Red Hat Enterprise Virtualization 3.6

Java SDK ガイド

Red Hat Enterprise Virtualization Java SDK の使用

Last Updated: 2017-09-22

Red Hat Enterprise Virtualization 3.6 Java SDK ガイド

Red Hat Enterprise Virtualization Java SDK の使用

Red Hat Enterprise Virtualization Documentation Team
Red Hat Customer Content Services
rhev-docs@redhat.com

法律上の通知

概要

本ガイドは、Red Hat Enterprise Virtualization の Java ソフトウェア開発キットについて説明します。

目次

第1章 概要

Java ソフトウェア開発キットは、Java ベースプロジェクトで Red Hat Enterprise Virtualization Manager との対話を可能にするクラスのコレクションです。これらのクラスをダウンロードしてプロジェクトに追加することにより、管理タスクを高度に自動化するさまざまな機能を利用することができます。

Java ソフトウェア開発キットは、rhevm-sdk-java パッケージを使用します。このパッケージは、Red Hat Subscription Manager の **Red Hat Enterprise Virtualization** エンタイトルメントプールをサブスクライブしているシステムに提供されています。

1.1. 前提条件

Java ソフトウェア開発キットをインストールする場合の前提条件は以下のとおりです。

- Red Hat Enterprise Linux 6.6 または 7 がインストールされたシステム。Server バージョンと Workstation バージョンがサポートされます。
- Red Hat Hat Enterprise Virtualization エンタイトルメントのサブスクリプション。

重要

rhevm-sdk-java パッケージは、ソフトウェア開発キットを使用するスクリプトを実行する各システムにインストールする必要があります。

重要

ソフトウェア開発キットは、Red Hat Enterprise Virtualization REST API のインターフェースです。このように、Red Hat Enterprise Virtualization 環境のバージョンに適したソフトウェア開発キットのバージョンを使用する必要があります。たとえば、Red Hat Enterprise Virtualization 3.5 を使用している場合には、3.5 向けに設計されたソフトウェア開発キットのバージョンを使用する必要があります。

第2章 ビルド環境の構築

本章では、Java ソフトウェア開発キットを使用してアプリケーションを開発するためのビルド環境を構築する方法を説明します。

2.1. JAVA ソフトウェア開発キットのインストール

Java ソフトウェア開発キットと付属ドキュメントをインストールします。

手順2.1 Java ソフトウェア開発キットのインストール

1. Red Hat Subscription Manager でお使いのシステムが **Red Hat Enterprise Virtualization** のエンタイトルメントにサブスクライブされていることを確認してください。

```
# subscription-manager list --available | grep -A8 "Red Hat
Enterprise Virtualization"
# subscription-manager attach --pool=pool_id
# subscription-manager repos --enable=rhel-6-server-rpms
# subscription-manager repos --enable=rhel-6-server-rhevm-3.6-rpms
# subscription-manager repos --enable=jb-eap-6-for-rhel-6-server-
rpms
```

2. 必要なパッケージをインストールします。

```
# yum install rhevm-sdk-java
# yum install rhevm-sdk-java-javadoc
```

Java ソフトウェア開発キットと付属ドキュメントが **/usr/share/java/rhevm-sdk-java** のディレクトリーにダウンロードされ、Java プロジェクトに追加できる状態となりました。

2.2. 依存関係

Java アプリケーションで Java ソフトウェア開発キットを使用するには、これらのアプリケーションのクラスパスに以下の JAR ファイルを追加する必要があります。

- **commons-beanutils.jar**
- **commons-codec.jar**
- **httpclient.jar**
- **httpcore.jar**
- **jakarta-commons-logging.jar**
- **log4j.jar**

これらの JAR ファイルを提供するパッケージが rhevm-sdk-java パッケージの依存関係としてインストールされます。デフォルトでは、Red Hat Enterprise Linux 6 および Red Hat Enterprise Linux 7 システムでは、**/usr/share/java** から入手できます。

