upload-service Project

This project uses Quarkus, the Supersonic Subatomic Java Framework.

If you want to learn more about Quarkus, please visit its website: https://quarkus.io/ .

Running the application in dev mode

You can run your application in dev mode that enables live coding using:

```
./gradlew quarkusDev
```

> ***NOTE:*** Quarkus now ships with a Dev UI, which is available in dev mode only at http://localhost:8080/q/dev/.

Packaging and running the application

The application can be packaged using:

```
./gradlew build
```

It produces the `quarkus-run.jar` file in the `build/quarkus-app/` directory. Be aware that it’s not an *über-jar* as the dependencies are copied into the `build/quarkus-app/lib/` directory.

The application is now runnable using `java -jar build/quarkus-app/quarkus-run.jar`.

If you want to build an *über-jar*, execute the following command:

```
./gradlew build -Dquarkus.package.type=uber-jar
```

The application, packaged as an *über-jar*, is now runnable using `java -jar build/*-runner.jar`.

Creating a native executable

You can create a native executable using:

```
./gradlew build -Dquarkus.package.type=native
```

Or, if you don't have GraalVM installed, you can run the native executable build in a container using:

```

```

```
./gradlew build -Dquarkus.package.type=native -Dquarkus.native.container-build=true
```

You can then execute your native executable with: `./build/upload-service-1.0.0-SNAPSHOT-runner`

If you want to learn more about building native executables, please consult https://quarkus.io/guides/gradle-tooling.

Related Guides

- Hibernate ORM (guide): Define your persistent model with Hibernate ORM and JPA
- Debezium Quarkus Outbox (guide): Implement the outbox pattern for reliable microservices data exchange with Debezium and change data capture
- SmallRye Reactive Messaging - Kafka Connector (guide): Connect to Kafka with Reactive Messaging
- RESTEasy Classic Multipart (guide): Multipart support for RESTEasy Classic
- Quarkus Extension for Spring Data JPA API (guide): Use Spring Data JPA annotations to create your data access layer
- Elasticsearch REST client (guide): Connect to an Elasticsearch cluster using the REST low level client
- JDBC Driver - PostgreSQL (guide): Connect to the PostgreSQL database via JDBC

Provided Code

Hibernate ORM

Create your first JPA entity

Related guide section...

Reactive Messaging codestart

Use SmallRye Reactive Messaging

Related Apache Kafka guide section...

RESTEasy JAX-RS

Easily start your RESTful Web Services

Related guide section...