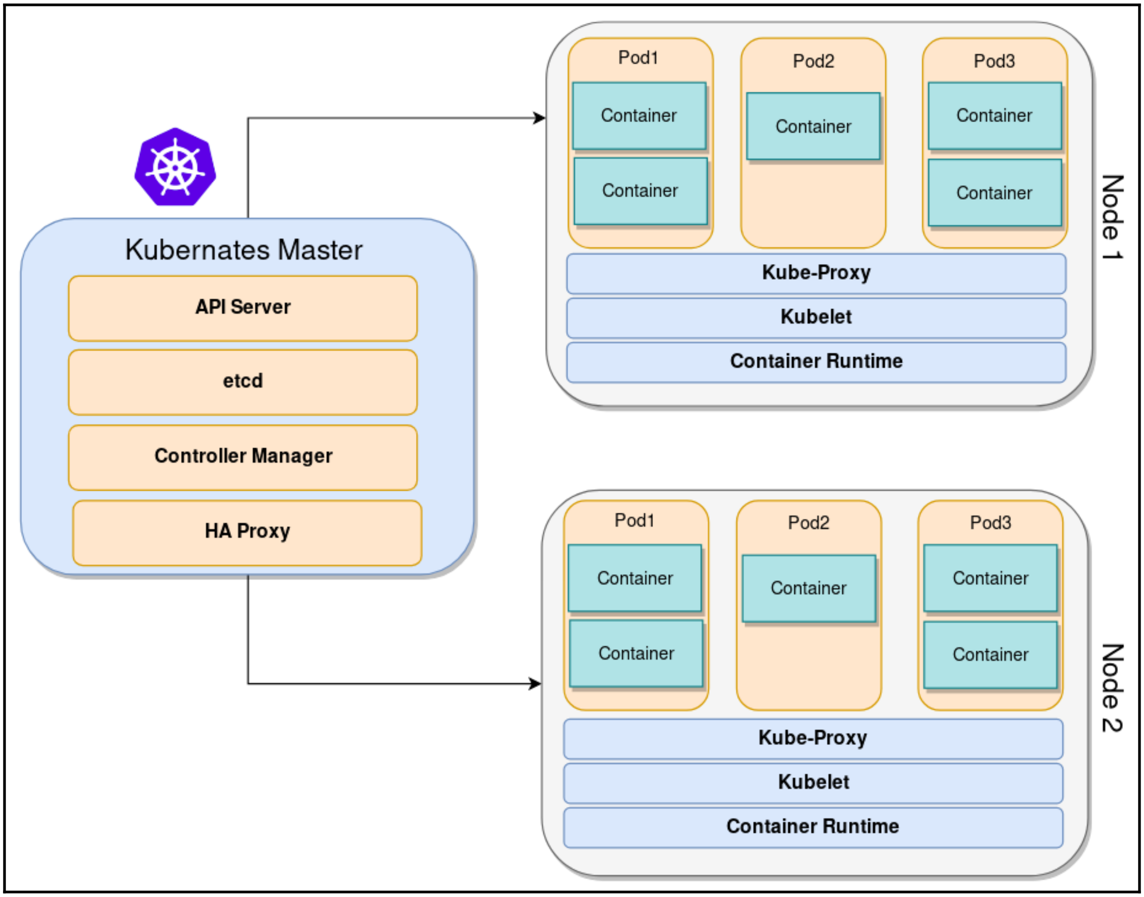
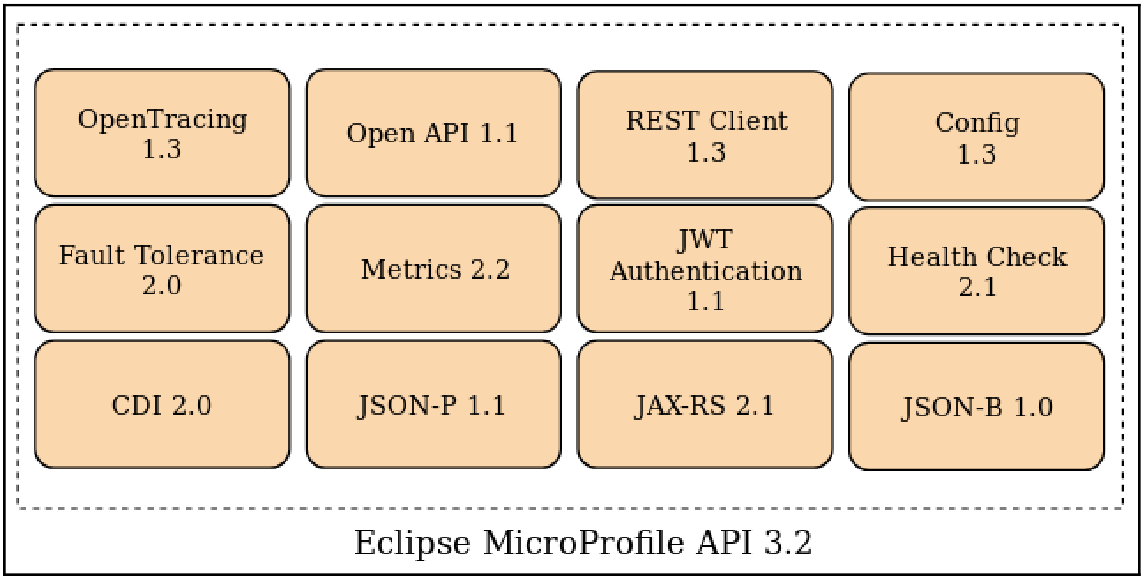
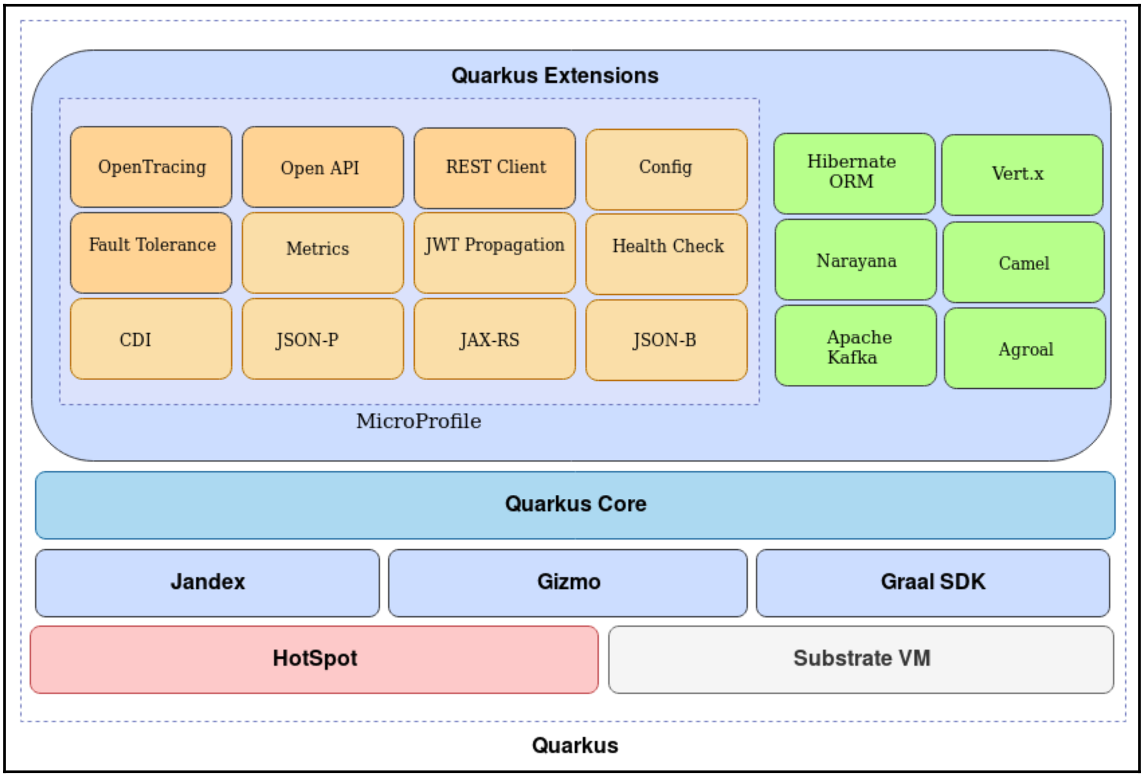
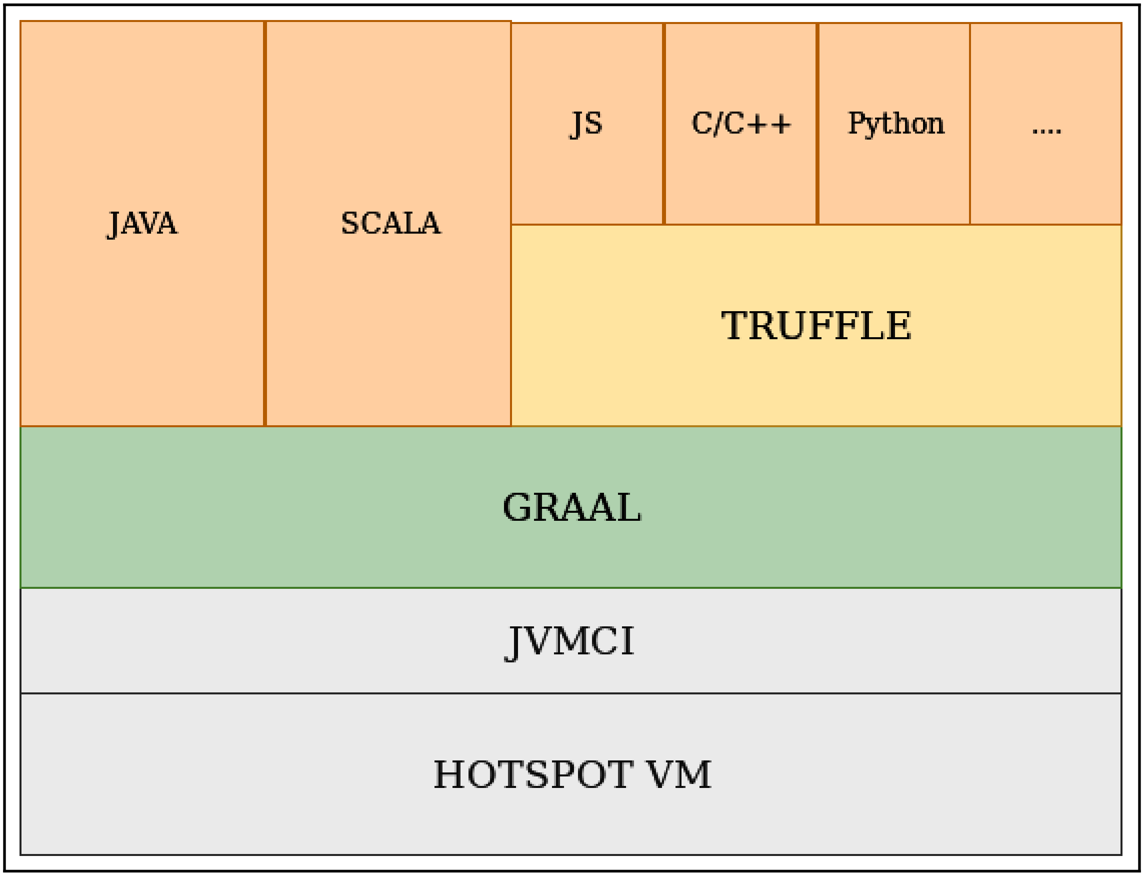
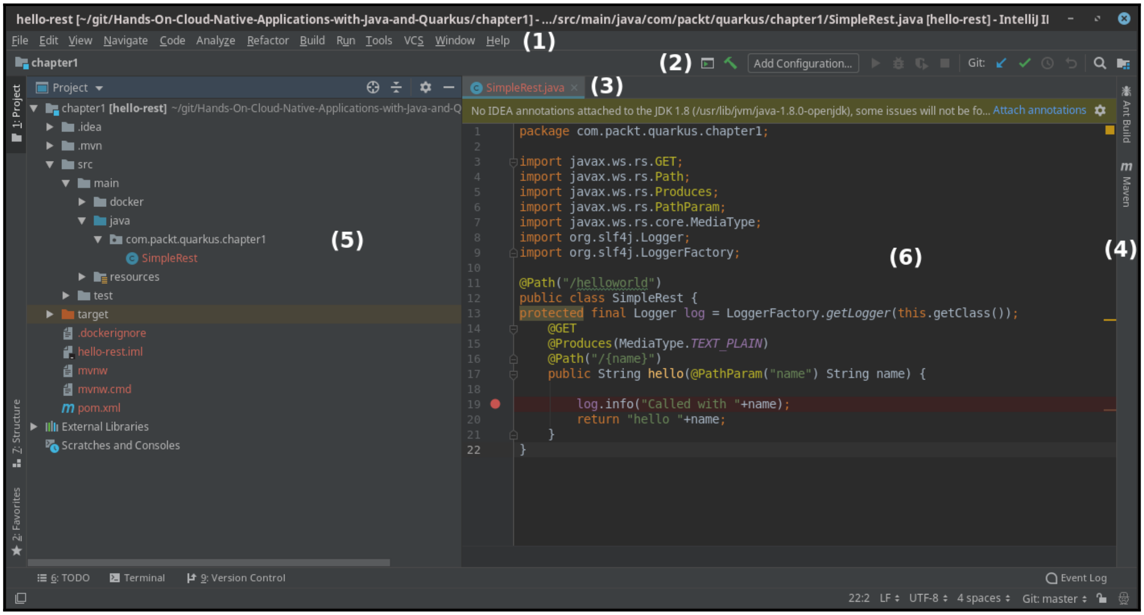
Chapter 1: Introduction to Quarkus Core Concepts




OpenTracing 1.3
Open API 1.1
REST Client 1.3
Config 1.3
Fault Tolerance 2.0
Metrics 2.2
JWT Authentication 1.1
Health Check 2.1
CDI 2.0
JSON-P 1.1
JAX-RS 2.1
JSON-B 1.0
Eclipse MicroProfile API 3.2


Quarkus Extensions
OpenTracing
Open API
REST Client
Config
Fault Tolerance
Metrics
JWT Propagation
Health Check
CDI
JSON-P
JAX-RS
JSON-B
MicroProfile
Hibernate ORM
Vert.x
Narayana
Camel
Apache Kafka
Agroal
Quarkus Core
Jandex
Gizmo
Graal SDK
HotSpot
Substrate VM
Quarkus


JAVA
SCALA
JS
C/C++
Python
....
TRUFFLE
GRAAL
JVMCI
HOTSPOT VM



hello-rest [~/git/Hands-On-Cloud-Native-Applications-with-Java-and-Quarkus/chapter1] - .../src/main/java/com/packt/quarkus/chapter1/SimpleRest.java [hello-rest] - IntelliJ II

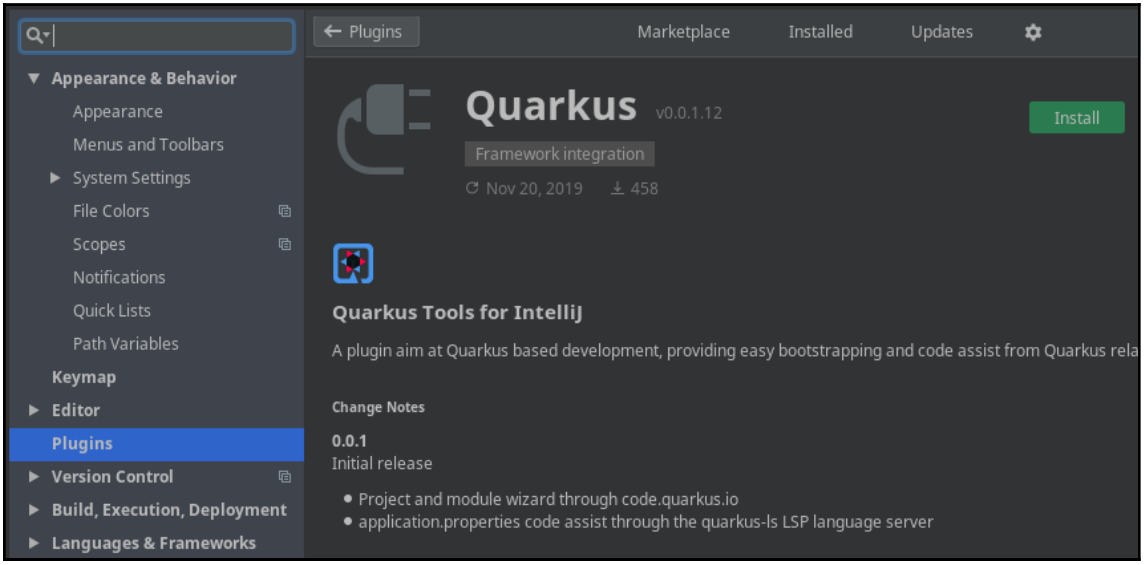
No IDEA annotations attached to the JDK 1.8 (/usr/lib/jvm/java-1.8.0-openjdk), some issues will not be fo... Attach annotations

```java
package com.packt.quarkus.chapter1;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.PathParam;
import javax.ws.rs.core.MediaType;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@Path("/helloworld")
public class SimpleRest {
protected final Logger log = LoggerFactory.getLogger(this.getClass());
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    @Path("/{name}")
    public String hello(@PathParam("name") String name) {

        log.info("Called with "+name);
        return "hello "+name;
    }
}
```



Quarkus v0.0.1.12

Framework integration

Nov 20, 2019 458

Quarkus Tools for IntelliJ

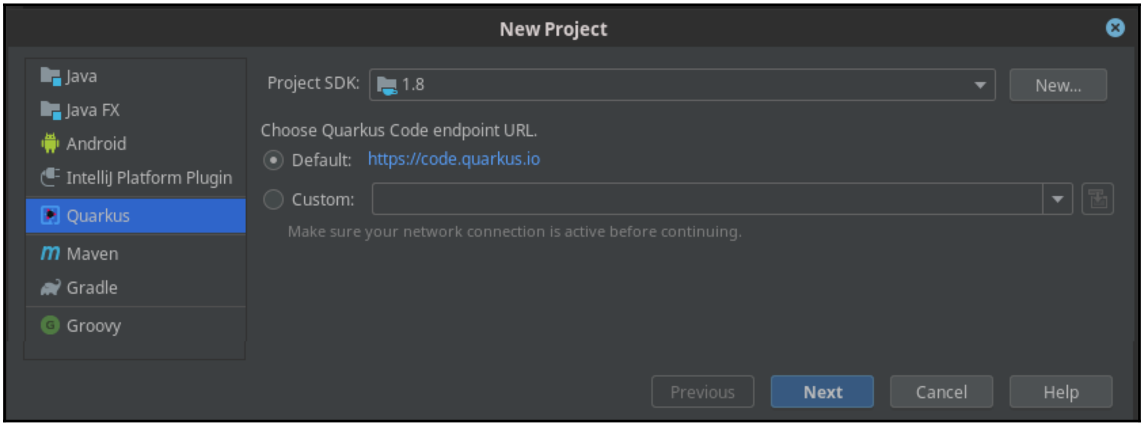
A plugin aim at Quarkus based development, providing easy bootstrapping and code assist from Quarkus rela

Change Notes

0.0.1

Initial release

- Project and module wizard through code.quarkus.io
- application.properties code assist through the quarkus-ls LSP language server



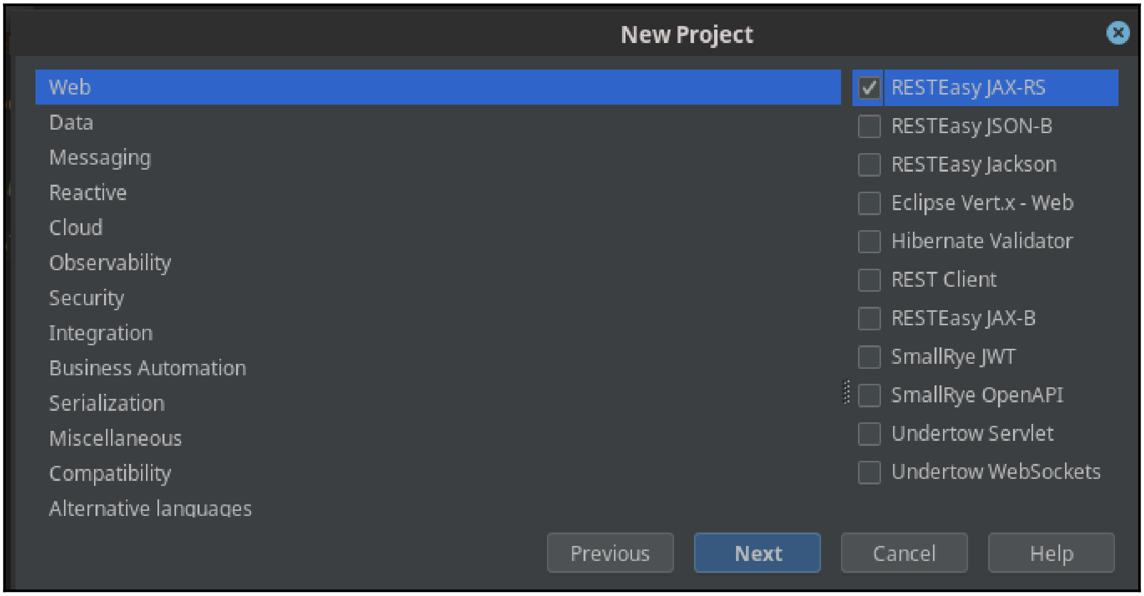
Project SDK: 1.8

Choose Quarkus Code endpoint URL.

● Default: https://code.quarkus.io

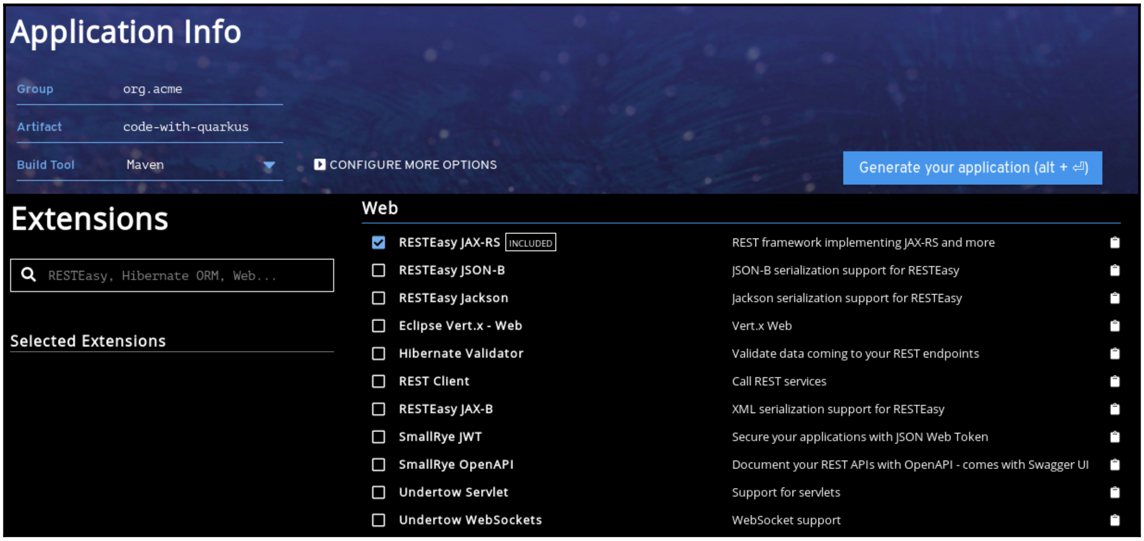
○ Custom:

Make sure your network connection is active before continuing.



