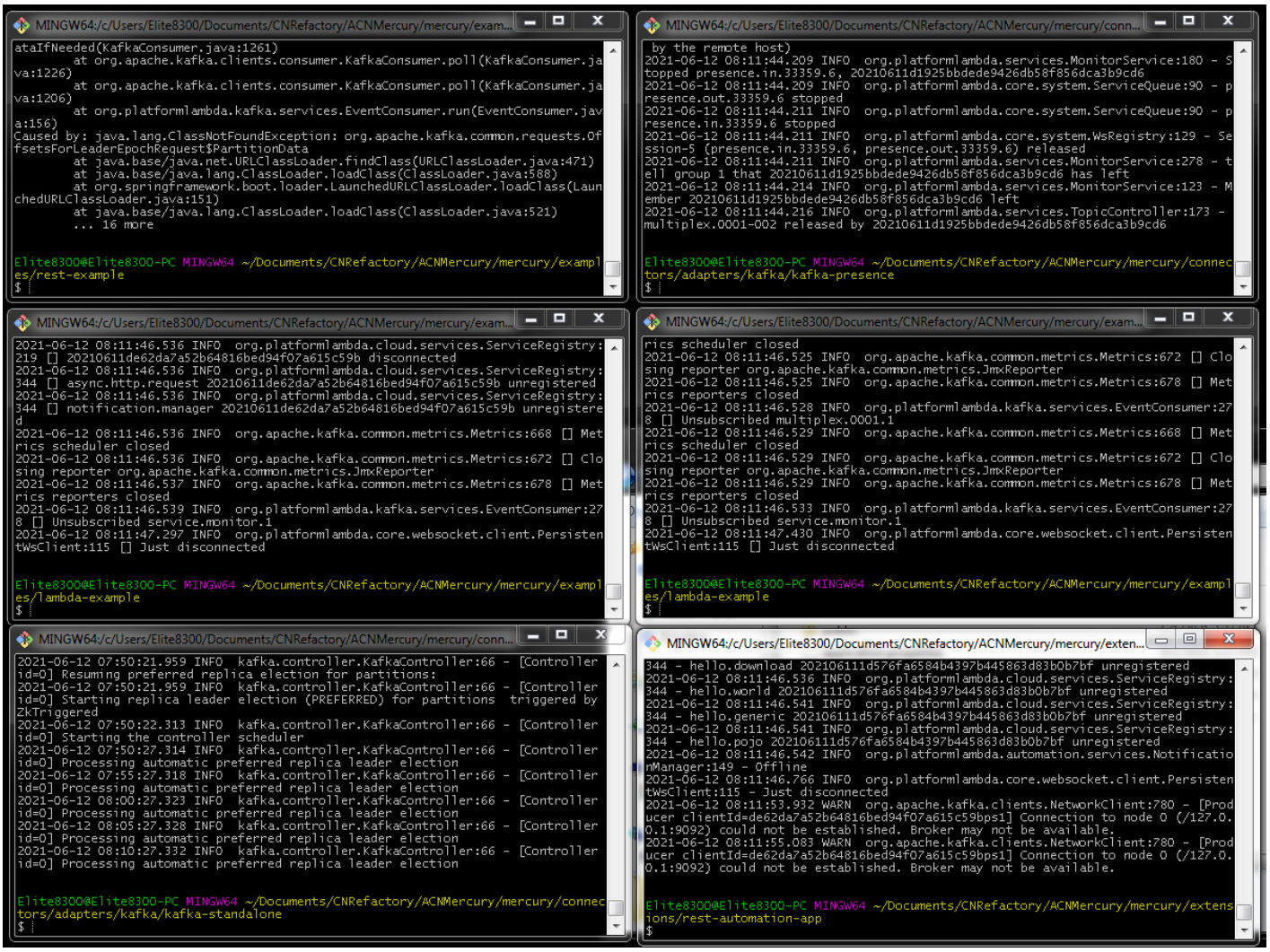
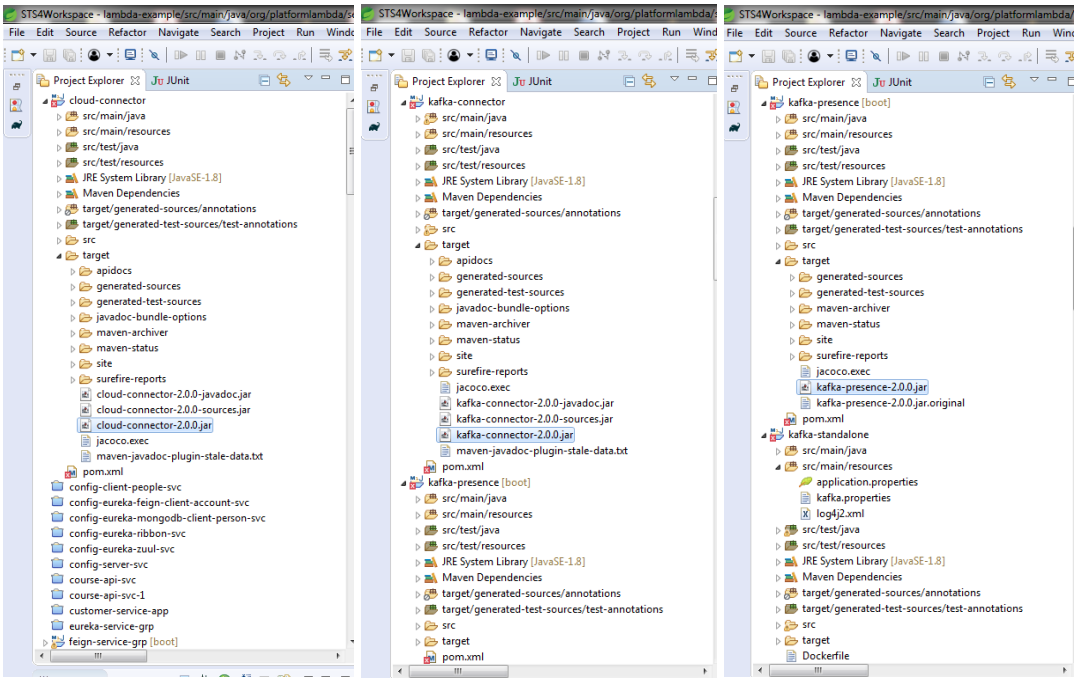
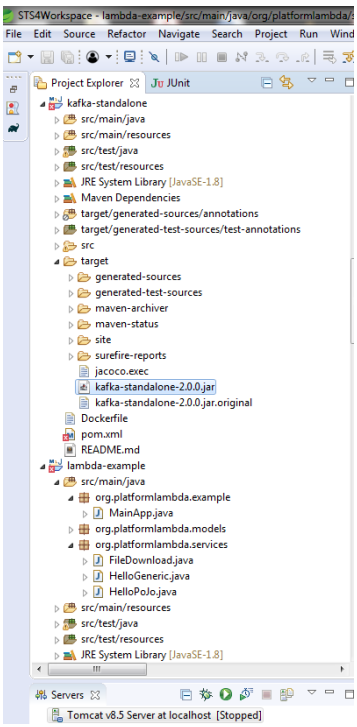
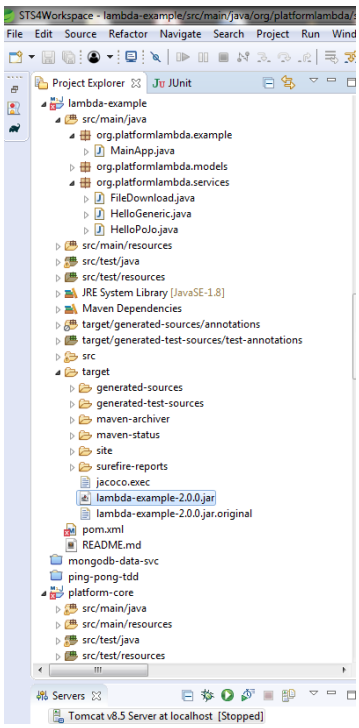
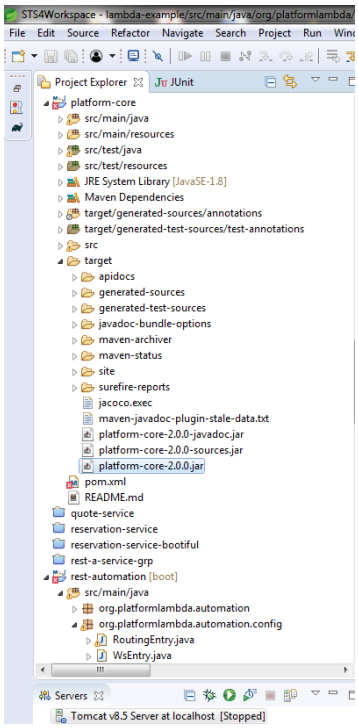
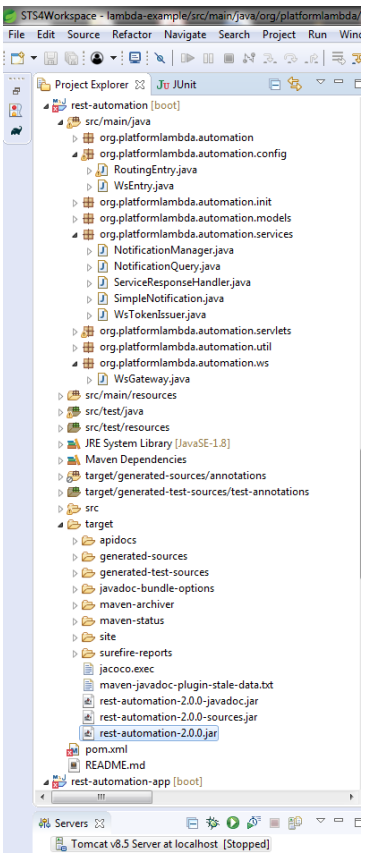

