

Python für ObjectScript Entwickler

Hands-on Aufgaben

1 Einleitung

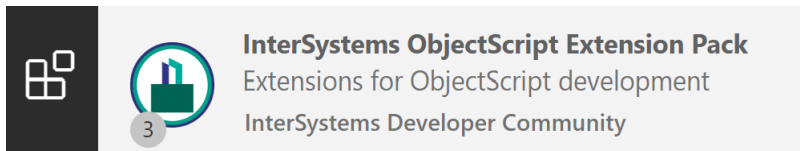
InterSystems IRIS bietet mit Embedded Python neben ObjectScript eine weitere serverseitige Sprache an. Die folgenden Aufgaben bieten einige praktische Beispiele für den Einsatz von Python und Python Bibliotheken im Zusammenspiel mit ObjectScript. Wir verwenden dafür eine gemeinsame IRIS Instanz, mit der Sie sich von Ihrem Laptop aus verbinden werden. Die Voraussetzungen dafür werden im folgenden Abschnitt erläutert.

2 Voraussetzungen

2.1 Visual Studio Code


Falls noch nicht geschehen, installieren Sie bitte die aktuelle Version von Visual Studio Code auf Ihrem Laptop, erhältlich unter <https://code.visualstudio.com/download>.

Anschließend installieren Sie bitte das „InterSystems ObjectScript Extension Pack“. Dieses Paket umfasst drei Erweiterungen. Es ist leicht zu finden, indem man im Suchfeld des Extension Marketplace den Begriff „InterSystems“ eingibt.



2.2 Verbindung zum Server

2.2.1 IRIS-Instanz

Verbinden Sie sich mit der IRIS-Instanz, indem Sie in VSCode links das InterSystems-Symbol  auswählen und unter SERVERS auf das +-Symbol klicken. Bitte füllen Sie die Felder wie folgt aus:

Name: iris-symposium

Optional description: Leer lassen



Hostname: DEP7780DEMOPC1

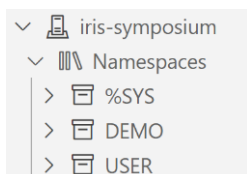
Port: 80

Optional path prefix: /iris

Username: demo

Connection type: http

Gehen Sie wieder in das InterSystems-Menü  und klicken Sie unter „All Servers“ auf die soeben erstellte Verbindung. Bestätigen Sie die Anmeldeanfrage mit „Zulassen“ und geben Sie als Passwort „demo“ ein. Klicken Sie auf den Schlüssel-Button  oben rechts, um das Kennwort für zukünftige Aufrufe zu speichern. Wenn die Verbindung funktioniert hat, sollten Sie eine Liste der vorhandenen Namespaces sehen:



2.2.2 Geteilte Ordner

In den Übungen werden wir auf Verzeichnisse und Dateien auf dem Server zugreifen. Bitte verbinden Sie sich von Ihrem Laptop aus mit den geteilten Ordnern. Unter Windows verwenden Sie z.B. die Tastenkombination Win-R und geben \\DEP7780DEMOPC1 ein gefolgt von der Return-Taste.

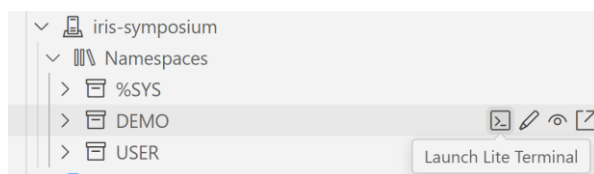
Benutzername: DEP7780DEMOPC1\demo

Kennwort: demo

Nun sollten Sie Zugriff auf zwei Ordner haben – python und symposium. Bitte legen Sie im Ordner symposium/aufgaben einen Unterordner an. Als Namen verwenden Sie den Anfangsbuchstaben Ihres Vornamens gefolgt vom Nachnamen, z.B. für Stephan Mohr den Ordner smohr.

2.3 Lite Terminal

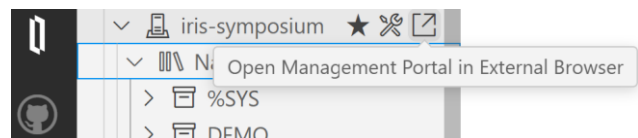
Im Rahmen der Übungen werden wir das IRIS-Terminal nutzen. Zum Öffnen eines Terminalfensters gehen Sie in das InterSystems-Menü von VSCode und klicken Sie in der Zeile des DEMO Namespaces auf das Terminal-Symbol. Daraufhin öffnet sich das IRIS-Terminal innerhalb von VSCode.