☑ RESTEasy JAX-RS
□ RESTEasy JSON-B
□ RESTEasy Jackson
□ Eclipse Vert.x - Web
□ Hibernate Validator
□ REST Client
□ RESTEasy JAX-B
□ SmallRye JWT
□ SmallRye OpenAPI
□ Undertow Servlet
□ Undertow WebSockets

Chapter 2: Developing Your First Application with Quarkus



Application Info

Group: org.acme

Artifact: code-with-quarkus

Build Tool: Maven

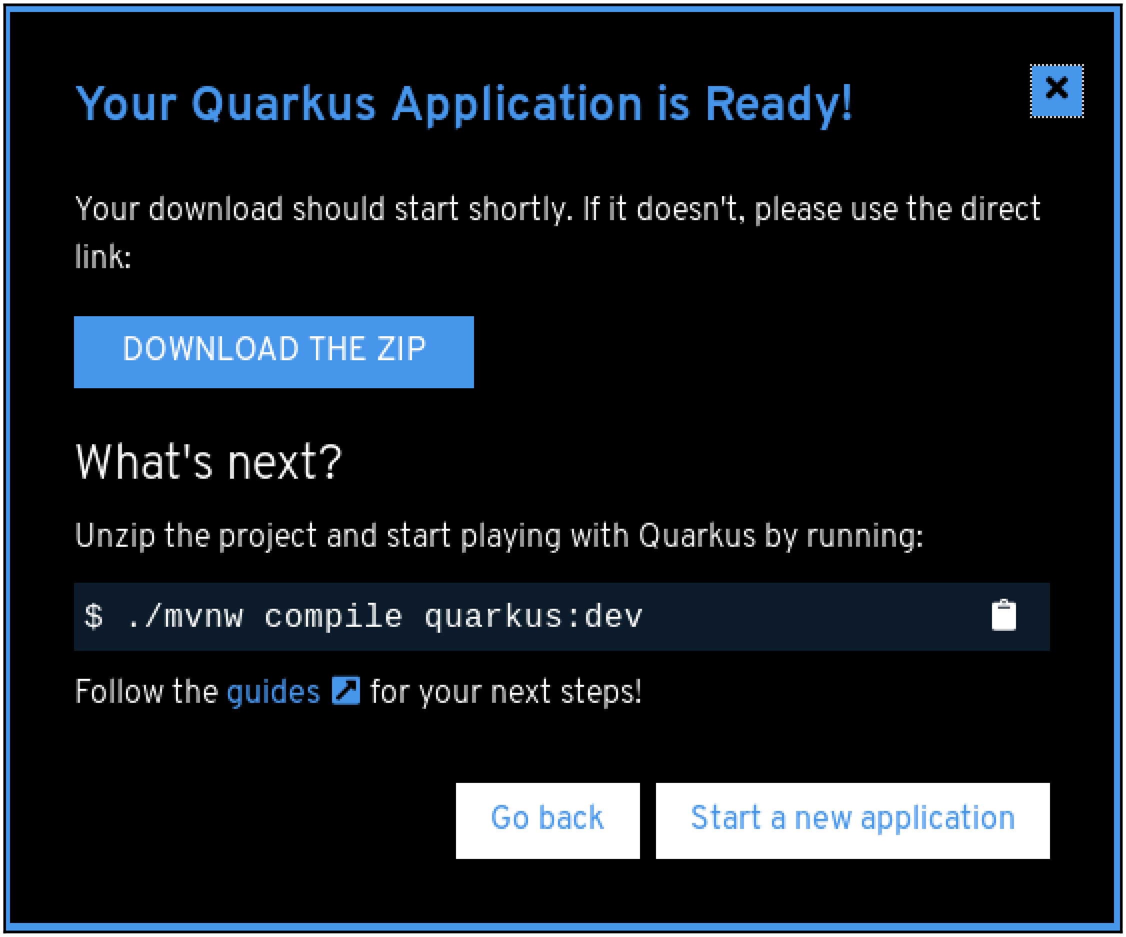
CONFIGURE MORE OPTIONS
Generate your application (alt + ⏎)

Extensions

RESTEasy, Hibernate ORM, Web...
Selected Extensions

Web

☑	RESTEasy JAX-RS INCLUDED	REST framework implementing JAX-RS and more
□	RESTEasy JSON-B	JSON-B serialization support for RESTEasy
□	RESTEasy Jackson	Jackson serialization support for RESTEasy
□	Eclipse Vert.x - Web	Vert.x Web
□	Hibernate Validator	Validate data coming to your REST endpoints
□	REST Client	Call REST services
□	RESTEasy JAX-B	XML serialization support for RESTEasy
□	SmallRye JWT	Secure your applications with JSON Web Token
□	SmallRye OpenAPI	Document your REST APIs with OpenAPI - comes with Swagger UI
□	Undertow Servlet	Support for servlets
□	Undertow WebSockets	WebSocket support



Your Quarkus Application is Ready!

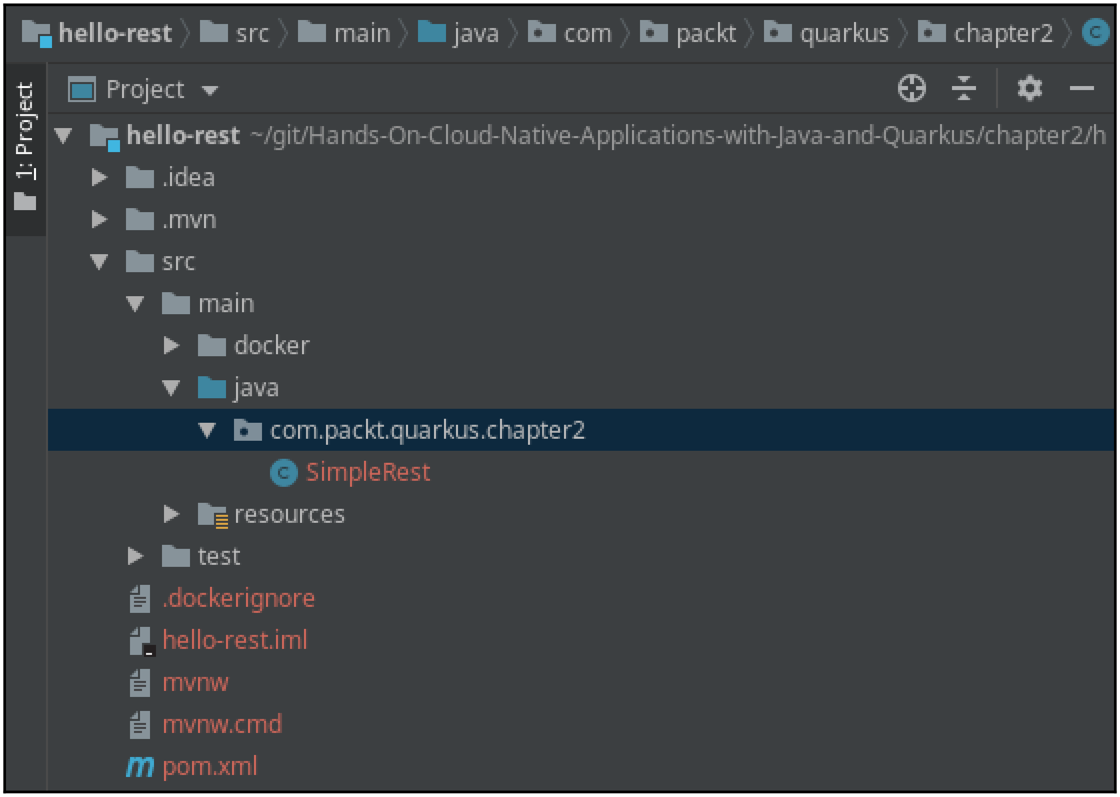
Your download should start shortly. If it doesn't, please use the direct link:

What's next?

Unzip the project and start playing with Quarkus by running:

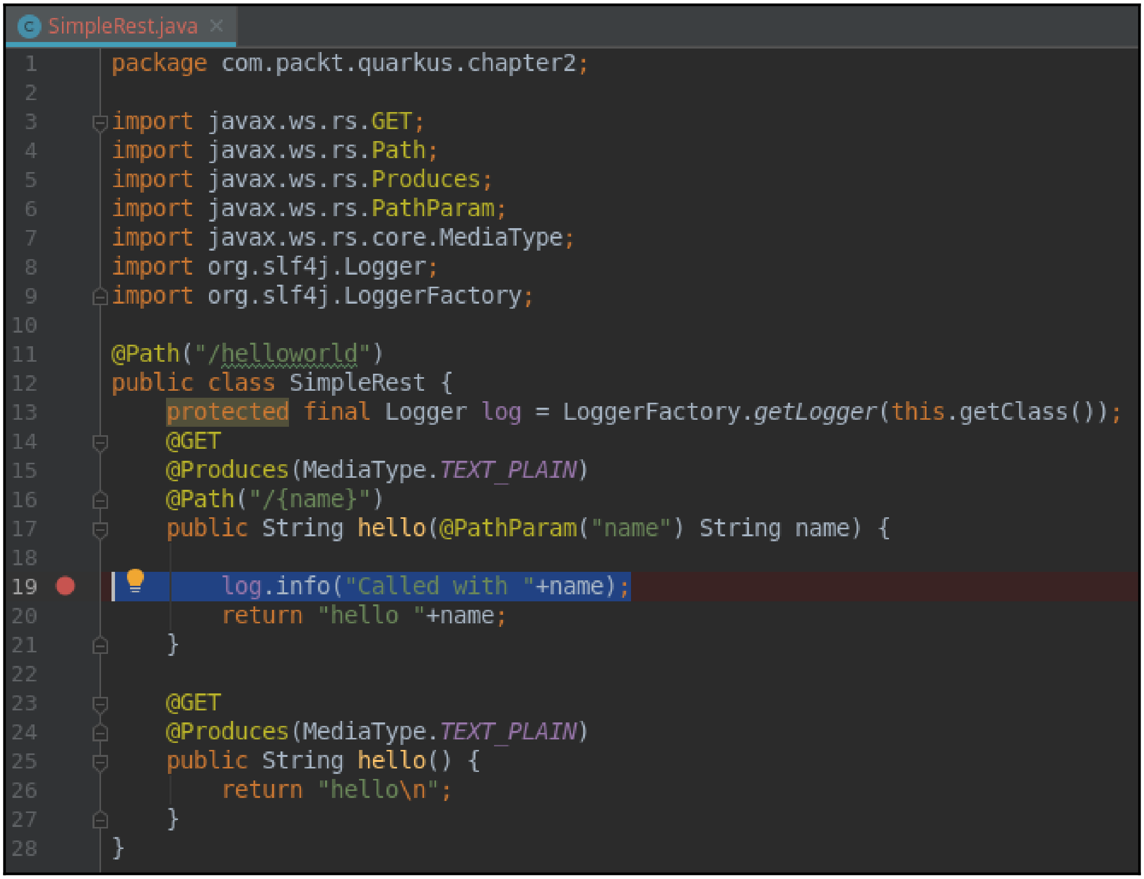
```
$ ./mvnw compile quarkus:dev
```

Follow the guides for your next steps!



hello-rest 〉 src 〉 main 〉 java 〉 com 〉 packt 〉 quarkus 〉 chapter2 〉
1: Project
Project

```
▼ hello-rest ~/git/Hands-On-Cloud-Native-Applications-with-Java-and-Quarkus/chapter2/h
    ▶ .idea
    ▶ .mvn
    ▼ src
        ▼ main
            ▶ docker
            ▼ java
                ▼ com.packt.quarkus.chapter2
                        SimpleRest
            ▶ resources
        ▶ test
      .dockerignore
      hello-rest.iml
      mvnw
      mvnw.cmd
      pom.xml
```



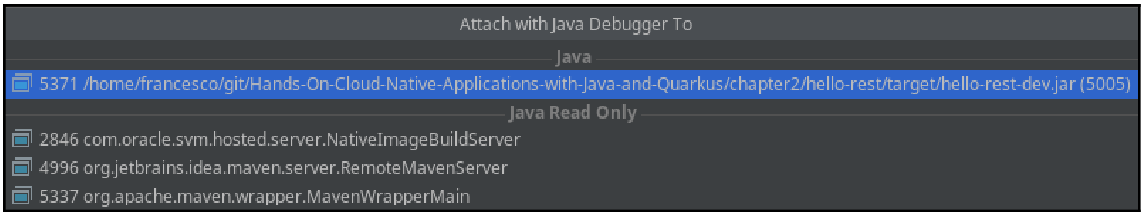
```java
package com.packt.quarkus.chapter2;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.PathParam;
import javax.ws.rs.core.MediaType;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@Path("/helloworld")
public class SimpleRest {
    protected final Logger log = LoggerFactory.getLogger(this.getClass());
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    @Path("/{name}")
    public String hello(@PathParam("name") String name) {

        log.info("Called with "+name);
        return "hello "+name;
    }

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String hello() {
        return "hello\n";
    }
}
```



Attach with Java Debugger To

Java

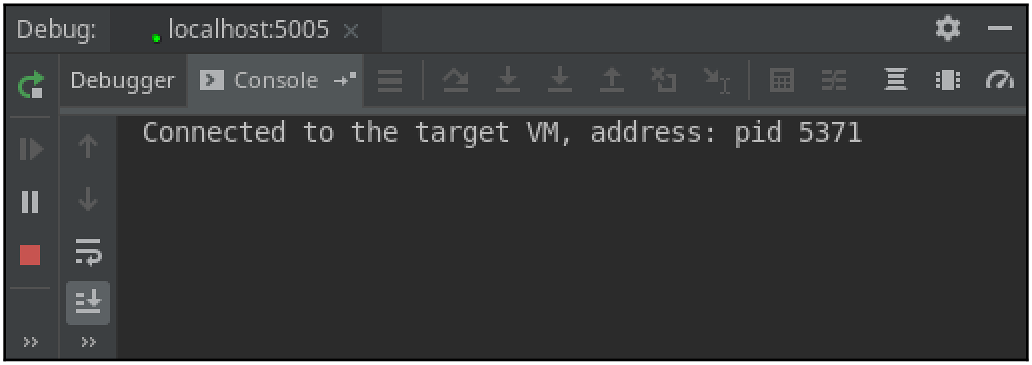
5371 /home/francesco/git/Hands-On-Cloud-Native-Applications-with-Java-and-Quarkus/chapter2/hello-rest/target/hello-rest-dev.jar (5005)

Java Read Only

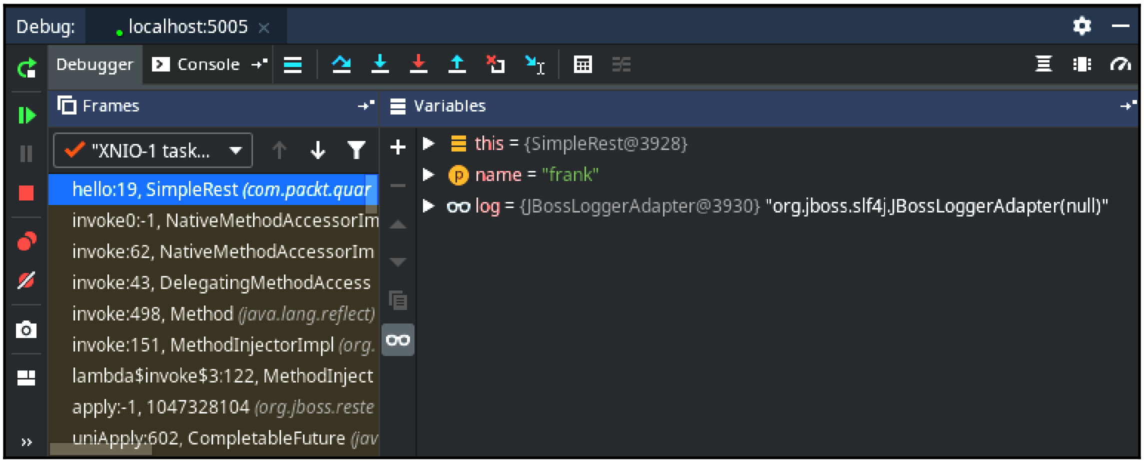
2846 com.oracle.svm.hosted.server.NativeImageBuildServer

4996 org.jetbrains.idea.maven.server.RemoteMavenServer

5337 org.apache.maven.wrapper.MavenWrapperMain



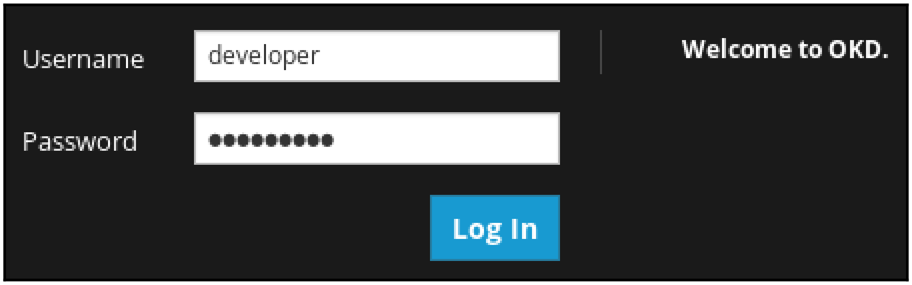
```
Connected to the target VM, address: pid 5371
```



```
hello:19, SimpleRest (com.packt.quar
invoke0:-1, NativeMethodAccessorIm
invoke:62, NativeMethodAccessorIm
invoke:43, DelegatingMethodAccess
invoke:498, Method (java.lang.reflect)
invoke:151, MethodInjectorImpl (org.
lambda$invoke$3:122, MethodInject
apply:-1, 1047328104 (org.jboss.reste
uniApply:602, CompletableFuture (jav
```

```
▶ this = {SimpleRest@3928}
▶ name = "frank"
▶ log = {JBossLoggerAdapter@3930} "org.jboss.slf4j.JBossLoggerAdapter(null)"
```

Chapter 3: Creating a Container Image of Your Application

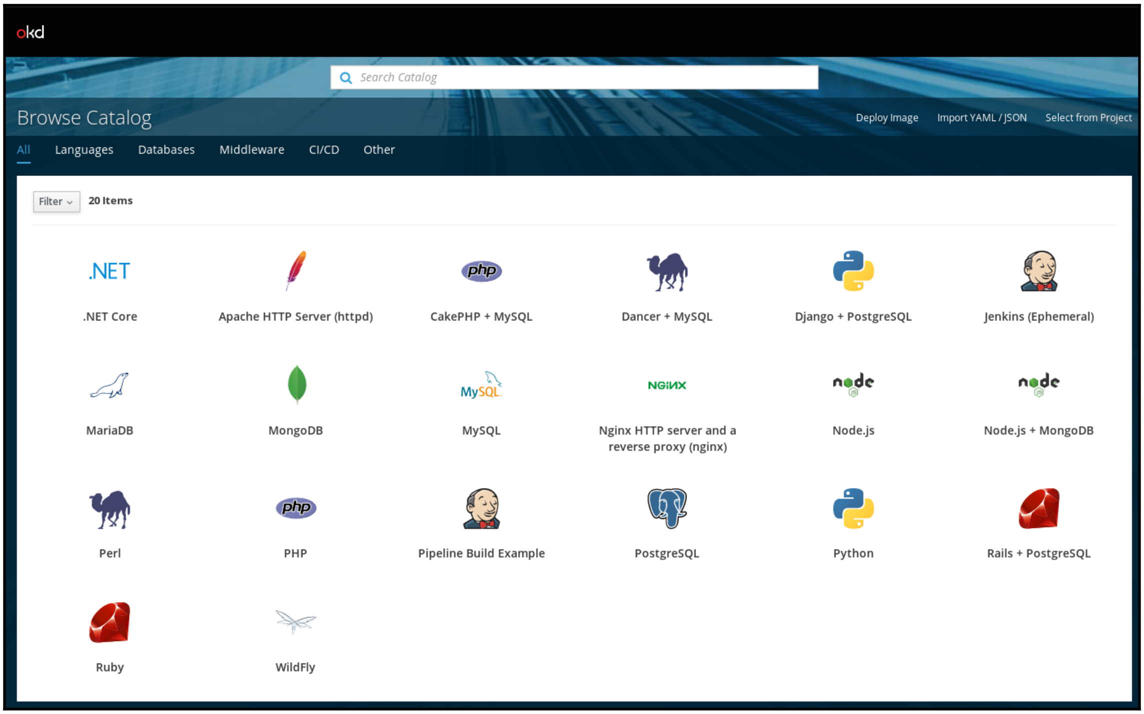


Username developer

Password ●●●●●●●●●●

Log In

Welcome to OKD.



okd

Search Catalog

Browse Catalog

Deploy Image | Import YAML / JSON | Select from Project

All | Languages | Databases | Middleware | CI/CD | Other

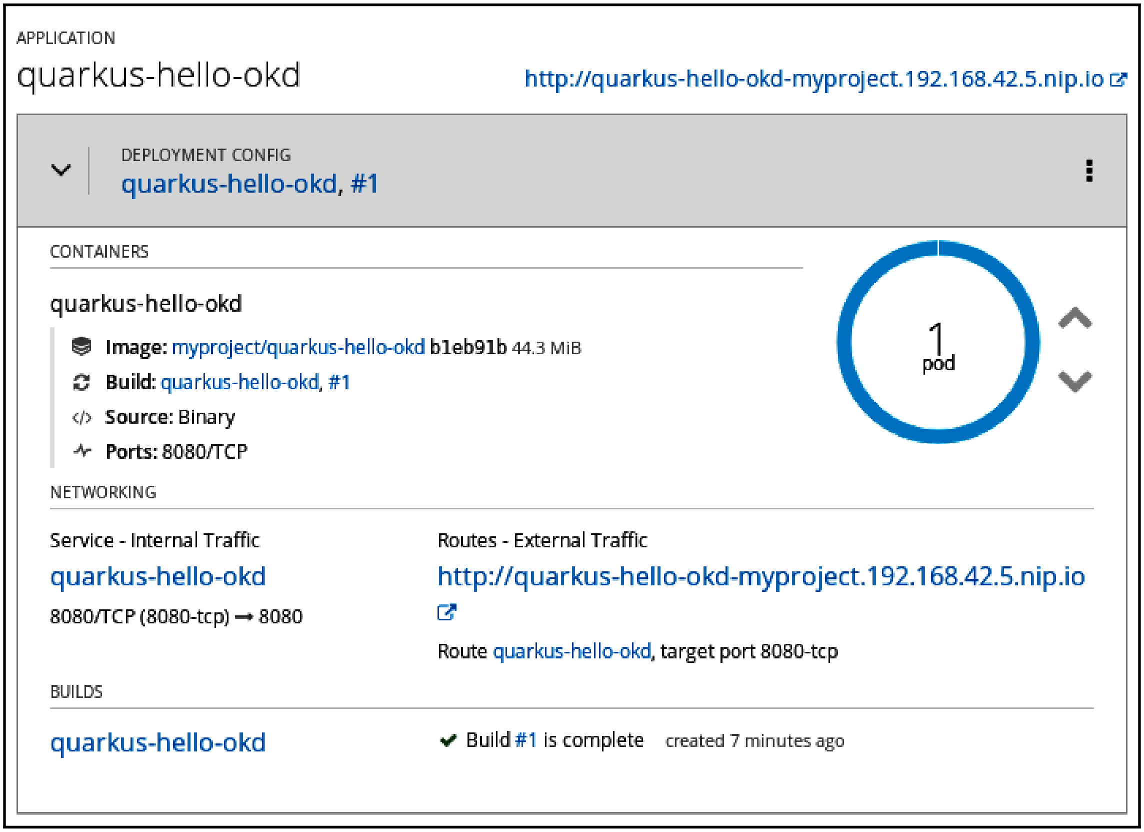
Filter ⌄ 20 Items

.NET Core | Apache HTTP Server (httpd) | CakePHP + MySQL | Dancer + MySQL | Django + PostgreSQL | Jenkins (Ephemeral)

MariaDB | MongoDB | MySQL | Nginx HTTP server and a reverse proxy (nginx) | Node.js | Node.js + MongoDB

Perl | PHP | Pipeline Build Example | PostgreSQL | Python | Rails + PostgreSQL

Ruby | WildFly



APPLICATION

quarkus-hello-okd

http://quarkus-hello-okd-myproject.192.168.42.5.nip.io

DEPLOYMENT CONFIG

quarkus-hello-okd, #1

CONTAINERS

quarkus-hello-okd

Image: myproject/quarkus-hello-okd **b1eb91b** 44.3 MiB

Build: quarkus-hello-okd, #1

Source: Binary

Ports: 8080/TCP

NETWORKING

Service - Internal Traffic

quarkus-hello-okd

8080/TCP (8080-tcp) → 8080

Routes - External Traffic

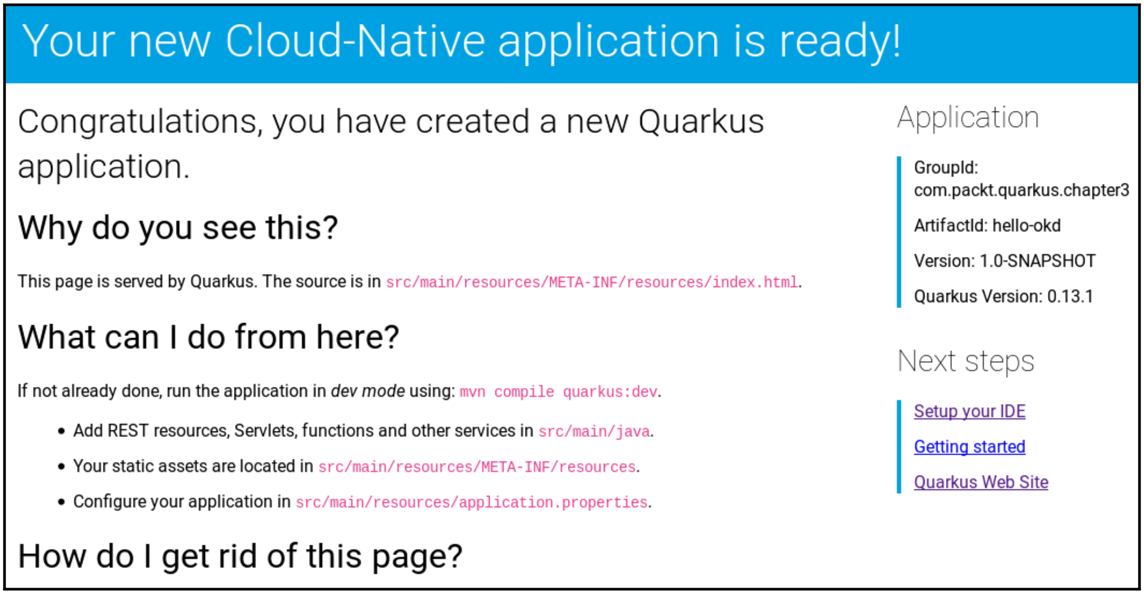
http://quarkus-hello-okd-myproject.192.168.42.5.nip.io

Route quarkus-hello-okd, target port 8080-tcp

BUILDS

quarkus-hello-okd

✓ Build #1 is complete created 7 minutes ago



Your new Cloud-Native application is ready!

