記事
Mihoko Iijima · 2021年5月28日 8m read

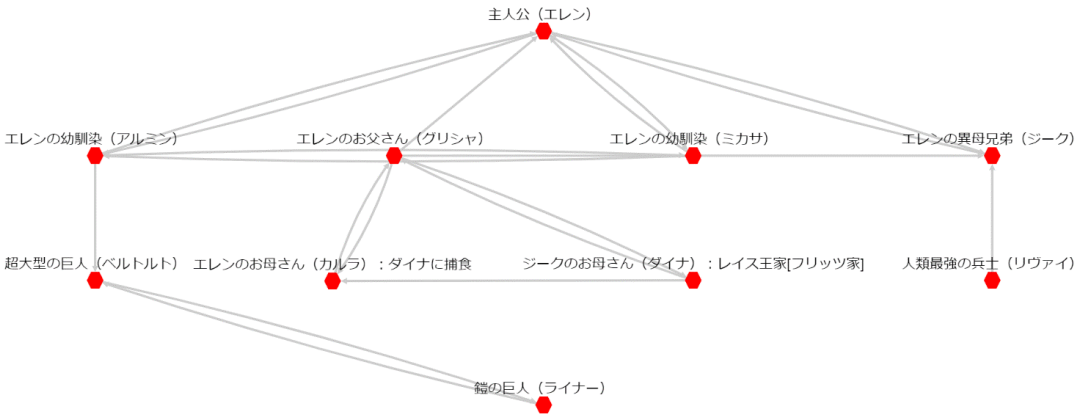
キーバリュー形式で Python / Node.js / Java から IRIS にアクセスできるテンプレート（グラフ構造によくある人物相関図を IRIS で表現しています）

開発者の皆さん、こんにちは　　今年は早い梅雨入りでした

さて、新しい　実行／開発環境テンプレートを作成しました。 Docker　　、docker-compose 、git がインストールされていれば、すぐにお試しいただけます。ぜひご利用ください！

今回は、ご存知の方が多いと思われる（？）某アニメの登場人物を使った人物相関図をテーマに【キーバリュー形式で IRIS に登録してグラフ構造で表示してみた】を体験できるテンプレートです（テンプレートは、Python／Node.js／Java からお試しいただける環境をご用意しています）。

人物相関図のイメージ



以下、今回のテーマについて、ビデオと文字でご紹介しています。最後までお付き合いいただければ幸いです！（ビデオは全体で 7 分 20 秒）

人物相関図と言えば、グラフデータベースをイメージされると思います。

IRIS はグラフデータベース**ではない**のですが、IRIS ネイティブのデータの「**グローバル**」を利用することで、グラフデータベースと似たような構造を表現することができます。

> IRIS の高パフォーマンスを支える「**グローバル**」は 40 年以上前（＝ InterSystems 創業）から InterSystems
> のコア技術であるデータベースとして提供さ
> れてきました。「**グローバル**」に対する操作方法は、現代のカテゴリに合わせるとしたら NoSQL

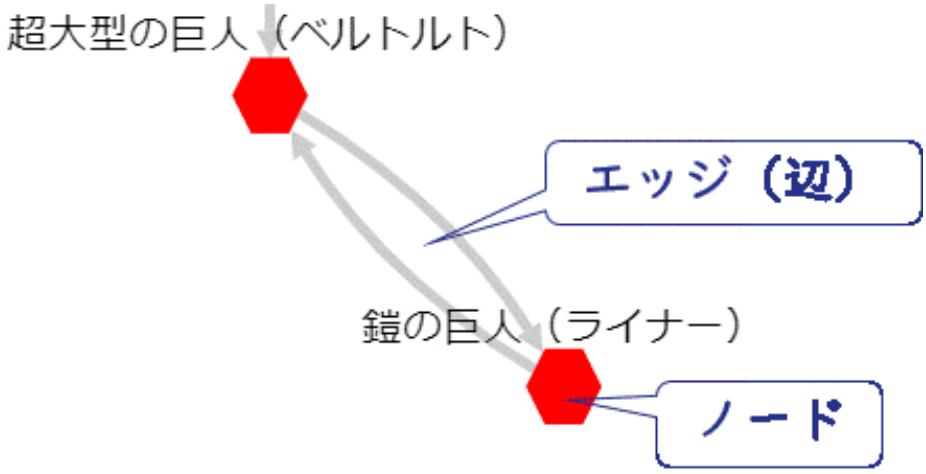
データベースと言えます。

では、どのようにグラフデータベースのような構造を表現しているか？についてですが、グラフ構造は、ノードと辺から構成されていて、辺は 2 つのノードを結び付けるものです。

