記事
Mihoko Iijima · 2021年6月1日 6m read

REST/JSON の簡単なサンプルご紹介

これは　InterSystems FAQ サイト　の記事です。

InterSystems 製品を利用した REST/JSON の操作方法を、簡単なサンプルを利用して解説します。

サンプルでは、REST
クライアント、HTML、ターミナルからデータ(JSON)を送信し、サーバ(InterSystems製品)で JSON
形式のデータを返す REST ディスパッチクラスを使った簡単な流れになっています。

サンプルは、　https://github.com/Intersystems-jp/FAQ-REST-SimpleSample からダウンロードいただけます。

サンプルの利用手順

- **(1) サンプルファイル(XML)のインポート**

RESTディスパッチクラスとサンプルHTMLファイルが含まれています

- **(2) ウェブアプリケーションパスの設定**
- **(3) 実行**

---

(1) サンプルファイル(XML)のインポート

管理ポータルからインポートします(スタジオを利用されている場合は、スタジオでもインポートできます)。

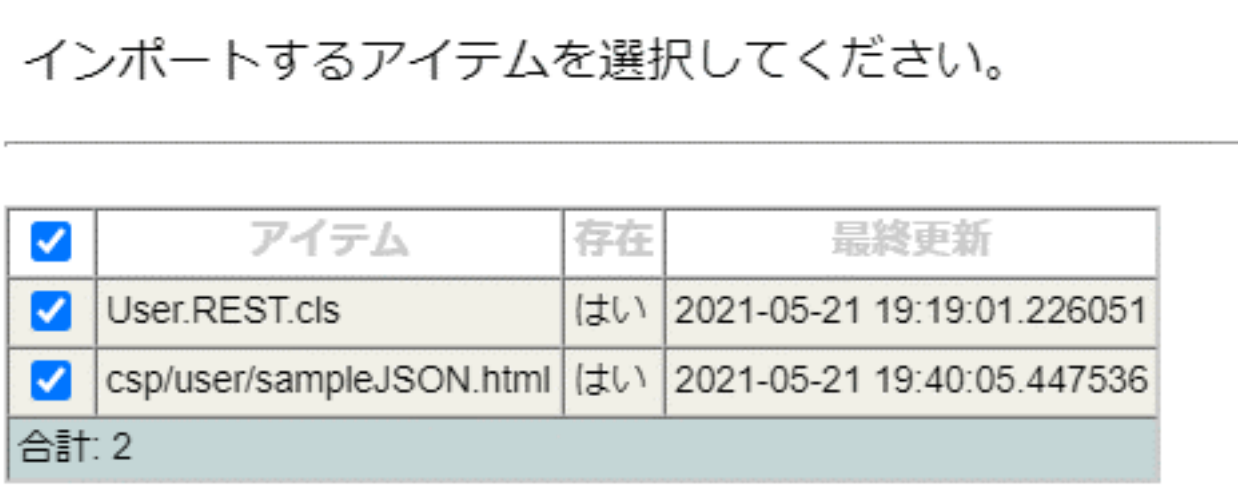
管理ポータルは　http://localhost:52773/csp/sys/UtilHome.csp でアクセスできます。