2.3. SSL の設定

Red Hat Enterprise Virtualization Manager Java SDK は、Java Secure Socket Extension (JSSE) を使用して、Secure Sockets Layer (SSL) および IETF Transport Layer Security (TLS) プロトコルを介した HTTP の完全なサポートを提供します。JSSE は Java 2 にバージョン 1.4 として実装されており、Java-SDK では追加設定なしで使用することができます。Java 2 の旧バージョンには、JSSE は手動でインストールおよび設定する必要があります。

手順2.2 SSL の設定

1. Red Hat Enterprise Virtualization Manager 用の証明書をダウンロードします。

```
https://[your manager's address]:[port]/ca.crt
```

2. キーストアを生成します。

```
keytool -import -alias "server.crt truststore" -file ca.crt -
keystore server.truststore
```

3. 以下のいずれかの方法で Java ソフトウェア開発キットにキーストアを認識させます。

   - キーストアのルックアップパスを作成します。

```
$ mkdir ~/.ovirtsdk/
$ cp server.truststore ~/.ovirtsdk/ovirtsdk-keystore.truststore
```

     ovirtsdk-keystore.truststore が **~/.ovirtsdk** ディレクトリーにコピーされると、通信先のホストとのハンドシェイクでホストのアイデンティティー検証に使用されるようになります。

   - **Api** クラスのインスタンスを宣言する際にカスタムのトラストストアを指定するには、以下の署名を使用します。

```
Api api = new Api(url, user, password,
"/path/server.truststore");
```

注記

ホストのアイデンティティー検証は無効にすることも可能です。ホストのアイデンティティーを検証しないことによってもたらされるセキュリティーへの影響を完全に認識した上での意識的な決断でない限りは、この方法は、セキュリティー上の理由から、実稼働システムで使用すべきではありません。ホストのアイデンティティー検証を無効にするには、**Api** クラスのインスタンスを宣言する際に以下の署名を使用します。

```
Api api = new Api(url, user, password, true);
```

第3章 ソフトウェア開発キットの使用

本章では、Java ソフトウェア開発キットの使用方法に関する例を記載します。

3.1. RED HAT ENTERPRISE VIRTUALIZATION MANAGER への接続

Api クラスは、Red Hat Enterprise Virtualization Manager 内のオブジェクトに接続して操作するためのゲートウェイとしての役割を果たします。プロジェクトで Manager と対話する際には毎回、このクラスのインスタンスを宣言する必要があります。

Api クラスのインスタンスを宣言するには、まず最初に**Api** クラスをプロジェクトにインポートする必要があります。

```
import org.ovirt.engine.sdk.Api;
```

次に、以下の署名を使用して、このクラスのインスタンスを宣言します。

```
Api api = new Api("[your manager's address]", "[user name]@[domain]",
"[password]", "[path to certificate]");
```

上記の署名で、**your manager's address** の箇所には、Red Hat Enterprise Virtualization Manager に接続するためのアドレスおよびポート名を指定します。**user name**、**domain**、**password** の箇所はログインの認証情報に置き換え、**path to certificate** の箇所には、お使いのシステムで SSL 通信に使用する Manager の証明書へのローカルパスを指定します。証明書の使用方法についての詳しい説明は 「SSL の設定」 を参照してください。

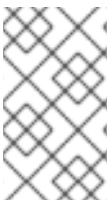
以下の例は、このプロセスをさらに詳しく示しています。

```
import org.ovirt.engine.sdk.Api;
import java.io.IOException;

public class Main
{
 private static final String URL = "https://localhost:443/api";

 @SuppressWarnings("unused")
 public static void main(String[] args) throws ClientProtocolException,
  ServerException, UnsecuredConnectionAttemptError, IOException
 {
  // #1 Authenticate using the user name and password
  Api api = new Api(URL, "admin@interal", "123456",
"/home/user/.ovirtsdk/");

  ...
 }
}
```