Congratulations, you have created a new Quarkus application.

Why do you see this?

This page is served by Quarkus. The source is in `src/main/resources/META-INF/resources/index.html`.

What can I do from here?

If not already done, run the application in *dev mode* using: `mvn compile quarkus:dev`.

- Add REST resources, Servlets, functions and other services in `src/main/java`.
- Your static assets are located in `src/main/resources/META-INF/resources`.
- Configure your application in `src/main/resources/application.properties`.

How do I get rid of this page?

Application

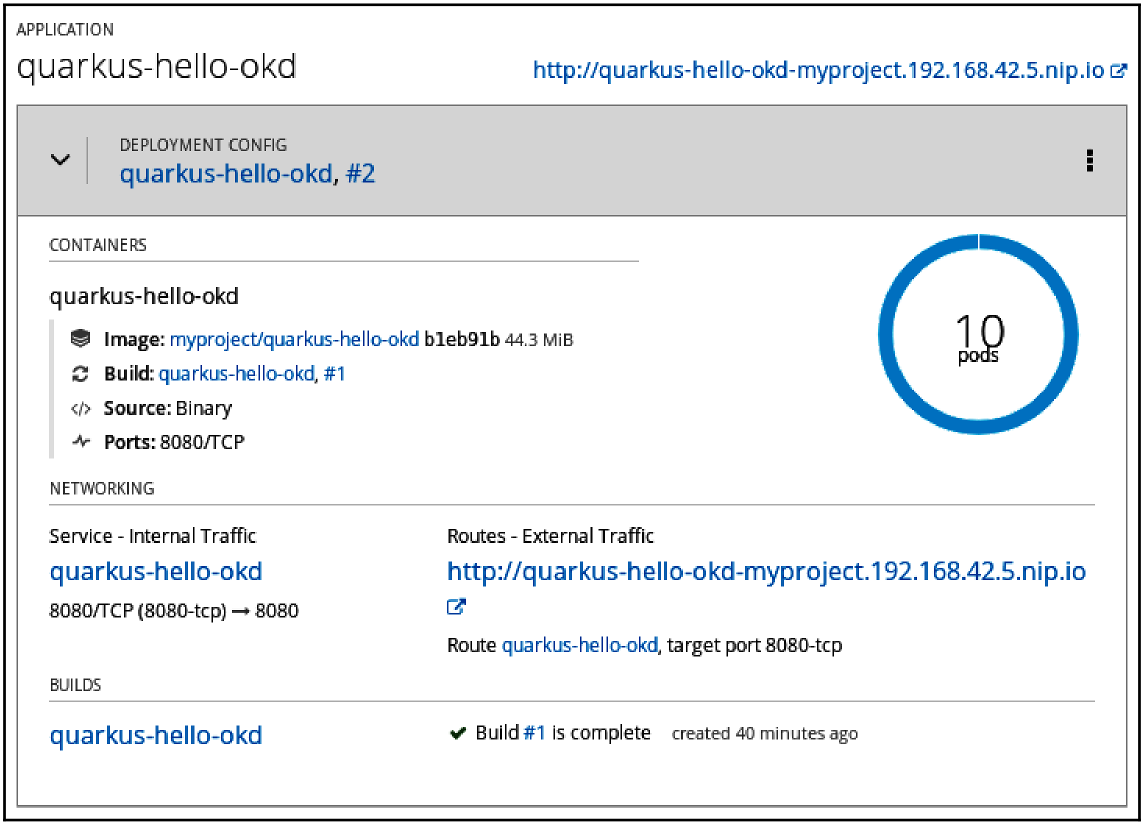
GroupId: com.packt.quarkus.chapter3

ArtifactId: hello-okd

Version: 1.0-SNAPSHOT

Quarkus Version: 0.13.1

Next steps


APPLICATION

quarkus-hello-okd

http://quarkus-hello-okd-myproject.192.168.42.5.nip.io

DEPLOYMENT CONFIG

quarkus-hello-okd, #2

CONTAINERS

quarkus-hello-okd

- **Image:** myproject/quarkus-hello-okd **b1eb91b** 44.3 MiB
- **Build:** quarkus-hello-okd, #1
- **Source:** Binary
- **Ports:** 8080/TCP

NETWORKING

Service - Internal Traffic

quarkus-hello-okd

8080/TCP (8080-tcp) → 8080

Routes - External Traffic

http://quarkus-hello-okd-myproject.192.168.42.5.nip.io

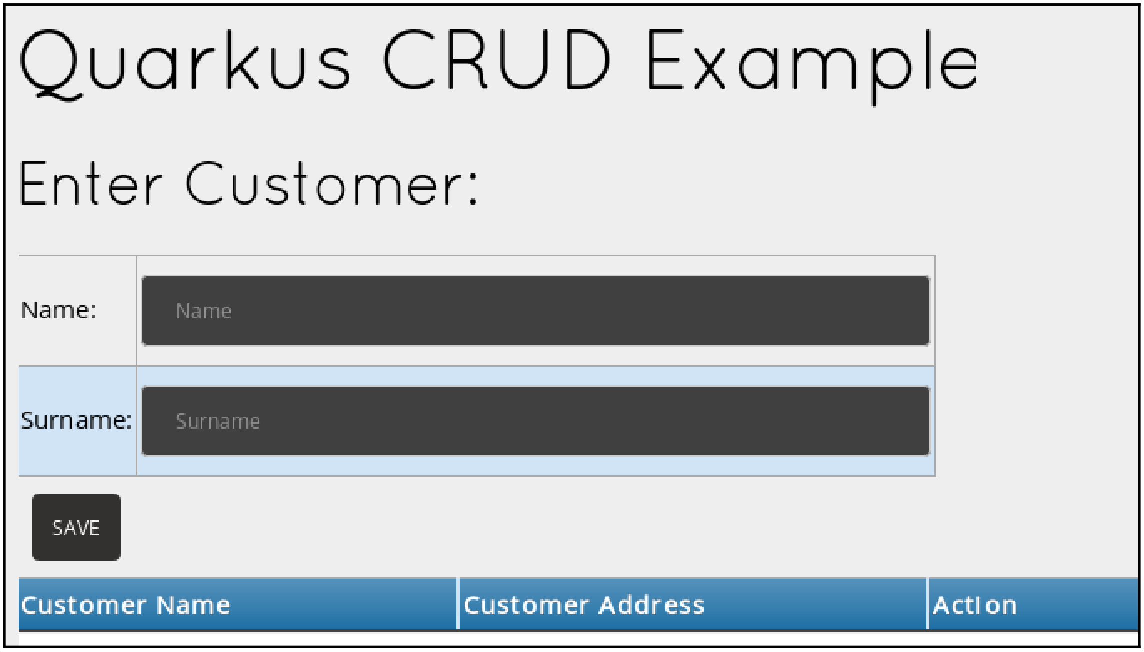
Route quarkus-hello-okd, target port 8080-tcp

BUILDS

quarkus-hello-okd

✓ Build #1 is complete created 40 minutes ago

Chapter 4: Adding Web Interfaces to Quarkus Services



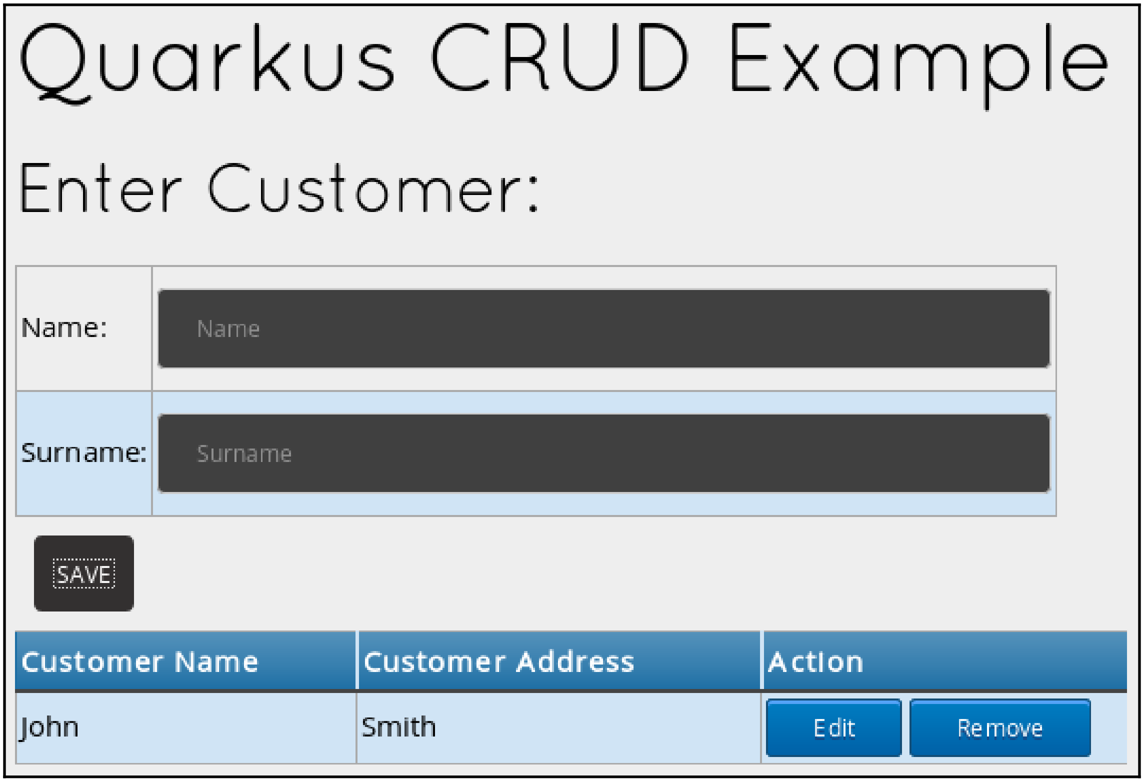
Quarkus CRUD Example

Enter Customer:

Name: Name

Surname: Surname

Customer Name	Customer Address	Action

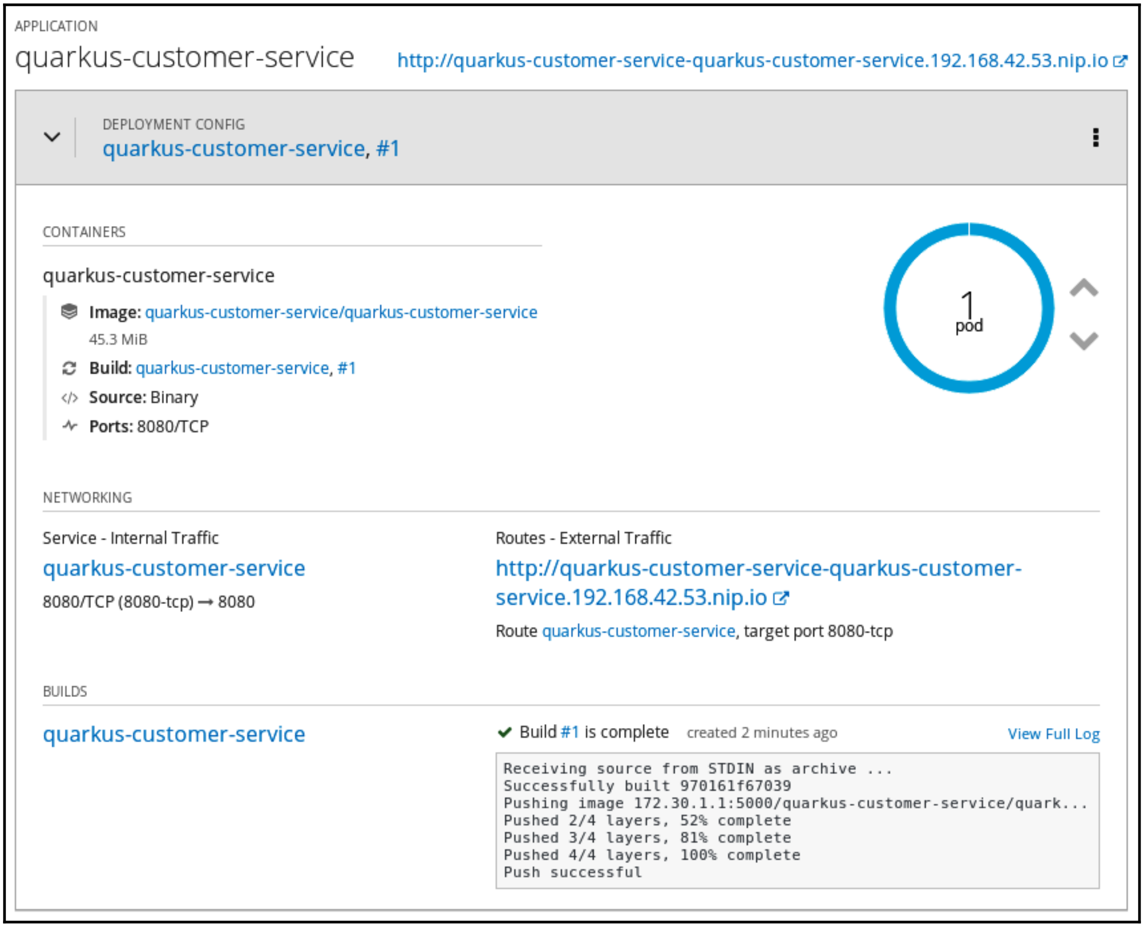


Quarkus CRUD Example

Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
John	Smith	Edit Remove



APPLICATION

quarkus-customer-service

http://quarkus-customer-service-quarkus-customer-service.192.168.42.53.nip.io

DEPLOYMENT CONFIG

quarkus-customer-service, #1

CONTAINERS

quarkus-customer-service

Image: quarkus-customer-service/quarkus-customer-service

45.3 MiB

Build: quarkus-customer-service, #1

Source: Binary

Ports: 8080/TCP

1 pod

NETWORKING

Service - Internal Traffic

quarkus-customer-service

8080/TCP (8080-tcp) → 8080

Routes - External Traffic

http://quarkus-customer-service-quarkus-customer-service.192.168.42.53.nip.io

Route quarkus-customer-service, target port 8080-tcp

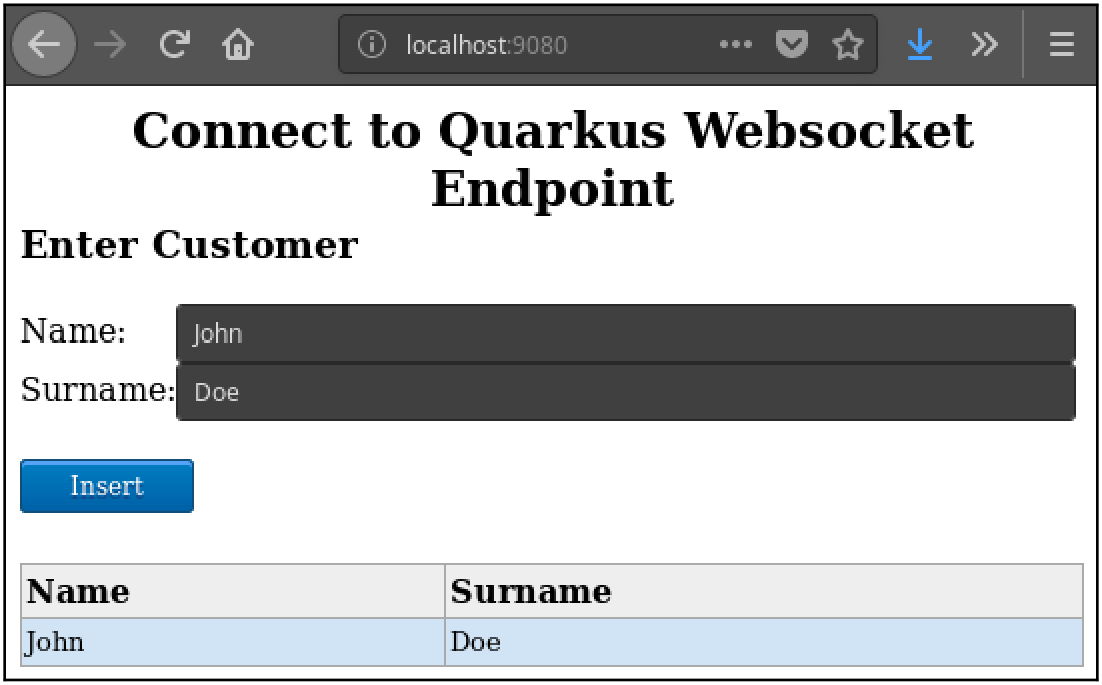
BUILDS

quarkus-customer-service

✓ Build #1 is complete created 2 minutes ago

View Full Log

```
Receiving source from STDIN as archive ...
Successfully built 970161f67039
Pushing image 172.30.1.1:5000/quarkus-customer-service/quark...
Pushed 2/4 layers, 52% complete
Pushed 3/4 layers, 81% complete
Pushed 4/4 layers, 100% complete
Push successful
```



Connect to Quarkus Websocket Endpoint

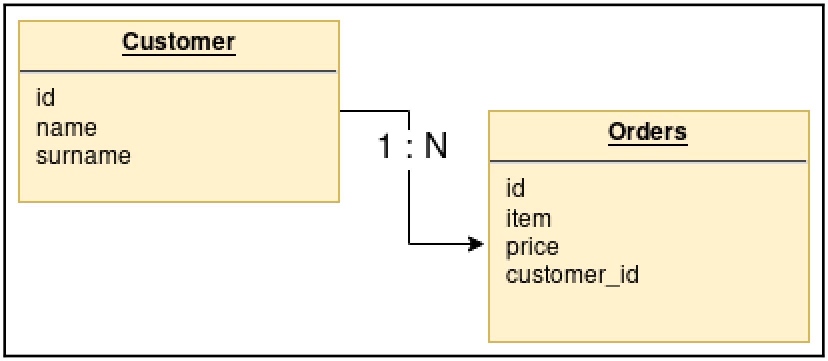
Enter Customer

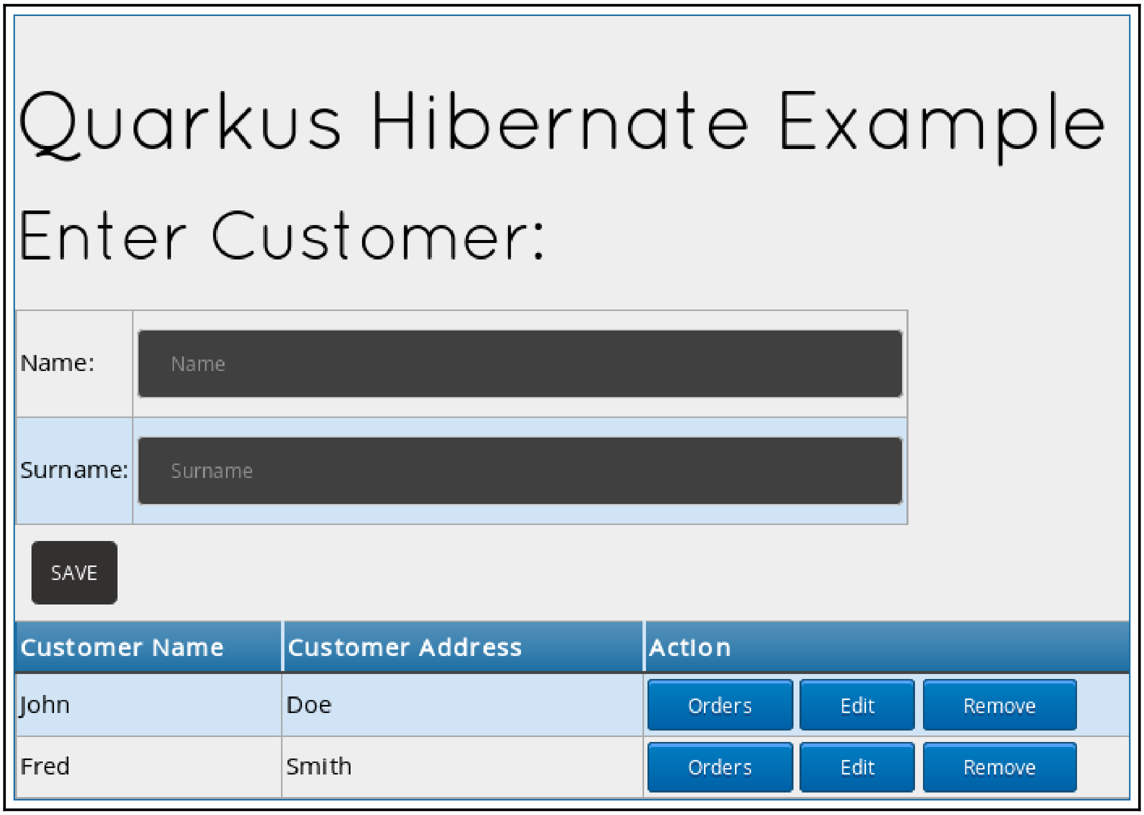
Name: John

Surname: Doe

Name	Surname
John	Doe

Chapter 5: Managing Data Persistence with Quarkus



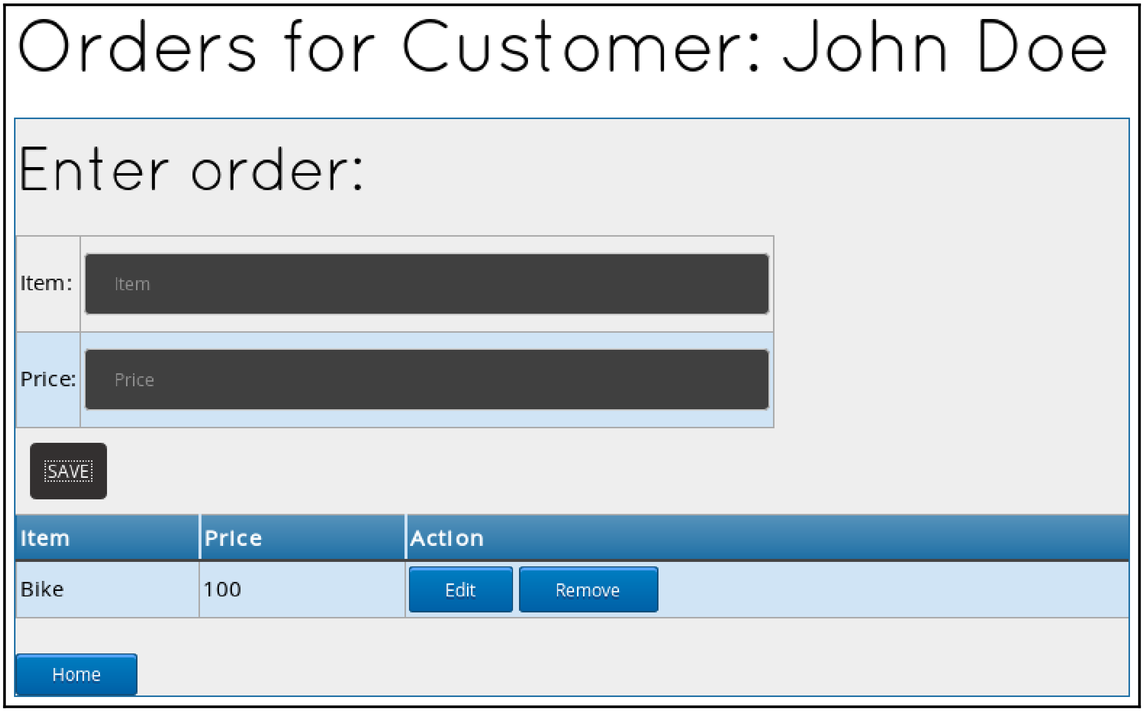


Quarkus Hibernate Example

Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
John	Doe	Orders Edit Remove
Fred	Smith	Orders Edit Remove



