記事

Mihoko lijima · 2021年6月1日 6m read

REST/JSON の簡単なサンプルご紹介

これは <u>InterSystems FAQ サイト</u>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、<u>https://github.com/Intersystems-jp/FAQ-REST-SimpleSample</u>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
- <u>(3) 実行</u>

(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 <u>http://localhost:52773/csp/sys/UtilHome.csp</u> 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 > <u>サンプルファイル</u>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)。

インポートするアイテムを選択してください。

 Image: A start of the start of	アイテム	存在	最終更新	
	User.REST.cls	はい	2021-05-21 19:19:01.226051	
	csp/user/sampleJSON.html	はい	2021-05-21 19:40:05.447536	
合計: 2				

クラス定義(User.REST)とHTMLファイル(sampleJSON.html)がインポートされます。

(2) **ウェブアプリケーションパスの設定**

(1) でインポートしたクラス定義(User.REST)を REST ディスパッチクラスに指定したウェブアプリケーションパスを定義します。

管理ポータルを開き、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 > ウェブ・アプリケーション > 「新しいウェブ・アプリケーションの作成」をクリックし、以下の図の設定を行ってください。

(サンプルでは、/simple を REST のベースパスとしています)

Published on InterSystems Developer Community (https://community.intersystems.com)

ウェブ・アプリケー	ション /simple の定義を編集:
アプリケーションを保存	しました。
/一般	アプリケーション・ロール マッチング・ロール
名前	/simple 必須です。(例 /csp/appname)
説明	
ネームスペース	USER USER のデフォルト・アプリケーション: /csp/user ロネームスペースのデフォルト・アプリケーション
アプリケーション有効	
有効	● <u>REST</u> ディスパッチ・クラス User.REST
	必須です。 ○ <u>CSP/ZEN</u> □ アナリティクス ■ 着信 Web サービス □ ログイン CSRF 攻撃を防ぐ
セキュリティの設定	必要なリソース ✓ ID でグループ化 許可された認証方法 □ 認証なし ☑ パスワード □ Kerberos □
セッションの設定	セッションタイムアウト 900 秒 イベントクラス セッションにクッキーを使用する 常時 マ セッションクッキーパス /simple/ マ Session Cookie Scope Stric

- 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。
 「許可さわた初知方法」は「「パスロード」を認定します。スクセス時、スーザタトパスロードの入る
- 「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:<u>system、パスワードは</u> 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:<u>sy</u>stem 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"/>
```

REST/JSON の簡単なサンプルご紹介

Published on InterSystems Developer Community (https://community.intersystems.com)

```
<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 reg1() 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/reg1?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 <u>%Net.HttpRequest クラス</u>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()
```

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。

write ans

 【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

REST/JSON の簡単なサンプルご紹介

Published on InterSystems Developer Community (https://community.intersystems.com)

- <u>【はじめてのInterSystems</u>
- 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
- <u>【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</u>

#HTML #JavaScript #REST API #ヒントとコツ #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