記事
Mihoko Iijima · 2021年2月5日 5m read

Java クラスのモジュールを IRIS から実行する方法

これは InterSystems FAQ サイトの記事です。

Java ゲートウェイを使用することで、Java クラスのモジュールを InterSystems IRIS から実行できます。

【メモ】Java ゲートウェイは、外部 Java オブジェクトを InterSystems IRIS
内のネイティブ・オブジェクトと同じようにインスタンス化し、Javaオブジェクトを操作するための方法です。

詳細は以下ドキュメントをご参照ください。
Java ゲートウェイについて

Java ゲートウェイを使用する手順は以下の通りです。

1) Java ゲートウェイサーバを開始する

2) プロキシオブジェクト（ダイナミック・プロキシ・オブジェクト）を作成して Java オブジェクトを操作する

---

1) Java ゲートウェイサーバを開始する

Java ゲートウェイサーバを開始する方法は 3 種類あります。以下のいずれかの方法で、Java
ゲートウェイサーバを開始します。

A) 管理ポータルの「オブジェクトゲートウェイ」を利用する

B) プロダクションにJavaゲートウェイサービスを追加する

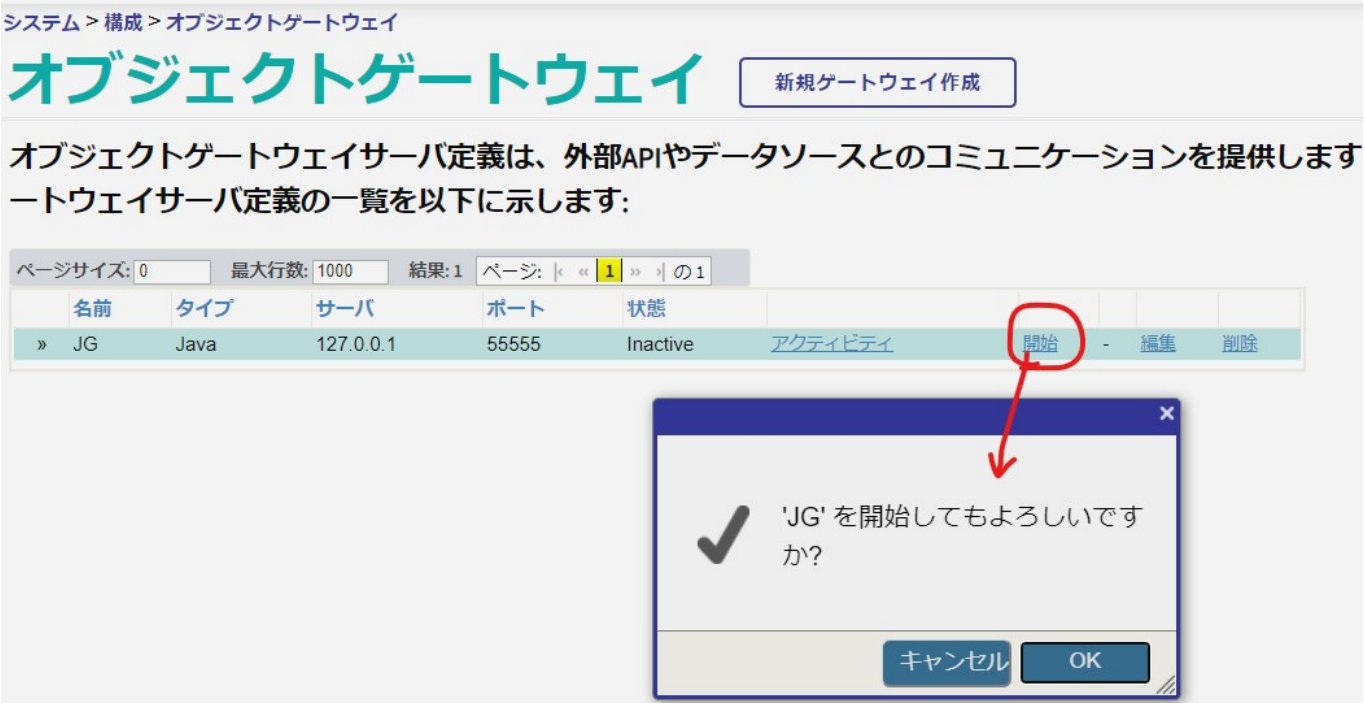
C) java コマンドを使用する

A)〜C) の方法は以下の通りです。

A) 管理ポータルの「オブジェクトゲートウェイ」を利用する

管理ポータル > [システム管理] > [構成] > [接続性] > [オブジェクトゲートウェイ] の画面で Java
ゲートウェイサーバを作成します。

設定保存後、「開始」をクリックして Java ゲートウェイサーバを開始します。



開始後の画面は以下の通りです。

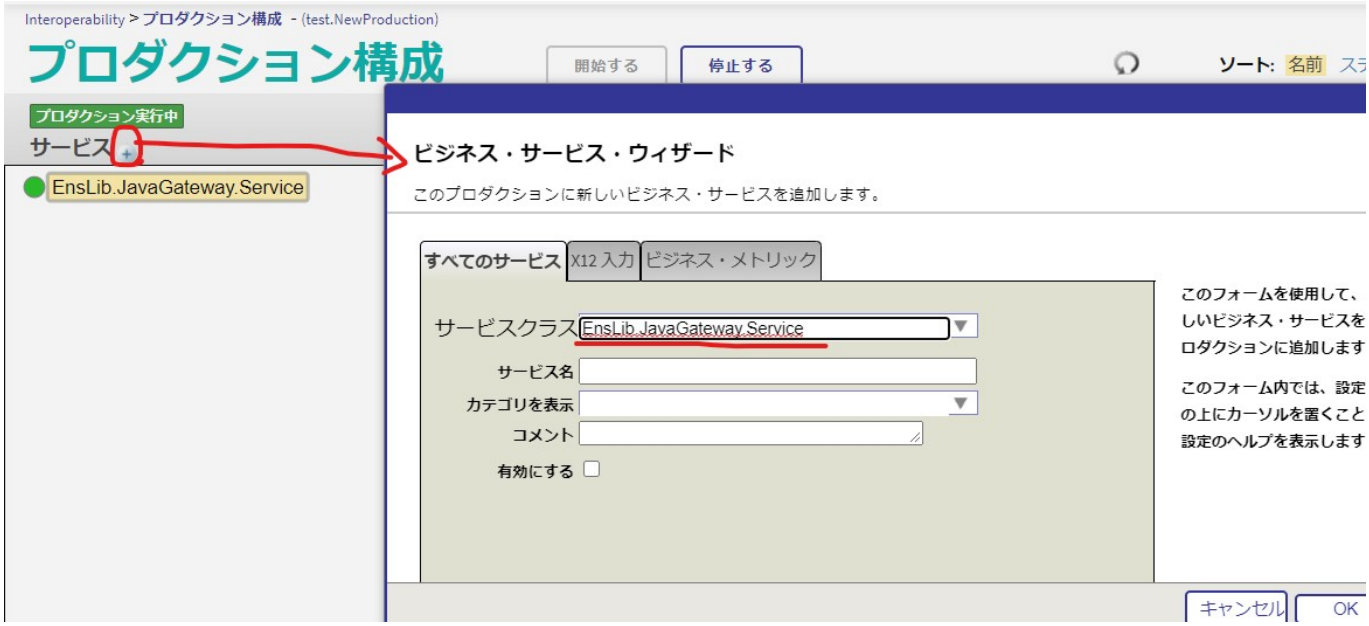
これで、Java ゲートウェイサーバの開始は終了です。停止する場合は、オブジェクトゲートウェイの設定画面で「停止」のリンクをクリックします。