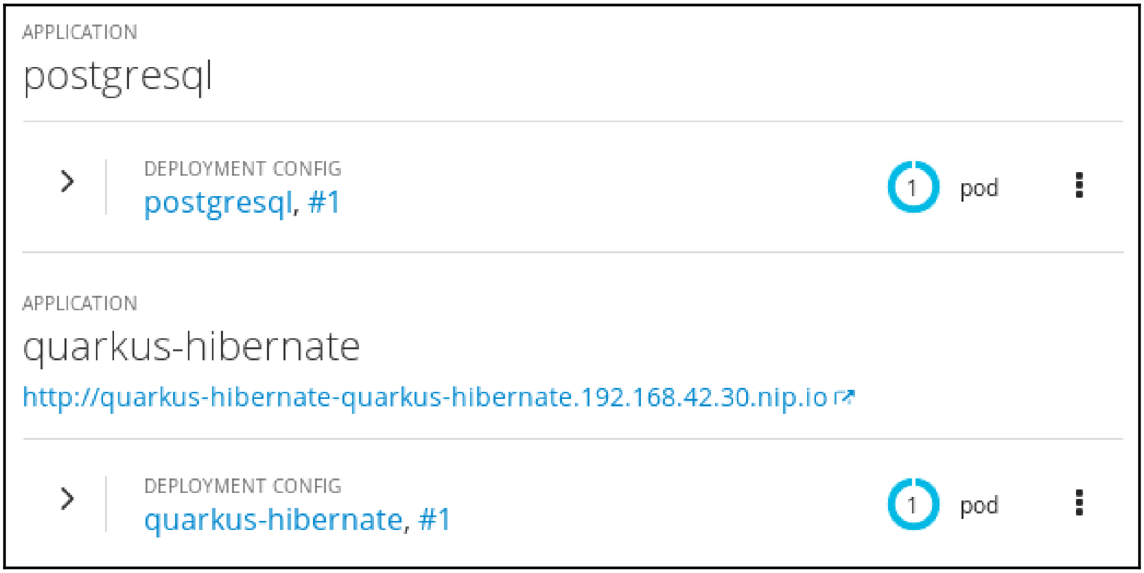
Orders for Customer: John Doe

Enter order:

Item: Item

Price: Price

Item	Price	Action
Bike	100	Edit Remove



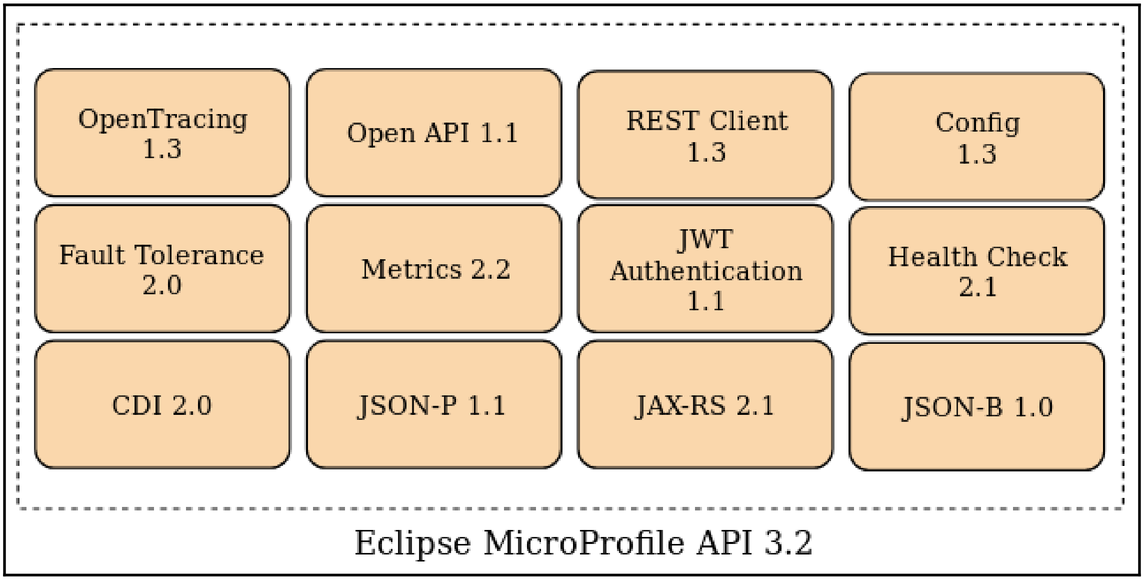
APPLICATION
postgresql

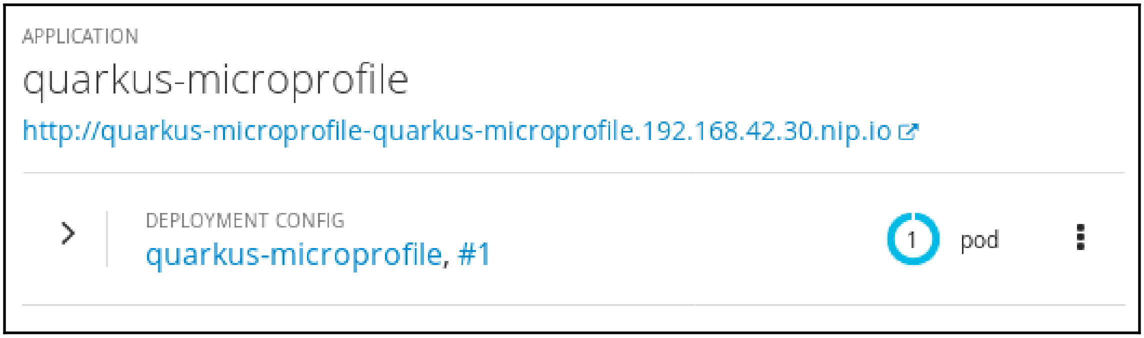
DEPLOYMENT CONFIG
postgresql, #1
1 pod

APPLICATION
quarkus-hibernate
http://quarkus-hibernate-quarkus-hibernate.192.168.42.30.nip.io

DEPLOYMENT CONFIG
quarkus-hibernate, #1
1 pod

Chapter 6: Building Applications Using the MicroProfile API




APPLICATION

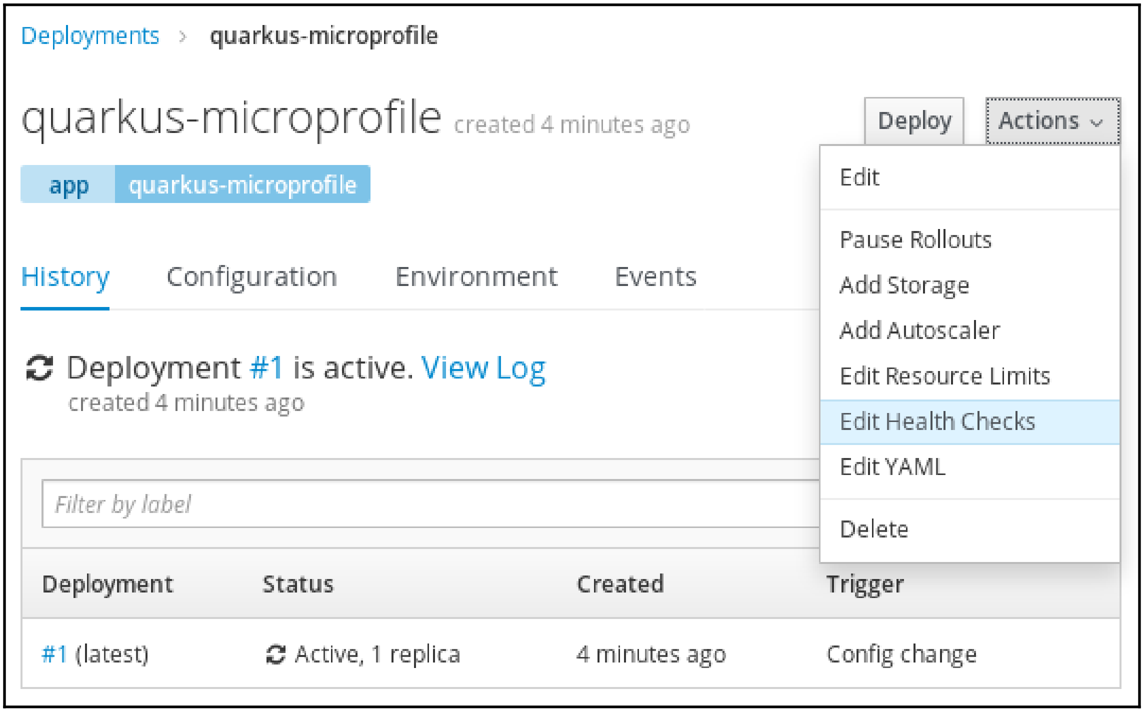
quarkus-microprofile

http://quarkus-microprofile-quarkus-microprofile.192.168.42.30.nip.io

DEPLOYMENT CONFIG

quarkus-microprofile, #1

1 pod

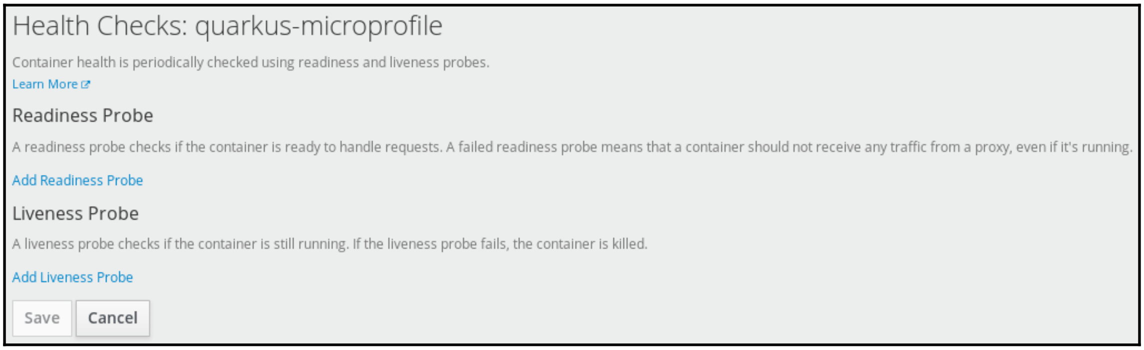


quarkus-microprofile created 4 minutes ago

app quarkus-microprofile

Deployment #1 is active. View Log
created 4 minutes ago

Deployment	Status	Created	Trigger
#1 (latest)	Active, 1 replica	4 minutes ago	Config change



Health Checks: quarkus-microprofile

Container health is periodically checked using readiness and liveness probes.

Learn More

Readiness Probe

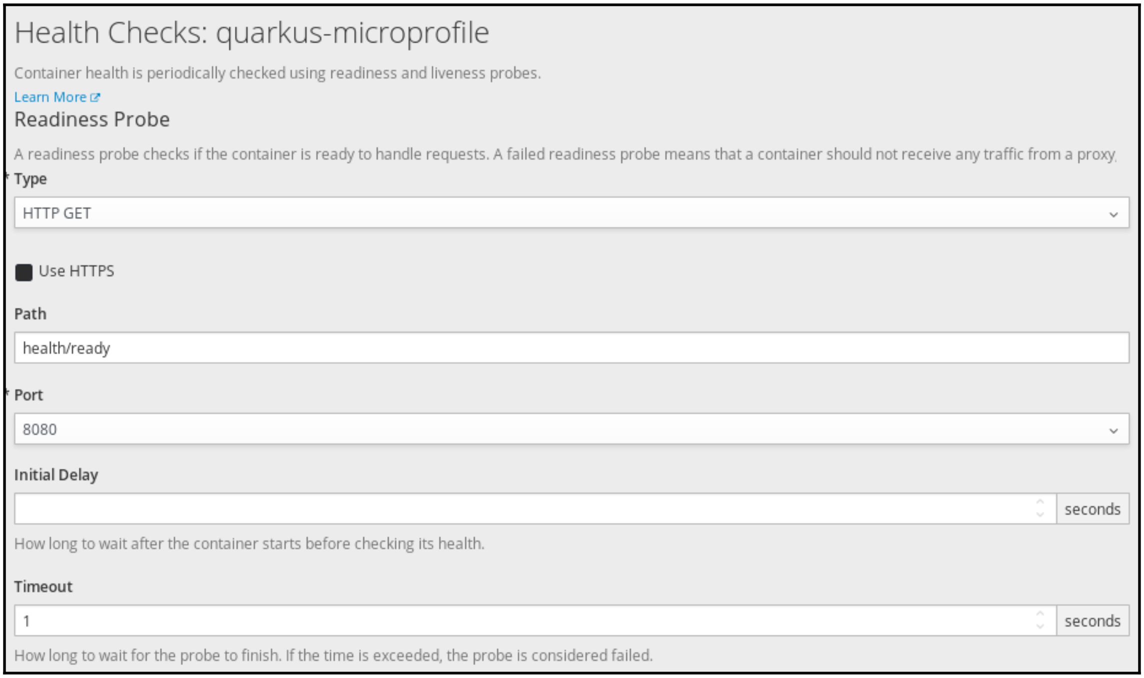
A readiness probe checks if the container is ready to handle requests. A failed readiness probe means that a container should not receive any traffic from a proxy, even if it's running.

Add Readiness Probe

Liveness Probe

A liveness probe checks if the container is still running. If the liveness probe fails, the container is killed.

Add Liveness Probe



Health Checks: quarkus-microprofile

Container health is periodically checked using readiness and liveness probes.

Learn More

Readiness Probe

A readiness probe checks if the container is ready to handle requests. A failed readiness probe means that a container should not receive any traffic from a proxy,

* Type

HTTP GET

■ Use HTTPS

Path

health/ready

* Port

8080

Initial Delay

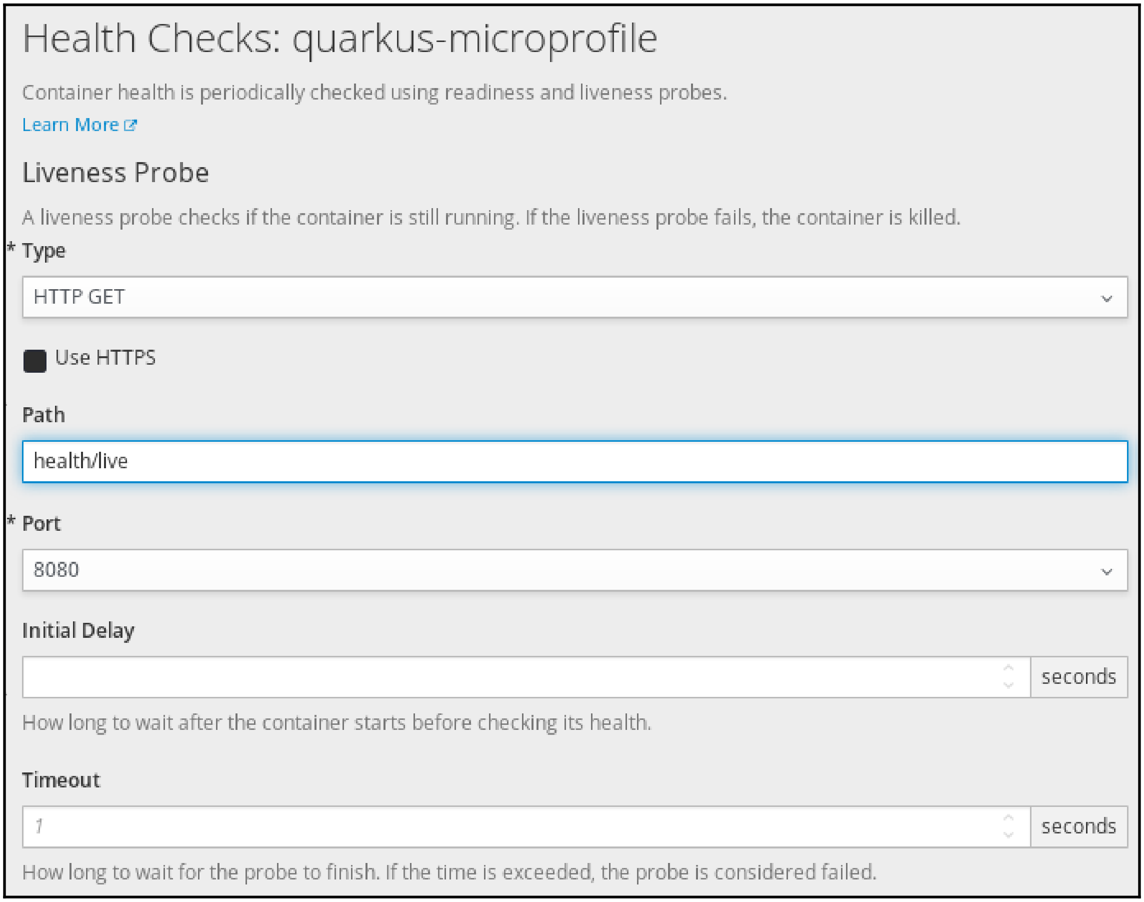
seconds

How long to wait after the container starts before checking its health.

Timeout

1 seconds

How long to wait for the probe to finish. If the time is exceeded, the probe is considered failed.



Health Checks: quarkus-microprofile

Container health is periodically checked using readiness and liveness probes.

Learn More

Liveness Probe

A liveness probe checks if the container is still running. If the liveness probe fails, the container is killed.

* Type

HTTP GET

■ Use HTTPS

Path

health/live

* Port

8080

Initial Delay

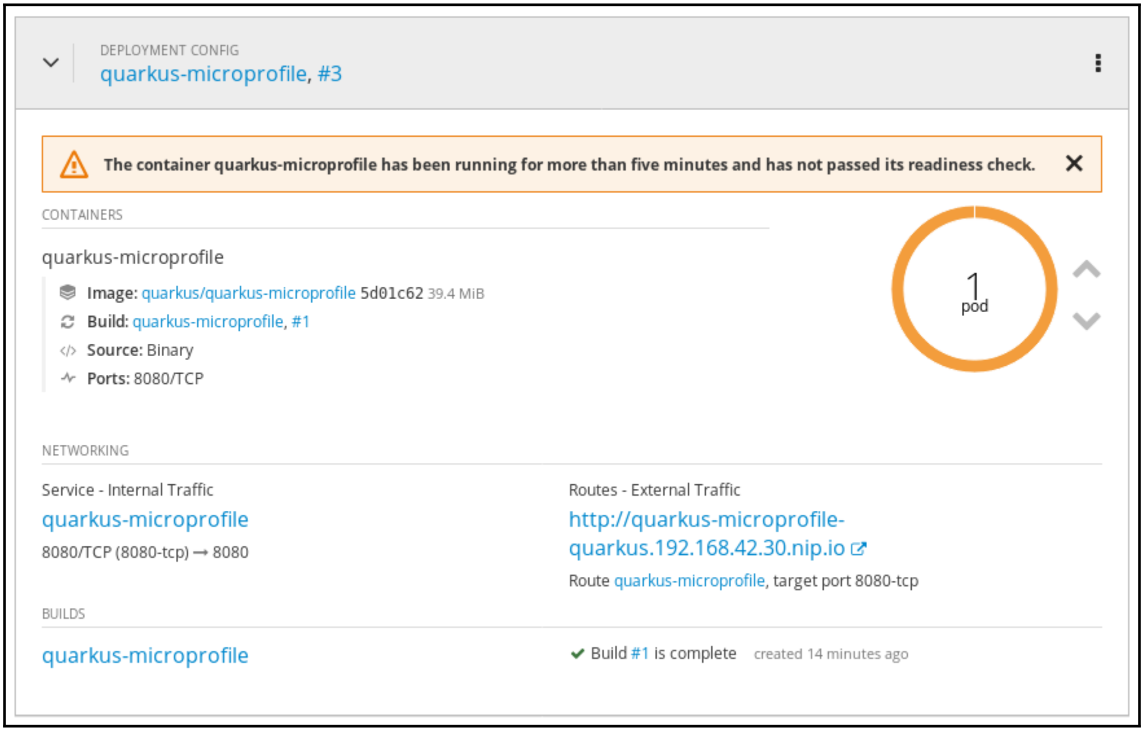
seconds

How long to wait after the container starts before checking its health.

Timeout

1 seconds

How long to wait for the probe to finish. If the time is exceeded, the probe is considered failed.



DEPLOYMENT CONFIG

quarkus-microprofile, #3

The container quarkus-microprofile has been running for more than five minutes and has not passed its readiness check.

