記事
Mihoko Iijima · 2020年8月5日 8m read

【はじめてのInterSystems IRIS】セルフラーニングビデオ：アクセス編：IRIS での JSON の操作

IRIS サーバ側で JSON の操作を行う方法を解説します（3つのビデオに分かれています）。

ビデオ① ：ダイナミックエンティティの操作練習

ビデオ② ：ダイナミックエンティティで利用できるメソッドの練習

ビデオ③ ：SQL関数と %JSON.Adapter の使い方

なお、このビデオには、以下の関連ビデオがあります。ぜひご参照ください。

- IRISの基本操作についてのビデオ（索引ページ）
- InterSystems開発者コミュニティ biginnerタグ一覧

ビデオ①

このビデオの目次は以下の通りです。

最初～ 復習ビデオ／関連ビデオについて など

- IRIS で 作成する REST サーバの仕組み
- 手動によるRESTディスパッチクラスの作成
- APIファーストによるRESTディスパッチクラスの作成

2:05～ JSONとは？

3:26～ JSONオブジェクト：ダイナミックエンティティの作成

```
//%DynamicObject??????
set json=##class(%DynamicObject).%New()
set json.Name="?????"
set json.Address="??????"
write json.%ToJSON()
// ??????????
//{"Name":"?????","Address":"??????"}
//????JSON??????? {} ??????
set json={}   // %DynamicObject???
set json.Name="?????",json.Address="??????"
write json.%ToJSON()
// ??????????
//{"Name":"????","Address":"??????"}
```

7:25～ JSON配列：ダイナミックエンティティでの作成例

```
//%DynamicArray??????
set array=##class(%DynamicArray).%New()
set array."0"="??1"  //???????????0??????
set array."1"="??2"
write array.%ToJSON()
// ??????????
//["??1","??2"]
//????JSON??????? [] ??????
set array=[]   //%DynamicArray???
set array."0"="??1",array."1"="??2"
write array.%ToJSON()
// ??????????
//["??1","??2"]
```

9:49～ ダイナミックエンティティ操作用のメソッド

※ ビデオ②に続きます

先頭へ戻る

ビデオ②

もくじは以下の通りです。

00:00～ %Set()、%Get()、%Remove() オブジェクト編

```
set obj={}
set obj.Name="????"
do obj.%Set("Zip","160-0023")
do obj.%Set("Tel","03-5321-6200")
write obj.%ToJSON()
write obj.%Get("Zip")," - ",obj.%Get("Name")
do obj.%Remove("Zip")
write obj.%ToJSON()
set obj.Pref="???"
write obj.%ToJSON()
do obj.%Set("City","???")
write obj.%ToJSON()
```

02:02～ %Set()、%Get()、%Remove() 配列編

```
set array=[]
do array.%Set(0,"??")
write array.%ToJSON()
set array."4"="??"   //set array."??"="?" ? array.%Set("??","?")???
```

```
do array.%Set(2,"??")
write array.%ToJSON()
do array.%Pop()
write array.%ToJSON()
do array.%Push("Push?????")
write array.%ToJSON()
do array.%Remove(1)   // ???2??? null ???
write array.%ToJSON()
```

03:34～ JSON配列 要素の操作：%Size()、%Get()

```
set array=["??",null,""]
do array.%Set(4,"??")   // ????????3 ?JSON?null???
write array.%ToJSON()
// ??????????
["??",null,"",null,"??"]

for i=0:1:array.%Size()-1 w array.%Get(i),!
// ??????????
??



??
```

05:20～ 値が有効値かどうか確認する %IsDefined()

```
set array=["??",null,""]
do array.%Set(4,"??")   // ????????3 ?JSON?null???
write array.%ToJSON()
// ??????????
["??",null,"",null,"??"]

for i=0:1:array.%Size()-1 {write array.%Get(i)," - ????",array.%IsDefined(i),!}
// ??????????
?? - ????1
 - ????1
 - ????1
 - ????0
?? - ????1
```

