

文章

[Qiao Peng](#) · 三月 28, 2021 阅读大约需 9 分钟

InterSystems IRIS医疗行业版创建FHIR服务器

如果您正打算学习FHIR、或者正在基于FHIR开发，最佳的学习和开发环境需要一个完备的FHIR服务器，帮助您理解FHIR标准和调试FHIR应用。使用InterSystems IRIS医疗行业版可以快速配置出FHIR服务器和FHIR资源仓库。这篇文章介绍如何在5分钟内在InterSystems IRIS医疗行业版上配置FHIR服务器和FHIR资源仓库；同时对初学者介绍FHIR测试数据生成与加载的方法，和基本的FHIR REST API操作。

软件准备

1. InterSystems IRIS医疗行业版

您可以使用已有的2020.1及以上的InterSystems IRIS医疗行业版。如果您还没有，那么就在社区里下载一个最新的、免费的[社区版](#)吧。

2. 测试用的FHIR数据

如果您正在做FHIR开发，应该有很多FHIR测试数据。如果您正在学习FHIR，那么[Synthea](#)是一个不错的FHIR测试数据生成器选项。

3. REST测试工具

您或许已经有趁手的REST测试工具了。如果还没有，POSTMAN是个不错的选择。我的例子里用到的就是POSTMAN。

FHIR服务器配置

1. 安装InterSystems IRIS医疗行业版或社区版

如果您还没有安装过InterSystems IRIS医疗行业版或社区版，社区里有很多很好的介绍文章。

2. 配置FHIR服务器

2.1 创建一个FHIR 服务器命名空间

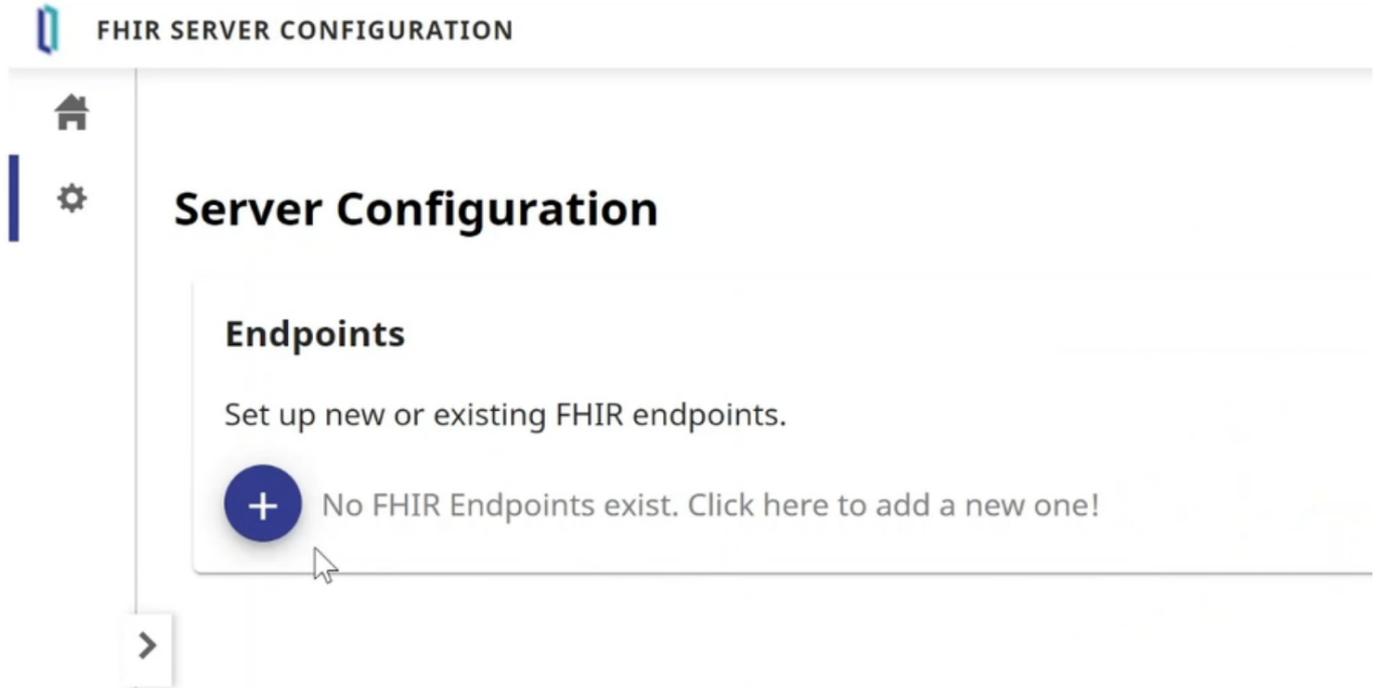
在InterSystems IRIS医疗行业版或社区版上，需要创建一个FHIR服务器命名空间。