CONTAINERS

quarkus-microprofile

Image: quarkus/quarkus-microprofile `5d01c62` 39.4 MiB

Build: quarkus-microprofile, #1

Source: Binary

Ports: 8080/TCP

NETWORKING

Service - Internal Traffic

quarkus-microprofile

8080/TCP (8080-tcp) → 8080

Routes - External Traffic

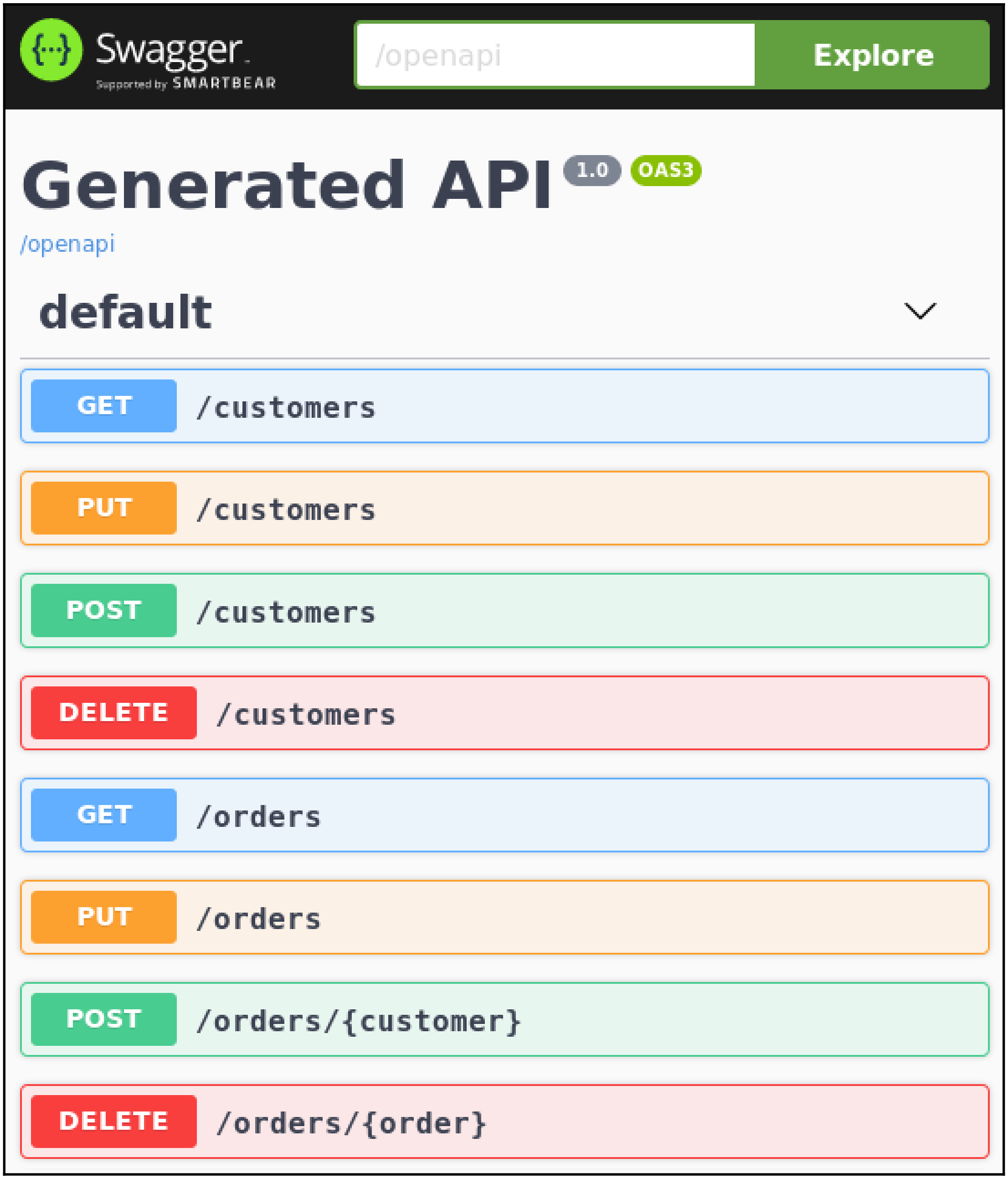
http://quarkus-microprofile-quarkus.192.168.42.30.nip.io

Route quarkus-microprofile, target port 8080-tcp

BUILDS

quarkus-microprofile

✓ Build #1 is complete   created 14 minutes ago

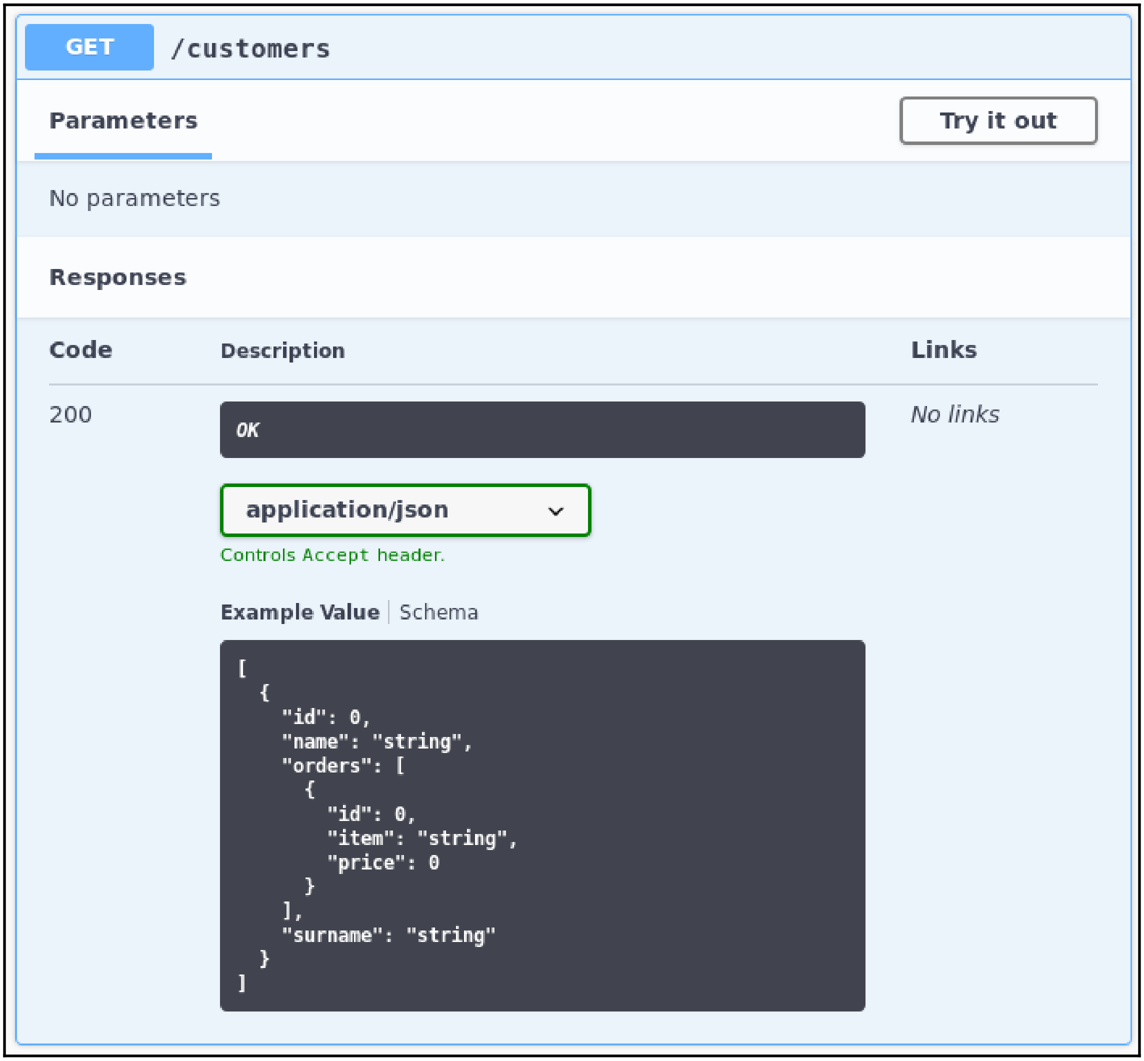


Generated API

/openapi

default

```
GET     /customers
PUT     /customers
POST    /customers
DELETE  /customers
GET     /orders
PUT     /orders
POST    /orders/{customer}
DELETE  /orders/{order}
```



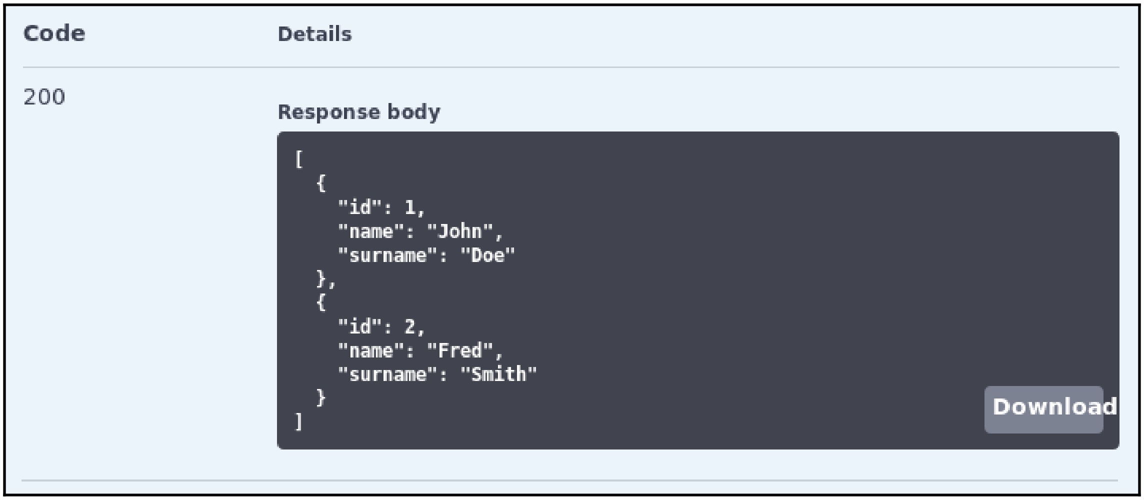
No parameters

Code	Description	Links
200	*OK*	*No links*

application/json

Controls Accept header.

```
[
  {
    "id": 0,
    "name": "string",
    "orders": [
      {
        "id": 0,
        "item": "string",
        "price": 0
      }
    ],
    "surname": "string"
  }
]
```



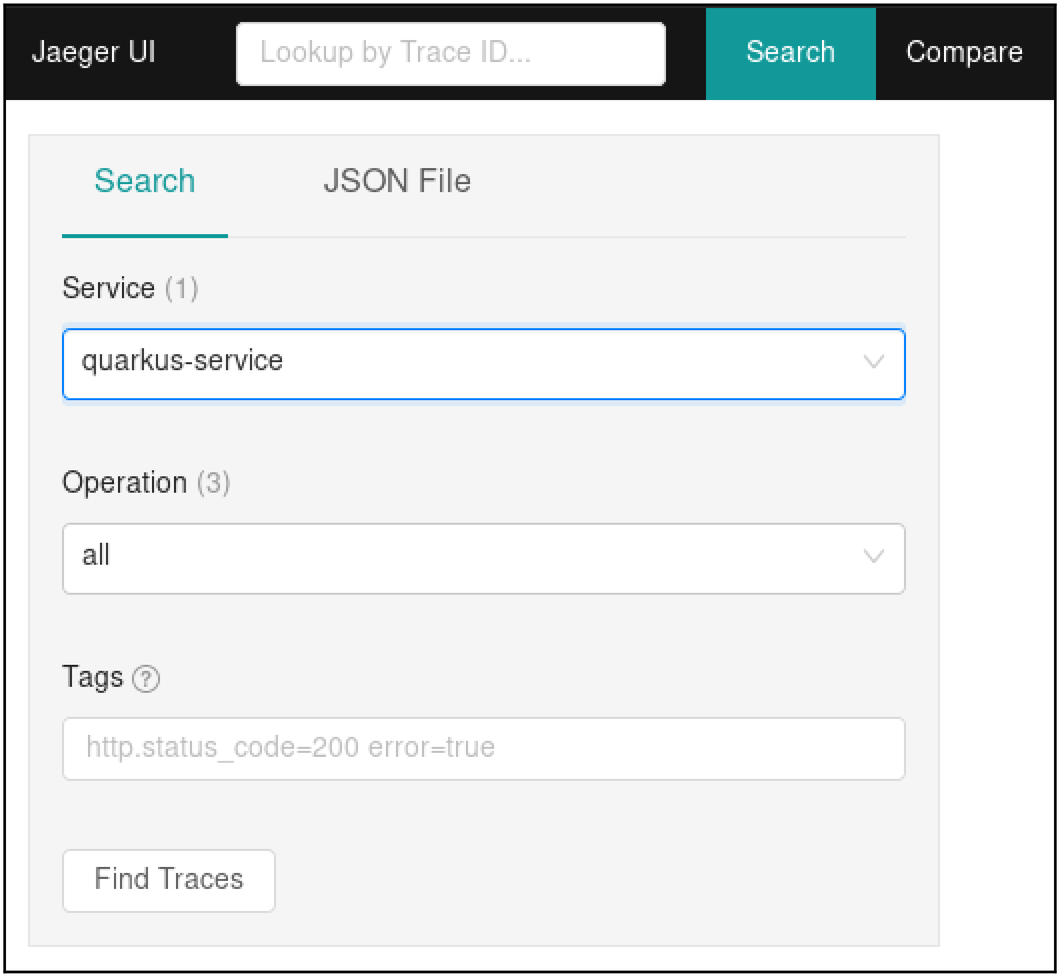
Code

Details

200

Response body

```
[
  {
    "id": 1,
    "name": "John",
    "surname": "Doe"
  },
  {
    "id": 2,
    "name": "Fred",
    "surname": "Smith"
  }
]
```

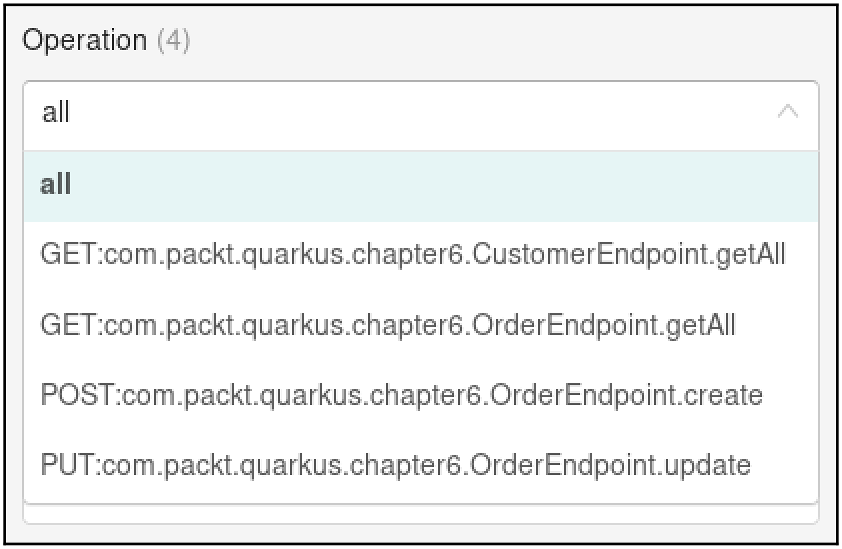


Service (1)

quarkus-service

Operation (3)

all



Operation (4)

all

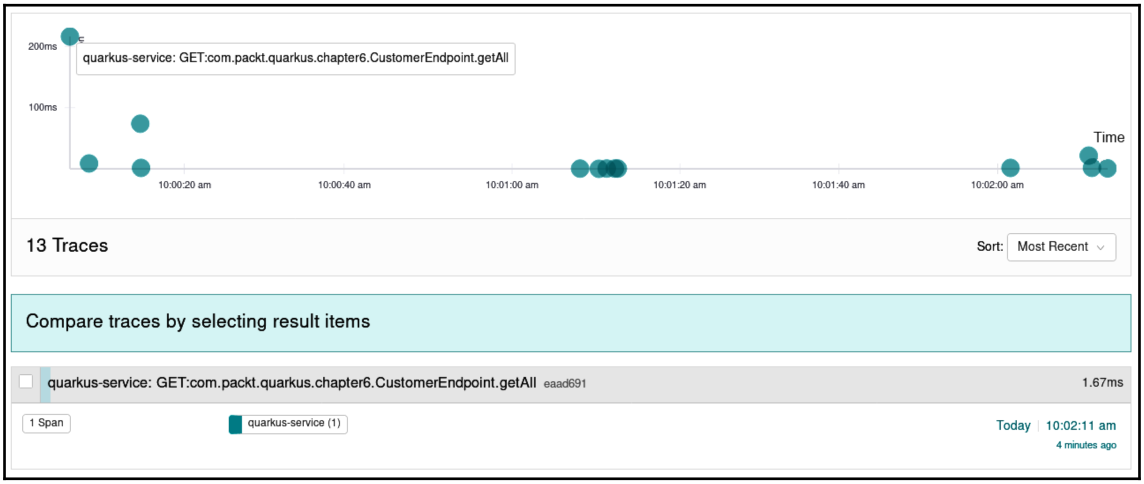
all

GET:com.packt.quarkus.chapter6.CustomerEndpoint.getAll

GET:com.packt.quarkus.chapter6.OrderEndpoint.getAll

POST:com.packt.quarkus.chapter6.OrderEndpoint.create

PUT:com.packt.quarkus.chapter6.OrderEndpoint.update



13 Traces

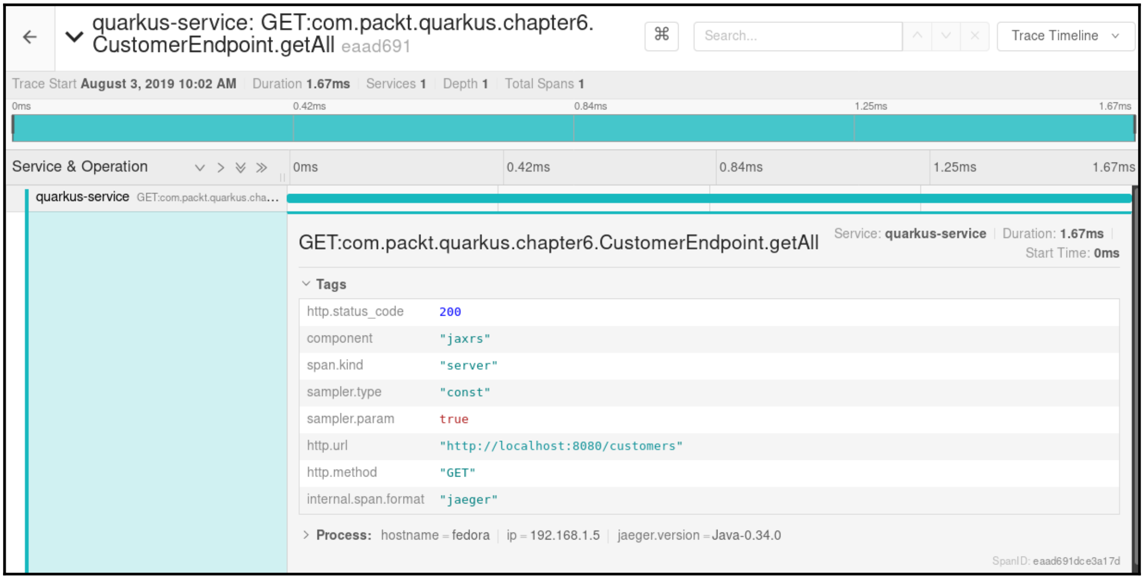
Sort: Most Recent

Compare traces by selecting result items

□ quarkus-service: GET:com.packt.quarkus.chapter6.CustomerEndpoint.getAll eaad691 1.67ms

1 Span quarkus-service (1) Today | 10:02:11 am

4 minutes ago



quarkus-service: GET:com.packt.quarkus.chapter6.CustomerEndpoint.getAll eaad691

Trace Start **August 3, 2019 10:02 AM** | Duration **1.67ms** | Services **1** | Depth **1** | Total Spans **1**

quarkus-service GET:com.packt.quarkus.cha...

GET:com.packt.quarkus.chapter6.CustomerEndpoint.getAll

Service: **quarkus-service** | Duration: **1.67ms** | Start Time: **0ms**

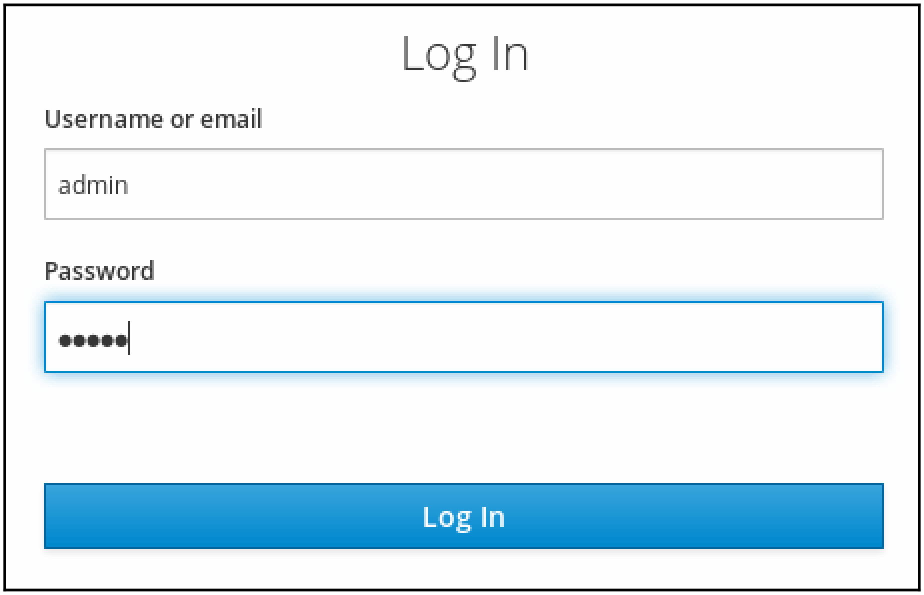
Tags

http.status_code	200
component	"jaxrs"
span.kind	"server"
sampler.type	"const"
sampler.param	true
http.url	"http://localhost:8080/customers"
http.method	"GET"
internal.span.format	"jaeger"

Process: hostname = fedora | ip = 192.168.1.5 | jaeger.version = Java-0.34.0

SpanID: eaad691dce3a17d

Chapter 7: Securing Applications

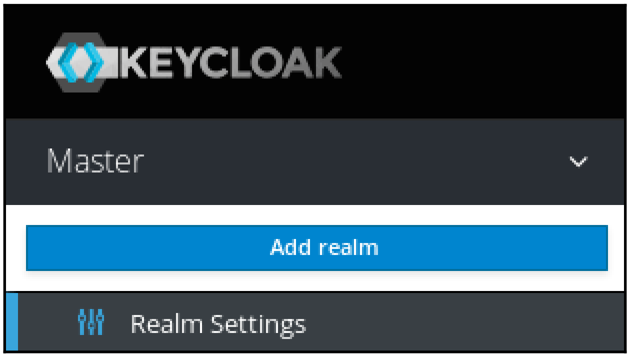

Log In

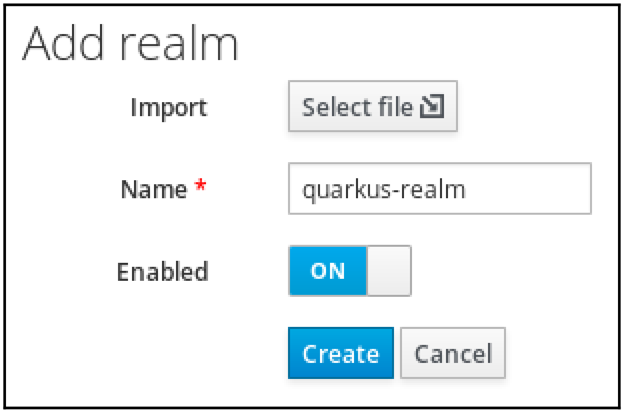
Username or email

admin

Password

●●●●●



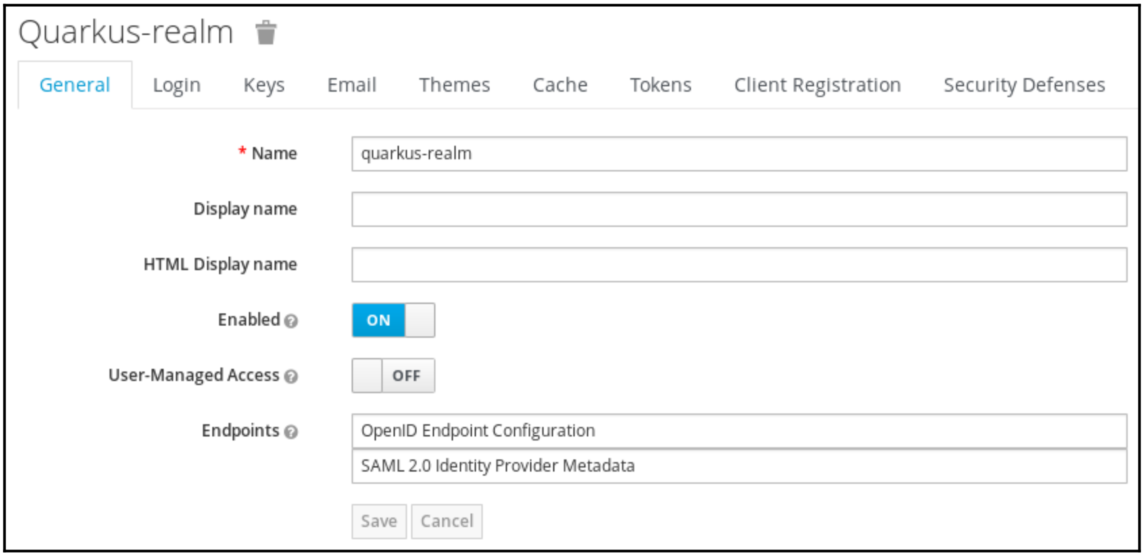


Add realm

Import

Name *: quarkus-realm

Enabled: ON



Quarkus-realm

* Name: quarkus-realm

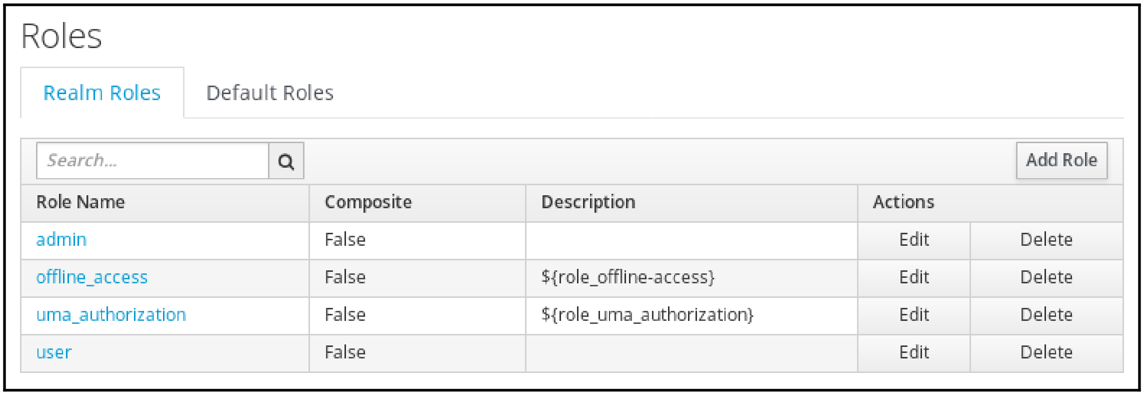
Display name:

HTML Display name:

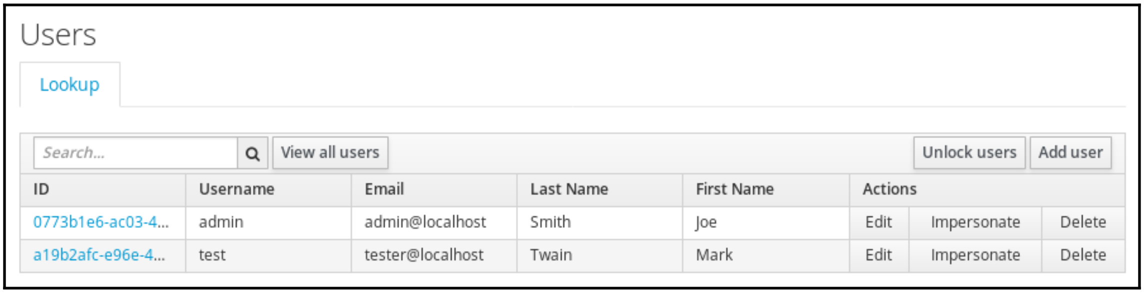
Enabled: ON

User-Managed Access: OFF

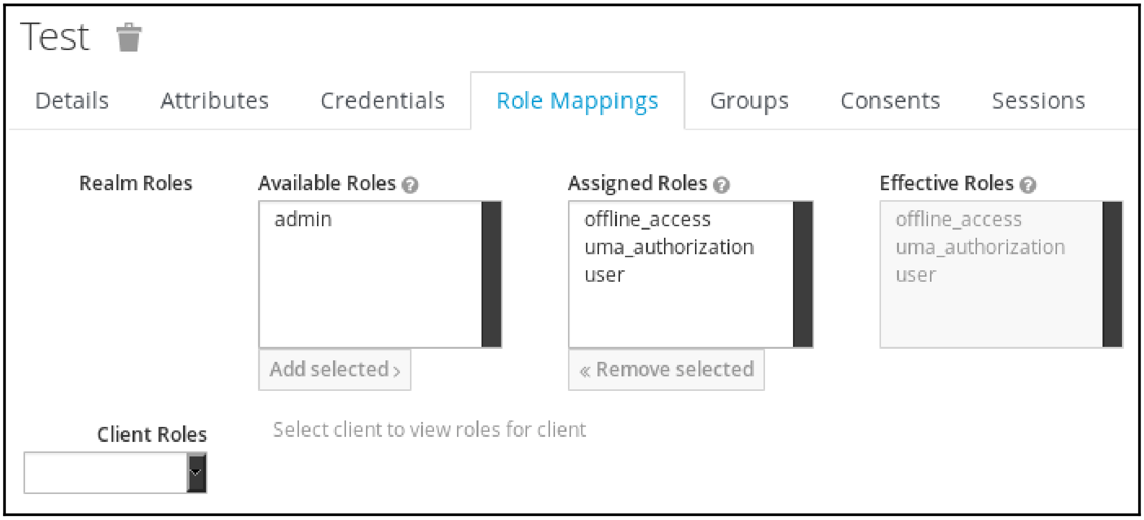
Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata



Role Name	Composite	Description	Actions	
admin	False		Edit	Delete
offline_access	False	${role_offline-access}	Edit	Delete
uma_authorization	False	${role_uma_authorization}	Edit	Delete
user	False		Edit	Delete



ID	Username	Email	Last Name	First Name	Actions		
0773b1e6-ac03-4...	admin	admin@localhost	Smith	Joe	Edit	Impersonate	Delete
a19b2afc-e96e-4...	test	tester@localhost	Twain	Mark	Edit	Impersonate	Delete



Realm Roles

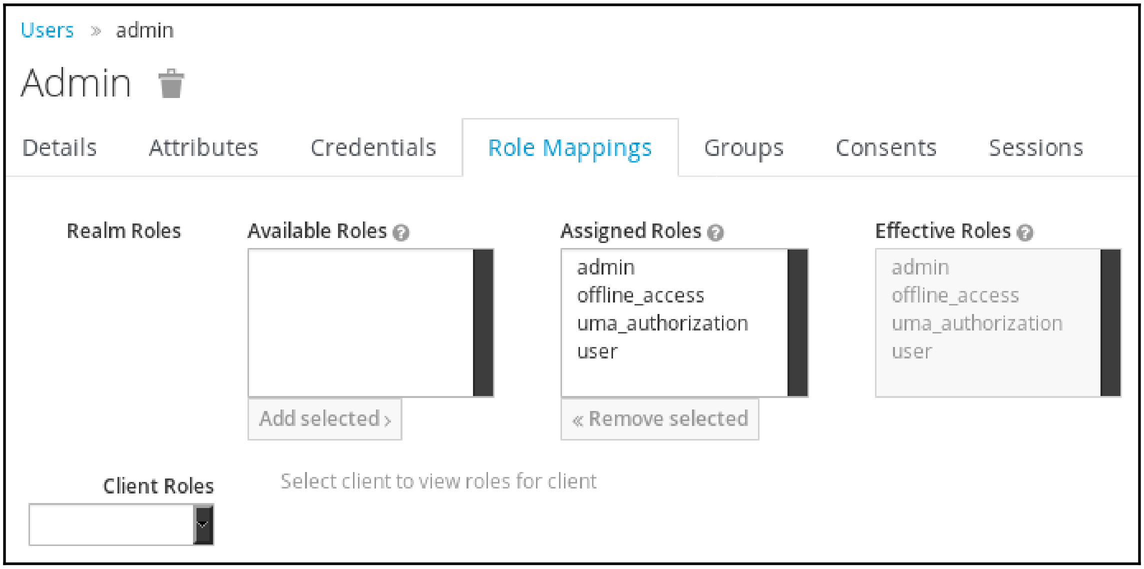
Available Roles: admin

Assigned Roles: offline_access, uma_authorization, user

Effective Roles: offline_access, uma_authorization, user

Client Roles

Select client to view roles for client



Realm Roles

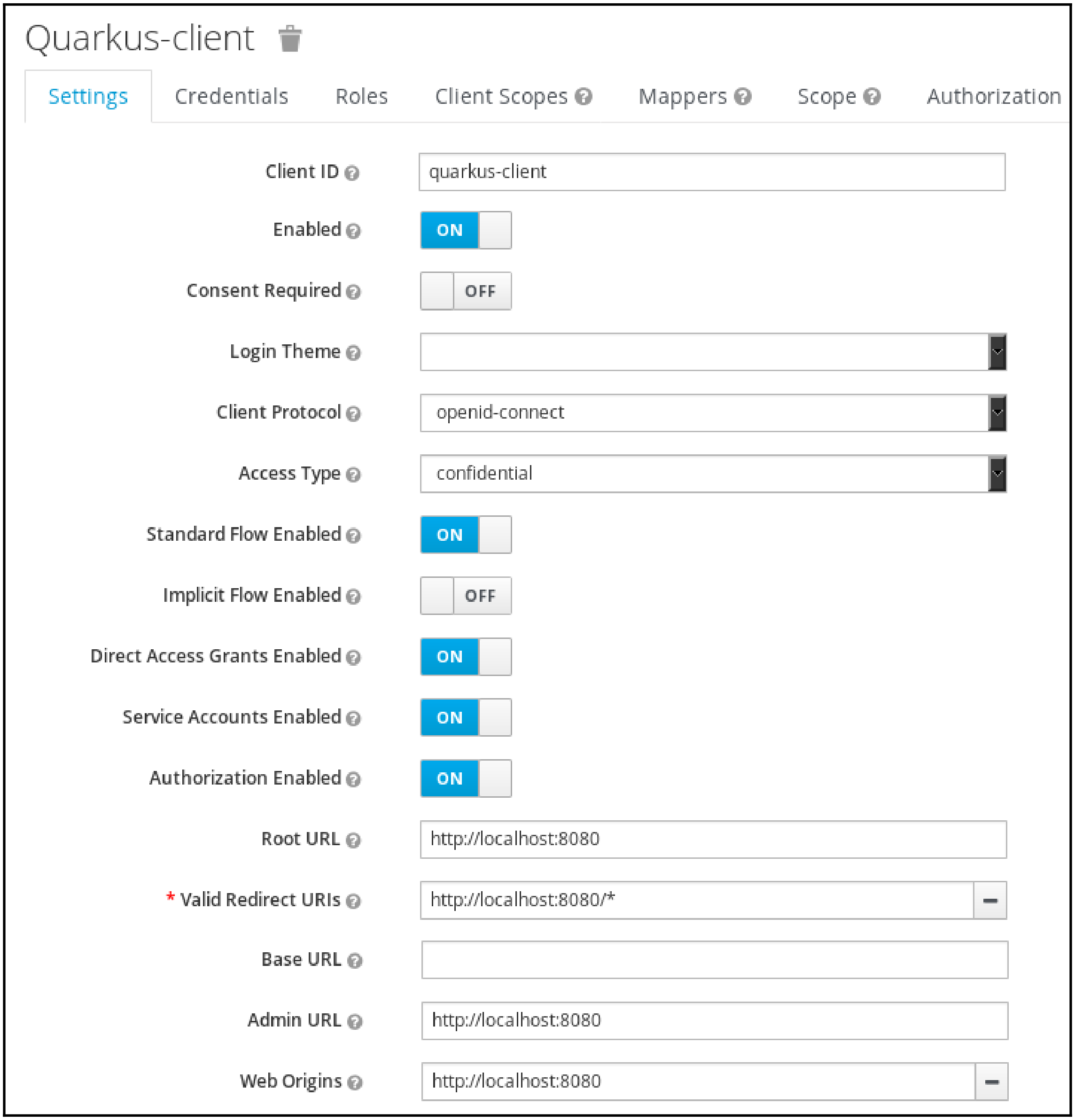
Available Roles

Assigned Roles
admin
offline_access
uma_authorization
user

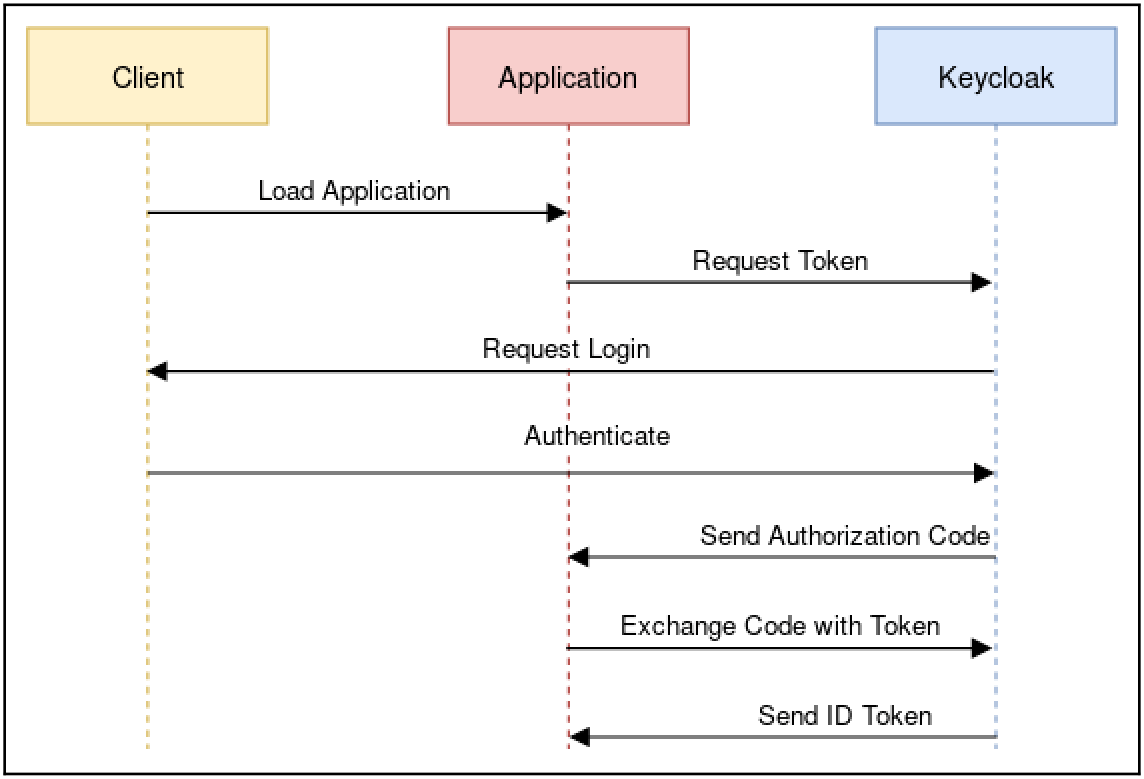
Effective Roles
admin
offline_access
uma_authorization
user

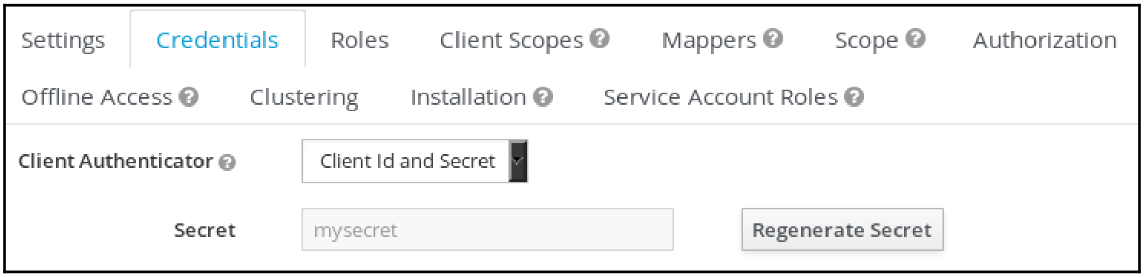
Client Roles

Select client to view roles for client



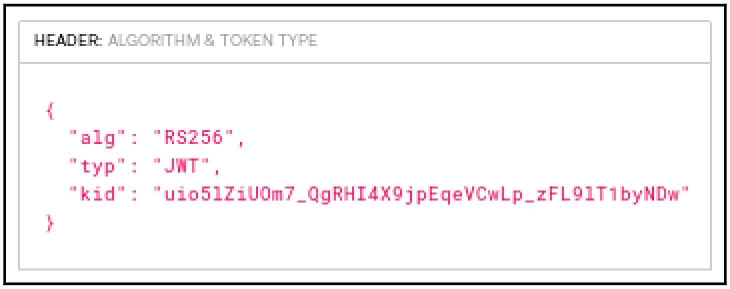
Client ID	quarkus-client
Enabled	ON
Consent Required	OFF
Login Theme	
Client Protocol	openid-connect
Access Type	confidential
Standard Flow Enabled	ON
Implicit Flow Enabled	OFF
Direct Access Grants Enabled	ON
Service Accounts Enabled	ON
Authorization Enabled	ON
Root URL	http://localhost:8080
* Valid Redirect URIs	http://localhost:8080/*
Base URL	
Admin URL	http://localhost:8080
Web Origins	http://localhost:8080





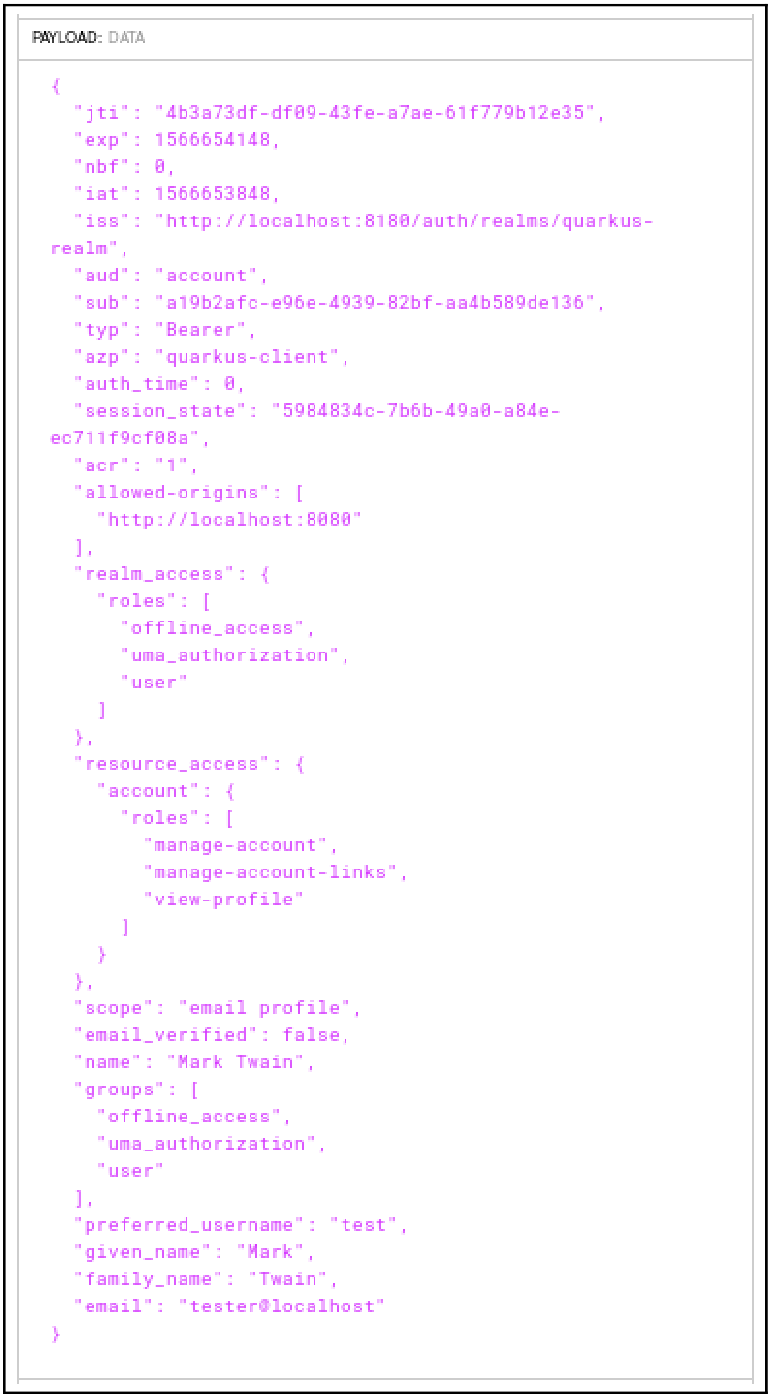
Client Authenticator: Client Id and Secret

Secret: mysecret

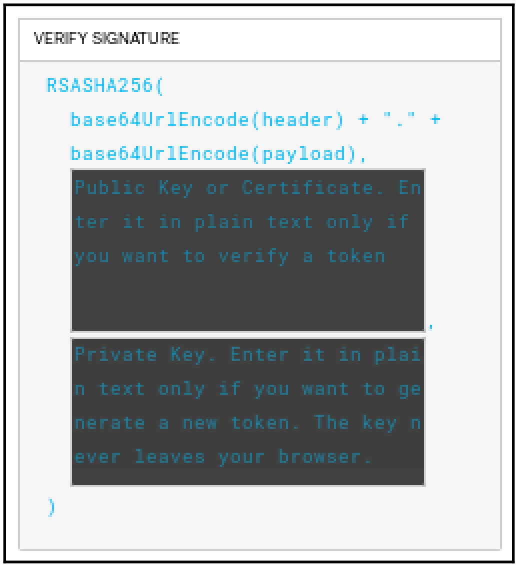


HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT",
  "kid": "uio51ZiUOm7_QgRHI4X9jpEqeVCwLp_zFL91T1byNDw"
}
```

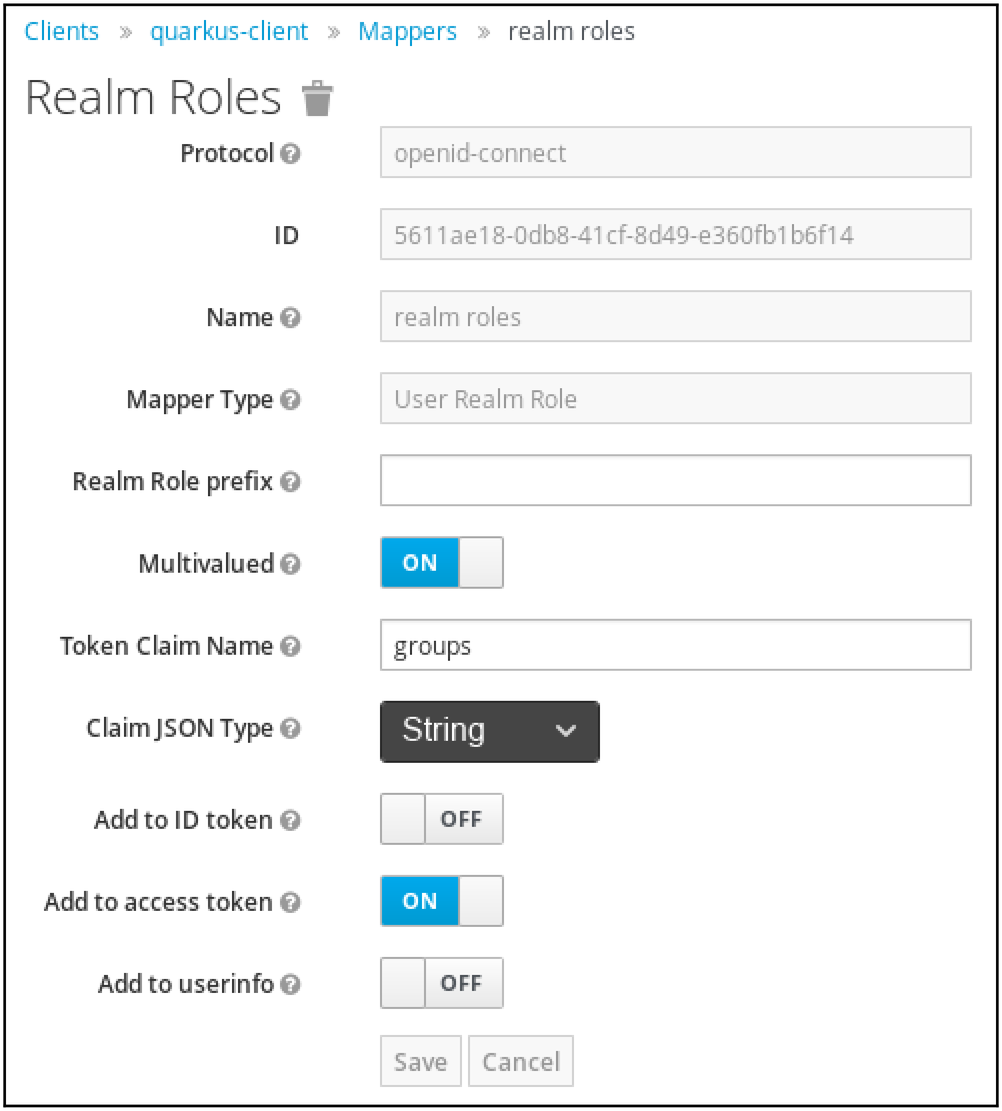