3 Python Installation und Konfiguration

Mit dem IRIS 2024.2 Installer wird keine Python Version mehr mitgeliefert. Da Windows standardmäßig kein Python enthält, muss man es selbst installieren. Einige Betriebssysteme unterstützen das neue **Flexible Python Runtime Feature** von InterSystems. Das bedeutet, man kann die Version, die für Embedded Python verwendet wird, nun frei wählen. Auf dem Server für unsere Übungen haben wir bereits Python 3.11 installiert und in IRIS konfiguriert. Hierfür wurde das Installationspaket von <https://www.python.org> verwendet und für „Alle Benutzer“ installiert.

Überprüfen Sie die Python-Konfiguration, indem Sie sich mit dem System Management Portal verbinden. Dazu klicken Sie im InterSystems-Menü von VSCode in der Zeile "iris-symposium" auf das Symbol ganz rechts oder tippen Sie die folgende URL im Webbrowser selbst ein (user/pw:



demo/demo) <http://dep7780demopc1/iris/csp/sys/UtilHome.csp>

Gehen Sie zu Systemadministration->Konfiguration->Zusätzliche Einstellungen->Erweiterter Speicher und sehen Sie sich die Einstellungen für *PythonRuntimeLibrary* und *PythonRuntimeLibraryVersion* an.

Wie lauten die Werte?

4 Python Packages installieren

Python hat bereits mit einer Reihe von Packages vorinstalliert. Zusätzlich stehen eine Vielzahl von öffentlichen Packages für die freie Nutzung zur Verfügung, die man in dem Fall zuvor installieren muss.

Öffnen Sie die Python Package Index unter <https://www.pypi.org>

Wie viele Projekte gibt es laut der Homepage aktuell in diesem Index? _____

Geben Sie im Projektsuchfenster den Begriff „geopy“ ein und klicken Sie auf das erste Ergebnis (geopy 2.4.1). In der Projektbeschreibung finden Sie Details darüber, wie das Paket installiert wird und wie man

es verwendet. Da wir auf einem gemeinsamen Server arbeiten, wurde das Paket bereits vorinstalliert. Damit es von Embedded Python in InterSystems IRIS verwendet werden kann, liegt das Paket in einem speziellen Verzeichnis. Zur Info: Das dabei verwendete Kommando lautet:

```
C:\>python -m pip install --target C:\InterSystems\IRIS\mgr\python geopy
```

Standardmäßig sucht IRIS im Unterordner mgr/python des IRIS Installationsverzeichnisses nach selbst installierten Python Packages. Schauen Sie sich den Inhalt des geteilten python-Ordners aus Aufgabe 2.2.2 kurz an.

5 Terminal Übungen

5.1 Python Shell

Gehen Sie in VSCode zum IRIS-Terminal im Namespace DEMO (siehe Abschnitt 2.3) und starten Sie eine Python-Shell mit folgendem Kommando:

```
Do $system.Python.Shell()
```

Innerhalb der Shell können Sie interaktiv Python Code ausführen, ähnlich wie Sie es vom ObjectScript Terminal her kennen. Das ist für Testzwecke sehr praktisch.

Falls Sie noch nie in Python programmiert haben, tippen Sie folgende Zeile ein gefolgt von der Return-Taste:

```
print("Hello Embedded Python")
```

Herzlichen Glückwunsch, nun haben auch Sie schon in Python programmiert 😊

5.1.1 GPS Koordinaten ermitteln

Testen Sie das geopy Package mit folgenden Kommandos und verwenden Sie Ihre eigenen Werte für user_agent und die Adresse:

```
import geopy
geolocator = geopy.Nominatim(user_agent="Symposium smohr")
location = geolocator.geocode("Robert-Bosch-Str. 16a, Darmstadt, Germany")
print(location.latitude)
print(location.longitude)
```

Wie lautet das Ergebnis für Breitengrad (latitude) und Längengrad (longitude)? _____

5.1.2 Adresse ermitteln

Schauen Sie sich die Dokumentation zum geopy Package an (siehe Abschnitt 4) und ermitteln Sie die Adresse zu folgenden GPS-Koordinaten: Latitude = 48.85844, Longitude = 2.294528. Eine Musterlösung finden Sie im geteilten Ordner „symposium/code“ und im Abschnitt 11.

