# construir api rest

|  |  |
| --- | --- |
|  | * configurar virtual machine   credenciales: micro/micro2019  <https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>   * + https://drive.google.com/drive/folders/11VxGYwbJoRYk1GPa7zuU1Kz6dVzdgGTw   + https://drive.google.com/drive/folders/11VxGYwbJoRYk1GPa7zuU1Kz6dVzdgGTw |
| 1 | git clone https://github.com/javidep/curso-microservicios.git  ﻿verificar archivos en  .../sesion2/demo-rest |
| 2 | Ejecutar:  cd demo-rest  mvn spring-boot:run |
|  | Con postman hacer las peticiones: |
| 3 | options http://localhost:8080/v1/customers/ |
| 4 | Get http://localhost:8080/v1/customers/1  Get http://localhost:8080/v1/customers/9 |
| 5 | Post http://localhost:8080/v1/customers/  ingresar como body:  {  "id": 9,  "firstName": "Mark",  "lastName": "Fisher"  } |
| 6 | Delete http://localhost:8080/v1/customers/1  http://localhost:8080/v1/customers/0 |
| 7 | Head http://localhost:8080/v1/customers/2  http://localhost:8080/v1/customers/0 |

|  |  |
| --- | --- |
|  | Demo confsvr |
| 1 | verificar objetos en  ../demo-config |
| 2 | Iniciar servicio config:  cd demo-config/confsvr  mvn spring-boot:run  verificar que el servicio inicie correctamente |
| 3 | usar postman:  Get http://localhost:8888/licensingservice/default |
| 4 | Get http://localhost:8888/licensingservice/dev |

|  |  |
| --- | --- |
|  | integrar cliente |
| 1 | Iniciar licensing service  cd demo-config/licensing-service  mvn spring-boot:run  Veriificar que hay errores al iniciar. |
| 2 | comentar propiedades de base de datos en confsvr  1. confsvr/src/main/resources/config/  licensingservice/licensingservice.yml  2. confsvr/src/main/resources/config/  licensingservice/licensingservice-dev.yml  3. confsvr/src/main/resources/config/  licensingservice/licensingservice-prod.yml |
| 3 | reiniciar servicio confsvr  cd confsvr  mvn spring-boot:run |
| 4 | ReIniciar licensing-service  cd licensing-service  mvn spring-boot:run  verificar que inicio correctamente |
| 5 | ejecutar  Get http://localhost:8080/env |
| 6 | Reiniciar licensing sobreescribiendo algunas propiedades  java -Dspring.cloud.config.uri=http://localhost:8888 \  -Dspring.profiles.active=dev \  -jar target/licensing-service-0.0.1-SNAPSHOT.jar |
| 7 | ejecutar y verificar  Get http://localhost:8080/env |
|  | refrescado de propiedades |
|  |  |
| 8 | agregar anotacion refreshscope  licensing-service/src/main/java/com/thoughtmechanix/  licenses/Application.java  @RefreshScope |
| 9 | reiniciar licensing  cd licensing-service  mvn spring-boot:run |
| 10 | Refrescar una propiedad de la configuracion en el archivo  confsvr/src/main/resources/config/  licensingservice/licensingservice.yml  y reiniciar  cd confsvr  mvn spring-boot:run |
| 11 | ejecutar  Get http://localhost:8080/env  Verificar si la propiedad se actualiza |
| 12 | ejecutar  post http://localhost:8080/refresh |
| 13 | ejecutar  Get http://localhost:8080/env  Verificar que se actualizo la propiedad |

|  |  |
| --- | --- |
|  | Git+Docker |
| 1 | modificar configuracion. de confsvr para apuntar a github  confsvr/src/main/resources/application.yml  descomentar  ...  # server:  # port: 8888  # spring:  # cloud:  # config:  # server:  # encrypt.enabled: false  # git:  # uri: https://github.com/luitoz/curso-microservicios/  # searchPaths: config-repo/licensingservice,config-repo/organizationservice  ... |
| 2 | reiniciar confsvr  cd confsvr  mvn spring-boot:run |
| 3 | ejecutar  Get http://localhost:8888/licensingservice/dev |

|  |  |
| --- | --- |
|  | dockerizar |
| 4 | construir la imagen desde el carpeta padre  cd demo-config  mvn clean package docker:build |
| 5 | ejecutar los servicios "dockerizados"  docker-compose -f docker/common/docker-compose.yml down  docker-compose -f docker/common/docker-compose.yml up |
| 6 | modificar docker-compose para agregar configuracion de base de datos  docker/common/docker-compose.yml  descomentar  # environment:  # POSTGRES\_USER: "postgres"  # POSTGRES\_PASSWORD: "p0stgr@s"  # POSTGRES\_DB: "eagle\_eye\_local" |
| 7 | ejecutar nuevamente los servicios "dockerizados"  docker-compose -f docker/common/docker-compose.yml up |
| 8 | eliminar contenedores  docker-compose -f docker/common/docker-compose.yml down |
| 9 | ejecutar nuevamente los servicios "dockerizados"  docker-compose -f docker/common/docker-compose.yml up  verificar que inician correctamente |
| 10 | ejecutar  Get http://localhost:8080/env |
| 11 | consultar licencias  get http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/ |
| 12 | ingresar un registro a licensing  post http://localhost:8080/v1/organizations/1/licenses/  {  "organizationId": 1,  "productName":"name",  "licenseType":"type",  "licenseMax":1,  "licenseAllocated":2  } |
| 13 | verificar el registro ingresado  get http://localhost:8080/v1/organizations/1/licenses/ |