MINGW64:/c/Users/Elite8300/Documents/CNRefactory/ACNMercury/mercury/exam...
ataIfNeeded(KafkaConsumer.java:1261)
at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1226)
at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1206)
at org.platformlambda.kafka.services.EventConsumer.run(EventConsumer.java:156)
Caused by: java.lang.ClassNotFoundException: org.apache.kafka.common.requests.OffsetsForLeaderEpochRequest$PartitionData
at java.base/java.net.URLClassLoader.findClass(URLClassLoader.java:471)
at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:588)
at org.springframework.boot.loader.LaunchedURLClassLoader.loadClass(LaunchedURLClassLoader.java:151)
at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
... 16 more
Elite8300@Elite8300-PC MINGW64 ~/Documents/CNRefactory/ACNMercury/mercury/examples/rest-example
$
MINGW64:/c/Users/Elite8300/Documents/CNRefactory/ACNMercury/mercury/conn...
by the remote host)
2021-06-12 08:11:44.209 INFO org.platformlambda.services.MonitorService:180 - Stopped presence.in.33359.6, 20210611d1925bbdede9426db58f856dca3b9cd6
2021-06-12 08:11:44.209 INFO org.platformlambda.core.system.ServiceQueue:90 - presence.out.33359.6 stopped
2021-06-12 08:11:44.211 INFO org.platformlambda.core.system.ServiceQueue:90 - presence.in.33359.6 stopped
2021-06-12 08:11:44.211 INFO org.platformlambda.core.system.WsRegistry:129 - Session-5 (presence.in.33359.6, presence.out.33359.6) released
2021-06-12 08:11:44.211 INFO org.platformlambda.services.MonitorService:278 - tell group 1 that 20210611d1925bbdede9426db58f856dca3b9cd6 has left
2021-06-12 08:11:44.214 INFO org.platformlambda.services.MonitorService:123 - Member 20210611d1925bbdede9426db58f856dca3b9cd6 left
2021-06-12 08:11:44.216 INFO org.platformlambda.services.TopicController:173 - multiplex.0001-002 released by 20210611d1925bbdede9426db58f856dca3b9cd6
Elite8300@Elite8300-PC MINGW64 ~/Documents/CNRefactory/ACNMercury/mercury/connectors/adapters/kafka/kafka-presence
$
MINGW64:/c/Users/Elite8300/Documents/CNRefactory/ACNMercury/mercury/exam...
2021-06-12 08:11:46.536 INFO org.platformlambda.cloud.services.ServiceRegistry:219 [] 20210611de62da7a52b64816bed94f07a615c59b disconnected
2021-06-12 08:11:46.536 INFO org.platformlambda.cloud.services.ServiceRegistry:344 [] async.http.request 20210611de62da7a52b64816bed94f07a615c59b unregistered
2021-06-12 08:11:46.536 INFO org.platformlambda.cloud.services.ServiceRegistry:344 [] notification.manager 20210611de62da7a52b64816bed94f07a615c59b unregistered
2021-06-12 08:11:46.536 INFO org.apache.kafka.common.metrics.Metrics:668 [] Metrics scheduler closed
2021-06-12 08:11:46.536 INFO org.apache.kafka.common.metrics.Metrics:672 [] Closing reporter org.apache.kafka.common.metrics.JmxReporter
2021-06-12 08:11:46.537 INFO org.apache.kafka.common.metrics.Metrics:678 [] Metrics reporters closed
2021-06-12 08:11:46.539 INFO org.platformlambda.kafka.services.EventConsumer:278 [] Unsubscribed service.monitor.1
2021-06-12 08:11:47.297 INFO org.platformlambda.core.websocket.client.PersistentWsClient:115 [] Just disconnected
Elite8300@Elite8300-PC MINGW64 ~/Documents/CNRefactory/ACNMercury/mercury/examples/lambda-example
$
MINGW64:/c/Users/Elite8300/Documents/CNRefactory/ACNMercury/mercury/exam...
rics scheduler closed
2021-06-12 08:11:46.525 INFO org.apache.kafka.common.metrics.Metrics:672 [] Closing reporter org.apache.kafka.common.metrics.JmxReporter
2021-06-12 08:11:46.525 INFO org.apache.kafka.common.metrics.Metrics:678 [] Metrics reporters closed
2021-06-12 08:11:46.528 INFO org.platformlambda.kafka.services.EventConsumer:278 [] Unsubscribed multiplex.0001.1
2021-06-12 08:11:46.529 INFO org.apache.kafka.common.metrics.Metrics:668 [] Metrics scheduler closed
2021-06-12 08:11:46.529 INFO org.apache.kafka.common.metrics.Metrics:672 [] Closing reporter org.apache.kafka.common.metrics.JmxReporter
2021-06-12 08:11:46.529 INFO org.apache.kafka.common.metrics.Metrics:678 [] Metrics reporters closed
2021-06-12 08:11:46.533 INFO org.platformlambda.kafka.services.EventConsumer:278 [] Unsubscribed service.monitor.1
2021-06-12 08:11:47.430 INFO org.platformlambda.core.websocket.client.PersistentWsClient:115 [] Just disconnected
Elite8300@Elite8300-PC MINGW64 ~/Documents/CNRefactory/ACNMercury/mercury/examples/lambda-example
$
MINGW64:/c/Users/Elite8300/Documents/CNRefactory/ACNMercury/mercury/conn...
2021-06-12 07:50:21.959 INFO kafka.controller.KafkaController:66 - [Controller id=0] Resuming preferred replica election for partitions:
2021-06-12 07:50:21.959 INFO kafka.controller.KafkaController:66 - [Controller id=0] Starting replica leader election (PREFERRED) for partitions triggered by ZkTriggered
2021-06-12 07:50:22.313 INFO kafka.controller.KafkaController:66 - [Controller id=0] Starting the controller scheduler
2021-06-12 07:50:27.314 INFO kafka.controller.KafkaController:66 - [Controller id=0] Processing automatic preferred replica leader election
2021-06-12 07:55:27.318 INFO kafka.controller.KafkaController:66 - [Controller id=0] Processing automatic preferred replica leader election
2021-06-12 08:00:27.323 INFO kafka.controller.KafkaController:66 - [Controller id=0] Processing automatic preferred replica leader election
2021-06-12 08:05:27.328 INFO kafka.controller.KafkaController:66 - [Controller id=0] Processing automatic preferred replica leader election
2021-06-12 08:10:27.332 INFO kafka.controller.KafkaController:66 - [Controller id=0] Processing automatic preferred replica leader election
Elite8300@Elite8300-PC MINGW64 ~/Documents/CNRefactory/ACNMercury/mercury/connectors/adapters/kafka/kafka-standalone
$
MINGW64:/c/Users/Elite8300/Documents/CNRefactory/ACNMercury/mercury/exten...
344 - hello.download 20210611d576fa6584b4397b445863d83b0b7bf unregistered
2021-06-12 08:11:46.536 INFO org.platformlambda.cloud.services.ServiceRegistry:344 - hello.world 20210611d576fa6584b4397b445863d83b0b7bf unregistered
2021-06-12 08:11:46.541 INFO org.platformlambda.cloud.services.ServiceRegistry:344 - hello.generic 20210611d576fa6584b4397b445863d83b0b7bf unregistered
2021-06-12 08:11:46.541 INFO org.platformlambda.cloud.services.ServiceRegistry:344 - hello.pojo 20210611d576fa6584b4397b445863d83b0b7bf unregistered
2021-06-12 08:11:46.542 INFO org.platformlambda.automation.services.NotificationManager:149 - Offline
2021-06-12 08:11:46.766 INFO org.platformlambda.core.websocket.client.PersistentWsClient:115 - Just disconnected
2021-06-12 08:11:53.932 WARN org.apache.kafka.clients.NetworkClient:780 - [Producer clientId=de62da7a52b64816bed94f07a615c59bps1] Connection to node 0 (/127.0.0.1:9092) could not be established. Broker may not be available.
2021-06-12 08:11:55.083 WARN org.apache.kafka.clients.NetworkClient:780 - [Producer clientId=de62da7a52b64816bed94f07a615c59bps1] Connection to node 0 (/127.0.0.1:9092) could not be established. Broker may not be available.
Elite8300@Elite8300-PC MINGW64 ~/Documents/CNRefactory/ACNMercury/mercury/extensions/rest-automation-app
$

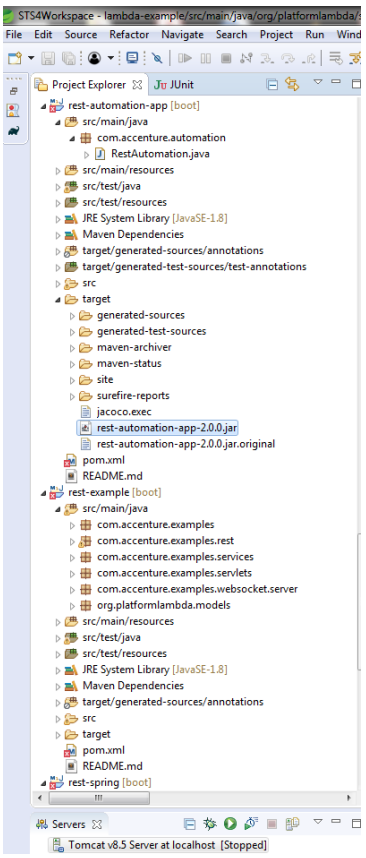

STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run
Project Explorer
JUnit
cloud-connector
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
apidocs
generated-sources
generated-test-sources
javadoc-bundle-options
maven-archiver
maven-status
site
surefire-reports
cloud-connector-2.0.0-javadoc.jar
cloud-connector-2.0.0-sources.jar
cloud-connector-2.0.0.jar
jacoco.exec
maven-javadoc-plugin-stale-data.txt
pom.xml
config-client-people-svc
config-eureka-feign-client-account-svc
config-eureka-mongodb-client-person-svc
config-eureka-ribbon-svc
config-eureka-zuul-svc
config-server-svc
course-api-svc
course-api-svc-1
customer-service-app
eureka-service-grp
feign-service-grp [boot]
kafka-connector
kafka-connector-2.0.0-javadoc.jar
kafka-connector-2.0.0-sources.jar
kafka-connector-2.0.0.jar
kafka-presence [boot]
kafka-presence-2.0.0.jar.original
kafka-standalone
application.properties
kafka.properties
log4j2.xml
Dockerfile

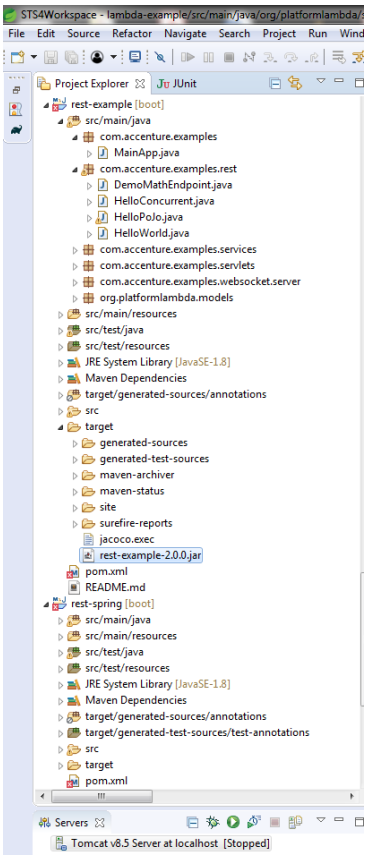

STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run Wind
Project Explorer
JUnit
kafka-standalone
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
generated-sources
generated-test-sources
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
kafka-standalone-2.0.0.jar
kafka-standalone-2.0.0.jar.original
Dockerfile
pom.xml
README.md
lambda-example
src/main/java
org.platformlambda.example
MainApp.java
org.platformlambda.models
org.platformlambda.services
FileDownload.java
HelloGeneric.java
HelloPoJo.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Servers
Tomcat v8.5 Server at localhost [Stopped]

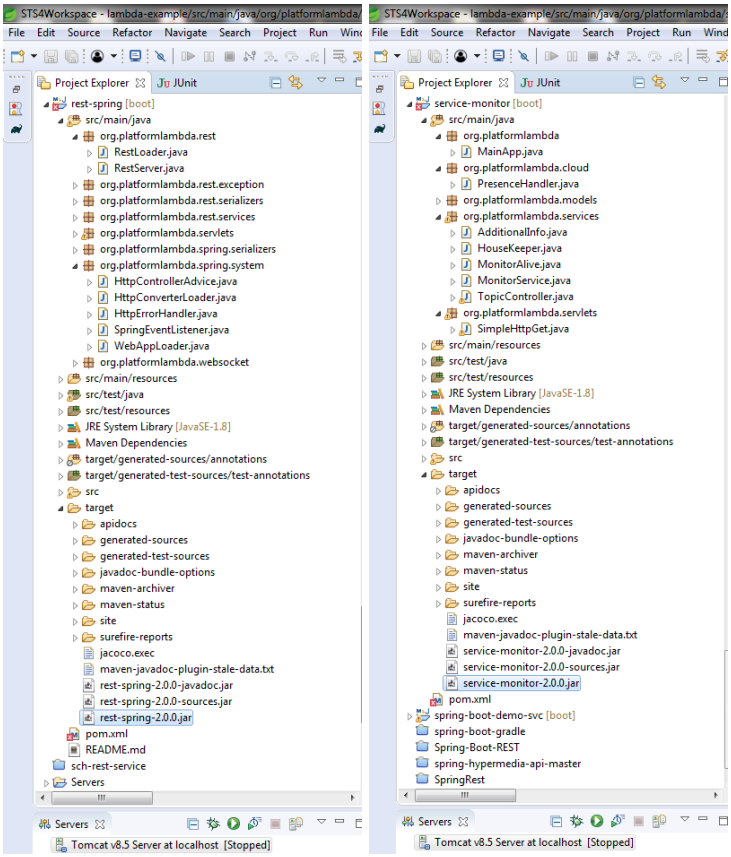

STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run Wind
Project Explorer
JUnit
lambda-example
src/main/java
org.platformlambda.example
MainApp.java
org.platformlambda.models
org.platformlambda.services
FileDownload.java
HelloGeneric.java
HelloPoJo.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
generated-sources
generated-test-sources
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
lambda-example-2.0.0.jar
lambda-example-2.0.0.jar.original
pom.xml
README.md
mongodb-data-svc
ping-pong-tdd
platform-core
src/main/java
src/main/resources
src/test/java
src/test/resources
Servers
Tomcat v8.5 Server at localhost [Stopped]


STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run Wind
Project Explorer
JUnit
platform-core
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
apidocs
generated-sources
generated-test-sources
javadoc-bundle-options
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
maven-javadoc-plugin-stale-data.txt
platform-core-2.0.0-javadoc.jar
platform-core-2.0.0-sources.jar
platform-core-2.0.0.jar
pom.xml
README.md
quote-service
reservation-service
reservation-service-bootiful
rest-a-service-grp
rest-automation [boot]
src/main/java
org.platformlambda.automation
org.platformlambda.automation.config
RoutingEntry.java
WsEntry.java
Servers
Tomcat v8.5 Server at localhost [Stopped]

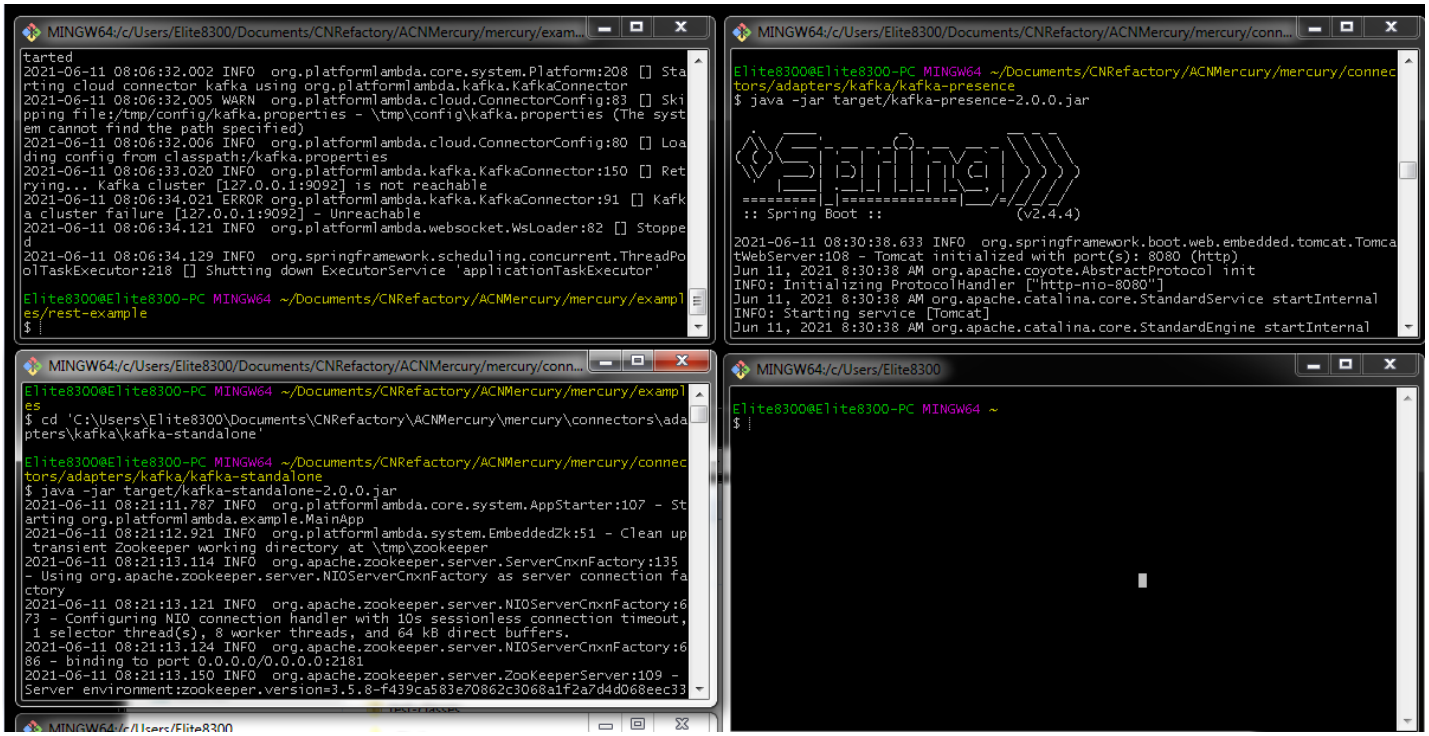

STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run Wind
Project Explorer
JUnit
rest-automation [boot]
src/main/java
org.platformlambda.automation
org.platformlambda.automation.config
RoutingEntry.java
WsEntry.java
org.platformlambda.automation.init
org.platformlambda.automation.models
org.platformlambda.automation.services
NotificationManager.java
NotificationQuery.java
ServiceResponseHandler.java
SimpleNotification.java
WsTokenIssuer.java
org.platformlambda.automation.servlets
org.platformlambda.automation.util
org.platformlambda.automation.ws
WsGateway.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
apidocs
generated-sources
generated-test-sources
javadoc-bundle-options
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
maven-javadoc-plugin-stale-data.txt
rest-automation-2.0.0-javadoc.jar
rest-automation-2.0.0-sources.jar
rest-automation-2.0.0.jar
pom.xml
README.md
rest-automation-app [boot]
Servers
Tomcat v8.5 Server at localhost [Stopped]

