

GABRIELE ROMANATO

Menu

Usare le Docker Engine API con Java

Le Docker Engine API consentono di interagire con Docker a livello programmatico, permettendo di gestire container, immagini e volumi. In questo articolo, vedremo come utilizzare Java con l'SDK per interfacciarsi con Docker Engine.

Prerequisiti

- Docker installato e in esecuzione
- Java Development Kit (JDK) 8 o superiore
- Maven o Gradle per la gestione delle dipendenze

Configurazione del progetto

Per usare le Docker Engine API in Java, possiamo utilizzare la libreria `docker-java`. Aggiungiamo la dipendenza in `pom.xml` se usiamo Maven:

```
<dependency>
  <groupId>com.github.docker-java</groupId>
  <artifactId>docker-java</artifactId>
  <version>3.2.13</version>
</dependency>
```

Se usiamo Gradle, aggiungiamo:

```
dependencies {
    implementation 'com.github.docker-java:docker-
java:3.2.13'
}
```

Connessione a Docker

Ora possiamo scrivere un semplice programma Java per connettersi a Docker e ottenere informazioni sul demone in esecuzione.

```
import com.github.dockerjava.api.DockerClient;
import com.github.dockerjava.core.DockerClientConfig;
import com.github.dockerjava.core.DockerClientImpl;
import
com.github.dockerjava.core.command.PullImageResultCallback;
import com.github.dockerjava.okhttp.OkDockerHttpClient;

public class DockerJavaExample {
    public static void main(String[] args) {
        DockerClientConfig config =
DockerClientConfig.createDefaultConfigBuilder().build();

        try (DockerClient dockerClient =
DockerClientImpl.getInstance(config,
            new
OkDockerHttpClient.Builder().dockerHost(config.getDockeHost(
)).build())) {

            System.out.println("Versione Docker: " +
dockerClient.versionCmd().exec().getVersion());
        }
    }
}
```

Gestione dei container

Creiamo ora un container basato su un'immagine esistente:

```
import
com.github.dockerjava.api.command.CreateContainerResponse;

CreateContainerResponse container =
dockerClient.createContainerCmd("nginx")
```

```
        .withName("my-nginx-container")
        .exec();

dockerClient.startContainerCmd(container.getId()).exec();
System.out.println("Container avviato con ID: " +
    container.getId());
```

Elenco dei container

Possiamo recuperare la lista dei container attivi con il seguente codice:

```
dockerClient.listContainersCmd().exec().forEach(c ->
    System.out.println("Container ID: " + c.getId() + ",
    Nome: " + c.getNames()[0])
);
```

Conclusione

Abbiamo visto come usare le Docker Engine API con Java e l'SDK `docker-java`. Questa libreria consente di interagire facilmente con Docker per creare, gestire ed eseguire container in modo programmatico.