```
{
  "jti": "4b3a73df-df09-43fe-a7ae-61f779b12e35",
  "exp": 1566654148,
  "nbf": 0,
  "iat": 1566653848,
  "iss": "http://localhost:8180/auth/realms/quarkus-
realm",
  "aud": "account",
  "sub": "a19b2afc-e96e-4939-82bf-aa4b589de136",
  "typ": "Bearer",
  "azp": "quarkus-client",
  "auth_time": 0,
  "session_state": "5984834c-7b6b-49a0-a84e-
ec711f9cf08a",
  "acr": "1",
  "allowed-origins": [
    "http://localhost:8080"
  ],
  "realm_access": {
    "roles": [
      "offline_access",
      "uma_authorization",
      "user"
    ]
  },
  "resource_access": {
    "account": {
      "roles": [
        "manage-account",
        "manage-account-links",
        "view-profile"
      ]
    }
  },
  "scope": "email profile",
  "email_verified": false,
  "name": "Mark Twain",
  "groups": [
    "offline_access",
    "uma_authorization",
    "user"
  ],
  "preferred_username": "test",
  "given_name": "Mark",
  "family_name": "Twain",
  "email": "tester@localhost"
}
```



VERIFY SIGNATURE

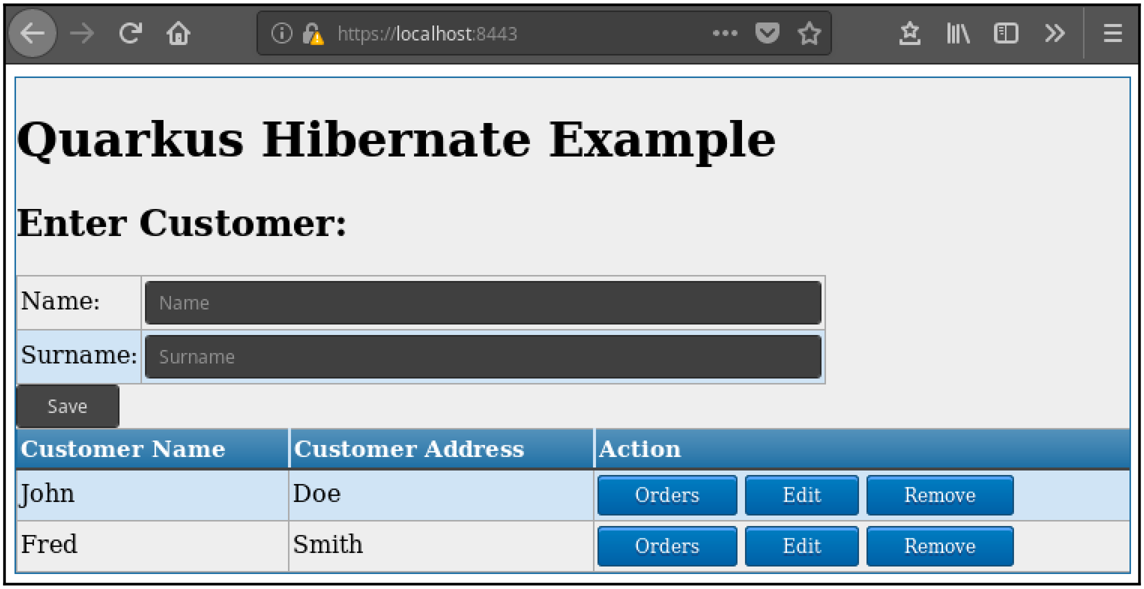
```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  Public Key or Certificate. En
  ter it in plain text only if
  you want to verify a token
                                ,
  Private Key. Enter it in plai
  n text only if you want to ge
  nerate a new token. The key n
  ever leaves your browser.
)
```



Clients » quarkus-client » Mappers » realm roles

Realm Roles

Protocol	openid-connect
ID	5611ae18-0db8-41cf-8d49-e360fb1b6f14
Name	realm roles
Mapper Type	User Realm Role
Realm Role prefix	
Multivalued	ON
Token Claim Name	groups
Claim JSON Type	String
Add to ID token	OFF
Add to access token	ON
Add to userinfo	OFF



Quarkus Hibernate Example

Enter Customer:

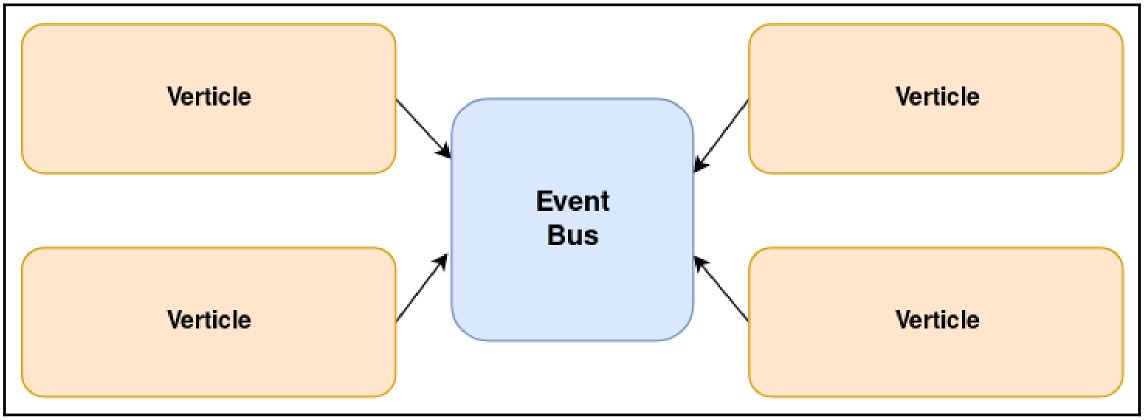
Name:	Name
Surname:	Surname

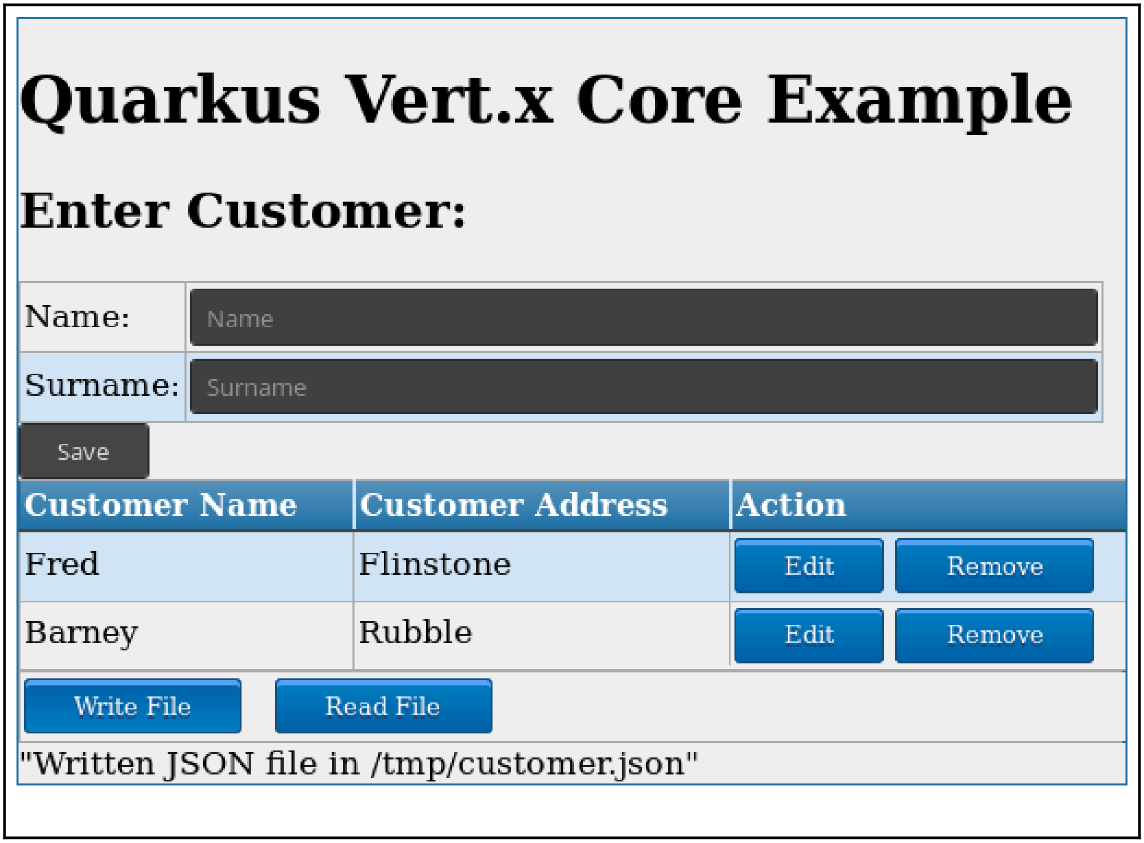
Customer Name	Customer Address	Action
John	Doe	Orders Edit Remove
Fred	Smith	Orders Edit Remove

Chapter 8: Advanced Application Development

No images.

Chapter 9: Unifying Imperative and Reactive with Vert.x





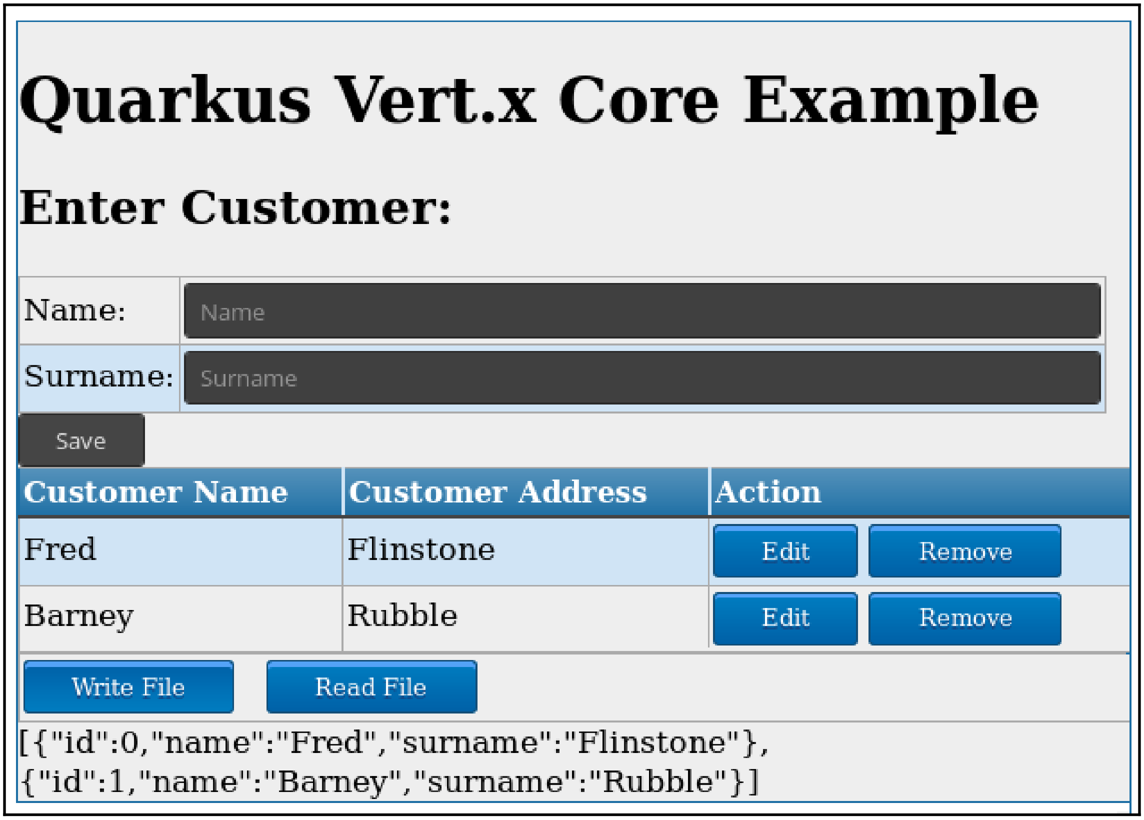
Quarkus Vert.x Core Example

Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Flinstone	Edit Remove
Barney	Rubble	Edit Remove

"Written JSON file in /tmp/customer.json"



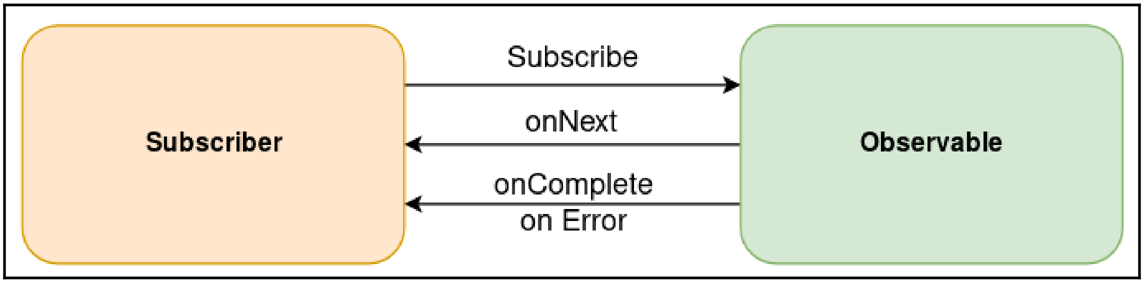
Quarkus Vert.x Core Example

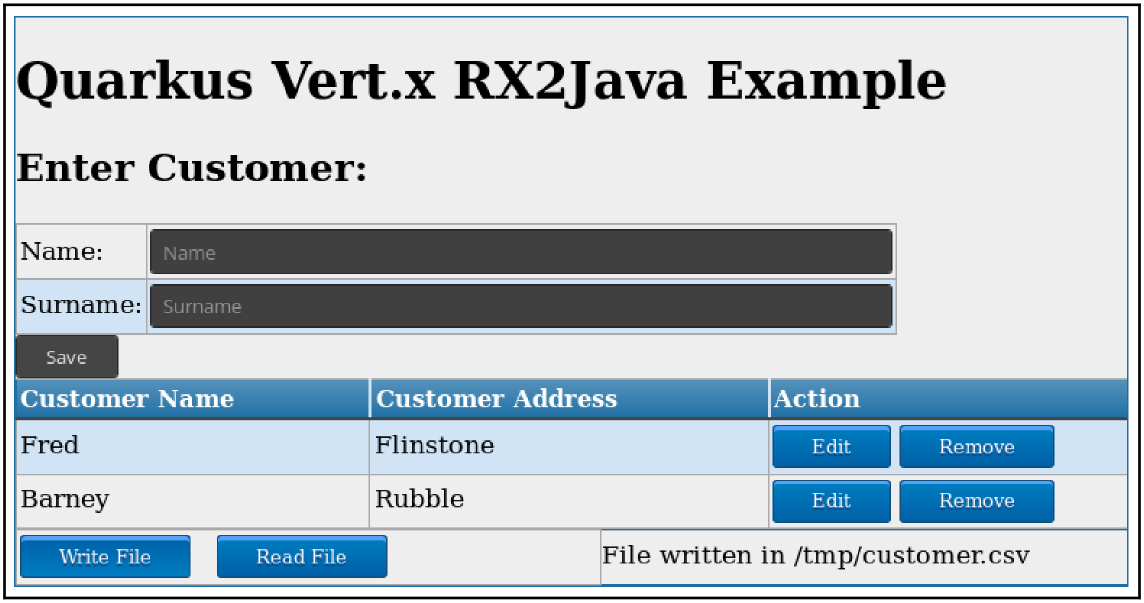
Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Flinstone	Edit Remove
Barney	Rubble	Edit Remove

```
[{"id":0,"name":"Fred","surname":"Flinstone"},
{"id":1,"name":"Barney","surname":"Rubble"}]
```





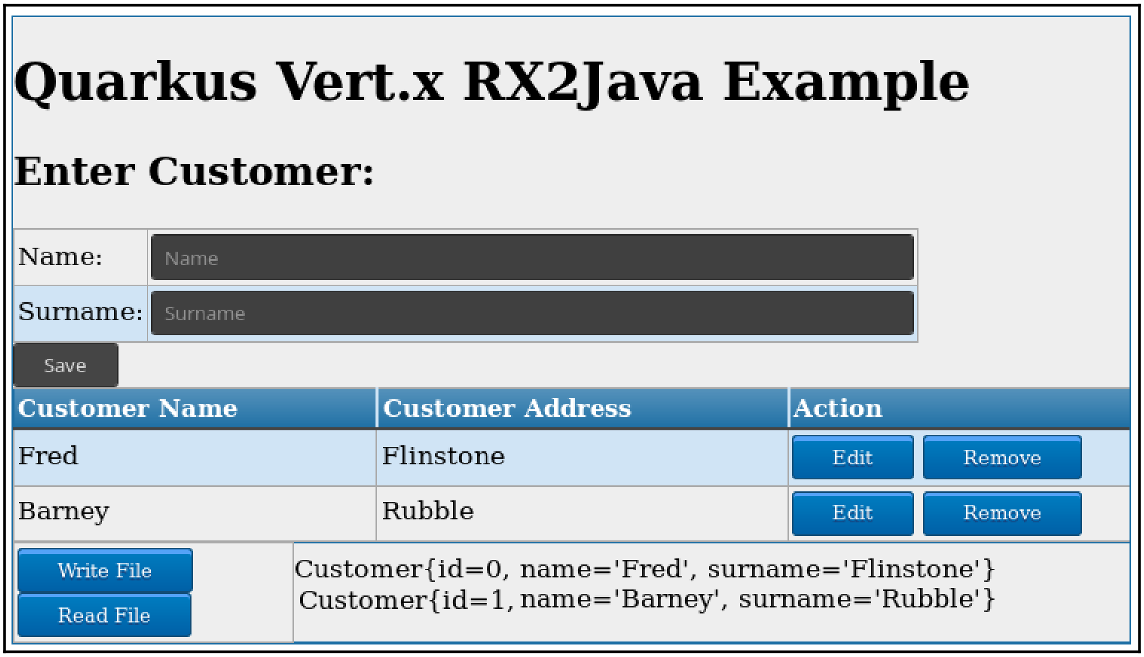
Quarkus Vert.x RX2Java Example

Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Flinstone	Edit Remove
Barney	Rubble	Edit Remove

File written in /tmp/customer.csv



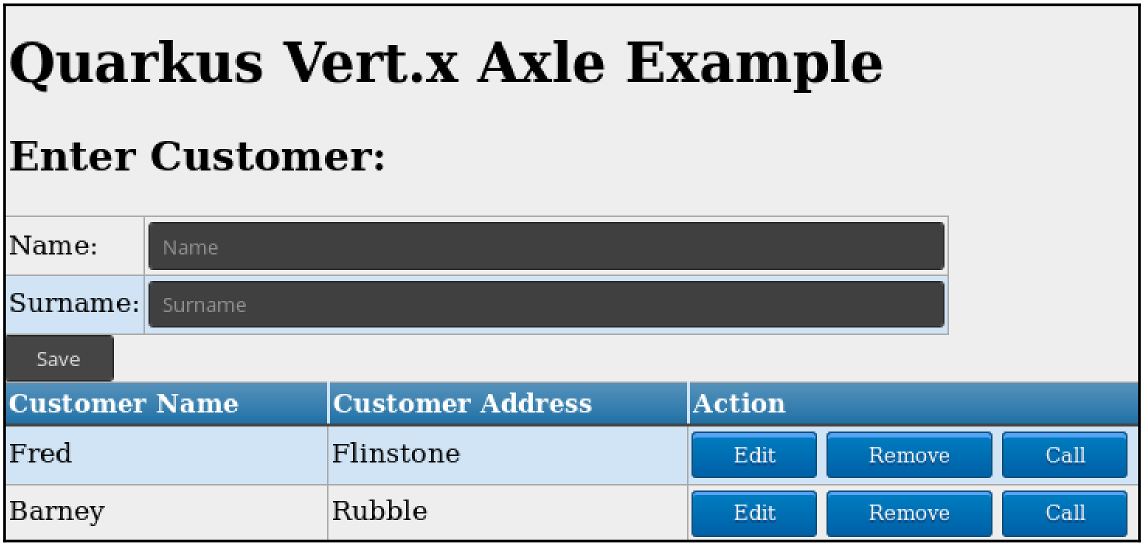
Quarkus Vert.x RX2Java Example

Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Flinstone	Edit Remove
Barney	Rubble	Edit Remove

