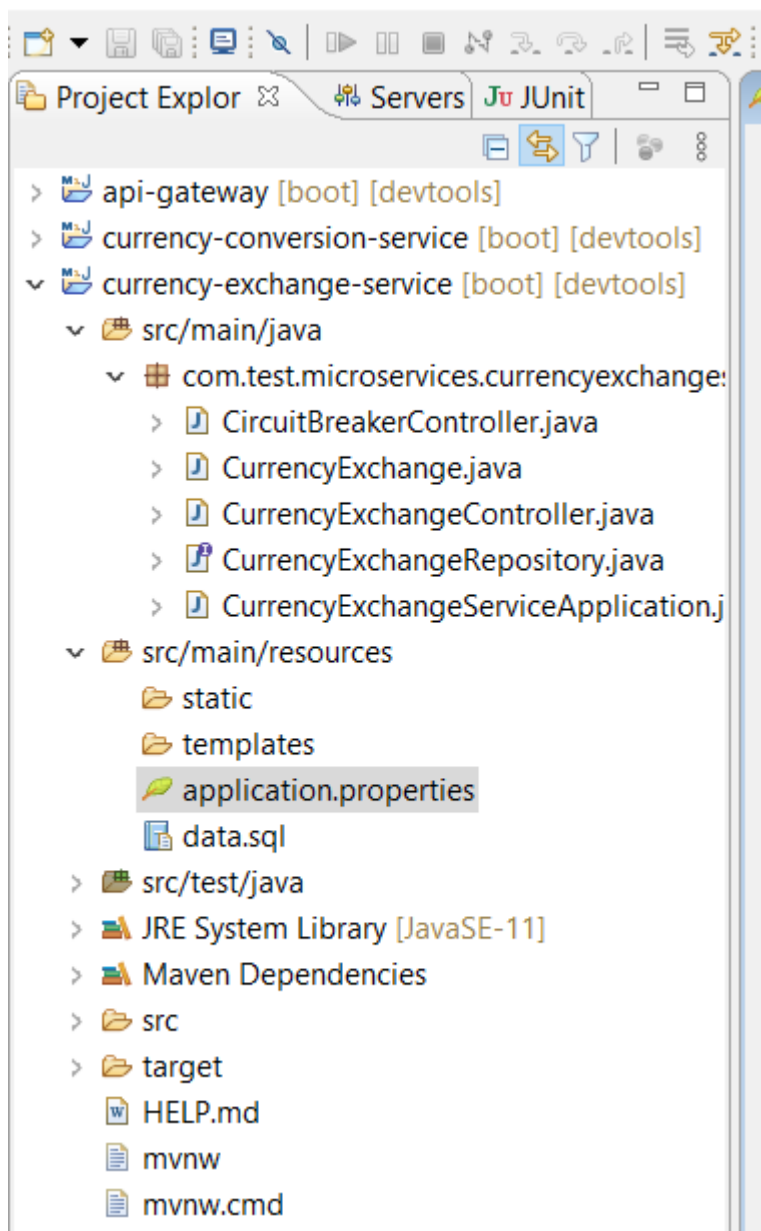


Currency exchange



```
application.properties x currency-exchange-service/pom.xml
1 ##application file for currency exchange service
2
3 spring.application.name=currency-exchange
4 server.port=8000
5 spring.config.import=optional:configserver:http://localhost:8888
6 #spring.config.import=optional:configserver:
7
8 #spring.cloud.config.enabled=false
9
10 spring.jpa.show-sql=true
11 spring.datasource.url=jdbc:h2:mem:testdb
12 #spring.datasource.url=jdbc:h2:mem:testdb;DB_CLOSE_ON_EXIT=FALSE
13 spring.h2.console.enabled=true
14 ##If the spring tries to load the data in the table before creation of the table (To change it we need to defe
15 spring.jpa.defer-datasource-initialization=true
16
17 #incase the app is not registered as client with Eureka
18 eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
19
20 #circuit breaker retry #NEW
21 resilience4j.retry.instances.sample-api.maxAttempts=5
22 #resilience4j.retry.instances.sample-api.max-attempts=5
23 resilience4j.retry.instances.sample-api.wait-duration=2s
24 resilience4j.retry.instances.sample-api.enable-exponential-backoff=true
25
26 #resilience4j.ratelimiter.instances.default.limit-for-period=2
27 #resilience4j.ratelimiter.instances.default.limit-refresh-period=10s
28
29 spring.sleuth.sampler.probability=1.0|
30
31
```