注記

Api クラスは、**Autocloseable** インターフェースを実装しないため、Red Hat Enterprise Virtualization Manager が正常にシャットダウンされるように、**finally** ブロック内の **Api** クラスのインスタンスをシャットダウンすることを推奨します。

3.2. エンティティーの一覧表示

以下の例では、Red Hat Enterprise Virtualization Manager のエンティティーを一覧表示する方法を説明します。この例では、一覧表示するエンティティーは仮想マシンで、**Api** クラスの **getVMs()** メソッドを使用して一覧表示します。

手順3.1 エンティティーの一覧表示

- 一覧表示するエンティティーのタイプの **List** を宣言し、対応するメソッドを使用してエンティティーを一覧表示します。

```
List<VM> vms = api.getVMs().list();
```

3.3. リソースの属性の変更

以下の例では、リソースの属性を変更する方法について説明します。この例では、変更する属性は、「test」という名前の仮想マシンの説明で、これを「java_sdk」に変更します。

手順3.2 リソースの属性の変更

1. 属性を変更するリソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. 新しい属性値を設定します。

```
vm.setDescription("java_sdk");
```

3. 変更を適用する仮想マシンを更新します。

```
VM newVM = vm.update();
```

3.4. リソースの取得

Java ソフトウェア開発キットでは、リソースは**name** および **UUID** の 2 つの属性で参照することができます。そのオブジェクトが存在している場合には、いずれも指定した属性のオブジェクトが返されます。

name 属性値を使用してリソースを取得するには、以下の方法を使用します。

```
VM vm = api.getVMs().get("test");
```

UUID 属性値を使用してリソースを取得するには、以下の方法を使用します。

```
VM vm = api.getVMs().get(UUID.fromString("5a89a1d2-32be-33f7-a0d1-
f8b5bc974ff6"));
```

3.5. リソースの追加

以下の例では、Red Hat Enterprise Virtualization Manager にリソースを追加する 2 つの方法について説明します。これらの例では、追加するリソースは仮想マシンです。

例 1

以下の例では、**VM** クラスを宣言して、追加する新規仮想マシンを表現し、次にその仮想マシンの属性を希望する値に設定した後、最後に新規仮想マシンを Manager に追加します。

```
org.ovirt.engine.sdk.entities.VM vmParams = new
org.ovirt.engine.sdk.entities.VM();

vmParams.setName("myVm");
vmParams.setCluster(api.getClusters().get("myCluster"));
vmParams.setTemplate(api.getTemplates().get("myTemplate"));
...
```

```
VM vm = api.getVMs().add(vmParams);
```

例 2

以下の例では、**VM** クラスのインスタンスは、例 1 と同じ方法で宣言しますが、**get** メソッドを使用して Manager 内の既存のオブジェクトを参照する方法の代わりに、各属性のインスタンスを宣言することによってその属性を参照します。最後に新規仮想マシンを Manager に追加します。

```
org.ovirt.engine.sdk.entities.VM vmParams = new
org.ovirt.engine.sdk.entities.VM();

vmParams.setName("myVm");
org.ovirt.engine.sdk.entities.Cluster clusterParam = new Cluster();
clusterParam.setName("myCluster");
vmParams.setCluster(clusterParam);
org.ovirt.engine.sdk.entities.Template templateParam = new Template();
templateParam.setName("myTemplate");
vmParams.setTemplate(templateParam);
...
```

```
VM vm = api.getVMs().add(vmParams);
```

3.6. リソースに対するアクションの実行

以下の例では、リソースに対してアクションを実行する方法について説明します。この例では、「**test**」という名前の仮想マシンを起動します。

手順3.3 リソースに対するアクションの実行

1. リソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. リソースに送るアクションパラメーターを宣言します。

```
Action actionParam = new Action();
org.ovirt.engine.sdk.entities.VM vmParam = new
org.ovirt.engine.sdk.entities.VM();
```

```
actionParam.setVm(vmParam);
```