HS.HC.Util.Installer类的方法InstallFoundation可以创建这样的命名空间，例如，我们创建一个名为FHIRSERVER的命名空间：

```
HSLIB>D ##class(HS.HC.Util.Installer).InstallFoundation("FHIRSERVER")
```

2.2 为这个命名空间配置FHIR访问端点 (FHIR endpoint)

进入管理门户，切换到FHIRSERVER命名空间，然后Health > FHIR Configuration，然后点击“Server Configuration”。这里会列出已有的FHIR服务器的访问端点，并可以配置新的端点

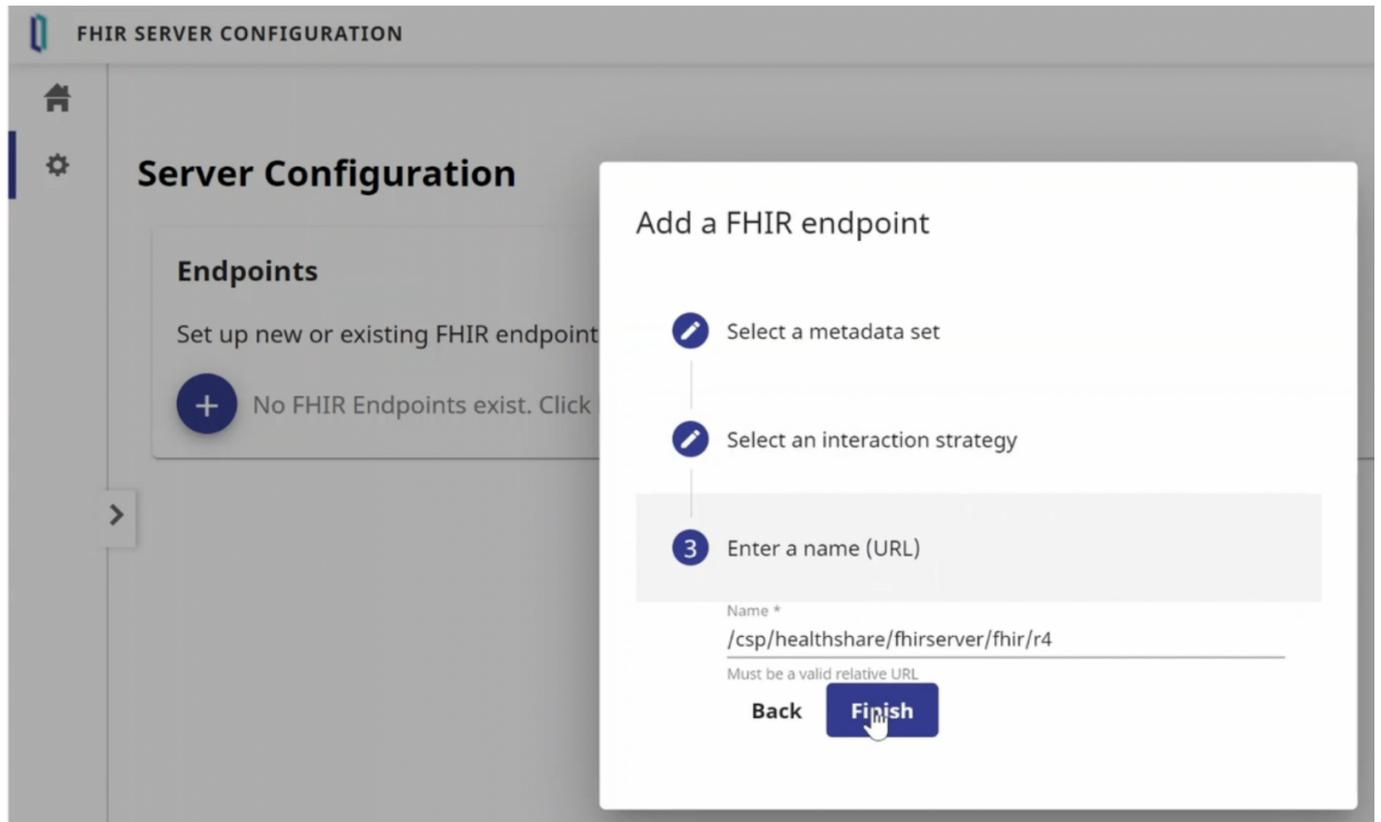


