

Article

[Lorenzo Scalese](#) · Sept 23, 2022 4m de lecture

[Open Exchange](#)

Création de Patient et d'Observations de Patient en utilisant l'application iris-fhir-client

Bonjour la communauté,

Cet article montre comment créer des ressources pour le patient et l'observation du patient en utilisant l'application [iris-fhir-client](#).

The screenshot displays the IRIS FHIR Client web application. The top navigation bar includes 'Dashboard', 'Management Portal', 'FHIR Official website', and 'FHIR Client Docs'. The main header shows 'InterSystem FHIR Accelerator Service' and the URL 'https://fhir.83z8498j30i6.static-test-account.iscloud.io'. A dropdown menu for 'FHIR Server' is open, showing options: 'InterSystem FHIR Accelerator Service', 'InterSystem FHIR Accelerator Service', 'SmartHealthIT Open FHIR Server', and 'INTERSYSTEMS FHIR Server'. The dashboard features four cards: '27 Patients', '100 Observations', '100 Practitioners', and '100 Encounters'. Below the dashboard are two tables: 'Patients' and 'Registered FHIR Servers'. The 'Patients' table has columns for ID, Family Name, Given Name, DOB, and Gender. The 'Registered FHIR Servers' table has columns for ID, Server Name, and Status. The system tray at the bottom shows the date and time as 1:38 PM 5/30/2022.

Je recommande de lire [mon premier article](#) sur cette application et de regarder [le Video Youtube](#) avant de continuer.

Donc commençons

1-Création d'une Ressource Patient

La fonction CreatePatient() de dc.FhirClient ci-dessous peut être utilisée pour créer une Ressource Patient

```
ClassMethod CreatePatient(givenName As %String, familyName As %String, birthDate As %String,gender As %String)
```

La fonction nécessite giveName, failyName, birthDate et sex pour créer la Resource Patient
La commande ci-dessous créera le Patient

```
do ##class(dc.FhirClient).CreatePatient("PatientGN","PatientFN","2000-06-01","male")
```

Voici la fonction python dans le fichier irisfhirclient.py qui va créer le patient

```
import json
from fhirpy import SyncFHIRClient
from tabulate import tabulate
from fhirpy.base.searchset import Raw
import requests

def CreatePatient(givenName,familyName,birthDate,gender,url,api_key):
    headers = {"Content-Type":contentType,"x-api-key":api_key}
    client = SyncFHIRClient(url = url, extra_headers=headers)

    patient = client.resource("Patient")
    patient['name'] = [
        {
            'given': [givenName],
            'family': familyName,
            'use': 'official'
        }
    ]

    patient['birthDate'] = birthDate
    patient['gender'] = gender
    try:
        patient.save()
    except Exception as e:
        print("Error while creating Patient:" +str(e))
        return
    print("Patient Created Successfully")
```

2- Création d'une ressource pour l'observation des patients

Créons une Observation sur notre ressource patient nouvellement créée

La fonction CreateObservation() de dc.FhirClient ci-dessous peut être utilisée pour la Création des observations de patients

```
ClassMethod CreateObservation(patientId As %String, loincCode As %String, ObrCategory
As %String, ObrValue As %Integer, ObrUOM As %String, effectiveDate As %String)
```

Paramètres

- patientId est l'identifiant du patient
- LioncCode est le Code Lionc, les détails sont présentés [ici](#)
- ObrCategory est une catégorie d'observation, les détails sont disponibles [ici](#)
- ObrValue est la valeur d'observation
- ObrUOM est l'unité d'observation
- EffectiveDate

La commande ci-dessous créera l'observation des Signes Vitaux du patient

```
do ##class(dc.FhirClient).CreateObservation("8111","8310-5","vital-
```

```
signs",96.8,"degF","2022-01-22")
```

Énumérons les observations des patients

```
do ##class(dc.FhirClient).GetPatientResources("Observation","8111")
```

Voici la fonction python dans le fichier irisfhirclient.py qui va créer le patient

```
import json
from fhirpy import SyncFHIRClient
from tabulate import tabulate
from fhirpy.base.searchset import Raw
import requests

#Function to create Patient Observation
def CreateObservation(patientId,loincCode,ObrCategory,ObrValue,ObrUOM,effectiveDate,url,api_key):
    headers = {"Content-Type":contentType,"x-api-key":api_key}
    client = SyncFHIRClient(url = url, extra_headers=headers)
    observation = client.resource(
        'Observation',
        status='preliminary',
        category=[{
            'coding': [{
                'system': 'http://hl7.org/fhir/observation-category',
                'code': ObrCategory
            }]
        }],
        code={
            'coding': [{
                'system': 'http://loinc.org',
                'code': loincCode
            }]
        })
    observation['effectiveDateTime'] = effectiveDate

    observation['valueQuantity'] = {
        'system': 'http://unitsofmeasure.org',
        'value': ObrValue,
        'code': ObrUOM
    }

    #find the patient
    patient = client.resources('Patient').search(_id=patientId).first()
    observation['subject'] = patient.to_reference()

    try:
        observation.save()
    except Exception as e:
        print("Error while creating observation :"+ str(e))
        return
    print("Patient Observation Created Successfully")
```

C'est comme ça

Si vous avez trouvé cette application utile, merci de voter pour mon application.

Merci!

[#Embedded Python](#) [#FHIR](#) [#InterSystems](#) [IRIS for Health](#)
[Voir l'application sur InterSystems Open Exchange](#)

URL de la
source: <https://fr.community.intersystems.com/post/cr%C3%A9ation-de-patient-et-dobservations-de-patient-en-utilisant-lapplication-iris-fhir-client>