3. アクションを実行します。

```
Action res = vm.start(actionParam);
```

代わりに、内部メソッドとしてアクションを実行することも可能です。

```
Action res = vm.start(new Action()
{
    {
        setVm(new org.ovirt.engine.sdk.entities.VM());
    }
});
```

3.7. サブリソースの一覧表示

以下の例では、リソースのサブリソースを一覧表示する方法について説明します。この例では、「**test**」という名前の仮想マシンのサブリソースを一覧表示します。

手順3.4 サブリソースの一覧表示

1. 一覧表示するサブリソースが含まれているリソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. サブリソースを一覧表示します。

```
List<VMDisk> disks = vm.getDisks().list();
```

3.8. サブリソースの取得

以下の例では、リソースのサブリソースを参照する方法について説明します。この例では、「**test**」という名前の仮想マシンに属する「**my disk**」という名前のディスクを参照します。

手順3.5 リソースのサブリソースの取得

1. 参照するサブリソースが含まれているリソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. 参照するサブリソースのインスタンスを宣言します。

```
VMDisk disk = vm.getDisks().get("my disk");
```

3.9. リソースへのサブリソース追加

以下の例では、リソースにサブリソースを追加する方法を説明します。この例では、サイズが「**1073741824L**」で、「**virtio**」インターフェースを使用する「**cow**」形式の新規ディスクを「**test**」という名前の仮想マシンに追加します。

手順3.6 リソースへのサブソースの追加

1. サブリソースを追加するリソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. そのリソースの属性を定義するパラメーターを作成します。

```
Disk diskParam = new Disk();
diskParam.setProvisionedSize(1073741824L);
diskParam.setInterface("virtio");
diskParam.setFormat("cow");
```

3. サブリソースを追加します。

```
Disk disk = vm.getDisks().add(diskParam);
```

3.10. サブリソースの変更

以下の例では、サブリソースの変更方法を説明します。この例では、「**test**」という名前の仮想マシンに属する「**test_Disk1**」という名前のディスクを「**test_Disk1_updated**」という名前に変更します。

手順3.7 サブリソースの更新

1. 変更するサブリソースが含まれているリソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. 変更するサブリソースのインスタンスを宣言します。

```
VMDisk disk = vm.getDisks().get("test_Disk1");
```

3. 新しい属性値を設定します。

```
disk.setAlias("test_Disk1_updated");
```

4. サブリソースを更新します。

```
VMDisk updateDisk = disk.update();
```

3.11. サブリソースに対するアクションの実行

以下の例では、サブリソースに対するアクションの実行方法を説明します。この例では、「**test**」という名前の仮想マシンに属する「**test_Disk1**」という名前のディスクがアクティブ化されます。

手順3.8 サブリソースに対するアクションの実行

1. アクションを実行する対象となるサブリソースが含まれているリソースのインスタンスを宣言します。

```
VM vm = api.getVMs().get("test");
```

2. サブリソースのインスタンスを宣言します。

```
VMDisk disk = vm.getDisks().get("test_Disk1");
```

3. サブリソースに送信するアクションパラメーターを宣言します。

```
Action actionParam = new Action();
```

4. アクションを実行します。

```
Action result = disk.activate(actionParam);
```

付録A 改訂履歴

改訂 3.6-2.1 **Sat Jan 23 2016** **Red Hat Localization Services**

翻訳ファイルを XML ソースバージョン 3.6-2 と同期

改訂 3.6-2 **Wed 18 Nov 2015** **Red Hat Enterprise Virtualization Documentation Team**

Red Hat Enterprise Virtualization 3.6 ベータリリースの最終版作成

改訂 3.6-1 **Mon 10 Aug 2015** **Red Hat Enterprise Virtualization Documentation Team**

Red Hat Enterprise Virtualization 3.6 リリースの初版作成