这里点击+号，在弹出的配置页面中进行设置，其配置项是：

选中支持的FHIR版本 (Select a metadata set): 如果要支持最新的FHIR版本4，选中HL7V40，就是FHIR R4；

选中交换策略 (Select an interaction strategy): 默认情况下只会有一个选项 - HS.FHIRServer.Storage.Json.InteractionsStrategy，它会将InterSystems IRIS同时配置为FHIR资源仓库。

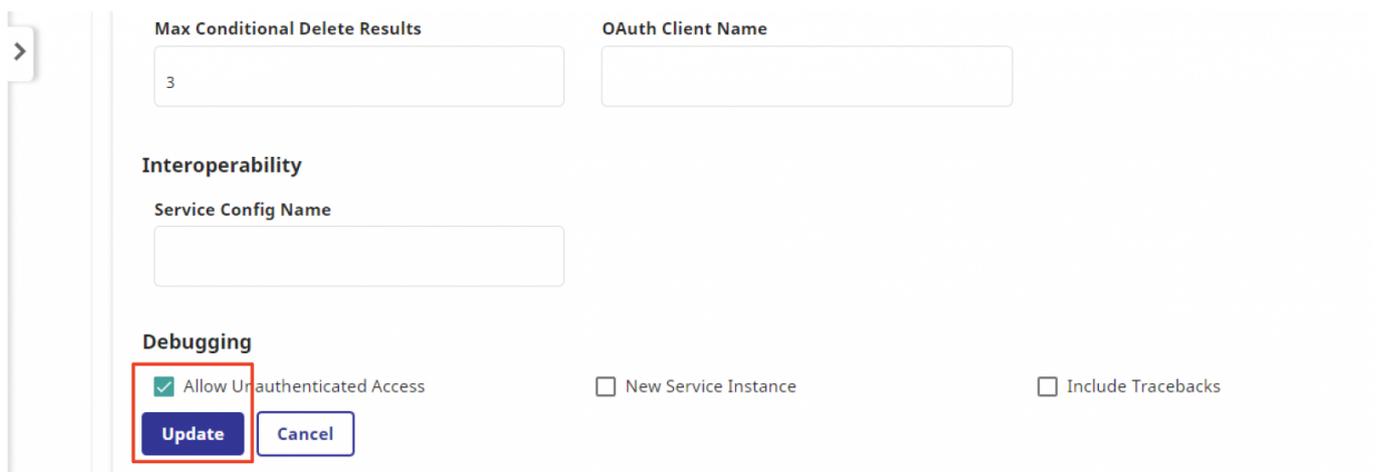
设置FHIR服务器REST服务端点 (Enter a name(URL)): 例如/csp/healthshare/fhirserver/fhir/r4



