文章
姚 鑫 · 六月 5, 2021 阅读大约需 5 分钟

第七章 Caché JSON %JSON快速参考

第七章 Caché JSON %JSON快速参考

%JSON快速参考

本节提供本章中讨论的%JSON方法、属性和参数的快速参考。

%JSON.Adaptor方法

这些方法提供了从JSON序列化和序列化到JSON的能力。

%JSONExport()

%JSON.Adaptor.%JSONExport()将启用JSON的类序列化为JSON文档，并将其写入当前设备。

```
method %JSONExport(%mappingName As %String = "") as %Status
```

- %mappingName(可选)-要用于导出的映射的名称。基本映射由"" 表示，并且是默认映射。

%JSONExportToStream()

%JSON.Adaptor.%JSONExportToStream()将启用JSON的类序列化为JSON文档并将其写入流。

```
method %JSONExportToStream(ByRef export As %Stream.Object,
   %mappingName As %String = "") as %Status
```

- export - 包含序列化的JSON文档的导出流。
- %mappingName(可选)-要用于导出的映射的名称。基本映射由""表示，并且是默认映射。

%JSONExportToString()

%JSON.Adaptor.%JSONExportToString()将启用JSON的类序列化为JSON文档，并将其作为字符串返回。

```
method %JSONExportToString(ByRef %export As %String,
   %mappingName As %String = "") as %Status
```

- Export-包含序列化的JSON文档的字符串。
- %mappingName可选)-要用于导出的映射的名称。基本映射由""表示，并且是默认映射。

%JSONImport()

%JSON.Adaptor.%JSONImport()将JSON或动态实体输入导入此对象。

```
method %JSONImport(input, %mappingName As %String = "") as %Status
```

- input -JSON可以是字符串或流，也可以是%DynamicAbstractObject的子类。
- %mappingName(可选)-要用于导入的映射的名称。基本映射由""表示，并且是默认映射。

%JSONNew()

%JSON.Adaptor.%JSONNew()获取启用JSON的类的实例。在返回此类的实例之前，可以重写此方法以执行自定义处理(如初始化对象实例)。但是，不应直接从用户代码调用此方法。

```
classmethod %JSONNew(dynamicObject As %DynamicObject,
   containerOref As %RegisteredObject = "") as %RegisteredObject
```

- dynamicObject -具有要分配给新对象的值的动态实体。
- containerOref (可选)-从%JSONImport()调用时的包含对象实例。

%JSON.Adaptor类和属性参数

除非另有说明，否则可以为类或单个属性指定参数。作为类参数，它指定相应属性参数的默认值。作为属性参数，它指定覆盖默认值的值。

%JSONENABLED

启用属性转换方法的生成。

```
parameter %JSONENABLED = 1;
```

- 1-(默认)将生成JSON启用方法。
- 0-方法生成器不会生成Runnable方法。

%JSONFIELDNAME (properties only)

设置要用作JSON内容中字段名的字符串。

```
parameter %JSONFIELDNAME
```

默认情况下，使用属性名称。

%JSONIGNOREINVALIDFIELD

控制对JSON输入中意外字段的处理。

```
parameter %JSONIGNOREINVALIDFIELD = 0;
```

- 0-(默认值)将意外字段视为错误。
- 1-意外字段将被忽略。

%JSONIGNORENULL

指定如何存储字符串属性的空字符串。此参数仅适用于真字符串(由 XSDTYPE = "string" 和 JSONTYPE="string"确定)。

```
parameter %JSONIGNORENULL = 0;
```

- 0-(默认)JSON输入中的空字符串存储为$char(0)，$char(0)作为字符串""写入JSON。JSON输入中缺少的字段始终存储为""，并且根据%JSONNULL参数，""始终输出到JSON。
- 1-空字符串和缺少的JSON字段都作为""输入，而""和$char(0)都作为字段值""输出。

%JSONINCLUDE (properties only)