```
Customer{id=0, name='Fred', surname='Flinstone'}
 Customer{id=1, name='Barney', surname='Rubble'}
```

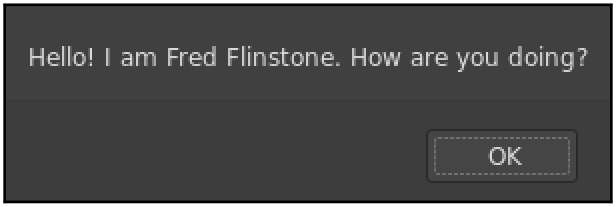


Quarkus Vert.x Axle Example

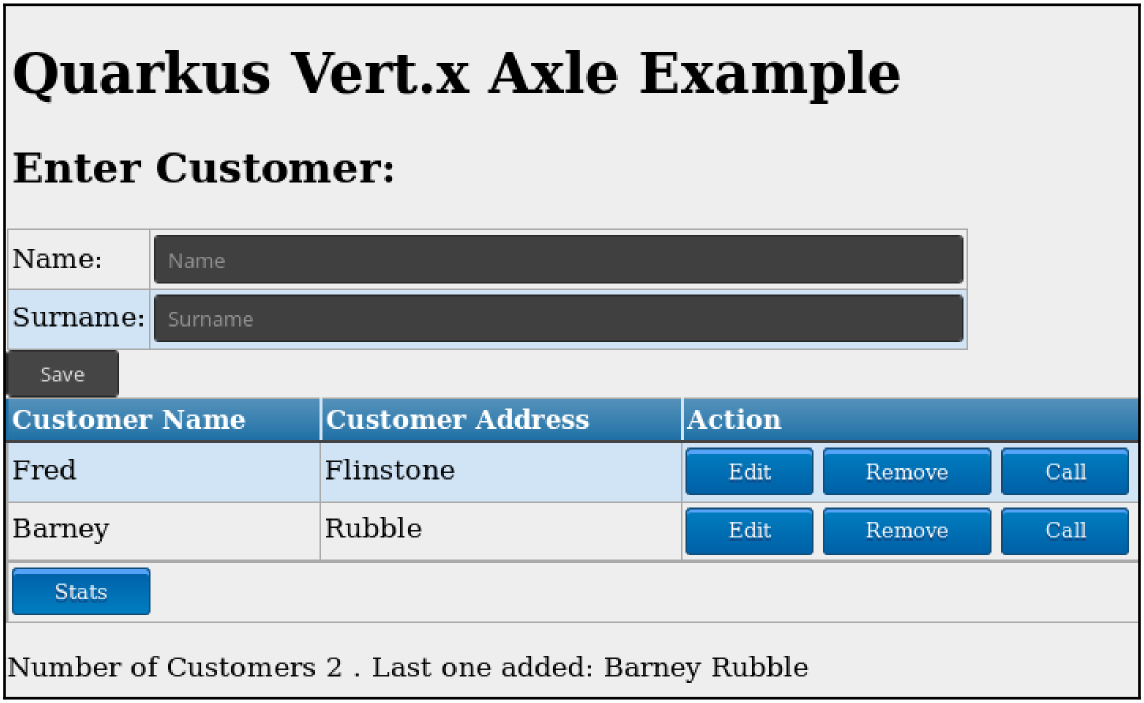
Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Flinstone	Edit Remove Call
Barney	Rubble	Edit Remove Call



Hello! I am Fred Flinstone. How are you doing?



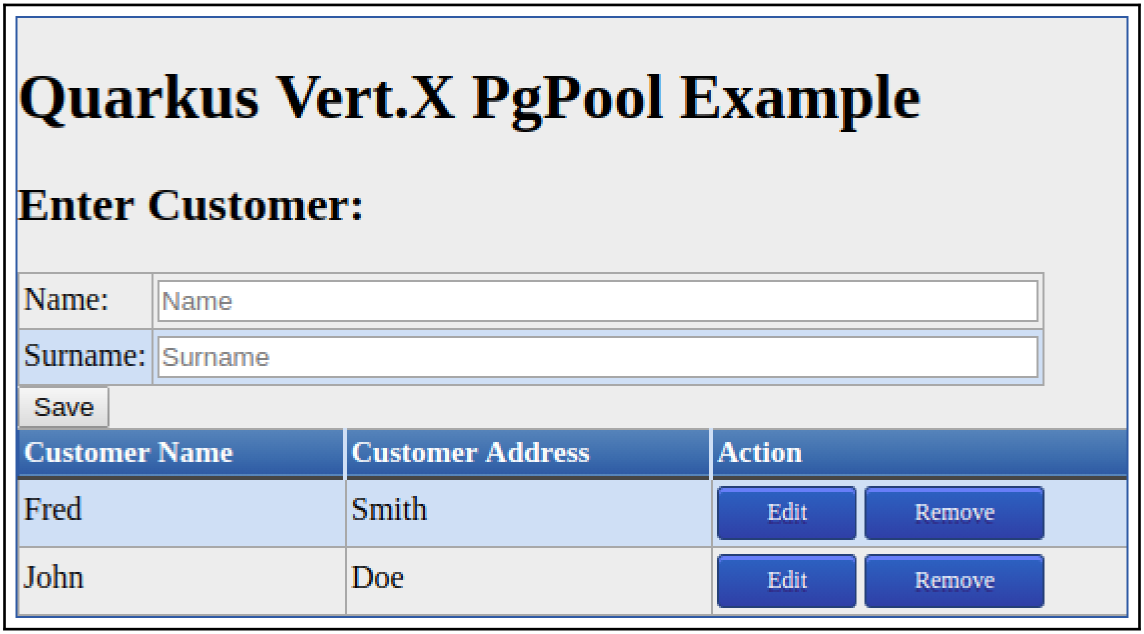
Quarkus Vert.x Axle Example

Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Flinstone	Edit Remove Call
Barney	Rubble	Edit Remove Call

Number of Customers 2 . Last one added: Barney Rubble



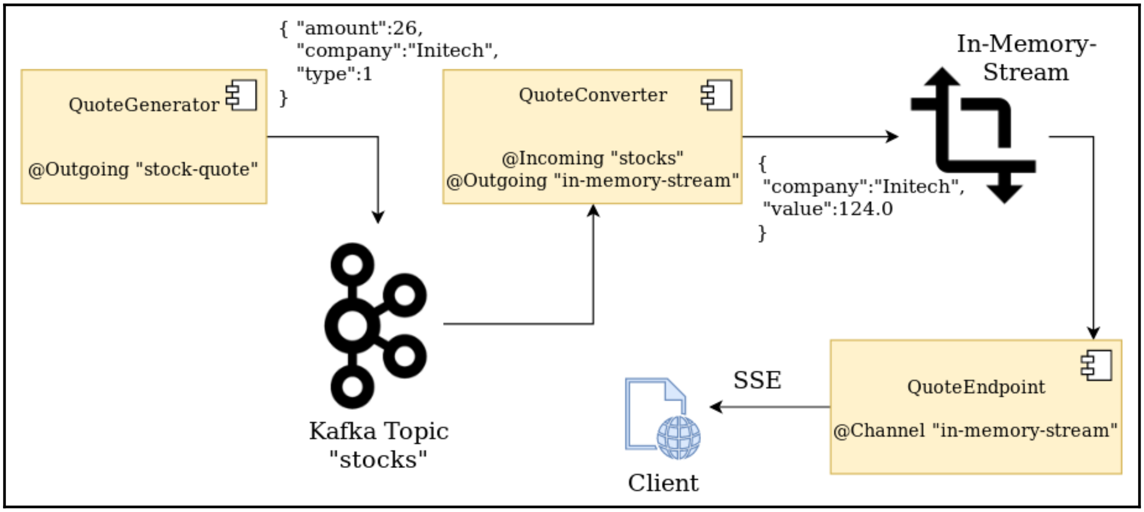
Quarkus Vert.X PgPool Example

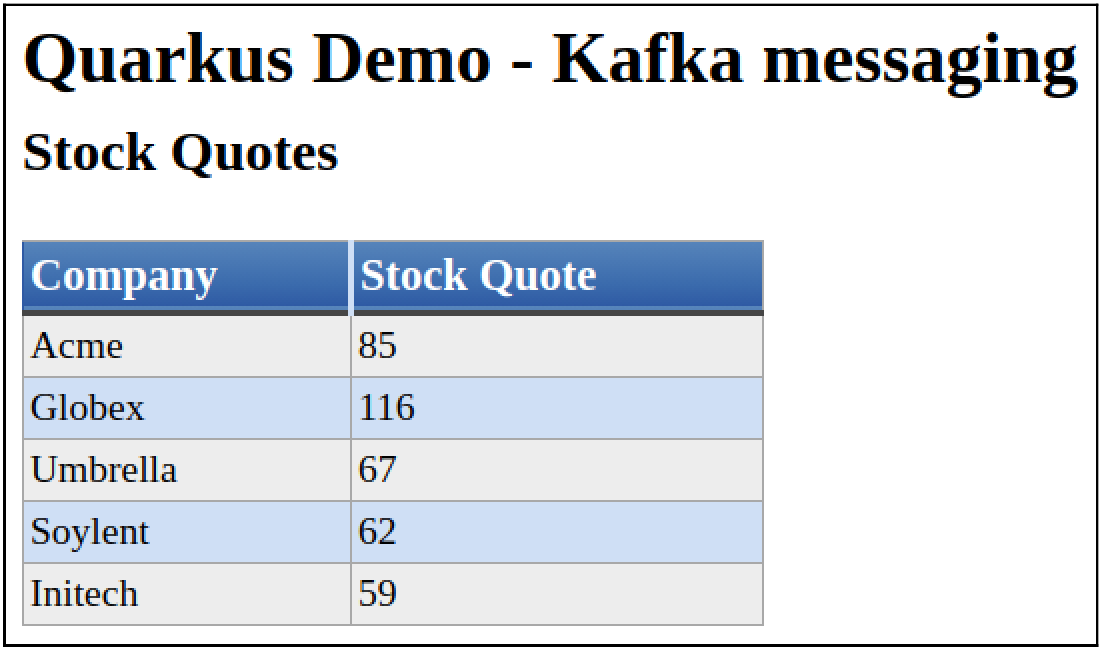
Enter Customer:

Name:	Name
Surname:	Surname

Customer Name	Customer Address	Action
Fred	Smith	Edit Remove
John	Doe	Edit Remove

Chapter 10: Reactive Messaging with Quarkus

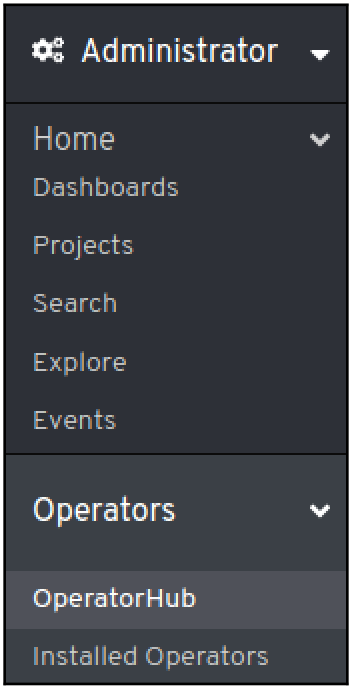


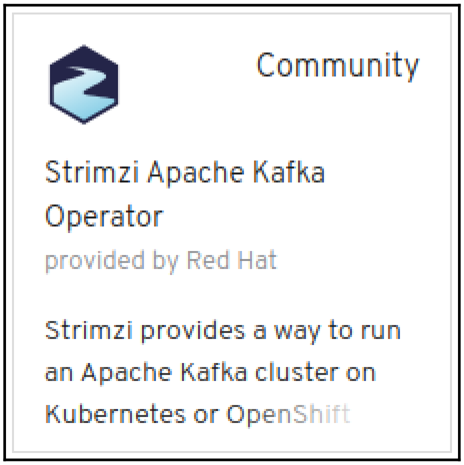


Quarkus Demo - Kafka messaging

Stock Quotes

Company	Stock Quote
Acme	85
Globex	116
Umbrella	67
Soylent	62
Initech	59

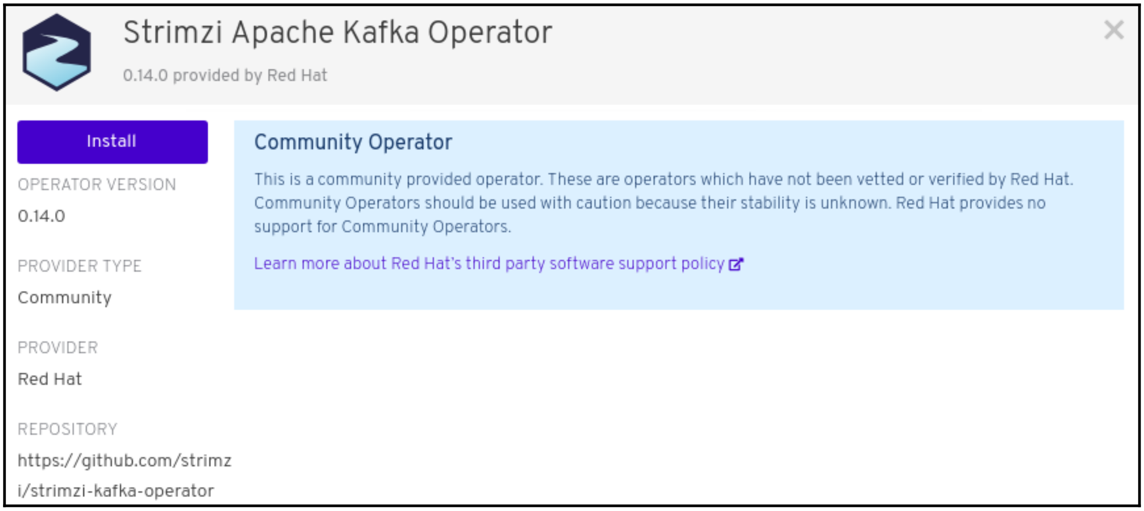




Strimzi Apache Kafka Operator

provided by Red Hat

Strimzi provides a way to run an Apache Kafka cluster on Kubernetes or OpenShift


Strimzi Apache Kafka Operator

0.14.0 provided by Red Hat

OPERATOR VERSION
0.14.0

PROVIDER TYPE
Community

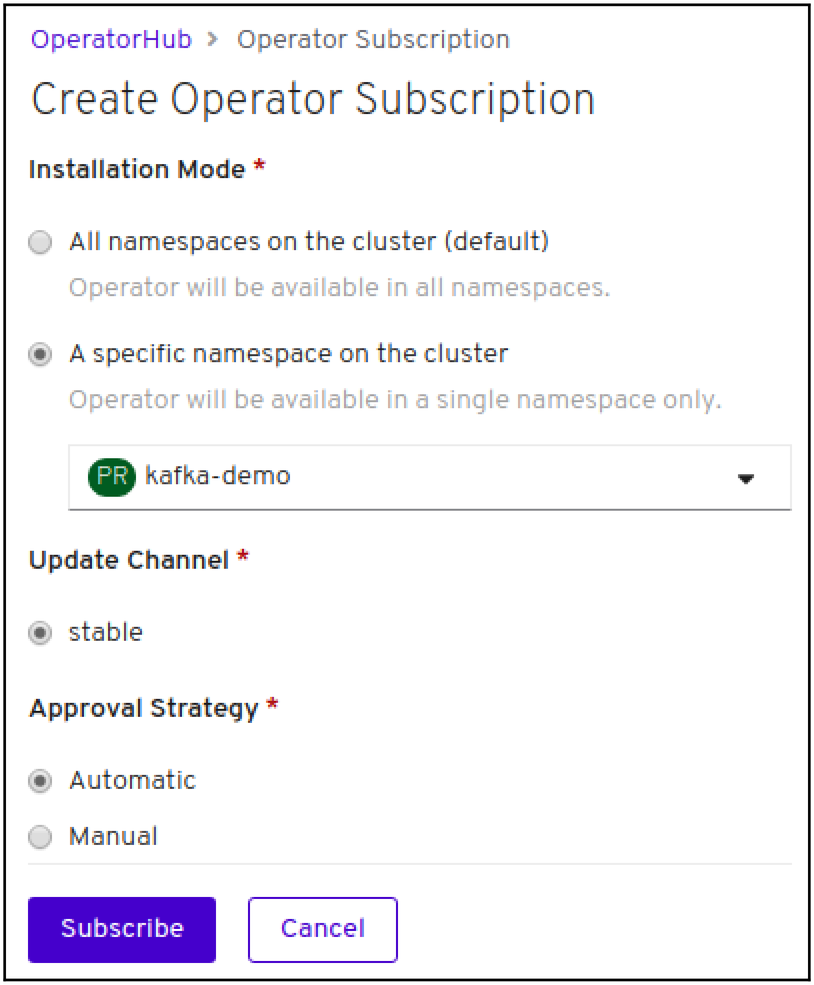
PROVIDER
Red Hat

REPOSITORY
https://github.com/strimzi/strimzi-kafka-operator

Community Operator

This is a community provided operator. These are operators which have not been vetted or verified by Red Hat. Community Operators should be used with caution because their stability is unknown. Red Hat provides no support for Community Operators.

Learn more about Red Hat's third party software support policy



OperatorHub > Operator Subscription

Create Operator Subscription

Installation Mode *

○ All namespaces on the cluster (default)

Operator will be available in all namespaces.

● A specific namespace on the cluster

Operator will be available in a single namespace only.

PR kafka-demo

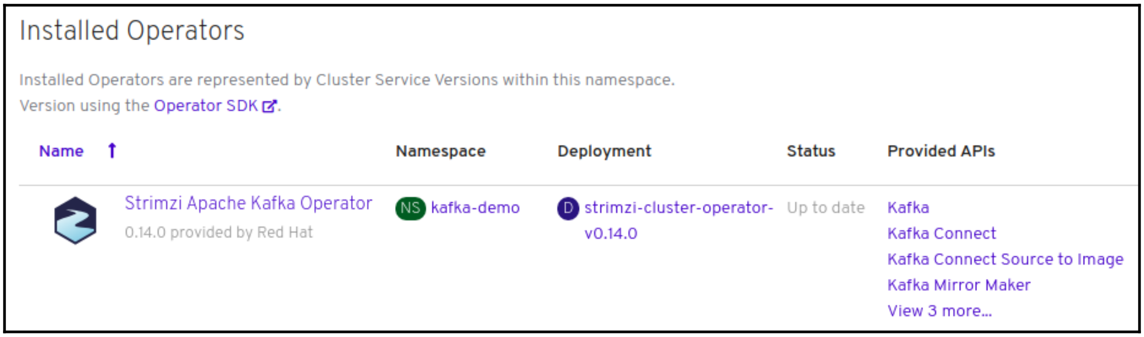
Update Channel *

● stable

Approval Strategy *

● Automatic

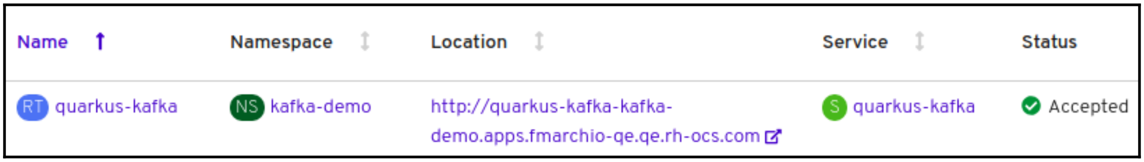
○ Manual



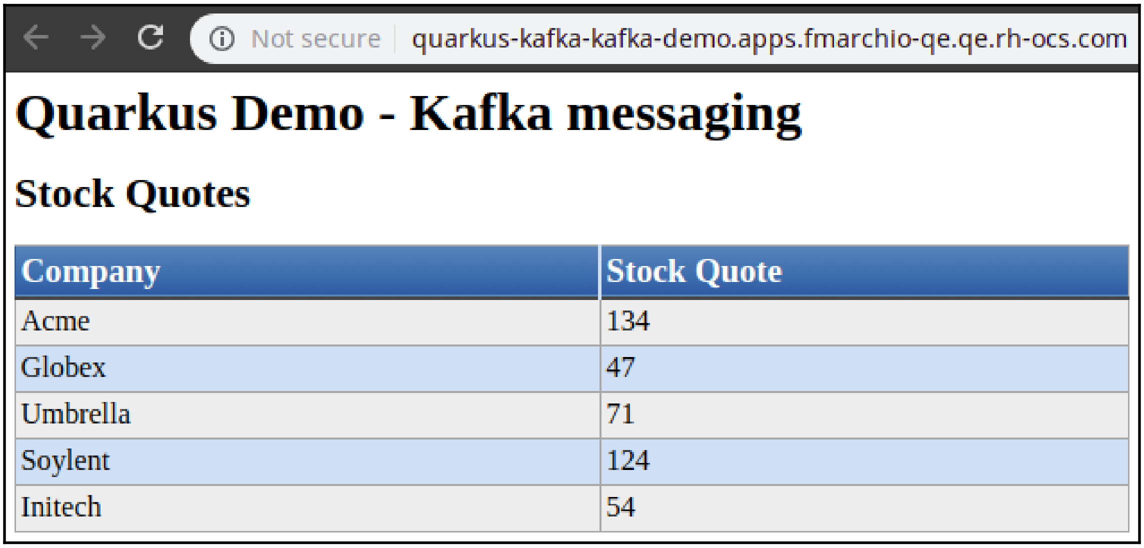
Installed Operators

Installed Operators are represented by Cluster Service Versions within this namespace.
Version using the Operator SDK.

Name ↑	Namespace	Deployment	Status	Provided APIs
Strimzi Apache Kafka Operator 0.14.0 provided by Red Hat	NS kafka-demo	D strimzi-cluster-operator-v0.14.0	Up to date	Kafka Kafka Connect Kafka Connect Source to Image Kafka Mirror Maker View 3 more...



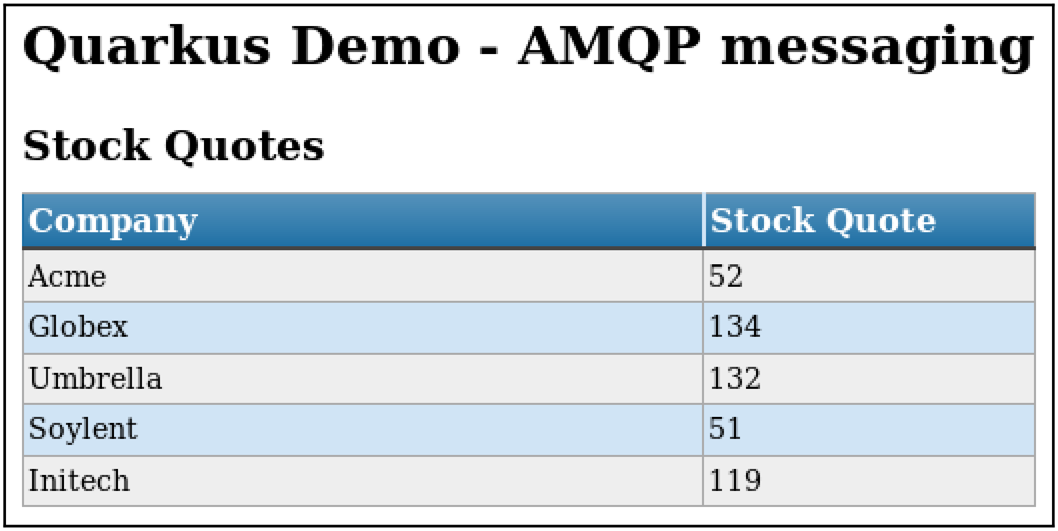
Name ↑	Namespace	Location	Service	Status
RT quarkus-kafka	NS kafka-demo	http://quarkus-kafka-kafka-demo.apps.fmarchio-qe.qe.rh-ocs.com	S quarkus-kafka	✓ Accepted



Quarkus Demo - Kafka messaging

Stock Quotes

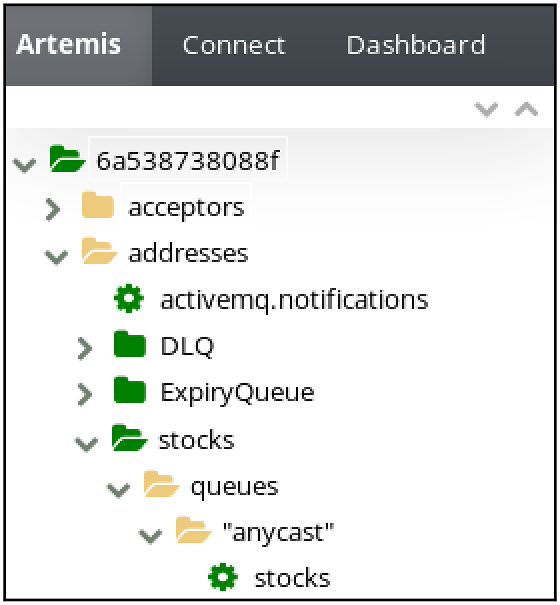
Company	Stock Quote
Acme	134
Globex	47
Umbrella	71
Soylent	124
Initech	54



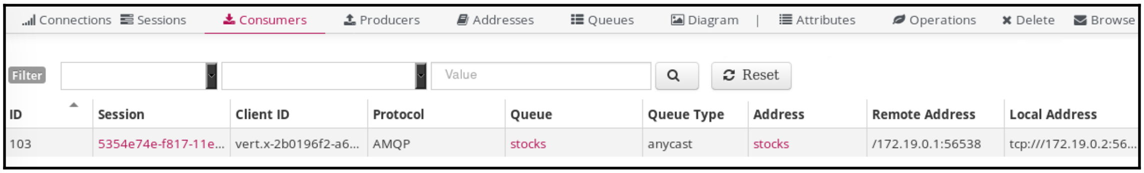
Quarkus Demo - AMQP messaging

Stock Quotes

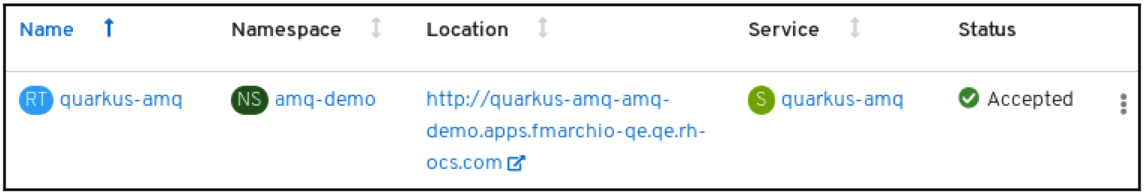
Company	Stock Quote
Acme	52
Globex	134
Umbrella	132
Soylent	51
Initech	119



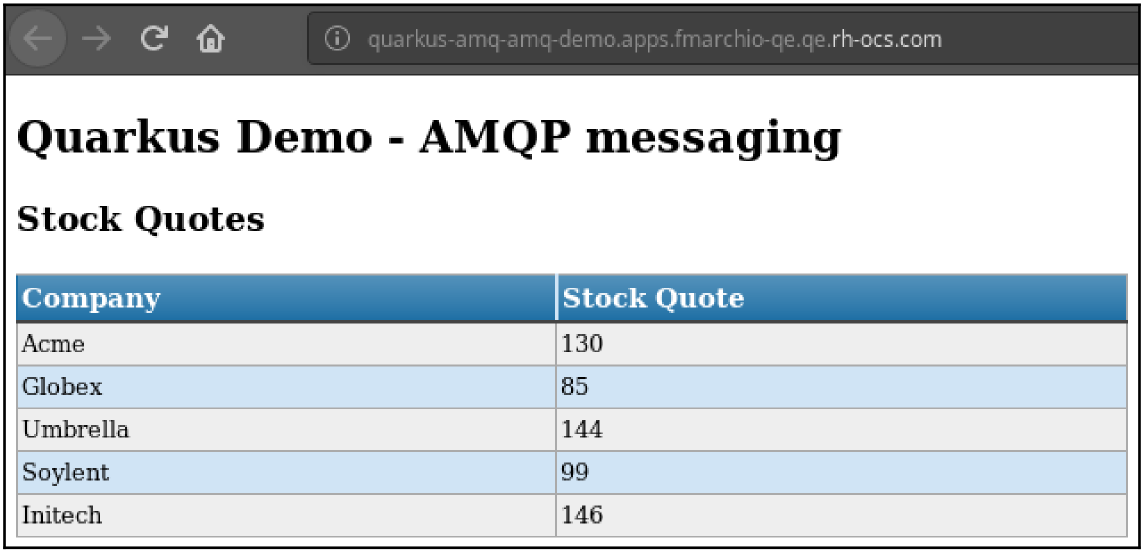
- 6a538738088f
  - acceptors
  - addresses
    - activemq.notifications
    - DLQ
    - ExpiryQueue
    - stocks
      - queues
        - "anycast"
          - stocks



ID	Session	Client ID	Protocol	Queue	Queue Type	Address	Remote Address	Local Address
103	5354e74e-f817-11e...	vert.x-2b0196f2-a6...	AMQP	stocks	anycast	stocks	/172.19.0.1:56538	tcp:///172.19.0.2:56...



Name	Namespace	Location	Service	Status
RT quarkus-amq	NS amq-demo	http://quarkus-amq-amq-demo.apps.fmarchio-qe.qe.rh-ocs.com	S quarkus-amq	Accepted



Quarkus Demo - AMQP messaging

Stock Quotes

Company	Stock Quote
Acme	130
Globex	85
Umbrella	144
Soylent	99
Initech	146