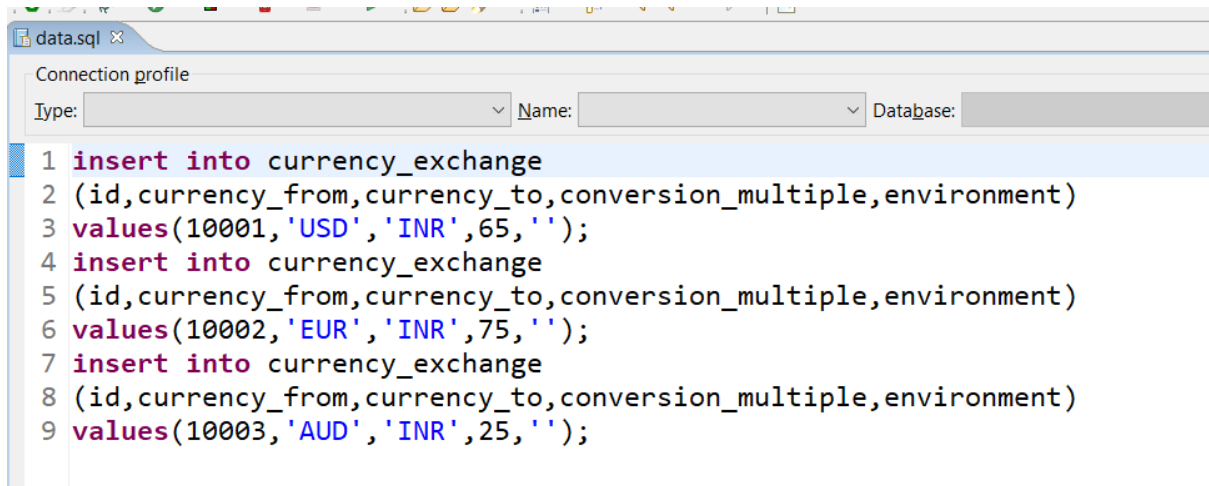
Pom

```
application.properties x currency-exchange-service/pom.xml x
22<dependencies>
23<!-- -->
24<dependency>
25<groupId>org.springframework.boot</groupId>
26<artifactId>spring-boot-starter-actuator</artifactId>
27</dependency>
28<dependency>
29<groupId>org.springframework.boot</groupId>
30<artifactId>spring-boot-starter-web</artifactId>
31</dependency>
32
33<dependency>
34<groupId>org.springframework.boot</groupId>
35<artifactId>spring-boot-starter-data-jpa</artifactId>
36</dependency>
37<dependency>
38<groupId>com.h2database</groupId>
39<artifactId>h2</artifactId>
40</dependency>
41<dependency>
42<groupId>org.springframework.cloud</groupId>
43<artifactId>spring-cloud-starter-config</artifactId>
44</dependency>
45<dependency>
46<groupId>org.springframework.cloud</groupId>
47<artifactId>spring-cloud-starter-sleuth</artifactId>
48</dependency>
49<dependency>
50<groupId>org.springframework.cloud</groupId>
51<artifactId>spring-cloud-sleuth-zipkin</artifactId>
52</dependency>
53</dependencies>
54</project>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

```
application.properties  currency-exchange-service/pom.xml
52      </dependency>
53      <dependency>
54          <groupId>org.springframework.amqp</groupId>
55          <artifactId>spring-rabbit</artifactId>
56      </dependency>
57
58      <dependency>
59          <groupId>org.springframework.cloud</groupId>
60          <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
61      </dependency>
62
63      <dependency>
64          <groupId>org.springframework.boot</groupId>
65          <artifactId>spring-boot-starter-aop</artifactId>
66      </dependency>
67      <dependency>
68          <groupId>io.github.resilience4j</groupId>
69          <artifactId>resilience4j-spring-boot2</artifactId>
70      </dependency>
71
72      <dependency>
73          <groupId>org.springframework.boot</groupId>
74          <artifactId>spring-boot-devtools</artifactId>
75          <scope>runtime</scope>
76          <optional>true</optional>
77      </dependency>
78      <dependency>
79          <groupId>org.springframework.boot</groupId>
80          <artifactId>spring-boot-starter-test</artifactId>
81          <scope>test</scope>
82      </dependency>
```

```
application.properties  currency-exchange-service/pom.xml
79          <groupId>org.springframework.boot</groupId>
80          <artifactId>spring-boot-starter-test</artifactId>
81          <scope>test</scope>
82      </dependency>
83
84  </dependencies>
85  <dependencyManagement>
86      <dependencies>
87          <dependency>
88              <groupId>org.springframework.cloud</groupId>
89              <artifactId>spring-cloud-dependencies</artifactId>
90              <version>${spring-cloud.version}</version>
91              <type>pom</type>
92              <scope>import</scope>
93          </dependency>
94      </dependencies>
95  </dependencyManagement>
96
97  <build>
98      <plugins>
99          <plugin>
100              <groupId>org.springframework.boot</groupId>
101              <artifactId>spring-boot-maven-plugin</artifactId>
102              <configuration>
103                  <image>
104                      <name>
105                          chaitanya510/ms.v1-${project.artifactId}:${project.version}
106                      </name>
107                  </image>
108                  <pullPolicy>IF_NOT_PRESENT</pullPolicy>
109              </configuration>
```



The image shows a screenshot of a SQL editor window. The window has a title bar with a file icon and the name 'data.sql'. Below the title bar is a 'Connection profile' section with fields for 'Type:', 'Name:', and 'Database:'. The main area of the window contains three SQL INSERT statements, each on a new line and numbered 1 through 9. The statements are: 1. 'insert into currency_exchange', 2. '(id,currency_from,currency_to,conversion_multiple,environment)', 3. 'values(10001, 'USD', 'INR', 65, '')';, 4. 'insert into currency_exchange', 5. '(id,currency_from,currency_to,conversion_multiple,environment)', 6. 'values(10002, 'EUR', 'INR', 75, '')';, 7. 'insert into currency_exchange', 8. '(id,currency_from,currency_to,conversion_multiple,environment)', 9. 'values(10003, 'AUD', 'INR', 25, '')';. The text is color-coded: 'insert into' is in purple, 'values' is in blue, and the currency codes 'USD', 'EUR', 'AUD' are in blue. The numbers 10001, 10002, and 10003 are in black. The conversion multiples 65, 75, and 25 are in black. The environment field is empty in all three statements.

```
1 insert into currency_exchange
2 (id,currency_from,currency_to,conversion_multiple,environment)
3 values(10001, 'USD', 'INR', 65, '');
4 insert into currency_exchange
5 (id,currency_from,currency_to,conversion_multiple,environment)
6 values(10002, 'EUR', 'INR', 75, '');
7 insert into currency_exchange
8 (id,currency_from,currency_to,conversion_multiple,environment)
9 values(10003, 'AUD', 'INR', 25, '');
```