然后点击Finish，让它完成后台的配置，这可能会花费1分钟。

2.3 修改这个FHIR服务端点

因为我们是想利用它来学习和测试FHIR，所以我们先跳过访问认证部分。在端点列表中选中刚才创建的/csp/healthshare/fhirserver/fhir/r4，滚动到最下方，找到Edit按钮，并选中“Debugging”下的“Allow Unauthenticated Access”，让FHIR服务器接受匿名访问。然后点击Update进行更新。



如果您按上述步骤执行完成，并且没有错误，那么FHIR服务器和FHIR资源仓库已经就绪了。看看表，您用了多久完成FHIR服务器的创建。

下面测试一下FHIR服务器是否工作正常。

测试FHIR服务器

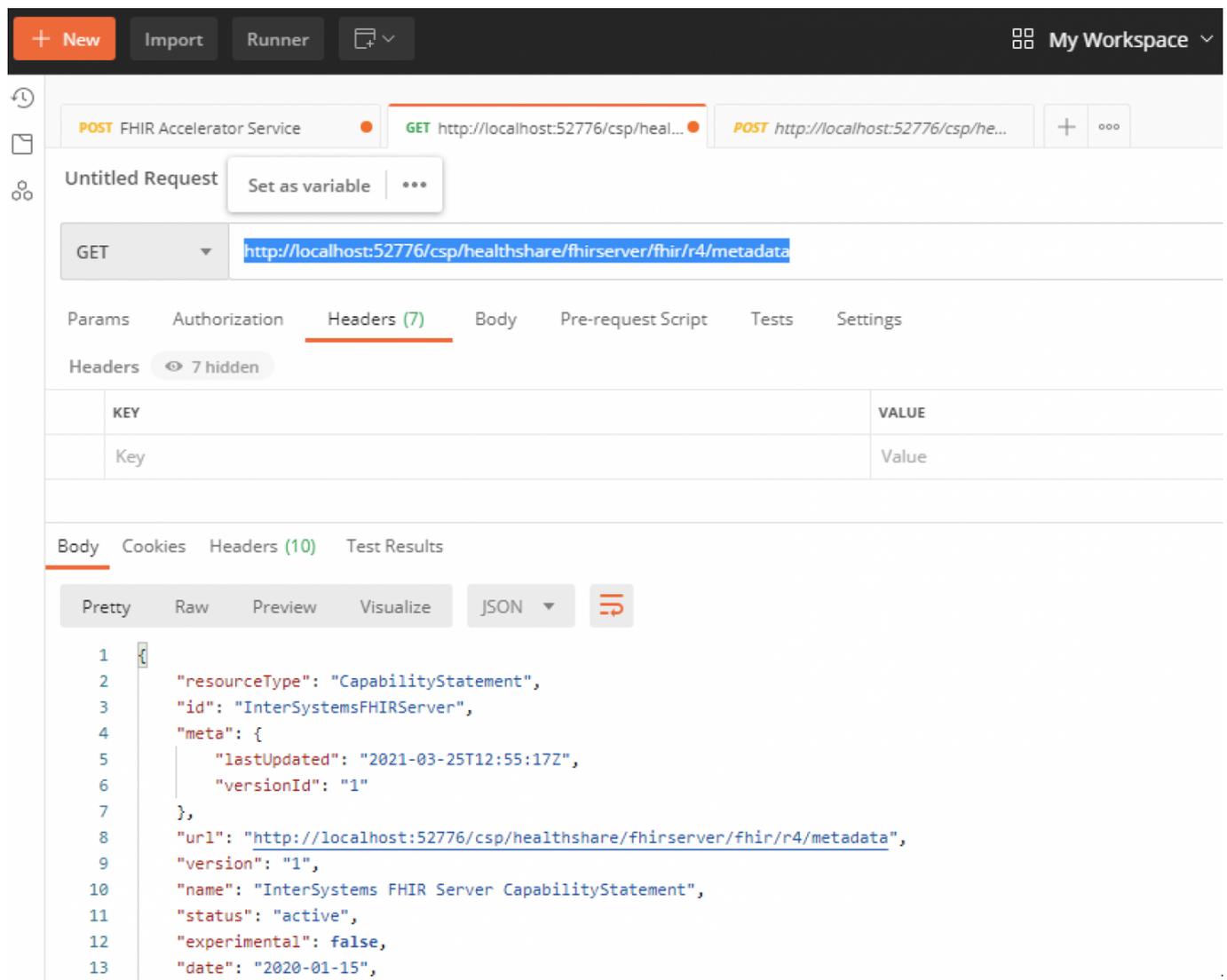
即便您还没有任何FHIR测试数据，也可以先看看这个FHIR服务器的能力声明。

通过POSTMAN，使用GET方法，进行REST调用：

<http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/metadata>。这里localhost:52776是我的测试InterSystems IRIS的服务器地址和Web服务端口号，您换成自己的就行。如果您不知道，那么用您访问InterSystems IRIS管理门户的地址和端口号。

因为我们在上一步配置时，允许匿名访问REST

API，所以POSTMAN无需设置任何登陆账户，非常方便测试。正常情况下，您应该看到和我一样的结果：



The screenshot shows the Postman interface with a GET request to `http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/metadata`. The response is a JSON object:

```
1 {
2   "resourceType": "CapabilityStatement",
3   "id": "InterSystemsFHIRServer",
4   "meta": {
5     "lastUpdated": "2021-03-25T12:55:17Z",
6     "versionId": "1"
7   },
8   "url": "http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/metadata",
9   "version": "1",
10  "name": "InterSystems FHIR Server CapabilityStatement",