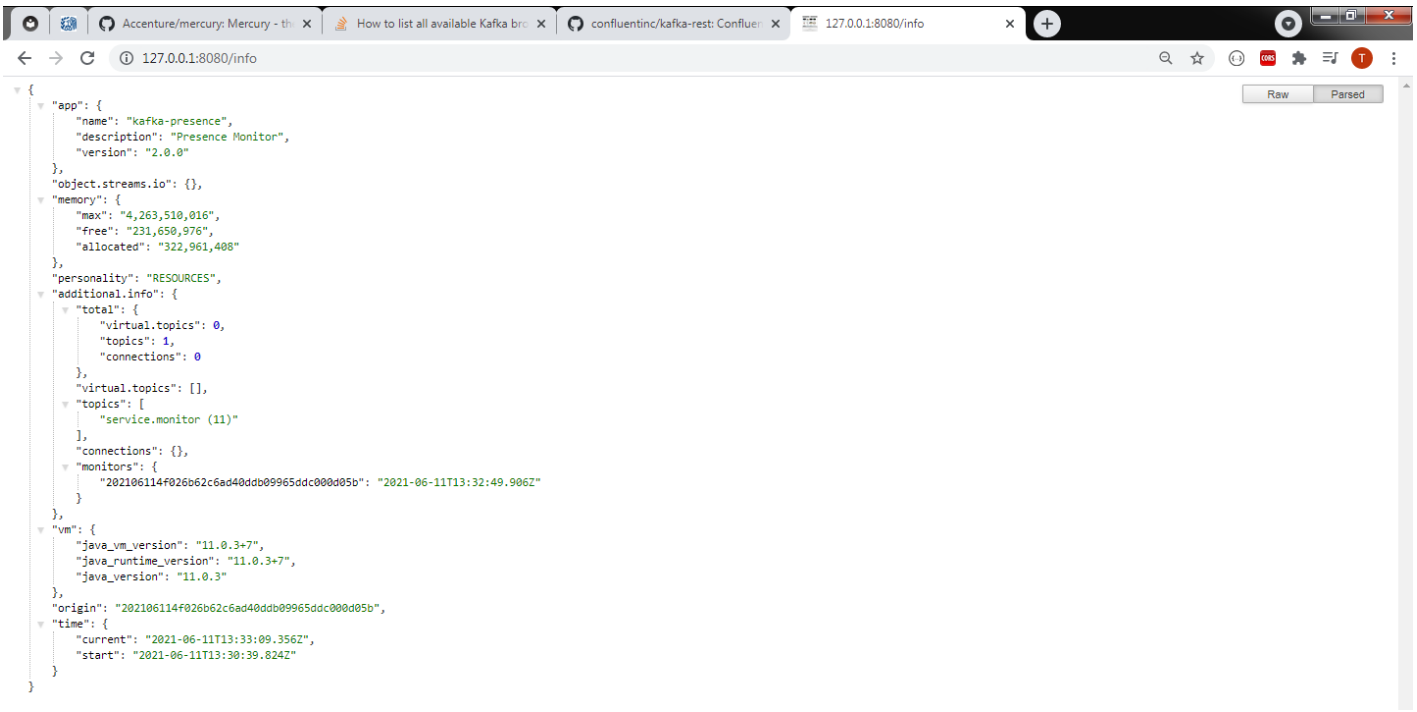

STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run Wind
Project Explorer
JUnit
rest-automation-app [boot]
src/main/java
com.accenture.automation
RestAutomation.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
generated-sources
generated-test-sources
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
rest-automation-app-2.0.0.jar
rest-automation-app-2.0.0.jar.original
pom.xml
README.md
rest-example [boot]
src/main/java
com.accenture.examples
com.accenture.examples.rest
com.accenture.examples.services
com.accenture.examples.servlets
com.accenture.examples.websocket.server
org.platformlambda.models
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
pom.xml
README.md
rest-spring [boot]
Servers
Tomcat v8.5 Server at localhost [Stopped]


STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run Wind
Project Explorer
JUnit
rest-example [boot]
src/main/java
com.accenture.examples
MainApp.java
com.accenture.examples.rest
DemoMathEndpoint.java
HelloConcurrent.java
HelloPoJo.java
HelloWorld.java
com.accenture.examples.services
com.accenture.examples.servlets
com.accenture.examples.websocket.server
org.platformlambda.models
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
src
target
generated-sources
generated-test-sources
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
rest-example-2.0.0.jar
pom.xml
README.md
rest-spring [boot]
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
pom.xml
Servers
Tomcat v8.5 Server at localhost [Stopped]

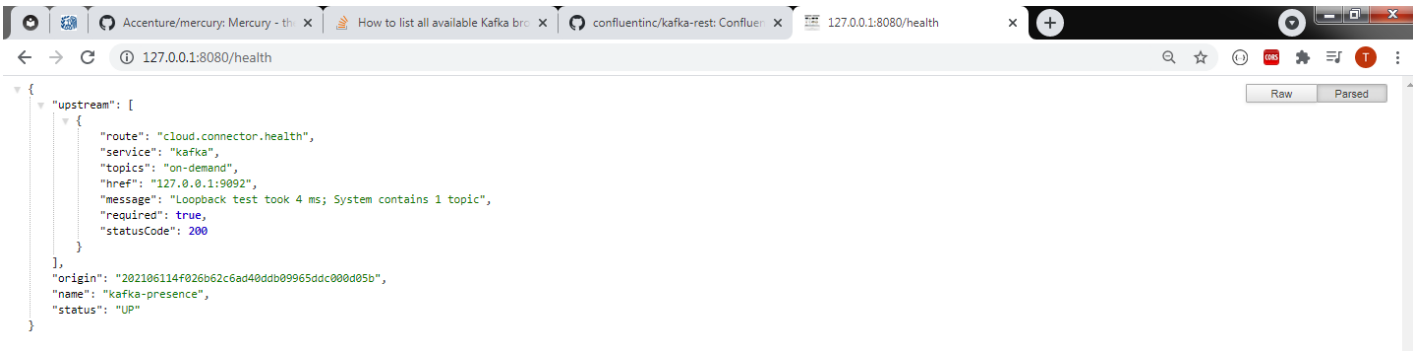

STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run
Project Explorer
JUnit
rest-spring [boot]
src/main/java
org.platformlambda.rest
RestLoader.java
RestServer.java
org.platformlambda.rest.exception
org.platformlambda.rest.serializers
org.platformlambda.rest.services
org.platformlambda.servlets
org.platformlambda.spring.serializers
org.platformlambda.spring.system
HttpControllerAdvice.java
HttpConverterLoader.java
HttpErrorHandler.java
SpringEventListener.java
WebAppLoader.java
org.platformlambda.websocket
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
apidocs
generated-sources
generated-test-sources
javadoc-bundle-options
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
maven-javadoc-plugin-stale-data.txt
rest-spring-2.0.0-javadoc.jar
rest-spring-2.0.0-sources.jar
rest-spring-2.0.0.jar
pom.xml
README.md
sch-rest-service
Servers
Tomcat v8.5 Server at localhost [Stopped]
STS4Workspace - lambda-example/src/main/java/org/platformlambda/
File Edit Source Refactor Navigate Search Project Run
Project Explorer
JUnit
service-monitor [boot]
src/main/java
org.platformlambda
MainApp.java
org.platformlambda.cloud
PresenceHandler.java
org.platformlambda.models
org.platformlambda.services
AdditionalInfo.java
HouseKeeper.java
MonitorAlive.java
MonitorService.java
TopicController.java
org.platformlambda.servlets
SimpleHttpGet.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
apidocs
generated-sources
generated-test-sources
javadoc-bundle-options
maven-archiver
maven-status
site
surefire-reports
jacoco.exec
maven-javadoc-plugin-stale-data.txt
service-monitor-2.0.0-javadoc.jar
service-monitor-2.0.0-sources.jar
service-monitor-2.0.0.jar
pom.xml
spring-boot-demo-svc [boot]
spring-boot-gradle
Spring-Boot-REST
spring-hypermedia-api-master
SpringRest
Servers
Tomcat v8.5 Server at localhost [Stopped]

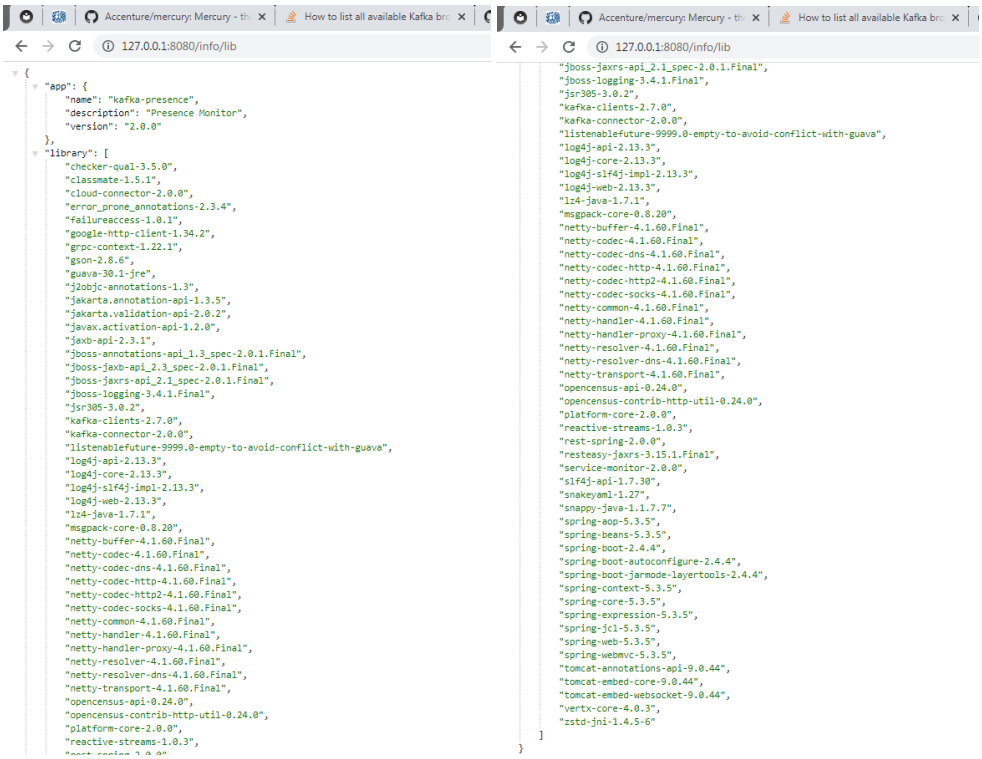

STS4Workspace - kafka-standalone/src/main/resources/kafka.properties - Spring Tool Suite 4
File Edit Navigate Search Project Run Window Help
Quick Access
Project Explorer
JUnit
Console
platform-core/pom.xml
rest-example/pom.xml
rest-spring/pom.xml
application.properties
kafka.properties
kafka-presence [boot]
kafka-standalone
application.properties
kafka.properties
log4j2.xml
Dockerfile
mongodb-data-svc
ping-pong-tdd
platform-core
Servers
Tomcat v8.5 Server at localhost [Stopped]
# For anything other than development testing, a value greater than 1 is recommended for to ensure availability such as 3.
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1
############################# Log Flush Policy #############################
Messages are immediately written to the filesystem but by default we only fsync() to sync
the OS cache lazily. The following configurations control the flush of data to disk.
There are a few important trade-offs here:
1. Durability: Unflushed data may be lost if you are not using replication.
2. Latency: Very large flush intervals may lead to latency spikes when the flush does occur as there will be a lot of data to flush.
3. Throughput: The flush is generally the most expensive operation, and a small flush interval may lead to excessive seeks.
The settings below allow one to configure the flush policy to flush data after a period of time or
every N messages (or both). This can be done globally and overridden on a per-topic basis.
The number of messages to accept before forcing a flush of data to disk
#log.flush.interval.messages=10000
The maximum amount of time a message can sit in a log before we force a flush
#log.flush.interval.ms=1000
############################# Log Retention Policy #############################
# The following configurations control the disposal of log segments. The policy can
# be set to delete segments after a period of time, or after a given size has accumulated.
# A segment will be deleted whenever *either* of these criteria are met. Deletion always happens
# from the end of the log.
# The minimum age of a log file to be eligible for deletion due to age
log.retention.hours=168
# A size-based retention policy for logs. Segments are pruned from the log unless the remaining
# segments drop below log.retention.bytes. Functions independently of log.retention.hours.
#log.retention.bytes=1073741824
# The maximum size of a log segment file. When this size is reached a new log segment will be created.
log.segment.bytes=1073741824
# The interval at which log segments are checked to see if they can be deleted according
# to the retention policies
log.retention.check.interval.ms=300000
############################# Zookeeper #############################
# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, each corresponding to a zk
# server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to the urls to specify the
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181
# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000
############################# Group Coordinator Settings #############################
The following configuration specifies the time, in milliseconds, that the GroupCoordinator will delay the initial consumer rebalance.
The rebalance will be further delayed by the value of group.initial.rebalance.delay.ms as new members join the group, up to a maximum of max.poll.interval.ms
The default value for this is 3 seconds.
We override this to 0 here as it makes for a better out-of-the-box experience for development and testing.
Network 6
Internet access
Writable
Insert
36 : 48