※ ポート番号はお使いの環境に合わせて変更してください。

管理ポータルにアクセスできたら以下メニューでインポートを行ってください。

システムエクスプローラ > クラス > ネームスペース選択 > インポートボタンクリック
> サンプルファイル　を選択してインポート

メモ：インポートファイルを選択すると、インポートアイテムが表示されます(下図)。



ｸﾗｽ定義(User.REST)ﾋHTMLﾌｧｲﾙ(sampleJSON.html)ｶｲﾝﾎﾟｰﾄされます。

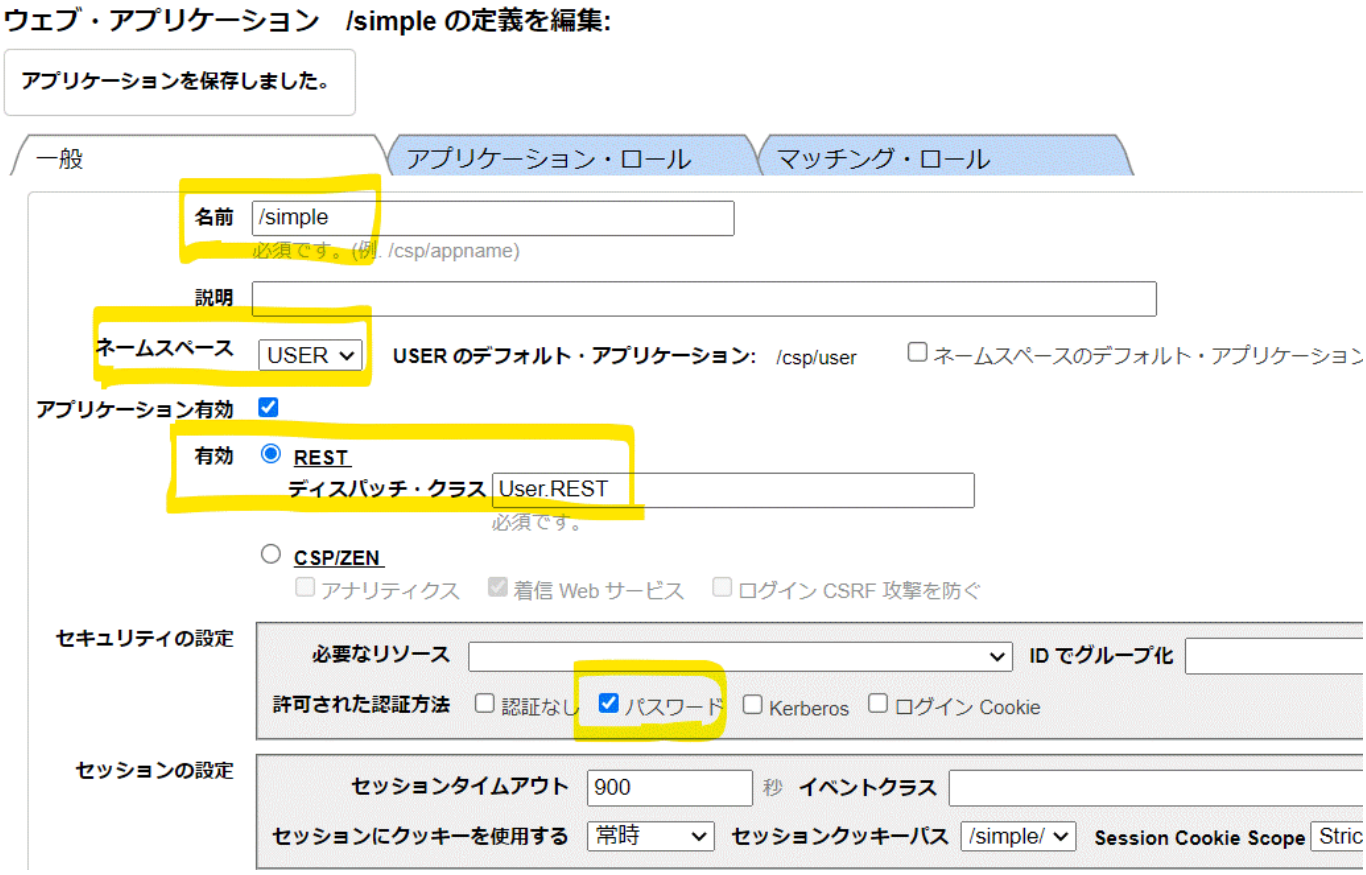
(2) ｳｪﾌﾞｱﾌﾟﾘｹｰｼｮﾝﾊﾟｽの設定

(1) でｲﾝﾎﾟｰﾄしｾﾀｸﾗｽ定義(User.REST)を REST
ﾃｨｽﾊﾟｯﾁｸﾗｽに指定しﾀｳｪﾌﾞｱﾌﾟﾘｹｰｼｮﾝﾊﾟｽを定義します。

管理ﾎﾟｰﾀﾙを開ｷ、以下ﾒﾆｭｰにｱｸｾｽします。

ｼｽﾃﾑ管理 > ｾｷｭﾘﾃｨ > ｳｪﾌﾞｱﾌﾟﾘｹｰｼｮﾝ >
「新しいｳｪﾌﾞｱﾌﾟﾘｹｰｼｮﾝの作成」ﾘｯｸし、以下の図の設定を行ってください。

(ｻﾝﾌﾟﾙでは、/simple を REST のﾍﾞｰｽﾊﾟｽにしています)



- ネームスペースはサンプルをインポートしたネームスペースを指定してください。
- ディスパッチクラス名には、(1)でインポートした User.REST を指定します。大文字小文字を区別します。記入時ご 注 意ください。
- 「許可された認証方法」は、「パスワード」を設定します。アクセス時、ユーザ名とパスワードの入力欄が出てきたらユーザ名：_system、パスワードは SYS（または設定したパスワード）を記入してログインします。

  メモ：管理ポータルアクセス時にユーザ名、パスワードの入力欄が出ないアクセス（認証なしアクセス）では、デフォルト設定としてユーザ：_system に対するパスワードが **SYS**（大文字）で設定されています。サンプルテスト時ご利用ください。