指定此属性是否包含在JSON输出或输入中。

```
parameter %JSONINCLUDE = "inout"
```

- "inout"(默认)-在输入和输出中都包含。
- "outputonly" -忽略该属性作为输入。
- "inputOnly" -忽略该属性作为输出。
- “ none ”—从不包含该属性。

%JSONNULL

控制未指定属性的处理。

```
parameter %JSONNULL = 0;
```

- 0 -(默认)在导出期间跳过与未指定属性对应的字段。
- 1 -未指定的属性作为空值导出。

%JSONREFERENCE

指定如何将对象引用投影到JSON字段。

```
parameter %JSONREFERENCE = "OBJECT";
```

- "OBJECT" -(默认)被引用类的属性用来表示被引用的对象。
- “ ID ”-持久或串行类的ID用于表示引用。
- “ OID ”——持久类或串行类的OID用于表示引用。
  oid以classname,id的形式投射到JSON中。
  -"GUID" -持久化类的GUID用来表示引用。

%JSON.ForMatter方法和属性

%JSON.ForMatter类可用于格式化%DynamicAbstractObject子类的JSON字符串、流或对象。

Format()

%JSON.Formatter.Format()使用指定的缩进格式化JSON文档并将其写入当前设备。

```
method Format(input) as %Status
```

- input -JSON可以是字符串或流，也可以是%DynamicAbstractObject的子类。

FormatToStream()

%JSON.Formatter.FormatToStream()使用指定的缩进格式化JSON文档并将其写入流。

```
method FormatToStream(input, ByRef export As %Stream.Object) as %Status
```

- input -JSON可以是字符串或流，也可以是%DynamicAbstractObject的子类。
- export -格式化的JSON流。

FormatToString()

%JSON.Formatter.FormatToString()使用指定的缩进格式化JSON文档并将其写入字符串，或将启用JSON的类序列化为JSON文档并将其作为字符串返回。

```
method FormatToString(input, ByRef export As %String = "") as %Status
```

- input-JSON可以是字符串或流，也可以是%DynamicAbstractObject的子类。
- export (可选)-格式化的JSON流。

Indent

%JSON.Formatter.Indent属性指定是否应缩进JSON输出。默认为true。

```
property Indent as %Boolean [ InitialExpression = 1 ];
```

IndentChars

%JSON.Formatter.IndentChars属性指定在启用缩进时用于每个缩进级别的字符序列。默认为一个空格。

```
property IndentChars as %String [ InitialExpression = " " ];
```

LineTerminator

%JSON.Formatter.LineTerminator属性指定缩进时终止每行的字符序列。默认为$char(13，10)。

```
property LineTerminator as %String [ InitialExpression = $char(13,10) ];
```

```
/// d ##class(PHA.TEST.Xml).Obj2FormatterJson()
ClassMethod Obj2FormatterJson()
{
    s event = ##class(Model.Event).%New()
    s event.Name = "yx"
```

```
    s location = ##class(Model.Location).%New()
    s location.City = "tianjin"
    s location.Country = "china"
    s event.Location = location
    d event.%JSONExportToString(.jsonEvent)
    s formatter = ##class(%JSON.Formatter).%New()
    s formatter.Indent = 1
    s formatter.IndentChars = " - "
    //s formatter.LineTerminator = "!"
    d formatter.Format(jsonEvent)
}


DHC-APP>d ##class(PHA.TEST.Xml).Obj2FormatterJson()
{
 - "Name":"yx",
 - "Location":{
 -  - "City":"tianjin",
 -  - "Country":"china"
 - }
}
DHC-APP>d ##class(PHA.TEST.Xml).Obj2FormatterJson()
{! - "Name":"yx",! - "Location":{! -  - "City":"tianjin",! -  - "Country":"china"! -
}!}
```

#JSON #Caché #InterSystems IRIS #InterSystems IRIS for Health

---

源URL:https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%83%E7%AB%A0-cach%C3%A9-json-json%E5%BF%AB%E9%80%9F%E5%8F%82%E8%80%83