SNS の「友達」で考えると、ノードは「ユーザ」、辺は「友達関係」で表現できます。

テンプレートで使用している人物相関図では、ノードは「登場人物」、辺は「登場人物との関係」を表現しています。

人物相関図のノードと辺（エッジ）



ノードと辺を、どのようにグローバル変数に設定しているでしょうか。

ノードは以下の通りです（配列には、画面表示に利用するノードの ID を設定し、右辺に人物名を登録しています）。

```
^Relation("Eren")="????????"
```

辺（エッジ）は以下の通りです（グローバル変数の配列を利用して、**登場人物** → **関係のある人** [**ソース**→**ターゲット**] を設定しています）。

主人公エレンは、アルミン、ミカサ、ジークと関係がある。を表現しています。

```
^Relation("Eren","Armin")=""
^Relation("Eren","Mikasa")=""
^Relation("Eren","Zeke")=""
```

両者で関係がある場合は、さらに以下のような配列を追加します。

```
^Relation("Mikasa")="???????????"
```

```
^Relation("Mikasa","Armin")=""
^Relation("Mikasa","Eren")=""
```

実際に、IRIS サーバ側で記述する場合には、ObjectScript の SET コマンドを使用してグローバル変数を設定します。

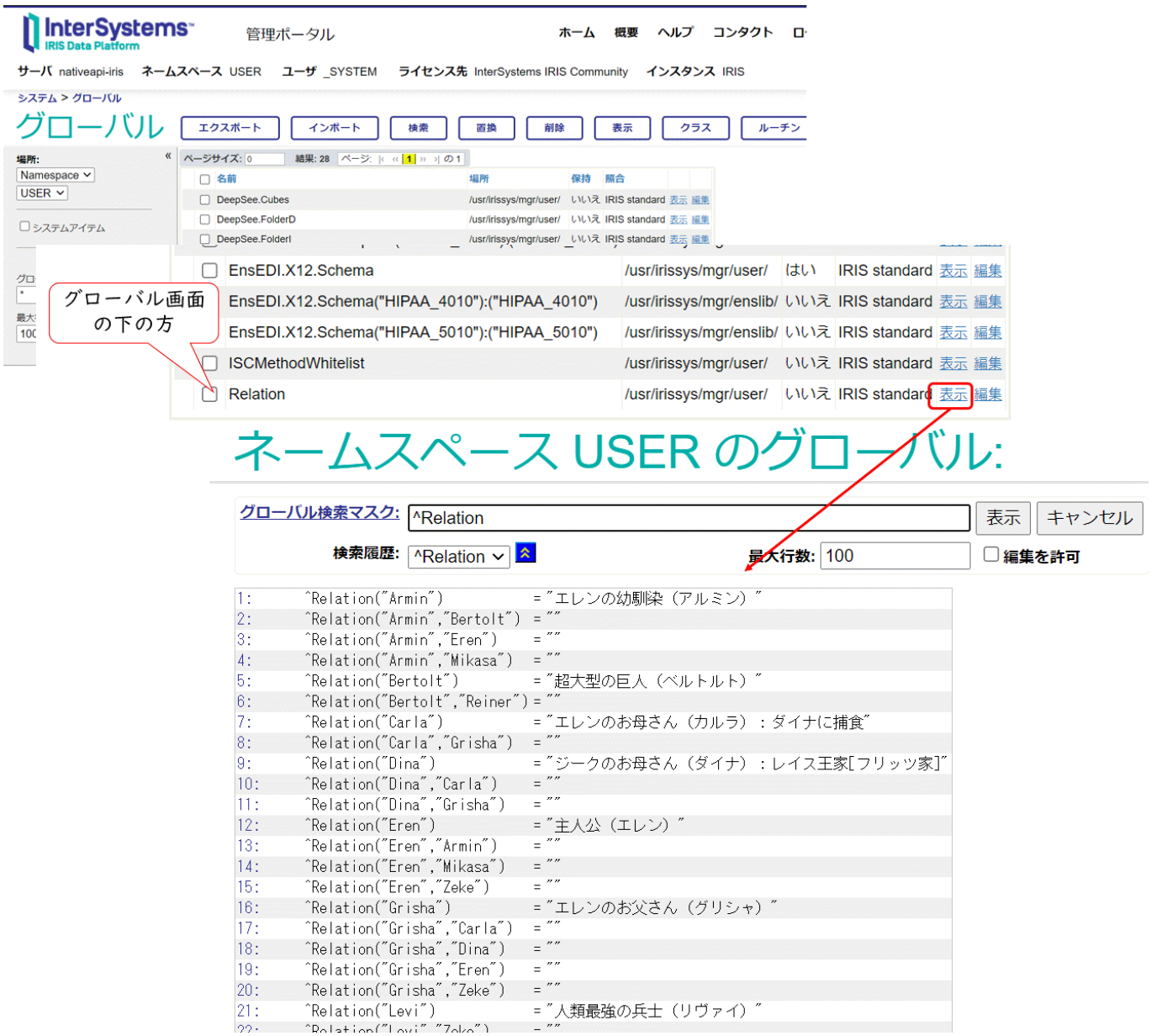
```
set ^Relation("Eren")="????????"
set ^Relation("Eren","Mikasa")=""
set ^Relation("Eren","Armin")=""
set ^Relation("Eren","Zeke")=""
set ^Relation("Mikasa")="???????????"
set ^Relation("Mikasa","Armin")=""
set ^Relation("Mikasa","Eren")=""
```