(3) 実行

(2) で作成したRESTディスパッチクラスを起動するためのベースパス（/simple）を利用してテストできます。

REST ディスパッチクラスには UrlMap という定義があり、ベースパスのあとに指定したパスに合わせて、どのメソッドが実行されるか定義されています。

```
XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="/req1" Method="GET" Call="req1"/>
```

```
<Route Url="/req2" Method="GET" Call="req2"/>
</Routes>
}
```

/simple/req1 を指定すると、以下のメソッドが実行されます。

```
ClassMethod req1() As %Status
{
 set name1=%request.Get("NAME1")
 set name2=%request.Get("NAME2")
 set age=%request.Get("AGE")

 set tdobject = {}
 set tdobject.name=name1_" "_name2
 set tdobject.age=age
 write tdobject.%ToJSON()

    quit $$$OK
}
```

[メモ]

HTTP 応答の Content-Type ヘッダの設定は、REST ディスパッチクラスのクラスパラメータ **CONTENTTYPE** に **"application/json"**、charset の指定は、クラスパラメータ **CHARSET** に **"utf-8"** を設定しているため、クラスメソッド内で指定する必要はありません。
また、今回の例にはありませんが、メッセージボディの情報をクラスパラメータ **CHARSET** で自動的に変換させたい場合は、クラスパラメータ **CONVERTINPUTSTREAM** に **1** を設定します（設定例は以下例文をご参照ください）。

```
Parameter CONTENTTYPE = "application/json";

Parameter CHARSET = "utf-8";

Parameter CONVERTINPUTSTREAM = 1;
```

以下 URL を利用して、/simple/req1 のテストが行えます（クエリ文字列に指定した内容が JSON として返送されます）。

http://localhost:52773/simple/req1?NAME1=Yamada&NAME2=Taro&AGE=20

※ ポート番号はお使いの環境に合わせて変更してください。

```
{"name":"Yamada Taro","age":"20"}
```

サンプル HTML　http://localhost:52773/csp/user/sampleJSON.html を利用してテストする場合は、Pushボタンをクリックすると、HTMLファイル内で設定したクエリ文字列を GET 要求で送信し、RESTディスパッチクラスで加工した文字列が JSON として返送されます（ポップアップで表示されます）。

以下、Pushボタン押下時の JavaScript です。

```
function go() {
  var url="/simple/req1";
  var params = {"NAME1" : "??", "NAME2":"??","AGE":20};
  var queryparams= new URLSearchParams(params);
  url=url+"?"+ queryparams
   fetch(url,{
   headers:{"Accept":"application/json"},
 }
   ).then(function(Response) {
          console.log("status=" + Response.status);
          return Response.json();
      }).then(function(json) {
        alert("????=" +JSON.stringify(json));
      });
 }
```

ポップアップに表示されるJSONは以下の通りです。

```
????={"name":"?? ??","age":"20"}
```

ターミナルで実行する場合は以下のように %Net.HttpRequest クラス を利用します。

```
set req=##class(%Net.HttpRequest).%New()
set req.Server="localhost"
set req.Port=52773  // ??????????????????????
set base64=$system.Encryption.Base64Encode("_system:SYS")
do req.SetHeader("Authorization","Basic "_base64)
do req.SetParam("NAME1", "Yamada")
do req.SetParam("NAME2", "Taro")
do req.SetParam("AGE", "20")
set sts=req.Get("/simple/req1")

set ans=req.HttpResponse.Data.Read()
write ans
```

関連するFAQトピックもご参照ください。

- jQueryでCachéのデータをJSON形式で取得するにはどうすればいいですか?

関連する記事もご参照ください。

- 【はじめてのInterSystems IRIS】セルフラーニングビデオ：アクセス編：IRIS で作成する REST サーバの仕組み

- 【はじめてのInterSystems IRIS】セルフラーニングビデオ：アクセス編：【REST】手動で作成するディスパッチクラス
- 【はじめてのInterSystems IRIS】セルフラーニングビデオ：アクセス編：【REST】APIファーストで作成するRESTディスパッチクラス
- 【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

#HTML #JavaScript #REST API #ヒント・コツ #初心者 #Caché #Ensemble #HealthShare #InterSystems IRIS #InterSystems IRIS for Health

ソースURL: https://jp.community.intersystems.com/post/restjson-%E3%81%AE%E7%B0%A1%E5%8D%98%E3%81%AA%E3%82%B5%E3%83%B3%E3%83%97%E3%83%AB%E3%81%94%E7%B4%B9%E4%BB%8B