11  "status": "active",
12  "experimental": false,
13  "date": "2020-01-15",
```

您的FHIR服务器已经正常工作了，如果还没有测试数据，是时候加载一些FHIR数据进来了。

产生FHIR数据

如果您有自己的FHIR测试数据，请跳过此步，查看“加载FHIR数据”。

如果您还没有自己的FHIR数据

，试试Synthea，它能方便的产生大量FHIR测试数据。[Synthea](#)

有清晰的文档，告诉我们如何产生测试数据。（注：如果您是FHIR初学者，建议先少量创建FHIR测试数据文件，例如先创建5个）

通常它会产生3类JSON文件：

医院信息：hospitalInformation*****.json，每批次只会产生一个医院信息文件。

医生信息：practitionerInformation*****.json，每批次只会产生一个医生信息文件。

患者信息：<患者姓名>*****.json，每批次会根据设置产生一个或多个患者信息文件。

这些FHIR文件都是Bundle资源，Bundle里面会有很多FHIR资源数据。其中医院信息文件和医生信息文件的Bundle类型为batch，而患者信息的Bundle是transaction。

而Synthea产生的患者信息Bundle中的资源使用literal

reference来引用医院/地点和医生信息，但它产生的literal reference信息不完整，例如：

```
"serviceProvider": {  
  
  "reference":  
  "Organization?identifier=https://github.com/synthetichealth/synthea|731e0f3a-075d-37ab-9ba7-fbf1ab2f45e9",  
  
  "display": "THE GENERAL HOSPITAL CORPORATION"  
  
}
```

需要将reference改为完整的地址，例如：

```
"serviceProvider": {  
  
  "reference":  
  "http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Organization?identifier=https://github.com/synthetichealth/synthea|731e0f3a-075d-37ab-9ba7-fbf1ab2f45e9",  
  
  "display": "THE GENERAL HOSPITAL CORPORATION"  
  
}
```