07:38～ 反復処理：配列の場合：%GetIterator()

```
set array=["??",null,""]
do array.%Set(4,"??")   // ????????3 ?JSON?null???
write array.%ToJSON()
// ??????????
["??",null,"",null,"??"]

set iter=array.%GetIterator()
```

```
while iter.%GetNext(.key,.val) { write key," - value=  ",val,! }
// ??????????
0 - value=  ??
1 - value=
2 - value=
4 - value=  ??
```

09:55～ 反復処理：オブジェクトの場合：%GetIterator()

```
set obj={"Name":"????","Zip":"160-0023","Pref":"???","City":"???"}
write obj.%ToJSON()
// ??????????
{"Name":"????","Zip":"160-0023","Pref":"???","City":"???"} set iter=obj.
%GetIterator()

set iter=obj.%GetIterator()
while iter.%GetNext(.key,.val) { write key," - value=  ",val,! }
// ??????????
Name - value=  ????
Zip - value=  160-0023
Pref - value=  ???
City - value=  ???
```

10:11～ JSON null/true/false （ObjectScriptの中での対応）

```
set array=[null,true,false,1,0,""]
set array."7"="???"
for i=0:1:array.%Size()-1 { write i," - ",array.%Get(i),! }
// ??????????
0 -
1 - 1
2 - 0
3 - 1
4 - 0
5 -
6 -
7 - ???
```

11:40～ JSONのデータタイプを確認する＝ %GetTypeOf()メソッド

```
set array=[null,true,false,1,0,""]
set array."7"="???"
for i=0:1:array.%Size()-1 {write i," - ",array.%Get(i)," - ",array.%GetTypeOf(i),! }
// ??????????
0 -  - null
1 - 1 - boolean
2 - 0 - boolean
3 - 1 - number
4 - 0 - number
5 -  - string
```

```
6 -  - unassigned
7 - ??? - string
```

12:32～ JSONのデータタイプを指定して値を設定する

```
set array=[]
do array.%Set(0,"","null")  // ?2??????????null
do array.%Set(1,1,"number")  //???1?????
do array.%Set(2,0,"number")  //???0?????
do array.%Set(3,1,"boolean")  // boolean?1?true?????
do array.%Push(0,"boolean")  // boolean?0?false ?????
for i=0:1:array.%Size()-1 { write i," - ",array.%Get(i)," - ",array.%GetTypeOf(i),! }
// ??????????
0 -  - null
1 - 1 - number
2 - 0 - number
3 - 1 - boolean
4 - 0 - boolean

write array.%ToJSON()
// ??????????
[null,1,0,true,false]
```

13:38～ ObjectScriptの変数や表現式を [] や {} で使用する方法

```
set obj={"??":($ZDATE($H,16)),"??":($ZTIME($PIECE($H,",",2)))}
write obj.%ToJSON()
set mgr=$system.Util.ManagerDirectory()
set array=[($system.Util.InstallDirectory()),(mgr)]
write array.%ToJSON()
```

※ ビデオ③へつづきます

先頭へ戻る

ビデオ③

もくじは以下の通りです。

00:00～ テーブルデータをJSONオブジェクト、JSON配列で取得する方法 概要

関連ビデオのご紹介

- 基本操作についてのビデオ（索引ページ）
- 【はじめての InterSystems IRIS】セルフラーニングビデオ：基本その3：IRIS でクラス定義を作ろう（オブジェクト操作の練習）

（Test.Personの作り方を確認する場合に良いビデオ）

上記ビデオの 13:20～（スタジオでの操作）／18:44～（VS Code での操作） で作成方法を紹介しています。

00:54～ SQL:SELECTでの操作 JSONOBJECT()関数 説明と実演

管理ポータル→システムエクスプローラ→SQL を開き対象ネームスペースに移動後
クエリ実行タブで以下実行します。
SELECT JSONOBJECT('Name':Name,'Email':Email) ABSENT ON NULL as json from Test.Person