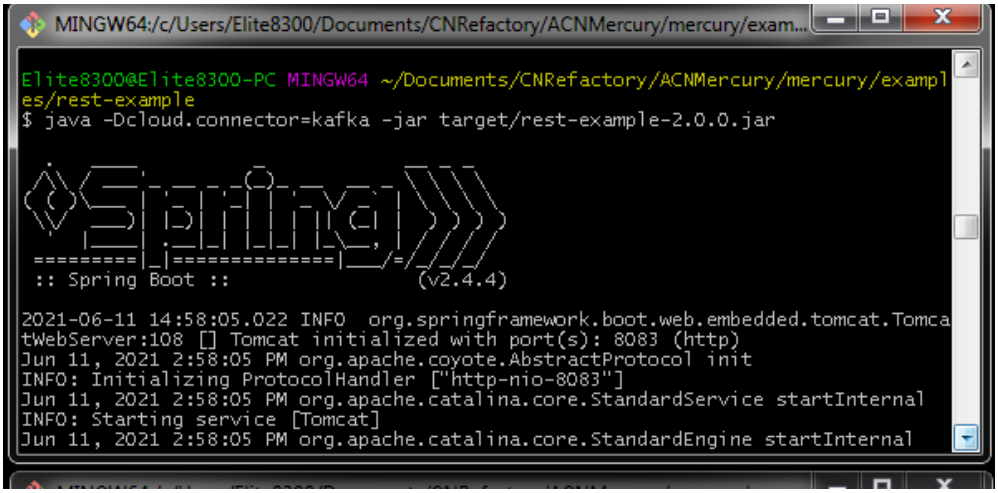




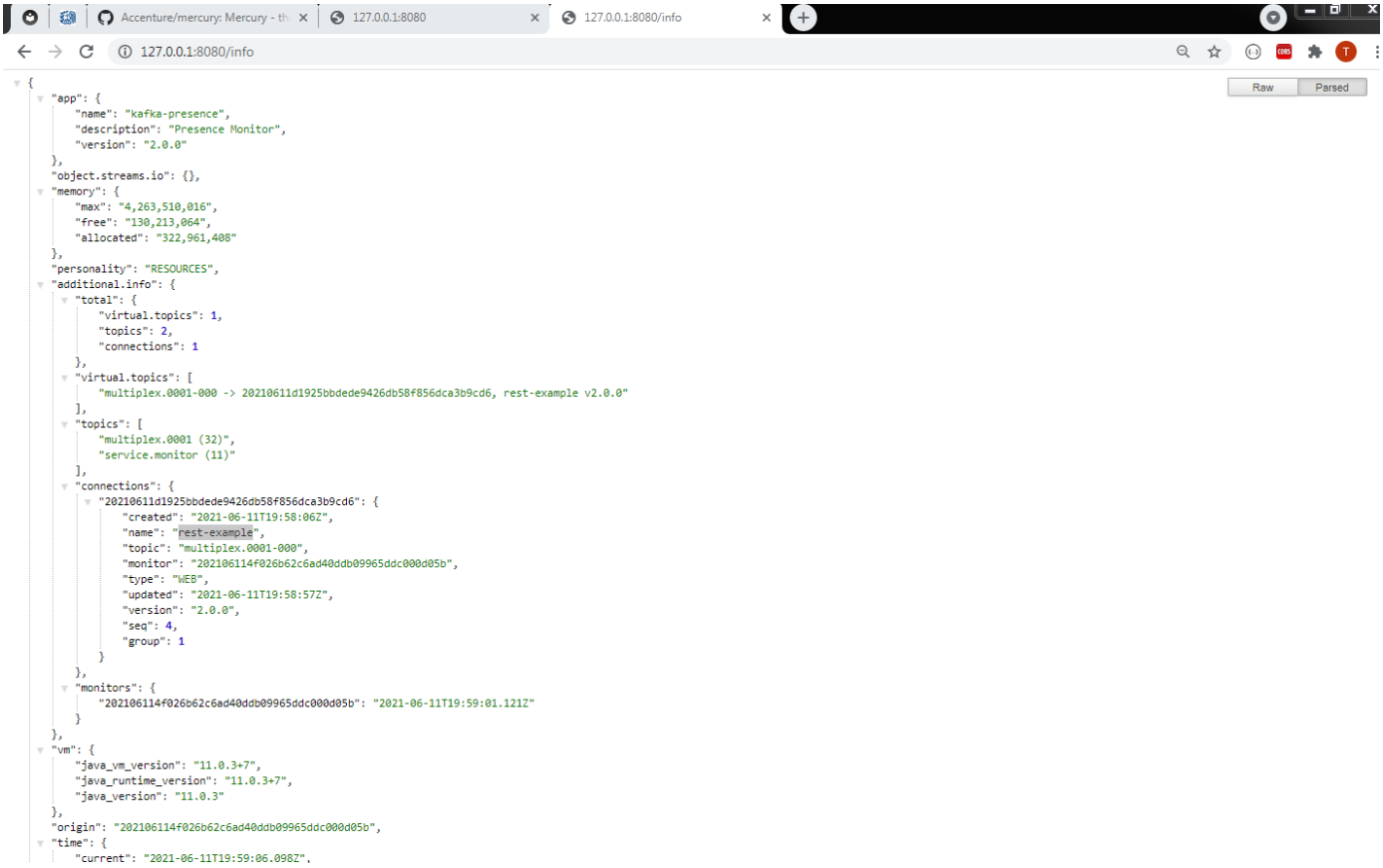
This is the kafka Broker_0 information at 127.0.0.1:8080/info



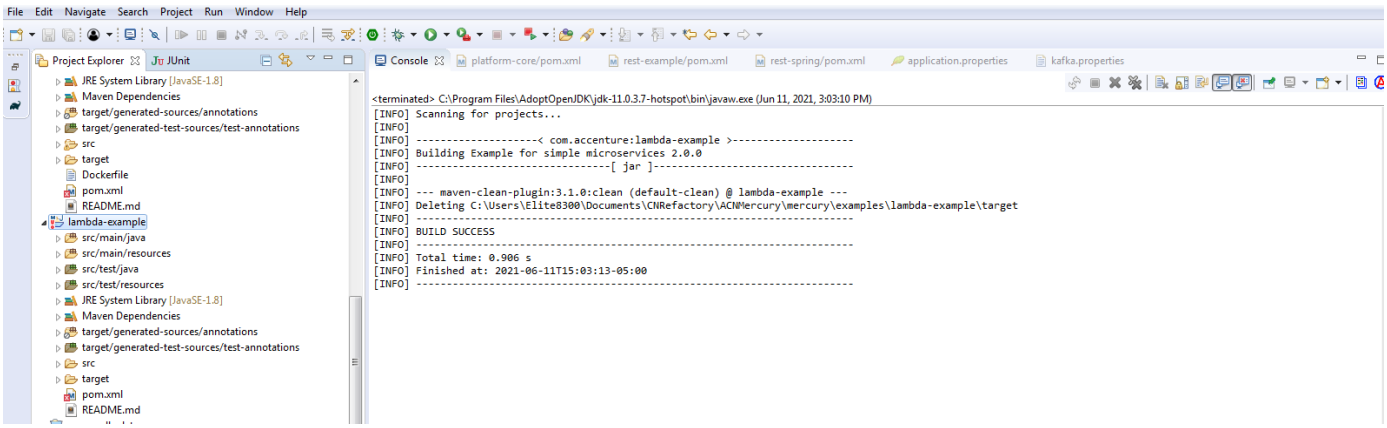




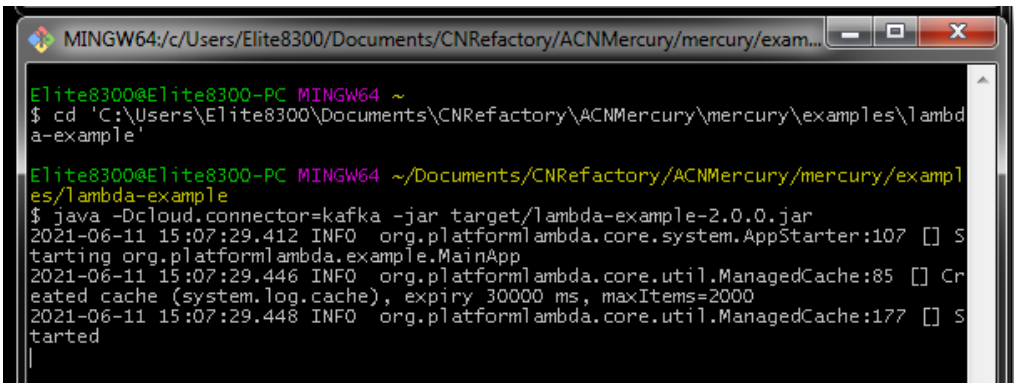
Start the rest-example using the JAR file



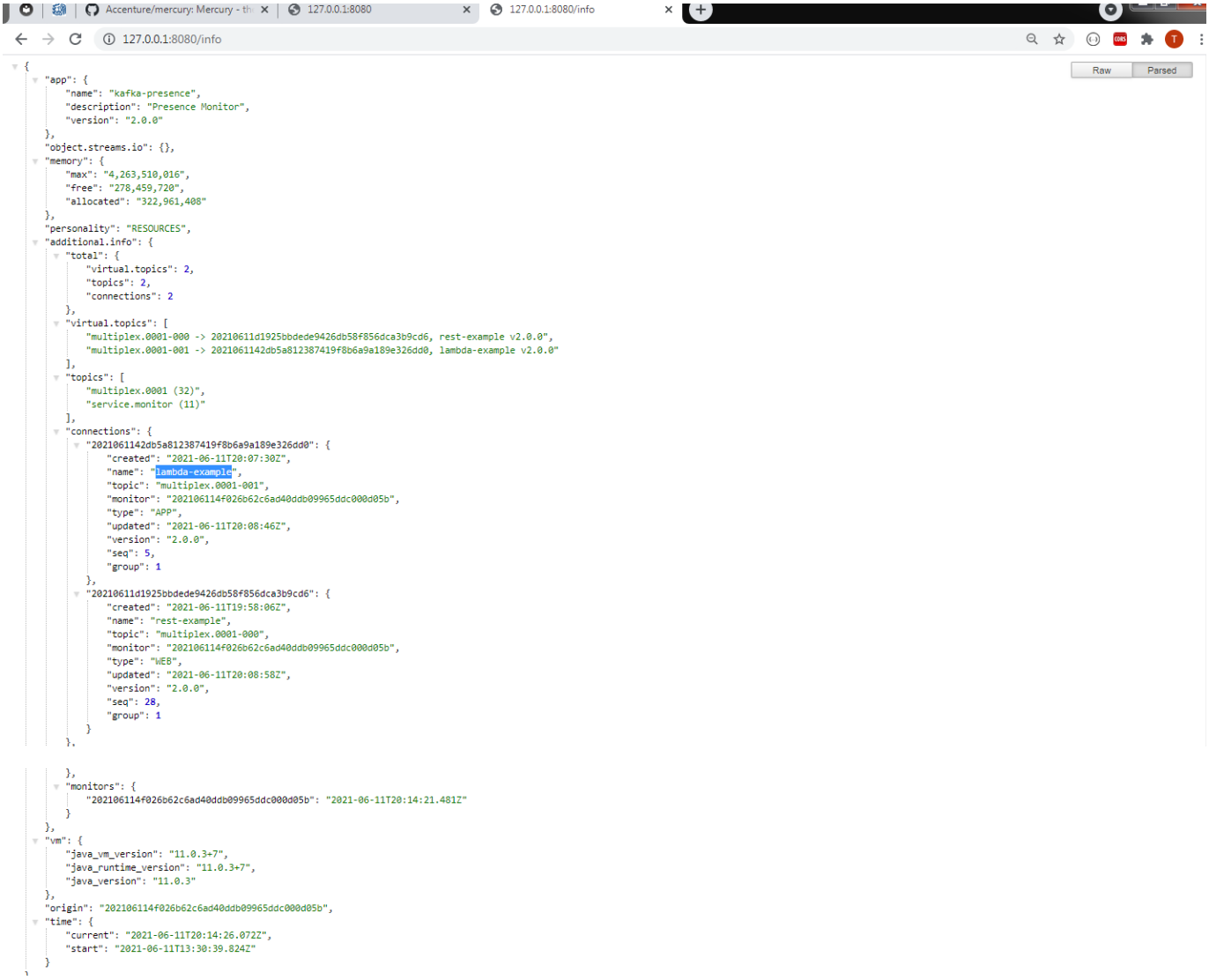
The rest-example app is now connected over the multiplex.0001.000 channel