配列のサブスクリプト（括弧の中身）は、配列のノード（例では、第 1 番目と第 2 番目）毎に Unicode 昇順でソートされます。

実行後、管理ポータルなどからグローバル変数一覧を参照すると、実行順に関係なく Unicode 昇順にソートされていることを確認できます。

> 管理ポータルは、http://localhost:52779/csp/sys/UtilHome.csp でアクセスできます（ユーザ名：system 、パスワード：SYS）。
>
> ※ポート番号はご利用環境に合わせてご変更ください。

管理ポータル > [システムエクスプローラ] > [グローバル] > 左画面で「ネームスペース」USER を選択 > ^Relation の「表示」をクリック



ここまでご紹介したグローバル変数に
対する各言語のコード例については、テンプレート用リポジトリの以下サブディレクトリをご参照ください。

https://github.com/Intersystems-jp/IRIS-NativeAPI-Template

- Python ：jupyter のコンテナを用意します（Jupyter は 8896 ポートでアクセスできます）。
- Node.js：Node 12 のコンテナを用意します（8080 ポートで確認用 Web ページを参照できます）。
- Java：OpenJDK 8 のコンテナを用意します。

1) テンプレートの処理概要

各言語ごとのシンプルなコード例とコンテナで提供している内容についての解説は、以下のビデオでもご紹介しています（全体で 9 分 20 秒）。

テンプレートでは、Python / Node.js / Java 用から IRIS
用コンテナへ、キーバリュー形式でのアクセスを行うため、Native API を使用しています。

> 各言語で必要な irisnative
> モジュールを使用するための手順は、言語ごとのサブディレクトリで解説しています。

- Python
- Node.js
- Java

ビデオでもご紹介していますが、シンプルなコード例は以下の通りです。

Python の例

```
#irisnative??????????
import irisnative

#IRIS??? (????,???????????,???????,???,?????)
connection = irisnative.createConnection("iris",1972,"user","_system","SYS")

#IRIS????????
iris_native = irisnative.createIris(connection)

#???
iris_native.set("????\n??????","Relation","Reiner")
iris_native.set("??????\n???????","Relation","Bertolt")
iris_native.set(None,"Relation","Reiner","Bertolt")
iris_native.set(None,"Relation","Bertolt","Reiner")

#Iterator()
for source,value in iris_native.iterator("Relation").items():
    print(source,"-",value)
    for target,value in iris_native.iterator("Relation",source).items():
        print("  ????",target)

connection.close()
```