也就是增加FHIR服务器的端点。

需要替换/补齐的引用分别是：Location、Organization和Practitioner。

您可以通过文本编辑器进行全局替换，将FHIR服务器的端点信息补充到reference里。如果您产生了很多FHIR数据文件，当然可以用您熟悉的语言写几行代码做这个补充替换。

这里我也附上使用IRIS的Object

Script做补充替换的代码示例，它会对指定目录下的所有json文件进行扫描和补充替换，供您参考：

```
Class Demo.FHIRTools Extends %RegisteredObject  
{
```

```

/// ??FHIR?????
/// pFilePath?FHIR json????
/// pFHIREndpoint?FHIR REST?????
ClassMethod CorrectSyntheaFiles(
    pFilePath As %String = "C:\Temp\Synthea\output\fhir",
    pFHIREndpoint As %String = "http://localhost:52776/csp/healthshare/fhirserver/fhir/r4") As %Status
{
    Set tSC = $$$OK
    // ??????json??
    Set tSM = ##class(%SQL.Statement).%New()
    Set tSC = tSM.%PrepareClassQuery("%File", "FileSet")
    If $$$ISERR(tSC)
    {
        Do $system.OBJ.DisplayError(tSC)
        Return tSC
    }
    Set tRS = tSM.%Execute(pFilePath, "*.json", "Name")
    While tRS.%Next()
    {
        // ??????
        Set tFileName = tRS.%Get("Name"),tTempFileName=$Replace(tFileName, ".json"
, "temp.json")
        Set tFile=##class(%Stream.FileCharacter).%New()
        Set tTempFile = ##class(%Stream.FileCharacter).%New()
        Set tSC=tTempFile.LinkToFile(tTempFileName)
        Set tSC=tFile.LinkToFile(tFileName)
        // ??????????????????????
        While 'tFile.AtEnd
        {
            Set tLine=tFile.ReadLine()
            // ?????
            For tKeyword="Location?identifier=", "Organization?identifier=", "Practitioner?identifier="
            {
                Set tLine = $Replace(tLine,tKeyword,pFHIREndpoint_"/"_tKeyword)
            }
            Set tSC = tTempFile.WriteLine(tLine)
        }
        Do tTempFile.%Save()
        Do tFile.%Close(), tTempFile.%Close()
        Kill tFile,tTempFile
        // ??????????????????????
        Set tSC = ##class(%File).Delete(tFileName, .tRtn)
        Set tSC = ##class(%File).Rename(tTempFileName, tFileName, .tRtn)
    }
    Return tSC
}
}
}

```

加载FHIR数据

当然可以通过FHIR REST API来加载数据，不过，InterSystems IRIS提供了后台的API，可以快速加载特定目录下的大量的FHIR数据文件。

这里我们用后台API：HS.FHIRServer.Tools.DataLoader的

SubmitResourceFiles方法进行大量FHIR文件加载：

```
FHIRSERVER>Set sc=##class(HS.FHIRServer.Tools.DataLoader).SubmitResourceFiles("C:\Temp\Synthea\upload", "FHIRSERVER", "/csp/healthshare/fhirserver/fhir/r4")
```

其中第一个入参是需要加载的文件目录；第二个参数是加载的FHIR服务器类型，这里用FHIRSERVER；第三个参数是FHIR REST服务端点。

这个方法同样适用于加载NDJSON格式的FHIR数据文件。

如果您是通过Synthea产生的测试FHIR JSON文件，那么应该先加载

医院信息文件(hospitalInformation*****.json)和医生信息文件(practitionerInformation*****.json)，以确保患者数据加载时，Location、Organization和Practitioner的引用能正确关联到FHIR资源仓库中已保存的对应资源上。