Do a Maven clean & install for the lambda-example project to get the JAR file created

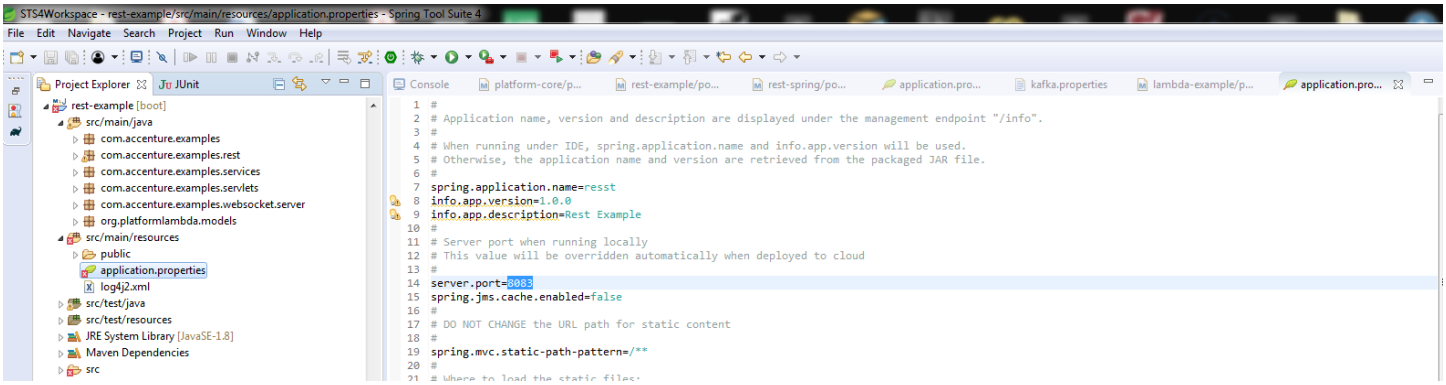


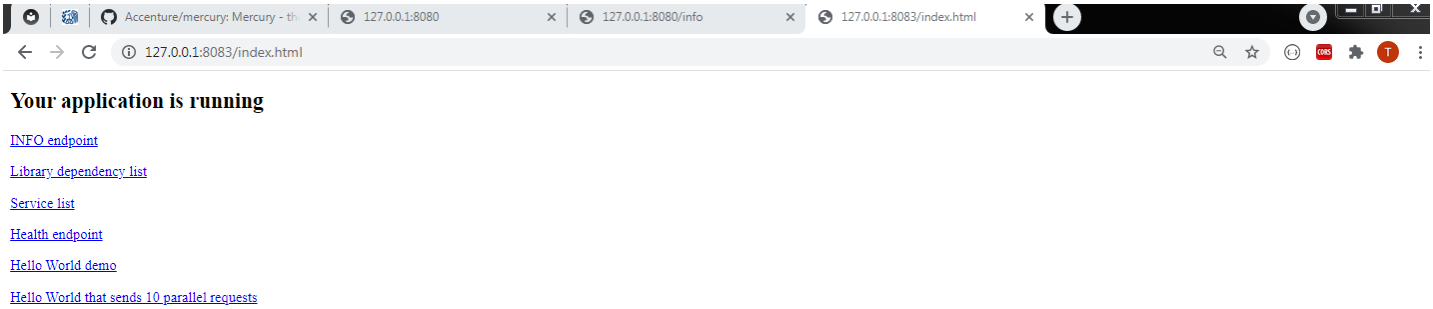
Run the lambda-example-2.0.0.jar file using the **$ java -Dcloud.connector=kafka -jar target/lambda-example-2.0.0.jar**



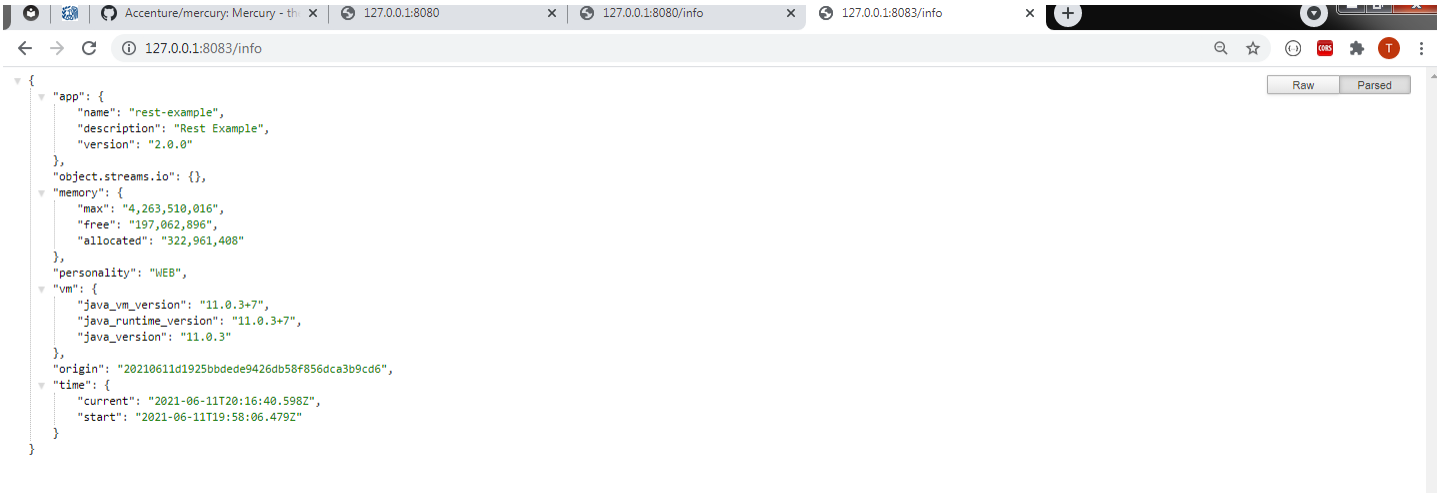
Both the rest-example web app and the lambda-example app are now running and detected by the presence monitor and assigned different channels

	Application Name	App Type	Copies	Virtual Topic
1	rest-example	Web		Multiplex.0001.000
2	Lambda-example	App		Multiplex.0001.001

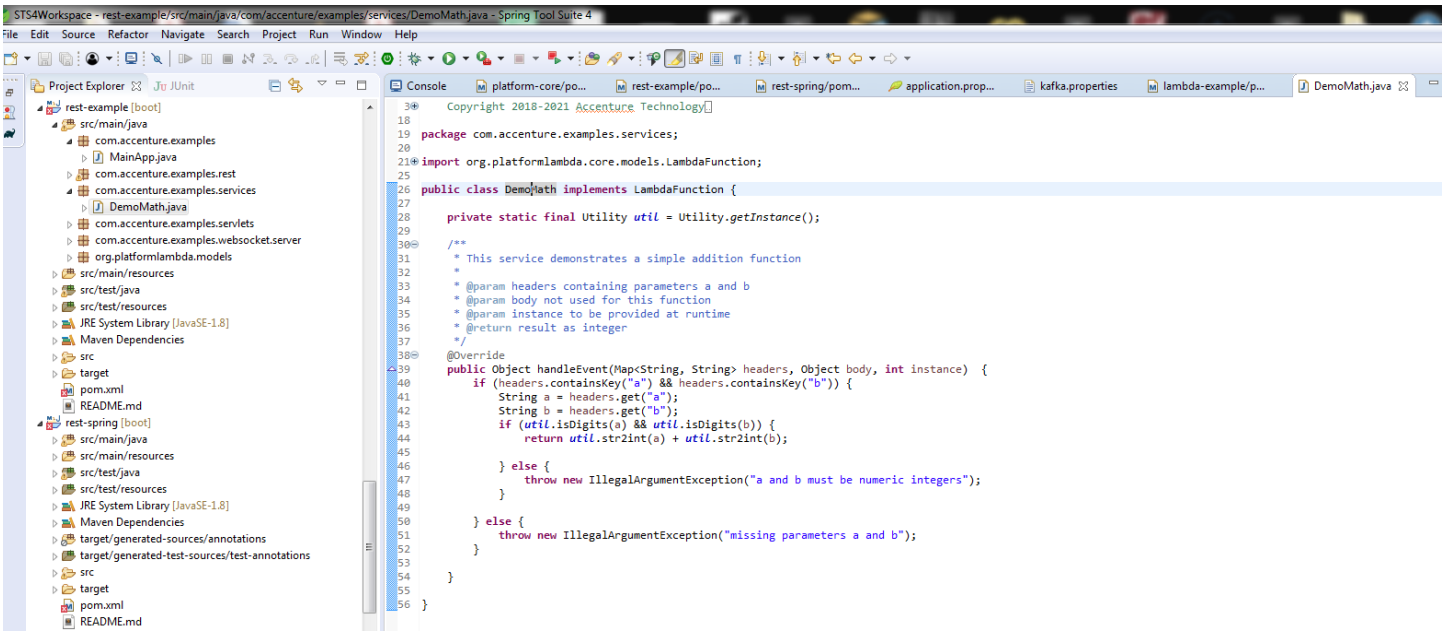




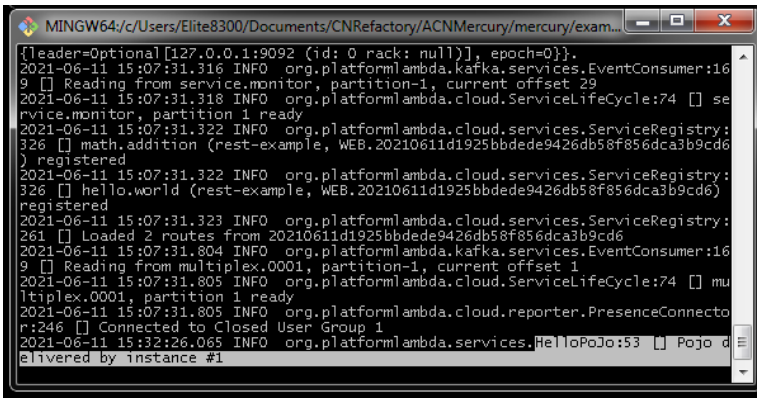
Access the rest-example landing page at **127.0.0.1:8083/index.html**



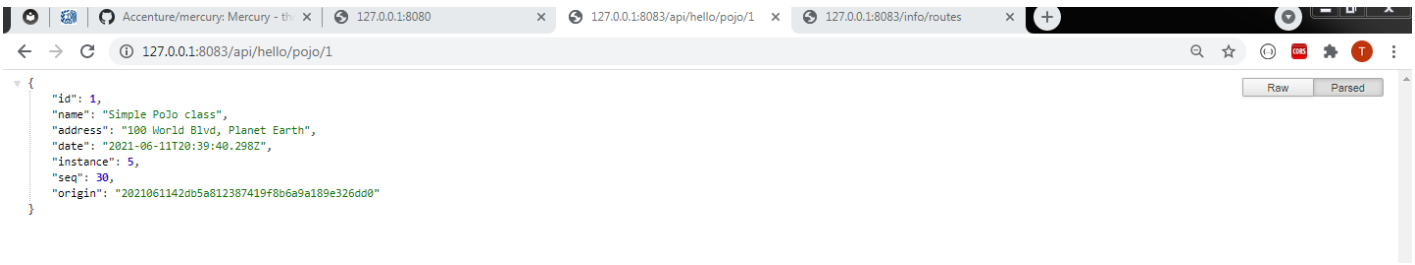
Access the rest-example 2 service routes (math.addition, hello.world) and the lambda-example's 4 (hello.download, hello.world, hello.generic, hello.pojo) **service routes** at **127.0.0.1:8083/info/routes**