B) プロダクションにJavaゲートウェイサービスを追加する

任意のネームスペースにプロダクションを作成し、EnsLib.JavaGateway.Service
クラスを「サービス」に追加します。

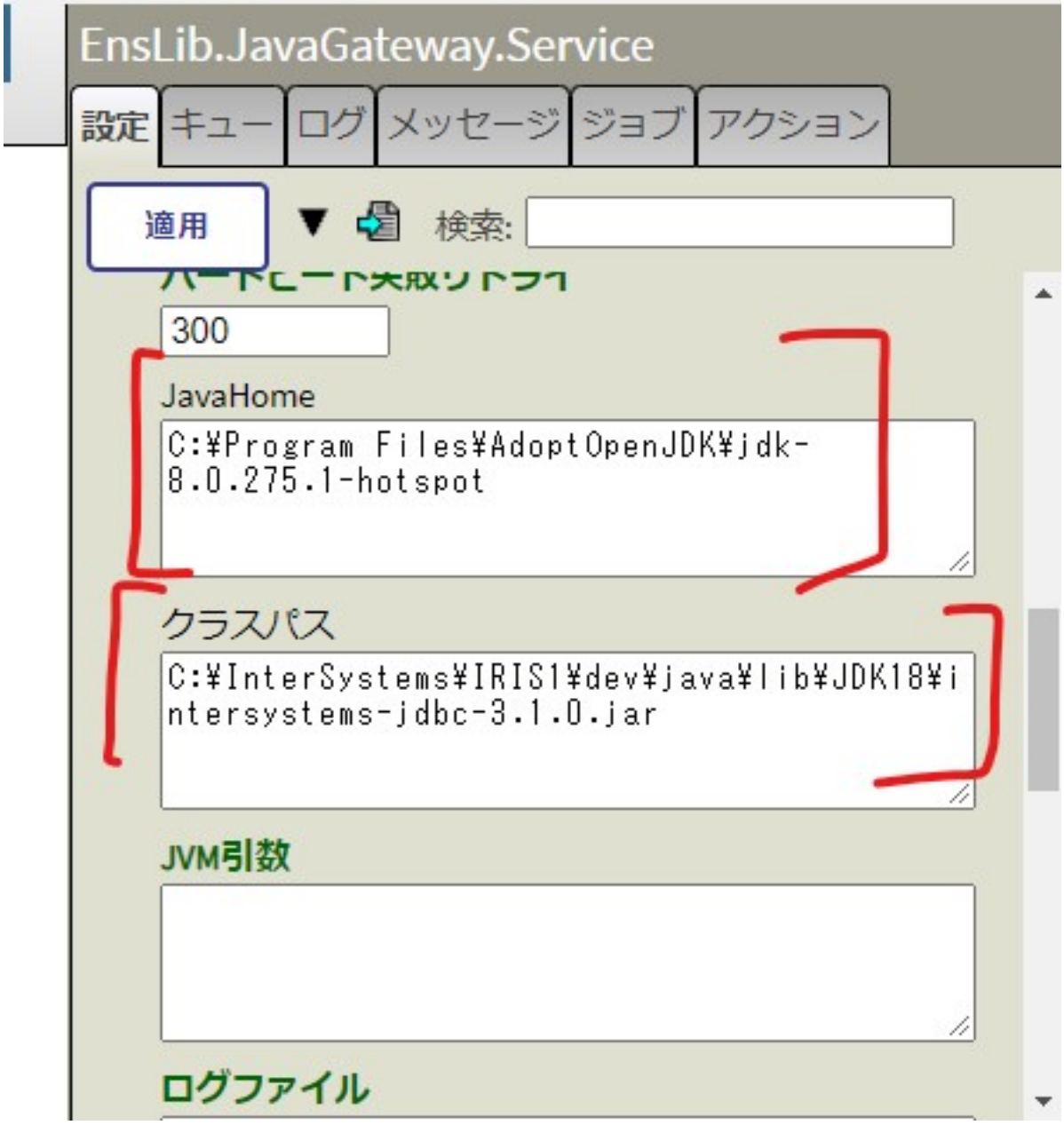
管理ポータル > [Interoperability] > [一覧] > [プロダクション] > [新規]ボタンクリック >
パッケージ名、プロダクション名（＝クラス名）を指定し OK ボタンクリック

