

---

## 記事

[Megumi Kakechi](#) · 2021年6月3日 5m read

# Javaから配列のような複数項目のデータを渡す方法

これは [InterSystems FAQ サイト](#) の記事です。

Java のプログラムで \$List 構造のデータを扱うには、InterSystems 製品が提供する IRISList クラスを使用することが可能です。

こちらは、IRIS Native と同様に JDBC ドライバの拡張機能の一部として提供しております。

[Native API for Java Quick Reference \(英語のみ\)](#)

こちらでは、InterSystems 製品上に \$List 構造を扱うクラスメソッドを作成し、それをストアードプロシージャとして公開し Java から呼び出すサンプルをご紹介します。

## 【使用可能バージョン】

IRIS 2020.1 以降に同梱されている JDBC ドライババージョン 3.1.0 以降。  
ただし、JDBC で使用する場合は 3.2.0 以降 (IRIS2021.1 以降)。

<IRIS インストールディレクトリ>/dev/java/lib/JDK(Ver.)/intersystems-jdbc-3.1.0.jar

## 【手順】

1. IRIS サーバの任意のネームスペースに以下のようなクラスを用意します。  
[SqlProc キーワード](#) を指定し、メソッドを SQL ストアドプロシージャとして呼び出せるようにします。  
こちらのサンプルでは、^scoreDetails グローバルに Java から渡されたリスト形式の gamesList をセットしています。

後程おまけのサンプルで使用するリスト型のプロパティ FavoriteColors も追加しています。

```
Class User.ListTest Extends %Persistent
{
Property Name As %String
Property FavoriteColors As list Of %String;
ClassMethod writeTest(name As %String, gamesList As %List) [ SqlProc ]
{
set ^scoreDetails(name,"gamesList") = gamesList
quit
}
}
```

2. Java のコードを用意します。今回は、JDBC を使用する方法と、[Native API for Java](#) を使用する方法をご紹介します。

JDBC でストアードプロシージャを実行するサンプル

```

import java.sql.*;
import java.util.*;
import com.intersystems.jdbc.*;

public class SListTestIRIS1 {

    public static void main(String[] args) throws Exception {
        String url = "jdbc:IRIS://127.0.0.1:1972/USER";

        IRISDataSource dataSource = new IRISDataSource();
        dataSource.setURL(url);
        dataSource.setUser("username");
        dataSource.setPassword("password");
        IRISConnection connection = (IRISConnection) dataSource.getConnection();

        IRISList list = new IRISList();
        list.add("basketball");
        list.add("tennis");
        //list.set(1,"basketball"); // add, set ????????OK
        //list.set(2,"tennis");
        PreparedStateme
nt cstmt = connection.prepareStatement("{ call SQLUser.ListTest_writeTest(?, ?) }");

        cstmt.setString(1, "David");
        cstmt.setBytes(2
,Arrays.copyOf(list.getBuffer(),list
.size())); // ?????IRISList????????????Arrays????

        //cstmt.setBinaryStream(2, new java.io.ByteArrayInputStream(list.getBuffer()),list.si
ze()); // 4.3MB???byte[] ????????????
        cstmt.execute();
        cstmt.close();

        return;
    }
}

```

実行結果：

```

USER>zw ^scoreDetails
^scoreDetails("David","gamesList")=$lb("basketball","tennis")

```

### Native APIでクラスメソッドを呼び出すサンプル

```

import com.intersystems.jdbc.*;

public class SListTestIRIS2 {

    public static void main(String[] args) throws Exception {
        String url = "jdbc:IRIS://127.0.0.1:1972/USER";

        IRISDataSource dataSource = new IRISDataSource();
        dataSource.setURL(url);
        dataSource.setUser("username");
    }
}

```

```

        dataSource.setPassword("password");
        IRISConnection connection = (IRISConnection) dataSource.getConnection();

        // create IRIS Native object
        IRIS iris = IRIS.createIRIS(connection);

        IRISList list = new IRISList();
        list.add("basketball");
        list.add("cricket");
        //list.set(1,"basketball");    // add, set ???????OK
        //list.set(2,"tennis");

        iris.classMethodVoid("User.ListTest", "writeTest", "Nicole", list);

        return;
    }
}

```

実行結果：

```

USER>zw ^scoreDetails
^scoreDetails("Nicole","gamesList")=$lb("basketball","cricket")

```

おまけ

INSERT で文字列のリストを設定する場合、以下のような単純な方法で行うことができます。

List Of %String

のプロパティは、値をカンマで区切り、二重引用符で囲んだ単純な文字列として渡すことができます。

```

import java.sql.*;
import com.intersystems.jdbc.*;

public class SListTestIRIS3 {

    public static void main(String[] args) throws Exception {
        String url = "jdbc:IRIS://127.0.0.1:1972/USER";

        IRISDataSource dataSource = new IRISDataSource();
        dataSource.setURL(url);
        dataSource.setUser("username");
        dataSource.setPassword("password");
        IRISConnection connection = (IRISConnection) dataSource.getConnection();

        PreparedStatement insert = connection.prepareStatement("INSERT INTO SQLUser.Li
stTest (name, favoritecolors) " + "VALUES (?,?)");

        insert.setString(1, "InterSystems, Japan");
        insert.setString(2, "\"red\", \"orange\", \"yellow\", \"green\", \"blue\", \"purple
\"");

        int rowsAffected = insert.executeUpdate();

        return;
    }
}

```

```
    }  
}
```

実行結果：

```
USER>d $SYSTEM.SQL.Shell()  
SQL Command Line Shell
```

```
-----  
  
The command prefix is currently set to: <<nothing>>.
```

```
Enter <command>, 'q' to quit, '?' for help.
```

```
[SQL]USER>>select * from SQLUser.ListTest
```

```
1.      select * from SQLUser.ListTest
```

```
ID      FavoriteColors  Name
```

```
1      $lb("red","orange","yellow","green","blue","purple")  InterSystems, Japan
```

```
1 Rows(s) Affected
```

```
statement prepare time(s)/globals/cmds/disk: 0.0629s/40013/185744/15ms
```

```
execute time(s)/globals/cmds/disk: 0.0014s/2/816/0ms
```

```
cached query class: %sqlcq.USER.cls9
```

```
-----  
[SQL]USER>>
```

[#Java #JDBC #ヒントとコツ #InterSystems IRIS #InterSystems IRIS for Health](#)

---

ソースURL:

<https://jp.community.intersystems.com/post/java%E3%81%8B%E3%82%89%E9%85%8D%E5%88%97%E3%81%AE%E3%82%88%E3%81%86%E3%81%AA%E8%A4%87%E6%95%B0%E9%A0%85%E7%9B%AE%E3%81%AE%E3%83%87%E3%83%BC%E3%82%BF%E3%82%92%E6%B8%A1%E3%81%99%E6%96%B9%E6%B3%95>