Make a hello.pojo request to http://127.0.0.1:8083/api/hello/pojo/1 endpoint

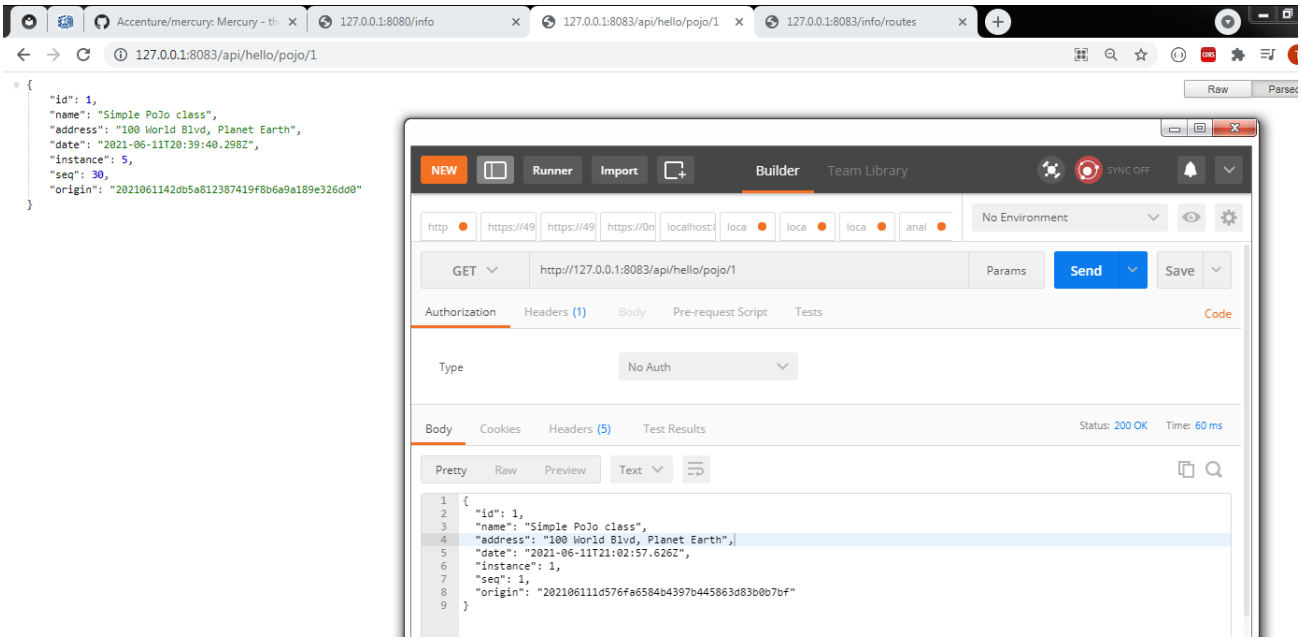


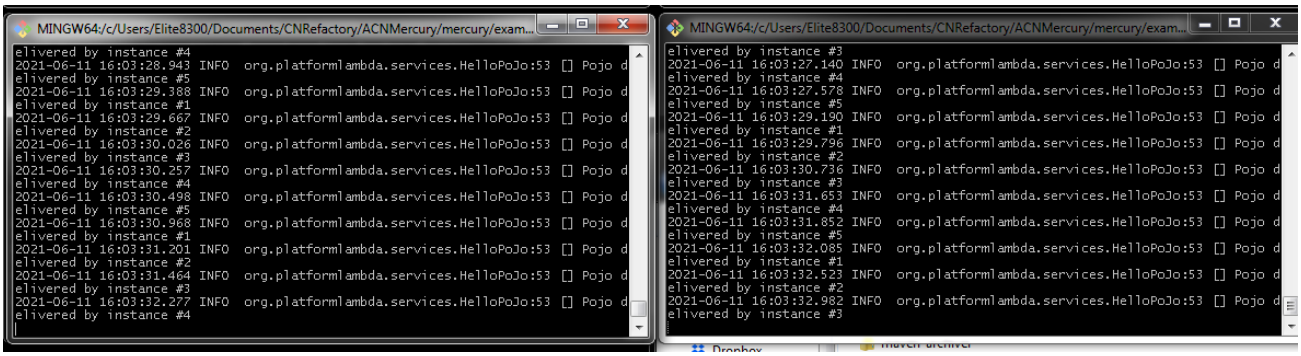
The request is routed from the rest-example web app's /api/hello.pojo controller endpoint to the lambda-example app's hello.pojo function via Kafka.



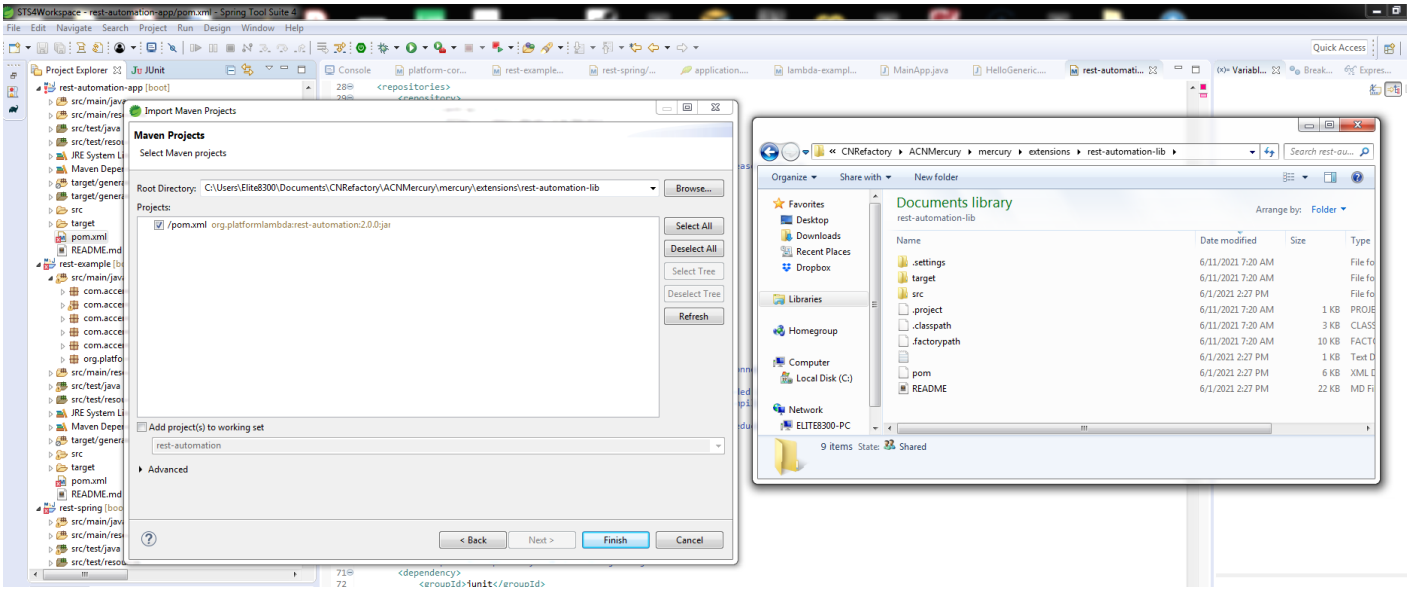
Start a 2nd instance of the lambda-example app, it should connect to the Kafka Presence and start serving requests too

Now we see 2 instances of the lambda-example functions available on topics multiplex.0001-000 and multiplex.0001-002.

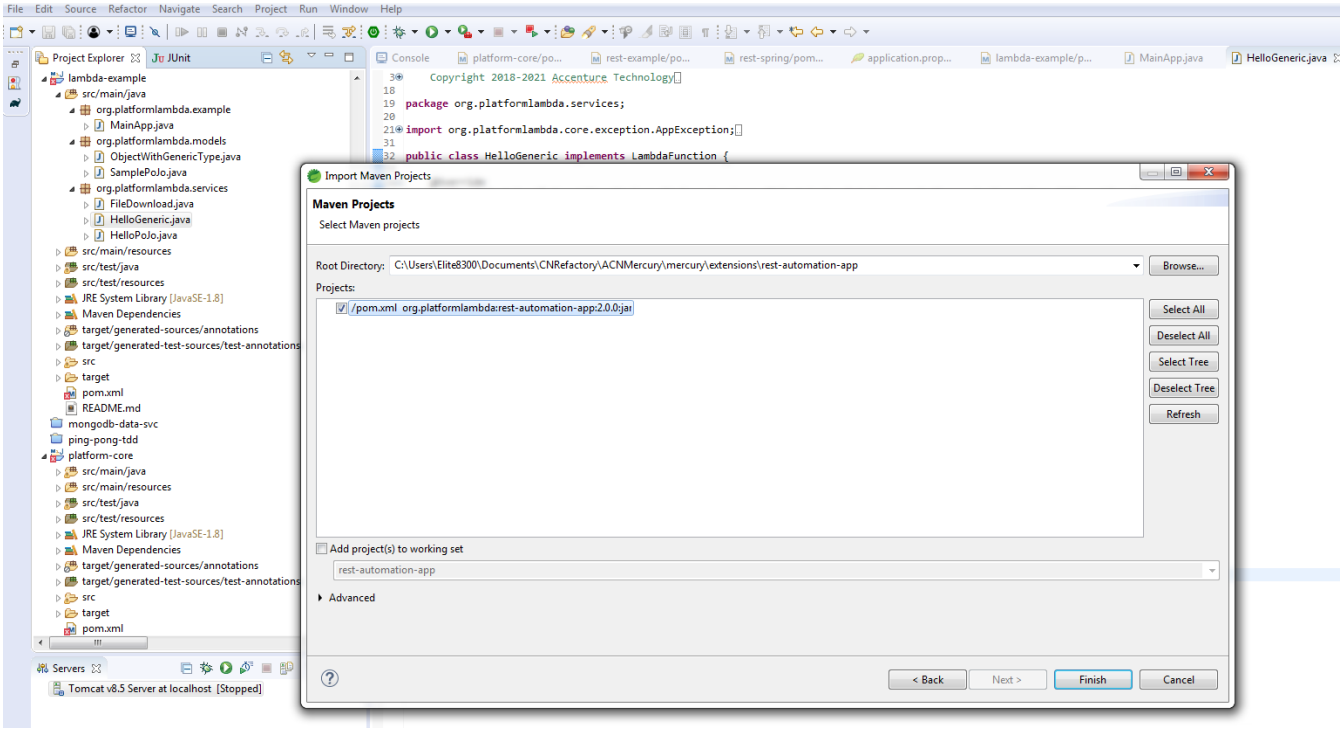




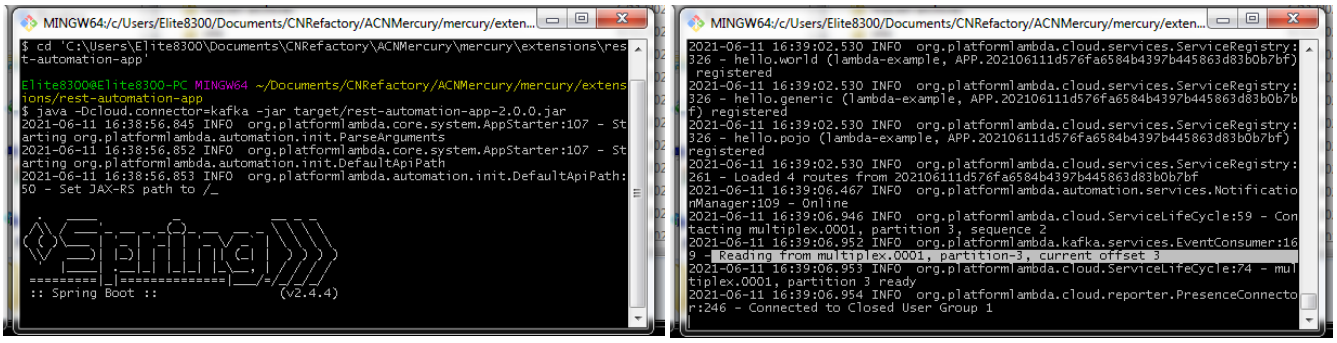
These 2 instances are now both serving the **hello.pojo** requests coming in



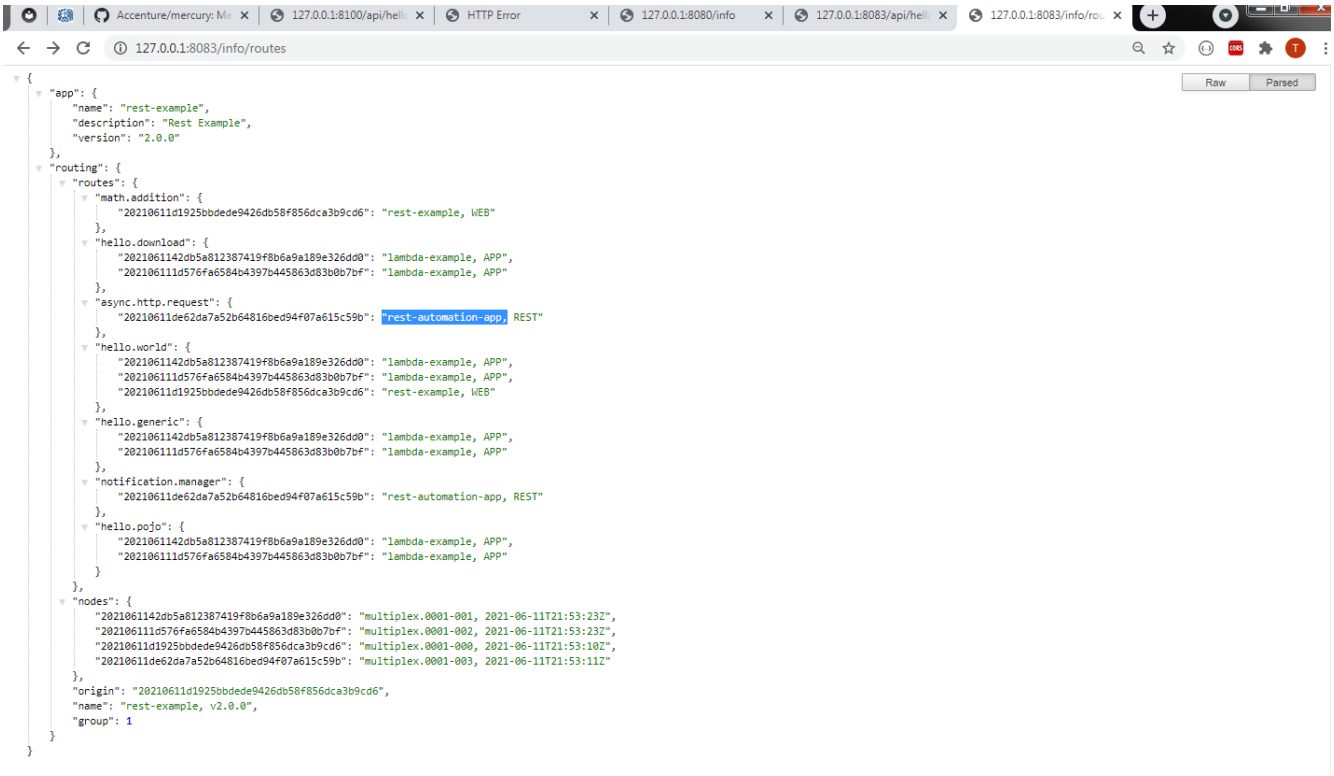
Import clean, build:compile and install the rest-automation-lib project, the JAR is needed for the rest-automation-app below



