

文章

王喆 · 九月 18 阅读大约需 4 分钟

## Rest实现Post、Get、Put、Delete几种操作方式

这是一篇笔记:

### 说明

Restful是一种基于Http的网络应用程序的设计风格 and 开发方式，可以使用xml格式或者Json格式定义。

RESTFUL特点包括：

- 1、每一个URI代表1种资源；
- 2、客户端使用GET、POST、PUT、DELETE4

个表示操作方式的动词对服务端资源进行操作：GET用来获取资源，POST用来新建资源（也可以用于更新资源），PUT用来更新资源，DELETE用来删除资源；

- 3、通过操作资源的表现形式来操作资源；
- 4、资源的表现形式是XML或者HTML；

5

、客户端与服务端之间的交互在请求之间是无状态的，从客户端到服务端的每个请求都必须包含理解请求所必需的信息。

### IRIS 中实现

```

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>

    <Route Url="/item" Method="GET" Call="item1" Cors="true"/>
    <Route Url="/item/:id" Method="GET" Call="item2" Cors="true"/>
    <Route Url="/items" Method="GET" Call="item3" Cors="true"/>

</Routes>
}

/// ??GET????
ClassMethod item1() As %Status
{
    Set sc = $$$OK
    s resp = {}
    d resp.%Set("Code",200)
    d resp.%Set("Msg","ok")
    s resp = resp.%ToJSON()
    w resp
}

```

```

    Return sc
}

/// ?????GET????
ClassMethod item2(id As %Integer) As %Status
{
    Set sc = $$$OK
    s resp = {}
    d resp.%Set("Code",200)
    d resp.%Set("Msg","ID?:"_id)
    s resp = resp.%ToJSON()
    w resp
    Return sc
}

/// n????GET????
ClassMethod item3() As %Status
{
    s sc = $$$OK
    s id = $GET(%request.Data("id",1),"")
    s dataName = $GET(%request.Data("dataName",1),"")
    s valueName = $GET(%request.Data("valueName",1),"")
    s resp = {}
    d resp.%Set("code",200)
    d resp.%Set("id",id)
    d resp.%Set("dataName",dataName)
    d resp.%Set("valueName",valueName)
    s resp = resp.%ToJSON()
    w resp
    Return sc
}

```

## POST :

```

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
    <Route Url="/item" Method="POST" Call="item1" Cors="true"/>
</Routes>
}

ClassMethod item1() As %Status
{
    #Dim JsonUtils as BKIP.Rest.Utils.JsonAndObj = ##class
(BKIP.Rest.Utils.JsonAndObj).%New()
    #Dim XmlUtils as BKIP.Rest.Utils.XmlAndObj = ##class
(BKIP.Rest.Utils.XmlAndObj).%New()
    s ContentType = ##class(%String).LogicalToDisplay($Get(
%request.CgiEnvs("CONTENT_TYPE")))
}

```

```

Set sc = $$$OK
Try {
  s message = ##class(%String).LogicalToDisplay(%request.Content.Read())
  If ContentType="application/xml" {
    s tSC = XmlUtils.Xml2Object(message, "Request",
"BKIP.Rest.MSG.item", .req) $$$ThrowOnError(tSC)
  }Else{
    s tSC = JsonUtils.Json2Object(message,
"BKIP.Rest.MSG.item", .req) $$$ThrowOnError(tSC)
  }
  s req.valueName="?????1"
  s tSC = JsonUtils.ObjectJson(req, .resp)
}
Catch ex {
  #; w "?????"
  s resp = {}
  d resp.%Set("Code", -1)
  d resp.%Set("Msg", ex.DisplayString())
  s resp = resp.%ToJSON()
}
w resp
Return sc
}

```

PUT :

```

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]

{
<Routes>
  <Route Url="/item" Method="PUT" Call="item1" Cors="true"/>
</Routes>
}
ClassMethod item1() As %Status
{
  #Dim JsonUtils as BKIP.Rest.Utils.JsonAndObj = ##class
(BKIP.Rest.Utils.JsonAndObj).%New()
  #Dim XmlUtils as BKIP.Rest.Utils.XmlAndObj = ##class
(BKIP.Rest.Utils.XmlAndObj).%New()
  s ContentType = ##class(%String).LogicalToDisplay($Get(
%request.CgiEnvs("CONTENT_TYPE")))
  Set sc = $$$OK
  Try {
    s message = ##class(%String).LogicalToDisplay(%request.Content.Read())
    If ContentType="application/xml" {
      s tSC = XmlUtils.Xml2Object(message, "Request",
"BKIP.Rest.MSG.item", .req) $$$ThrowOnError(tSC)
    }Else{
      s tSC = JsonUtils.Json2Object(message,
"BKIP.Rest.MSG.item", .req) $$$ThrowOnError(tSC)
    }
    s req.valueName="?????1"
    s tSC = JsonUtils.ObjectJson(req, .resp)
  }
}

```

```

    }
    Catch ex {
        #; w "?????"
        s resp = {}
        d resp.%Set("Code",-1)
        d resp.%Set("Msg",ex.DisplayString())
        s resp = resp.%ToJSON()
    }
    w resp
    Return sc
}

```

## DELETE :

```

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
    <Route Url="/item" Method="DELETE" Call="item1" Cors="true"/>
</Routes>
}

```

```

ClassMethod item1() As %Status
{
    #Dim JsonUtils as BKIP.Rest.Utils.JsonAndObj = ##class
(BKIP.Rest.Utils.JsonAndObj).%New()
    #Dim XmlUtils as BKIP.Rest.Utils.XmlAndObj = ##class
(BKIP.Rest.Utils.XmlAndObj).%New()
    s ContentType = ##class(%String).LogicalToDisplay($Get(
%request.CgiEnvs("CONTENT_TYPE")))

    Set sc = $$$OK
    Try {
        s message = ##class(%String).LogicalToDisplay(%request.Content.Read())
        If ContentType="application/xml"{
            s tSC = XmlUtils.Xml2Object(message,"Request",
"BKIP.Rest.MSG.item",.req) $$$ThrowOnError(tSC)
        }Else{
            s tSC = JsonUtils.Json2Object(message,
"BKIP.Rest.MSG.item",.req) $$$ThrowOnError(tSC)
        }
        s req.valueName="?????1"
        s tSC = JsonUtils.ObjectJson(req,.resp)
    }
    Catch ex {
        #; w "?????"
        s resp = {}
        d resp.%Set("Code",-1)
        d resp.%Set("Msg",ex.DisplayString())
        s resp = resp.%ToJSON()
    }
    w resp
    Return sc
}

```

Rest**实现**Post、Get、Put、Delete**几种操作方式**

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

---

[#API](#) [#ObjectScript](#) [#REST API](#) [#HealthShare](#) [#VSCode](#)

---

**源**

URL:

<https://cn.community.intersystems.com/post/rest%E5%AE%9E%E7%8E%B0post%E3%80%81get%E3%80%81put%E3%80%81delete%E5%87%A0%E7%A7%8D%E6%93%8D%E4%BD%9C%E6%96%B9%E5%BC%8F>