## Labo JWT

|  |  |
| --- | --- |
|  | si primera vez  git clone https://github.com/luitoz/curso-microservicios |
| 1 | Bajar fuentes  git pull  verificar que se hayan creado los objetos en  curso-microservicios/sesion6/demo-jwt |
| 2 | modificar authentication service   * authentication-service/pom.xml   agregar dependencia  ...  <dependency>  <groupId>org.springframework.security</groupId>  <artifactId>spring-security-jwt</artifactId>  </dependency>  ... |
| 3 | agregar   * authentication-service-jwt/src/main/java/com/thoughtmechanix/authentication/security/JWTTokenStoreConfig.java |
| 4 | reemplazar oath2config por   * authentication-service-jwt/src/main/java/com/thoughtmechanix/authentication/security/JWTOAuth2Config.java |
| 5 | verificar   * <https://github.com/luitoz/curso-microservicios/blob/master/config-repo2/authenticationservice/authenticationservice.yml>   ...  signing.key: "345345fsdfsf5345" |
| 6 | desplegar  mvn clean package docker:build && docker-compose -f docker/common/docker-compose.yml up |
| 7 | Obtener token jwt  Post localhost:5555/api/auth/auth/oauth/token  basic auth   * username/password= eagleeye/thisissecret   body-form data   * grant\_type: password * scope: webclient * username: admin.user * password: password2 |
| 8 | decodificar token jwt   1. ir a http://jsonwebtoken.io y copiar el token generado 2. ver token decodificado |
| 9 | licensing-service   * licensing-service/pom.xml * licensing-service/src/main/resources/application.yml * licensing-service/src/main/java/com/thoughtmechanix/licenses/security/JWTTokenStoreConfig.java |
| 10 | organization-service   * organization-service/pom.xml * organization-service/src/main/resources/application.yml * organization-service/src/main/java/com/thoughtmechanix/organization/security/JWTTokenStoreConfig.java |
| 11 | desplegar licensing y organization  mvn clean package docker:build -f licensing-service/pom.xml && docker-compose -f docker/common/docker-compose.yml up licensingservice  mvn clean package docker:build -f organization-service/pom.xml && docker-compose -f docker/common/docker-compose.yml up organizationservice |
| 12 | Invocar con token jwt  http://localhost:5555/api/licensing/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a  con token de admin.user |

## Labo Entender JWT

|  |  |
| --- | --- |
| 1 | Extender token jwt.  Agregar   * authentication-service/src/main/java/com/thoughtmechanix/authentication/security/JWTTokenEnhancer.java   Modificar   * authentication-service/src/main/java/com/thoughtmechanix/authentication/security/JWTTokenStoreConfig.java   …  @Bean  public TokenEnhancer jwtTokenEnhancer() {  return new JWTTokenEnhancer();  }  …   * authentication-service/src/main/java/com/thoughtmechanix/authentication/security/JWTOAuth2Config.java   …  @Autowired  private TokenEnhancer jwtTokenEnhancer;  ...  @Override  public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws Exception {  //5.  TokenEnhancerChain tokenEnhancerChain = new TokenEnhancerChain();  tokenEnhancerChain.setTokenEnhancers(Arrays.asList(jwtTokenEnhancer, jwtAccessTokenConverter));  //3.  endpoints.tokenStore(tokenStore) .accessTokenConverter(jwtAccessTokenConverter)  .tokenEnhancer(tokenEnhancerChain)  .authenticationManager(authenticationManager)  .userDetailsService(userDetailsService);  }  … |
| 2 | desplegar authentication  mvn clean package docker:build -f authentication-service/pom.xml && docker-compose -f docker/common/docker-compose.yml up authenticationservice |
| 3 | Obtener token jwt  Post localhost:5555/api/auth/auth/oauth/token |
| 4 | decodificar token jwt   1. ir a http://jsonwebtoken.io y copiar el token generado 2. ver token decodificado |
| 5 | propagar organizationid de jwt a todos los servicios  Modificar   * zuulsvr/pom.xml   agregar  …  <dependency>  <groupId>io.jsonwebtoken</groupId>  <artifactId>jjwt</artifactId>  <version>0.7.0</version>  </dependency>  …   * zuulsvr/src/main/java/com/thoughtmechanix/zuulsvr/filters/TrackingFilter.java   agregar  …  private String getOrganizationId(){  String result="";  if (filterUtils.getAuthToken()!=null){  //1  String authToken = filterUtils.getAuthToken().replace("Bearer ","");  try {  Claims claims = Jwts.parser()//2  .setSigningKey(serviceConfig.getJwtSigningKey().getBytes("UTF-8"))  .parseClaimsJws(authToken).getBody();//3  result = (String) claims.get("organizationId");//4  }  catch (Exception e){  e.printStackTrace();  }  }  return result;  }  //0  public Object run() {  RequestContext ctx = RequestContext.getCurrentContext();  if (isCorrelationIdPresent()) {  logger.debug("tmx-correlation-id found in tracking filter: {}. ", filterUtils.getCorrelationId());  }  else{  filterUtils.setCorrelationId(generateCorrelationId());  logger.debug("tmx-correlation-id generated in tracking filter: {}.", filterUtils.getCorrelationId());  }  logger.debug("The organization id from the token is : " + getOrganizationId());//5  // 6  filterUtils.setOrgId(getOrganizationId());  logger.debug("Processing incoming request for {}.", ctx.getRequest().getRequestURI());  return null;  }  … |
| 6 | Desplegar zuul  mvn clean package docker:build -f zuulsvr/pom.xml && docker-compose -f docker/common/docker-compose.yml up zuulserver |
| 7 | Invocar con token jwt  http://localhost:5555/api/licensing/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a  con token de admin.user |

## Labo testing Basics

|  |  |
| --- | --- |
| 1 | demo-testing/  ejecutar  cd src  mvn install |
| 2 | * importar proyectos a eclipse |
| 3 | * account-service/ |
| 4 | * account-service/src/test/java/demo/account/AccountServiceTests.java |
| 5 | * account-service/src/main/java/demo/account/AccountService.java |
| 6 | * account-service/src/main/java/demo/user/UserService.java |
| 7 | ejecutar AccountServiceTests con eclipse |

## Labo Test Slices

|  |  |
| --- | --- |
| 1 | * account-service/src/test/java/demo/account/AccountTests.java |
| 2 | * account-service/src/test/java/demo/account/AccountControllerTest.java |
| 3 | * account-service/src/test/java/demo/account/AccountRepositoryTest.java |
| 4 | * account-service/src/test/java/demo/user/UserServiceTests.java |