Import, clean, build:compile and install the rest-automation app in mercury/extensions/rest-automation-app



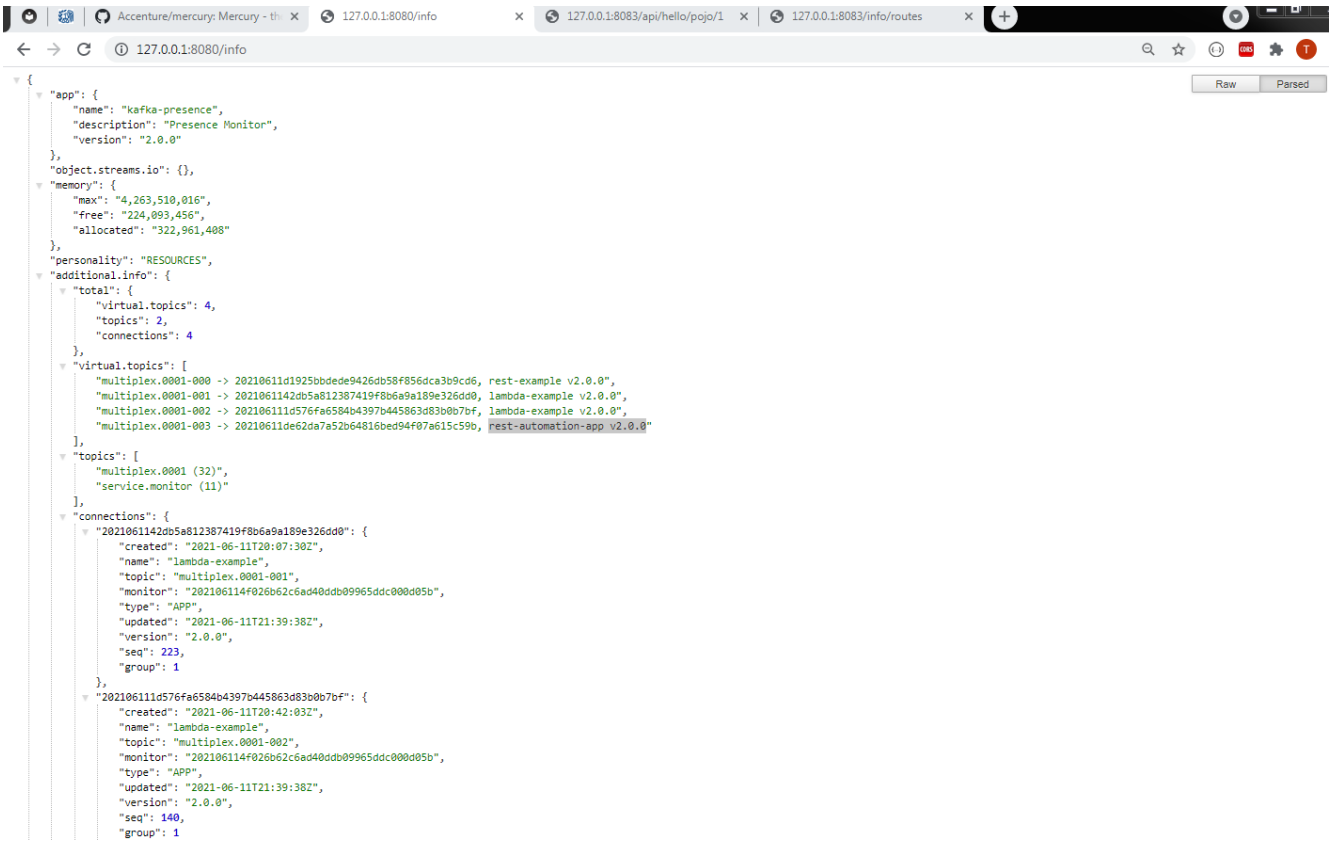
Run the rest-automation-app using the command **$ java -Dcloud.connector=kafka -jar target/rest-automation-app-2.0.0.jar**



127.0.0.1:8083/info/routes

Raw Parsed

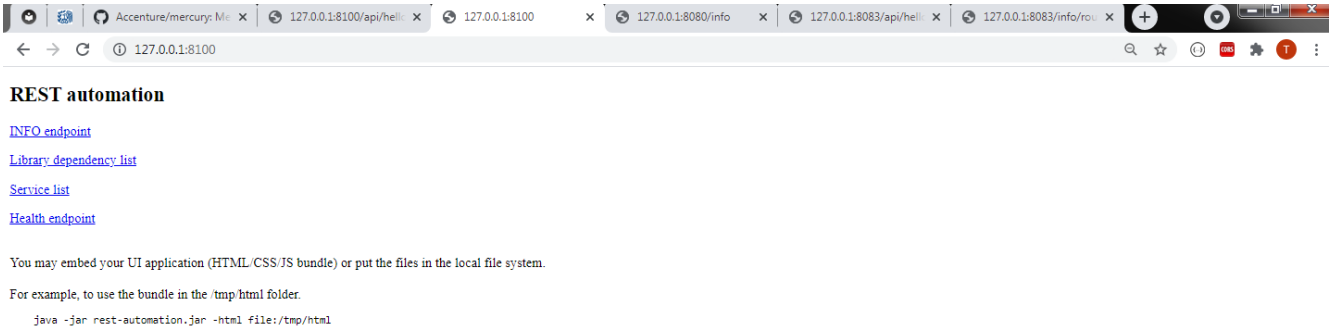

```
{
  "app": {
    "name": "rest-example",
    "description": "Rest Example",
    "version": "2.0.0"
  },
  "routing": {
    "routes": {
      "math.addition": {
        "20210611d1925bbdede9426db58f856dca3b9cd6": "rest-example, WEB"
      },
      "hello.download": {
        "2021061142db5a812387419f8b6a9a189e326dd0": "lambda-example, APP",
        "202106111d576fa6584b4397b445863d83b0b7bf": "lambda-example, APP"
      },
      "async.http.request": {
        "20210611de62da7a52b64816bed94f07a615c59b": "rest-automation-app, REST"
      },
      "hello.world": {
        "2021061142db5a812387419f8b6a9a189e326dd0": "lambda-example, APP",
        "202106111d576fa6584b4397b445863d83b0b7bf": "lambda-example, APP",
        "20210611d1925bbdede9426db58f856dca3b9cd6": "rest-example, WEB"
      },
      "hello.generic": {
        "2021061142db5a812387419f8b6a9a189e326dd0": "lambda-example, APP",
        "202106111d576fa6584b4397b445863d83b0b7bf": "lambda-example, APP"
      },
      "notification.manager": {
        "20210611de62da7a52b64816bed94f07a615c59b": "rest-automation-app, REST"
      },
      "hello.pojo": {
        "2021061142db5a812387419f8b6a9a189e326dd0": "lambda-example, APP",
        "202106111d576fa6584b4397b445863d83b0b7bf": "lambda-example, APP"
      }
    },
    "nodes": {
      "2021061142db5a812387419f8b6a9a189e326dd0": "multiplex.0001-001, 2021-06-11T21:53:23Z",
      "202106111d576fa6584b4397b445863d83b0b7bf": "multiplex.0001-002, 2021-06-11T21:53:23Z",
      "20210611d1925bbdede9426db58f856dca3b9cd6": "multiplex.0001-000, 2021-06-11T21:53:10Z",
      "20210611de62da7a52b64816bed94f07a615c59b": "multiplex.0001-003, 2021-06-11T21:53:11Z"
    },
    "origin": "20210611d1925bbdede9426db58f856dca3b9cd6",
    "name": "rest-example, v2.0.0",
    "group": 1
  }
}
```

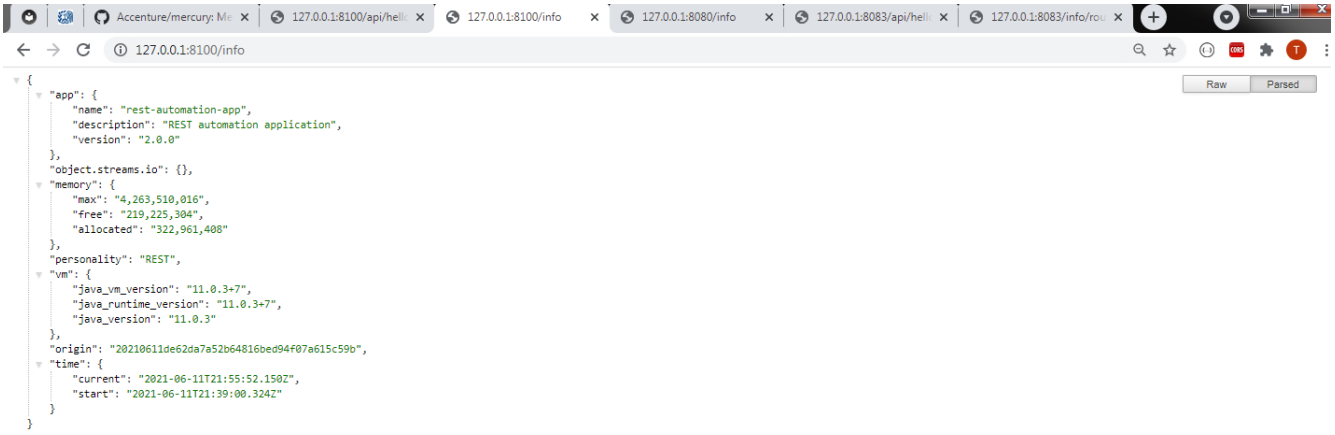
127.0.0.1:8080/info

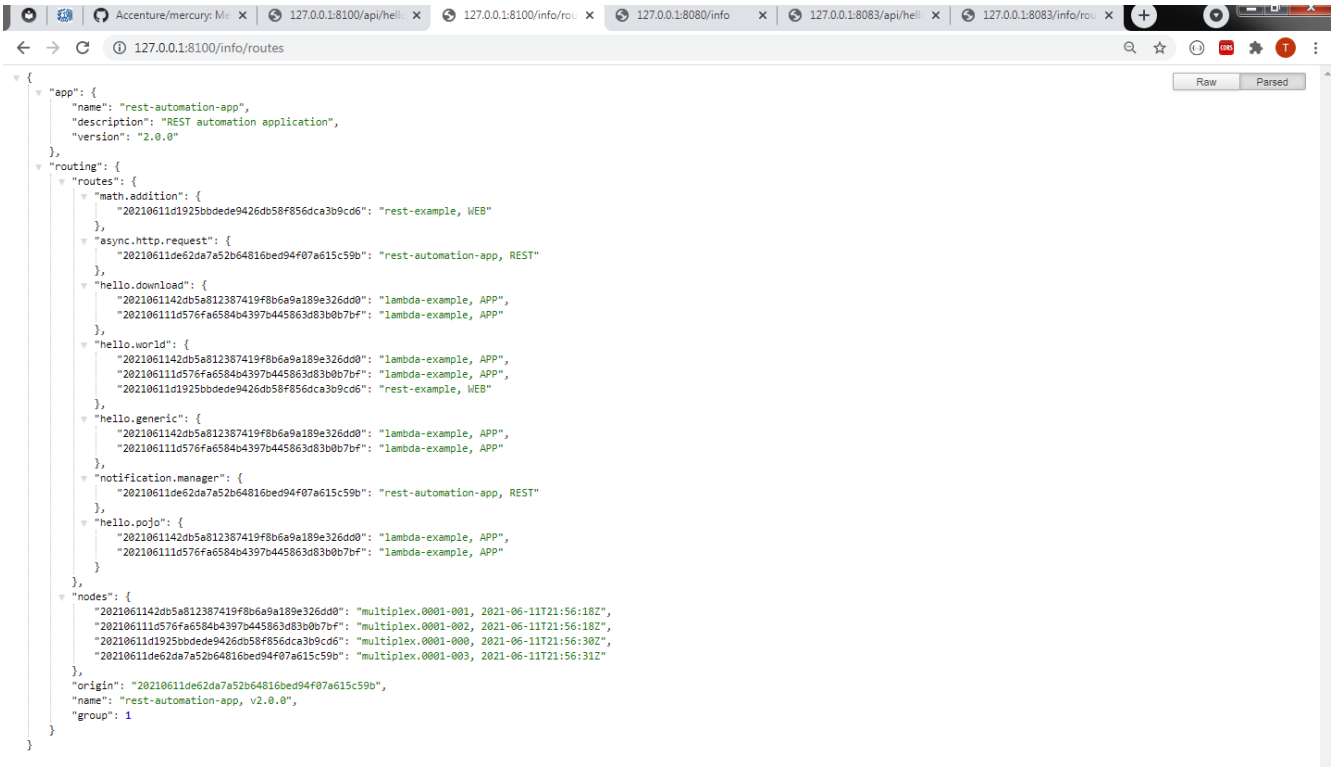
Raw Parsed


```
{
  "app": {
    "name": "kafka-presence",
    "description": "Presence Monitor",
    "version": "2.0.0"
  },
  "object.streams.io": {},
  "memory": {
    "max": "4,263,510,016",
    "free": "224,093,456",
    "allocated": "322,961,408"
  },
  "personality": "RESOURCES",
  "additional.info": {
    "total": {
      "virtual.topics": 4,
      "topics": 2,
      "connections": 4
    },
    "virtual.topics": [
      "multiplex.0001-000 -> 20210611d1925bbdede9426db58f856dca3b9cd6, rest-example v2.0.0",
      "multiplex.0001-001 -> 2021061142db5a812387419f8b6a9a189e326dd0, lambda-example v2.0.0",
      "multiplex.0001-002 -> 202106111d576fa6584b4397b445863d83b0b7bf, lambda-example v2.0.0",
      "multiplex.0001-003 -> 20210611de62da7a52b64816bed94f07a615c59b, rest-automation-app v2.0.0"
    ],
    "topics": [
      "multiplex.0001 (32)",
      "service.monitor (11)"
    ],
    "connections": {
      "2021061142db5a812387419f8b6a9a189e326dd0": {
        "created": "2021-06-11T20:07:30Z",
        "name": "lambda-example",
        "topic": "multiplex.0001-001",
        "monitor": "202106114f026b62c6ad40ddb09965ddc000d05b",
        "type": "APP",
        "updated": "2021-06-11T21:39:38Z",
        "version": "2.0.0",
        "seq": 223,
        "group": 1
      },
      "202106111d576fa6584b4397b445863d83b0b7bf": {
        "created": "2021-06-11T20:42:03Z",
        "name": "lambda-example",
        "topic": "multiplex.0001-002",
        "monitor": "202106114f026b62c6ad40ddb09965ddc000d05b",
        "type": "APP",
        "updated": "2021-06-11T21:39:38Z",
        "version": "2.0.0",
        "seq": 140,
        "group": 1
```

