**Usage and Demo**

**Step 1:** Create the Docker image according to [Dockerfile](https://github.com/miguno/java-docker-build-tutorial/blob/main/Dockerfile). This step uses Maven to build, test, and package the [Java application](https://github.com/miguno/java-docker-build-tutorial/blob/main/src/main/java/com/miguno/App.java) according to [pom.xml](https://github.com/miguno/java-docker-build-tutorial/blob/main/pom.xml). The resulting image is 176MB in size, of which 170MB are the underlying eclipse-temurin image.

# \*\*\*Creating an image may take a few minutes!\*\*\*

$ docker build --platform linux/x86\_64/v8 -t miguno/java-docker-build-tutorial:latest .

# You can also build with the new BuildKit.

# https://docs.docker.com/build/

$ docker buildx build --platform linux/x86\_64/v8 -t miguno/java-docker-build-tutorial:latest .

Optionally, you can check the size of the generated Docker image:

$ docker images miguno/java-docker-build-tutorial

REPOSITORY TAG IMAGE ID CREATED SIZE

miguno/java-docker-build-tutorial latest 1403a608d055 4 minutes ago 176MB

**Step 2:** Start a container for the Docker image.

$ docker run -p 8123:8123 miguno/java-docker-build-tutorial:latest

**Step 3:** Open another terminal and access the example API endpoint of the running container.

$ curl http://localhost:8123/status

{"status": "idle"}

**Usage with just**

If you have [just](https://github.com/casey/just) installed, you can run the commands above more conveniently as per this project's [justfile](https://github.com/miguno/java-docker-build-tutorial/blob/main/justfile):

$ just

Available recipes:

default # print available targets

docker-image-create # create a docker image (requires Docker)

docker-image-run # run the docker image (requires Docker)

docker-image-size # size of the docker image (requires Docker)

evaluate # evaluate and print all just variables

send-request-to-app # send request to the app's HTTP endpoint (requires running container)

system-info # print system information such as OS and architecture

Example:

$ just docker-image-create

**Notes**

You can also build, test, package, and run the Java application locally (without Docker) if you have JDK 17+ and Maven installed.

# Build, test, package the application locally

$ mvn clean package

# Run the example application locally

$ java -jar target/app.jar