIN')
ernate.SQL : insert into usuarios_roles (usuario_id, role_id) values (1,1)
ernate.SQL : insert into usuarios_roles (usuario_id, role_id) values (2,2)
ernate.SQL : insert into usuarios_roles (usuario_id, role_id) values (2,1)
ContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
```

5. Modifica el servidor Zuul y registra el servicio servicio-oauth. En el proyecto springboot-servicio-zuul-server edita el archivo de recursos application.properties.

```
spring.application.name=servicio-zuul-server
server.port=8090

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

zuul.routes.usuarios.service-id=servicio-usuarios
zuul.routes.usuarios.path=/api/usuarios/**

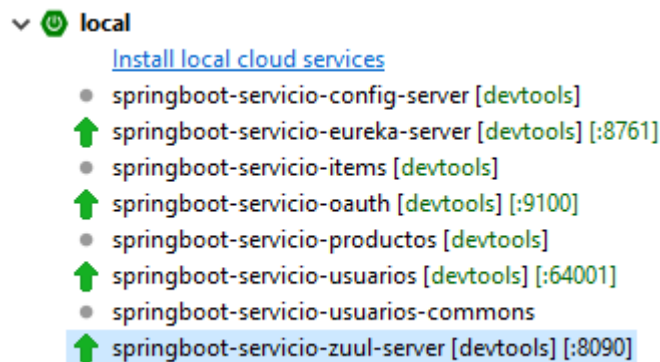
zuul.routes.productos.service-id=servicio-productos
zuul.routes.productos.path=/api/productos/**

zuul.routes.items.service-id=servicio-items
zuul.routes.items.path=/api/items/**

zuul.routes.oauth.service-id=servicio-oauth
zuul.routes.oauth.path=/api/security/**

#excluir las cookies de los headers
zuul.routes.oauth.sensitive-headers=Cookie,Set-Cookie
```

6. Inicia los servicios servicio-eureka-server, servicio-usuarios, servicio-oauth y servicio-zuul-server.

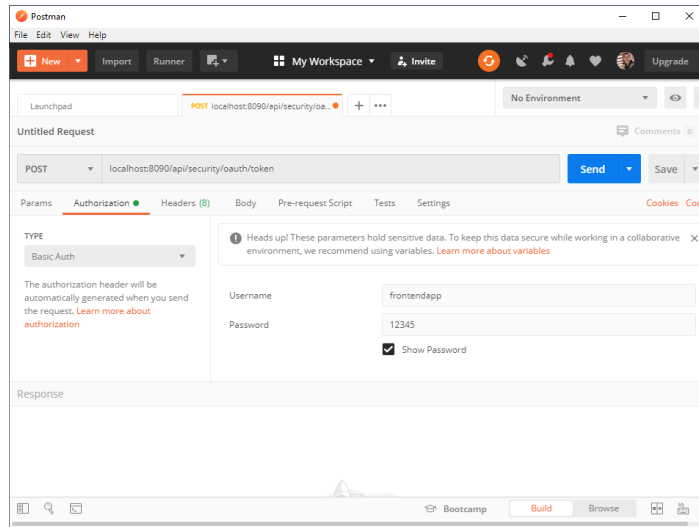


7. Realiza las siguientes pruebas con Postman.
 - a. Crea el request.

En la url del Request escribe:

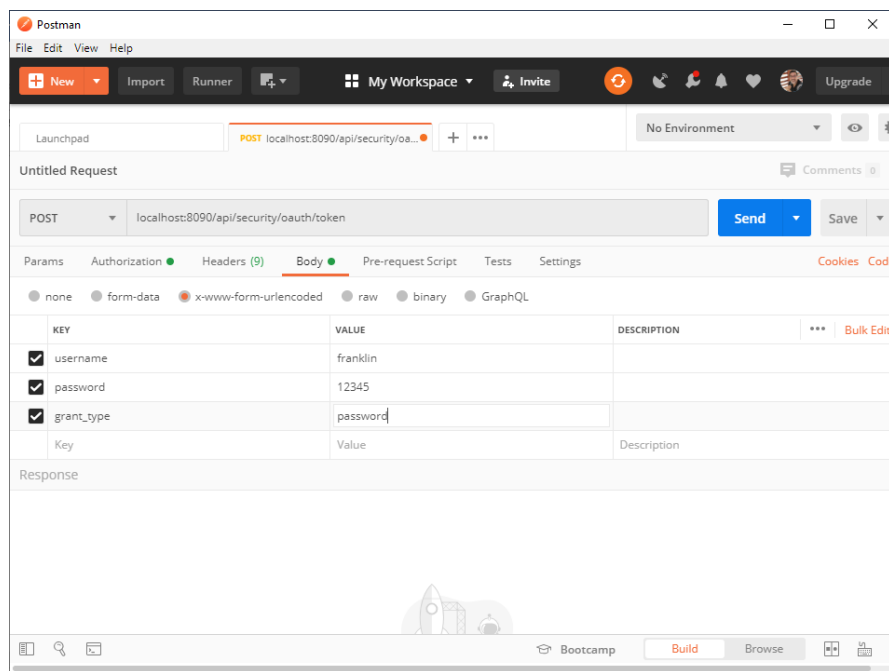
localhost:8090/api/security/oauth/token de tipo **POST**

Selecciona el tab **Authorization**, en TYPE, selecciona: **Basic OAuth**
y crea los headers: **username=frontend** y **password=12345**



- b. Crea el body e indica las credenciales.

Selecciona el tab Body , luego x-www-form-urlencoded y crea las credenciales, username=franklin, password=12345 y grant_type=password.



Haz clic en Send.

- c. Verificando el token.

Nota: En algunos casos, sobre todo la primera vez puede demorar y abortar por “time out” si fuera el caso intenta otra vez.