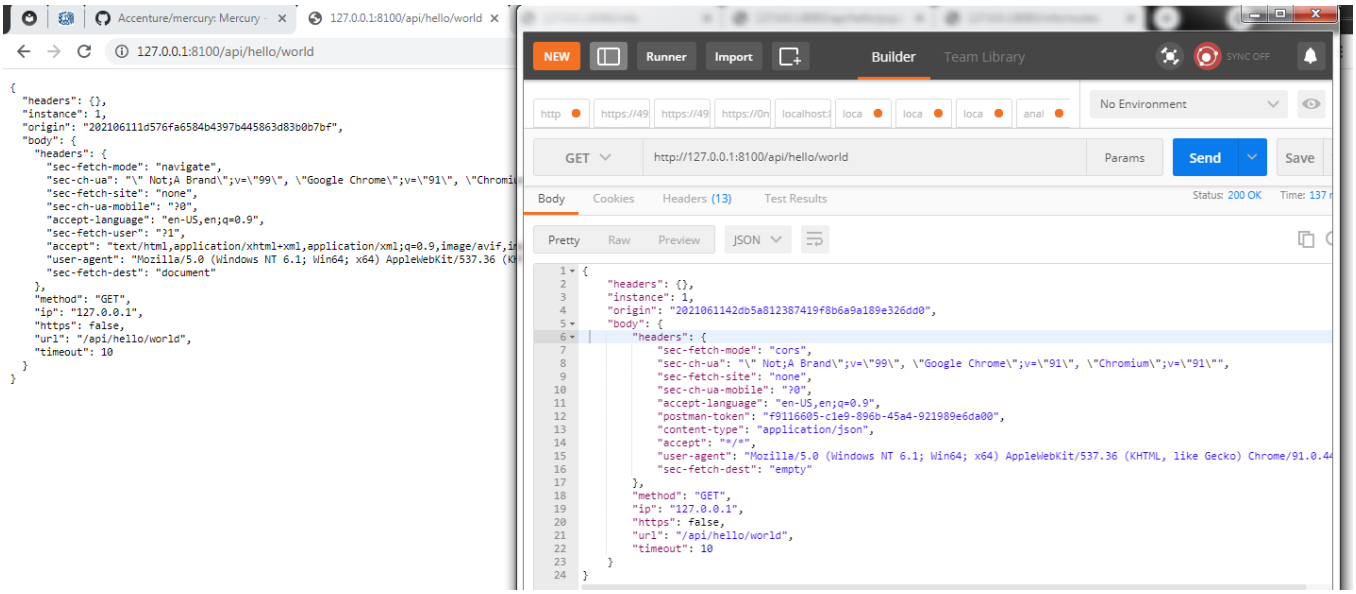


The rest-automation-app is now running on **port 8100** and registered with the Presence Monitor, it is using the virtual topic/channel multiplex.0001-003 for communication

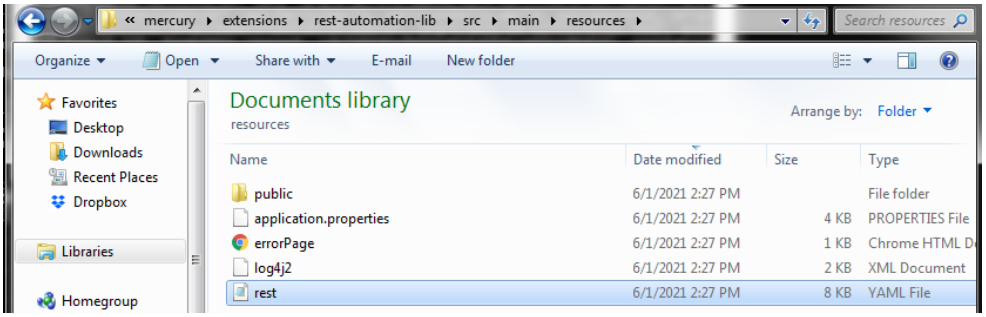




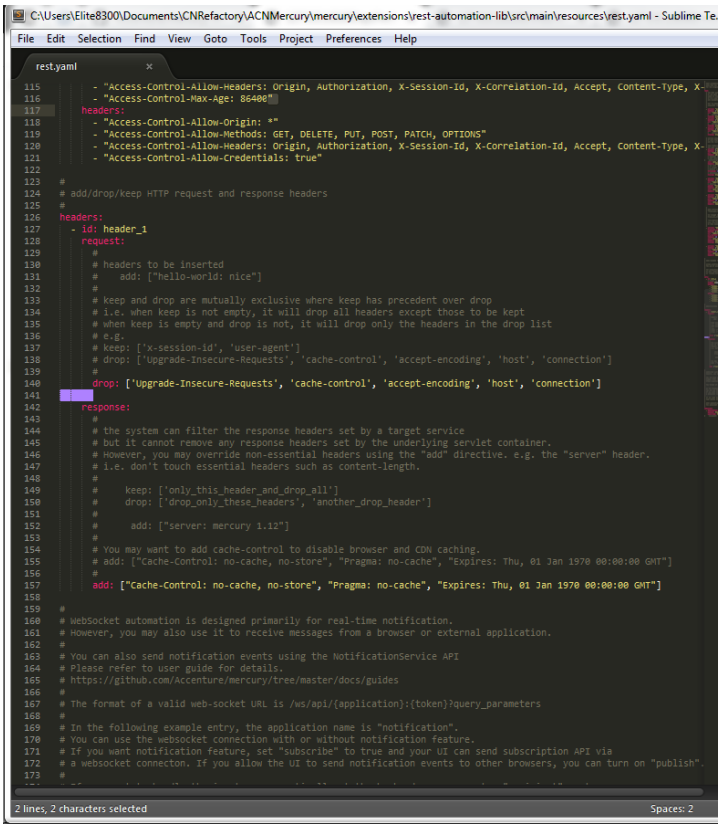
These are the routes available from the rest-automation-app instance on port 8100

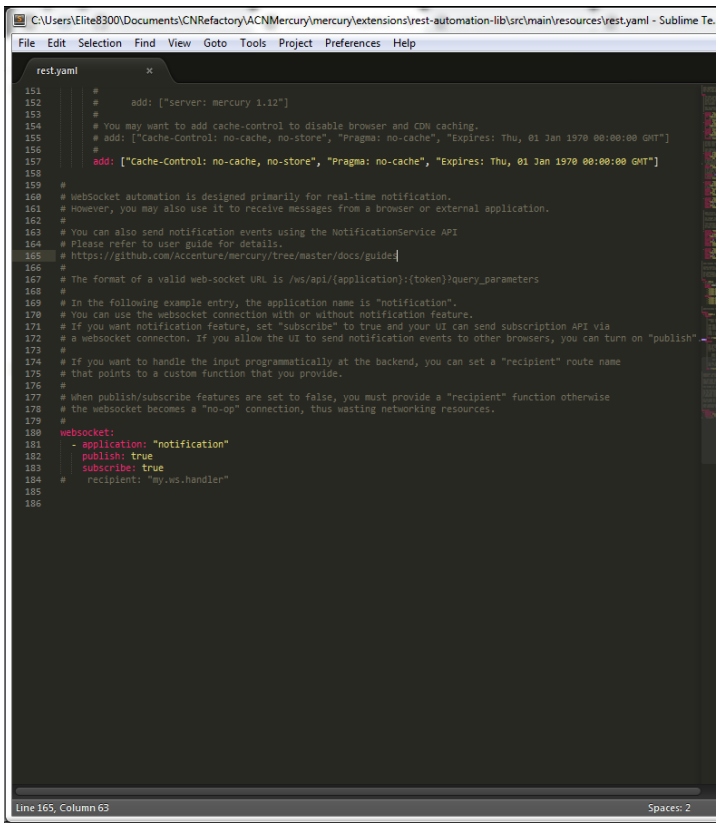


We can now access the hello.world function on the lambda-example app via the rest-automation-app on port 8100



Open the rest.yml file in the rest-automation-lib's resources folder that define the REST Automation application to see the available endpoints for calling the lambda functions available.


C:\Users\Elite8300\Documents\CNRefactory\ACNMercury\mercury\extensions\rest-automation-lib\src\main\resources\rest.yaml - Sublime Te
File Edit Selection Find View Goto Tools Project Preferences Help
rest.yaml
- "Access-Control-Allow-Headers: Origin, Authorization, X-Session-Id, X-Correlation-Id, Accept, Content-Type, X-
- "Access-Control-Max-Age: 86400"
headers:
- "Access-Control-Allow-Origin: *"
- "Access-Control-Allow-Methods: GET, DELETE, PUT, POST, PATCH, OPTIONS"
- "Access-Control-Allow-Headers: Origin, Authorization, X-Session-Id, X-Correlation-Id, Accept, Content-Type, X-
- "Access-Control-Allow-Credentials: true"
#
add/drop/keep HTTP request and response headers
#
headers:
- id: header_1
request:
#
headers to be inserted
add: ["hello-world: nice"]
#
keep and drop are mutually exclusive where keep has precedent over drop
i.e. when keep is not empty, it will drop all headers except those to be kept
when keep is empty and drop is not, it will drop only the headers in the drop list
e.g.
keep: ['x-session-id', 'user-agent']
drop: ['Upgrade-Insecure-Requests', 'cache-control', 'accept-encoding', 'host', 'connection']
#
drop: ['Upgrade-Insecure-Requests', 'cache-control', 'accept-encoding', 'host', 'connection']
response:
#
the system can filter the response headers set by a target service
but it cannot remove any response headers set by the underlying servlet container.
However, you may override non-essential headers using the "add" directive. e.g. the "server" header.
i.e. don't touch essential headers such as content-length.
#
keep: ['only_this_header_and_drop_all']
drop: ['drop_only_these_headers', 'another_drop_header']
#
add: ["server: mercury 1.12"]
#
You may want to add cache-control to disable browser and CDN caching.
add: ["Cache-Control: no-cache, no-store", "Pragma: no-cache", "Expires: Thu, 01 Jan 1970 00:00:00 GMT"]
#
add: ["Cache-Control: no-cache, no-store", "Pragma: no-cache", "Expires: Thu, 01 Jan 1970 00:00:00 GMT"]
#
WebSocket automation is designed primarily for real-time notification.
However, you may also use it to receive messages from a browser or external application.
#
You can also send notification events using the NotificationService API
Please refer to user guide for details.
https://github.com/Accenture/mercury/tree/master/docs/guides
#
The format of a valid web-socket URL is /ws/api/{application}:{token}?query_parameters
#
In the following example entry, the application name is "notification".
You can use the websocket connection with or without notification feature.
If you want notification feature, set "subscribe" to true and your UI can send subscription API via
a websocket connecton. If you allow the UI to send notification events to other browsers, you can turn on "publish".
#
2 lines, 2 characters selected
Spaces: 2


C:\Users\Elite8300\Documents\CNRefactory\ACNMercury\mercury\extensions\rest-automation-lib\src\main\resources\rest.yaml - Sublime Te
File Edit Selection Find View Goto Tools Project Preferences Help
rest.yaml
#
add: ["server: mercury 1.12"]
#
You may want to add cache-control to disable browser and CDN caching.
add: ["Cache-Control: no-cache, no-store", "Pragma: no-cache", "Expires: Thu, 01 Jan 1970 00:00:00 GMT"]
#
add: ["Cache-Control: no-cache, no-store", "Pragma: no-cache", "Expires: Thu, 01 Jan 1970 00:00:00 GMT"]
#
WebSocket automation is designed primarily for real-time notification.
However, you may also use it to receive messages from a browser or external application.
#
You can also send notification events using the NotificationService API
Please refer to user guide for details.
https://github.com/Accenture/mercury/tree/master/docs/guides
#
The format of a valid web-socket URL is /ws/api/{application}:{token}?query_parameters
#
In the following example entry, the application name is "notification".
You can use the websocket connection with or without notification feature.
If you want notification feature, set "subscribe" to true and your UI can send subscription API via
a websocket connecton. If you allow the UI to send notification events to other browsers, you can turn on "publish"
#
If you want to handle the input programmatically at the backend, you can set a "recipient" route name
that points to a custom function that you provide.
#
When publish/subscribe features are set to false, you must provide a "recipient" function otherwise
the websocket becomes a "no-op" connection, thus wasting networking resources.
#
websocket:
- application: "notification"
publish: true
subscribe: true
recipient: "my.ws.handler"
Line 165, Column 63
Spaces: 2