02:42～ JSONOBJECT()例（埋込SQLでの実行例）

```
Class Test.JSONTest
{
ClassMethod GetAllPerson()
{
 //??SQL
 &sql(declare C1 cursor for
 select JSON_OBJECT('Name':Name,'Email':Email) as json into :json from Test.Person)
 &sql(open C1)
 set array=[]
 for {
  &sql(fetch C1)
  if SQLCODE'=0 {
   quit
  }
  set obj={}.%FromJSON(json)
  do array.%Push(obj)
 }
 &sql(close C1)
 write array.%ToJSON()
}
}
```

```
//???
do ##class(Test.JSONTest).GetAllPerson()
```

07:00～ SQL：SELECTでの操作 JSONARRAY()関数

管理ポータル→システムエクスプローラ→SQL を開き対象のネームスペースに移動後
クエリ実行タブで以下実行します。
SELECT JSONARRAY(Name,Email ABSENT ON NULL) as array from Test.Person

07:50～ JSONARRAY()例（ダイナミックSQL実行例）

```
ClassMethod GetAllPersonArray()
{
    set sql="SELECT JSON_ARRAY(Name,Email ABSENT ON NULL) as array from Test.Person"
    set stmt=##class(%SQL.Statement).%New()
```

```
    set status=stmt.%Prepare(sql)
    set rset=stmt.%Execute()
    set root=[]
    while rset.%Next() {
        set array=[].%FromJSON(rset.%Get("array"))
        do root.%Push(array)
    }
    do root.%ToJSON()
}



//???
do ##class(Test.JSONTest).GetAllPersonArray()
```

08:59〜 JSONアダプタ（%JSON.Adapter）

```
set person=##class(Test.Person).%OpenId(1)
set status=person.%JSONExport()
write status
```

10:08〜 オブジェクト→JSON文字列を含むストリーム %JSONExportToStream() 説明と実演

```
set person=##class(Test.Person).%OpenId(1)
set st=person.%JSONExportToStream(.jstream)
write st
write jstream.Read()
write jstream.Rewind()
set jobj={}.%FromJSON(jstream.Read())
write jobj.Name
write jobj.Email
```

12:59〜 オブジェクト→JSON文字列にマッピング %JSONExportToString()

```
set person=##class(Test.Person).%OpenId(1)
set st=person.%JSONExportToString(.jstring)
write st
write jstring
set jobj={}.%FromJSON(jstring)
write jobj.Name
write jobj.Email
```

13:20〜 JSON文字列→オブジェクトへのマッピング %JSONImport()

```
set json={}
set json.Name="?????", json.Email="json@mail.com"
zwrite json
set p1=##class(Test.Person).%New()
set st=p1.%JSONImport(json)
```

```
write st
```

14:24〜 確認できたこと

《2024/1/16追記》ビデオには含まれていませんが、バージョン2023.3以降でJSON配列に情報を追加できる add()メソッド、JSON配列同士の結合に便利な addAll()メソッドが追加されました。

```
USER>set a1=["a","b","c"]

USER>do a1.add("??")

USER>zwrite a1
a1=["a","b","c","??"]  ; <DYNAMIC ARRAY>
USER>

USER>set a2=[1,2,3]

USER>do a1.addAll(a2)

USER>zwrite a1
a1=["a","b","c","??",1,2,3]  ; <DYNAMIC ARRAY>
```

先頭へ戻る

#JSON #ObjectScript #REST API #ビデオ #初心者 #Caché #Ensemble #InterSystems IRIS #InterSystems IRIS for Health

---

ソースURL:
https://jp.community.intersystems.com/post/%E3%80%90%E3%81%AF%E3%81%98%E3%82%81%E3%81%A6%E3%81%AEintersystems-iris%E3%80%91%E3%82%BB%E3%83%AB%E3%83%95%E3%83%A9%E3%83%BC%E3%83%8B%E3%83%B3%E3%82%B0%E3%83%93%E3%83%87%E3%82%AA%EF%BC%9A%E3%82%A2%E3%82%AF%E3%82%BB%E3%82%B9%E7%B7%A8%EF%BC%9Airis-%E3%81%A7%E3%81%AE-json-%E3%81%AE%E6%93%8D%E4%BD%9C