Node.js の例

```
const irisnativeapi = require('intersystems-iris-native');

//??
let connectionInfo = {"host": "nativeapi-
iris","port": 1972,"ns":"USER","user":"_SYSTEM","pwd":"SYS"};
const connection = irisnativeapi.createConnection(connectionInfo);

//IRIS????????
const irisNative = connection.createIris();

//???
irisNative.set("??????????","Relation","Reiner");
```

```
irisNative.set("?????????????","Relation","Bertolt");
irisNative.set(null,"Relation","Reiner","Bertolt");
irisNative.set(null,"Relation","Bertolt","Reiner");
let ite1=irisNative.iterator("Relation");

//Iterator()
for ([source,data] of ite1) {
    console.log("source"+"-"+data);
    let ite2=irisNative.iterator("Relation",source);
    for ([target] of ite2) {
        console.log("   ????",target);
    }
}

connection.close();
```

【Java の例】

```
package NativeAPI;
import com.intersystems.jdbc.IRIS;
import com.intersystems.jdbc.IRISConnection;
import com.intersystems.jdbc.IRISDataSource;
import com.intersystems.jdbc.IRISIterator;

public class Test{

    public static void main (String[] args) {
        try {
        //???????? IRISDataSource???
        IRISDataSource ds = new IRISDataSource();
        // jdbc:IRIS://????:??????????/????????
        ds.setURL("jdbc:IRIS://localhost:1972/user");
        ds.setUser("_SYSTEM");  //????
        ds.setPassword("SYS");  //?????
        IRISConnection dbconnection = (IRISConnection) ds.getConnection();
         //create irisNative object
        IRIS irisNative = IRIS.createIRIS(dbconnection);
        irisNative.set("??????????","Relation","Reiner");
        irisNative.set("?????????????","Relation","Bertolt");
        irisNative.set("","Relation","Reiner","Bertolt");
        irisNative.set("","Relation","Bertolt","Reiner");
        IRISIterator character=irisNative.getIRISIterator("Relation");
        while (character.hasNext()) {
            String source=character.next();
            System.out.println("\n?? = "+ source + " - ???"+ character.getValue());
            //??????????^Relation?
            IRISIterator correlate=irisNative.getIRISIterator("Relation",source);
            while (correlate.hasNext()) {
                String target=correlate.next();
                System.out.println("   ??? : "+ target);
            }
        }
        //iris????Close
        irisNative.close();
        dbconnection.close();
    }
```

```
    catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}
```

2) Native API について

Native API は、IRIS 内部のネイティブデータ（＝グローバル変数）を直接操作できる API
で Python、Node.js、Java、.NET からアクセスできます。

> グローバル変数の操作には、IRIS
> サーバーサイドプログラミングで使用する ObjectScript を利用しますが、Native API
> を利用することで、ObjectScript を使用せずにお好みの言語からアクセスすることができます。

また、Native API
は、グローバル変数の設定／取得の他にも、クラスメソッド、ルーチン、関数を実行することができます。

3) テンプレートの使用方法

言語ごとのサブディレクトリにある README をご参照ください。

- Python
- Node.js
- Java

テンプレートの例に使用した某アニメの人物相関図ですが、実は完成していません。

あの登場人物がいない！とお気づきの方、ぜひ追加いただき、よろしければ記事の返信欄に貼り付けていただければ・・。

最後までお読みいただきありがとうございました！

関連記事（NoSQLをテーマにした詳細説明付き記事）：Python Native APIでNoSQLデータベースにアクセス

#Java #Node.js #Python #キーバリュー #コンテナ化 #初心者 #InterSystems IRIS #InterSystems IRIS for Health

---

ソースURL:
https://jp.community.intersystems.com/post/%E3%82%AD%E3%83%BC%E3%83%90%E3%83%AA%E3%83%A5%E3%83%BC%E5%BD%A2%E5%BC%8F%E3%81%A7-python-nodejs-java-%E3%81%8B%E3%82%89-iris-%E3%81%AB%E3%82%A2%E3%82%AF%E3%82%BB%E3%82%B9%E3%81%A7%E3%81%8D%E3%82%8B%E3%83%86%E3%83%B3%E3%83%97%E3%83%AC%E3%83%BC%E3%83%88%EF%BC%88%E3%82%B0%E3%83%A9%E3%83%95%E6%A7%8B%E9%80%A0%E3%81%AB%E3%82%88%E3%81%8F%E3%81%82%E3%82%8B%E4%BA%BA%E7%89%A9%E7%9B%B8%E9%96%A2%E5%9B%B3%E3%82%92-iris-%E3%81%A7%E8%A1%A8%E7%8F%BE%E3%81%97%E3%81%A6%E3%81%84%E3%81%BE%E3%81%99%EF%BC%89