サービスの右側にある ＋ のボタンをクリックし [EnsLib.JavaGateway.Service] を追加します。



設定を行うため、EnsLib.JavaGateway.Service をクリックします（クラス名の字の部分をクリックします）。

画面右側の「設定」タブで、アドレス／ポート を設定します。



この後、プロダクションを「開始する」ボタンで開始することで、Java ゲートウェイサーバも開始します。
停止する場合は、プロダクションを「停止する」ボタンで停止してください。

詳細は、ドキュメントもご参照ください。
[Java ゲートウェイのビジネスサービスの追加]

C) java コマンドを使用する

以下のコマンドで開始できます。

java -classpath [IRISインストールディレクトリ]\dev\java\lib\JDK18\intersystems-jdbc-3.1.0.jar;[IRISインストールディレクトリ]\dev\java\lib\JDK18\intersystems-gateway-3.1.0.jar com.intersystems.gateway.JavaGateway ポート番号 ログファイルフルパス プロキシクラスの識別子 ホスト名

実行例） IRISのインストールディレクトリが c:\InterSystems\IRIS1 、ホスト名は localhost、ポート番号は 12345 で開始する例

```
java -classpath C:\InterSystems\IRIS1\dev\java\lib\JDK18\intersystems-jdbc-3.1.0.jar;
C:\InterSystems\IRIS1\dev\java\lib\JDK18\intersystems-gateway-3.1.0.jar com.intersyst
ems.gateway.JavaGateway 12345 c:\kit\javag.log abc localhost
```

詳細は、ドキュメントもご参照ください。
[コマンド・プロンプトの使用法]

2) プロキシオブジェクト（ダイナミック・プロキシ・オブジェクト）を作成して Java オブジェクトを操作する。

上記３手法のいずれか1つで Java ゲートウェイサーバを開始してから実行します。

なお、**Java 側のクラスや Jar ファイルが更新された場合、Java ゲートウェイサーバの再起動が必要です。**ご注意ください。

テストに使用している Java クラス（test.java）は以下の通りです。

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
public class test {
  public int num = 1234;
  public String data;

  // test1 ???????
  public static String test1() { return "abc"; }
  // test2 ? test3()?????????????
  public static int test2(List obj) { return obj.size(); }
  // test3 ? ??????????????????test??????????List???????
  public static List test3(String... input) {
    List testlist=new ArrayList();
    for (String str : input) {
      test obj=new test();
      obj.data=str;
      testlist.add(obj);
    }
    return testlist;
  }
}
```

例では、 c:\temp\test.java に配置した状態でコンパイルを実行し、c:\temp\test.class を作成しています。

IRIS から test.class のインスタンスを生成するコードは以下の通りです。

```
// ????JAR??????????????????????????
set class=##class(%ListOfDataTypes).%New()
// ????Java????????????
do class.Insert("c:\temp")
```

```
// Java??????????????????????????
set GW=##class(%Net.Remote.Gateway).%New()
// ?1??????????2??????????????3????????????????5?????????????????????
set st=GW.%Connect("localhost",55555,"USER",,class)
// 1?????????
write st
// 1??????????????????????
write $system.Status.GetErrorText(st)
// ?????????????????????? test.class?????????
set testobj=##class(%Net.Remote.Object).%New(GW,"test")
// test1()???????
write testobj.test1()
// ???????? num ???
write testobj.num
// Java???????? test3()????????? test???????List???????
set newobj=testobj.test3("?","?","abc")
// test3()???????????test2()?????????????????????????3??????
write testobj.test2(newobj)
// ????
set st=GW.%Disconnect()
```

#Java #JDBC #ObjectScript #相互運用性 #InterSystems IRIS #InterSystems IRIS for Health

---

ソースURL:
https://jp.community.intersystems.com/post/java-%E3%82%AF%E3%83%A9%E3%82%B9%E3%81%AE%E3%83%A2%E3%82%B8%E3%83%A5%E3%83%BC%E3%83%AB%E3%82%92-iris-%E3%81%8B%E3%82%89%E5%AE%9F%E8%A1%8C%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95