注意：2020.*版本中有一个bug，需要将医院信息文件(hospitalInformation*****.json)和医生信息文件(practitionerInformation*****.json)里的"type": "batch"改为"type": "transaction"。这个bug在2021.1版本中已经得到修复，我已经验证过，但如果您在使用2020的版本，请记得提前修改。

加载完医院信息文件(hospitalInformation*****.json)和医生信息文件(practitionerInformation*****.json)后，再执行相同的方法加载所有患者数据json文件。

使用FHIR 服务器

有了测试数据，现在您可以使用FHIR服务器进行学习或开发测试了。使用POSTMAN，您可以调用FHIR服务器的REST API，查询、更改、提交FHIR资源数据。

如果您是初学者，对FHIR的REST API不熟悉，建议您从[FHIR Cheat Sheet](#)开始，它是一个FHIR标准的汇总单页，里面告诉大家如何使用REST API进行操作。

例如：

1. 查询已经加载到FHIR资源仓库中的患者：

使用GET 方法调用REST：

```
http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Patient
```

2. 使用查询参数，查询名为Pedro316的患者：

使用GET 方法调用REST：

```
http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Patient?given=Pedro316
```

或使用POST 方法调用REST：

```
http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Patient/_search?given=Pedro316
```

3. 更复杂的查询参数用例，查询拥有观察项目编码为8302-2（体重）的患者：

使用GET 方法调用REST :

`http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Patient?_has:Observation:patient:code=8302-2`

4. 使用operation : \$everything 来获取特定患者的所有相关FHIR资源 (示例是id为95的患者) :

使用GET 方法调用REST :

`http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Patient/95/$everything`

5. 加载FHIR Bundle资源 :

使用POST 方法调用REST :

`http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/`

, 并在Body中直接贴入FHIR资源数据到raw , 或选择FHIR资源数据文件到binary。

注意 : 如果您是想让FHIR服务器解析

Bundle中的FHIR资源 , 应该用<http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/> , 而不是

<http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/Bundle>

。后者把POST的FHIR数据按Bundle进行处理 , 并不会解析其中包含的FHIR资源内容并逐一保存 !

The screenshot shows a REST client interface with the following details:

- Method: POST
- URL: `http://localhost:52773/csp/healthshare/fhirserver/fhir/r4/`
- Active tab: Headers (9)
- Header list:

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json+fhir
Key	Value

注意 : 记得根据FHIR数据的格式设置Content-Type为application/json+fhir

或 application/xml+fhir , 然后将数据贴在Body里 :

The screenshot shows a REST client interface with the following details:

- Method: POST
- URL: `http://localhost:52776/csp/healthshare/fhirserver/fhir/r4/`
- Body:

```
1 {
2   "resourceType": "Bundle",
3   "type": "transaction",
4   "entry": [
5     {
6       "fullUrl": "urn:uuid:f2a59bda-88aa-73ab-8126-5f6e69158eca",
7       "resource": {
8         "resourceType": "Patient",
9         "id": "f2a59bda-88aa-73ab-8126-5f6e69158eca",
10        "meta": {
11          "profile": [ "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient" ]
12        },
13        "text": {
14          "status": "generated",
15          "div": "<div xmlns=\\"http://www.w3.org/1999/xhtml\">Generated by <a href=\\"https://github.com/syntheti
16              1616739000644</div>"
17        },
18        "extension": [ {
19          "url": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-race",
20          "extension": [ {
21            "url": "ombCategory",
```

好了，开启您的FHIR之旅吧 :)

后续我还会更新InterSystems IRIS医疗版作为FHIR服务器和FHIR资源仓库的更多内容，欢迎关注。

[#InterSystems IRIS for Health](#)

源

URL: <https://cn.community.intersystems.com/post/intersystems-iris%E5%8C%BB%E7%96%97%E8%A1%8C%E4%B8%9A%E7%89%88%E5%88%9B%E5%BB%BAfhir%E6%9C%8D%E5%8A%A1%E5%99%A8>