Was befindet sich an dieser Position? _____

Welche Python-Codezeilen haben Sie dafür verwendet?

5.2 ObjectScript Terminal

5.2.1 GPS Koordinaten ermitteln

Wir können Python-Packages auch direkt in ObjectScript nutzen, ohne in Python zu programmieren.

Beenden Sie die Python Shell mit dem Kommando `exit()`, damit Sie zum ObjectScript-Prompt zurückkehren.

Testen Sie das geopy Package mit den gleichen Werten wie in 5.1.1, aber diesmal mit folgenden ObjectScript-Kommandos, aber verwenden Sie bitte Ihren eigenen Werte für `user_agent` und `location`:

```
set geopy = ##class(%SYS.Python).Import("geopy")
set args = { "user_agent": "Symposium smohr" }
set geolocator = geopy.Nominatim(args...)
set location = geolocator.geocode("Robert-Bosch-Str. 16a, Darmstadt, Germany")
write location.latitude, !, location.longitude
```

5.2.2 Adresse ermitteln


Passen Sie Ihren Python-Code aus 5.1.2 für ObjectScript an und ermitteln Sie die Adresse zu folgenden GPS-Koordinaten: Latitude = 40.6894, Longitude = -74.0445. Eine Musterlösung finden Sie im geteilten Ordner „symposium/code“ und in Abschnitt 11.

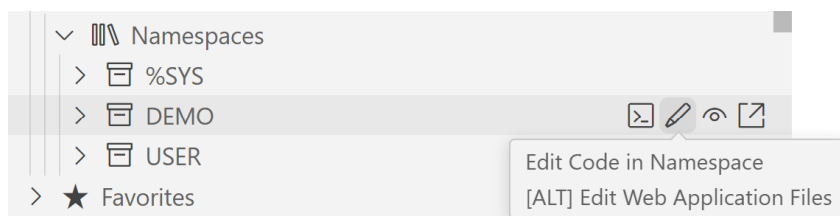
Was befindet sich an dieser Position? _____

Welche ObjectScript-Codezeilen haben Sie dafür verwendet?

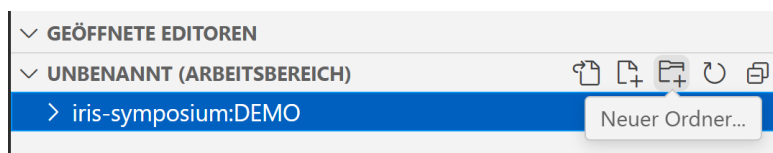
6 Testklasse anlegen

In dieser Übung erstellen Sie Methoden in einer eigenen ObjectScript-Klasse. Da wir eine gemeinsame IRIS Instanz verwenden, legen Sie die Klasse `Test.cls` in Ihrem eigenen Package an.

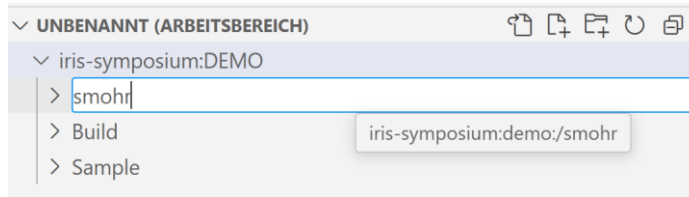
Gehen Sie in das InterSystems-Menü  von VSCode und klicken Sie in der Zeile des DEMO Namespaces auf das Stift-Symbol. Daraufhin gelangen Sie in den Explorer.



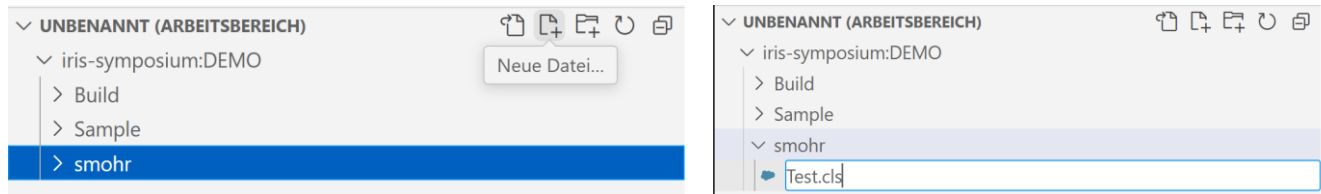
Markieren Sie die oberste Zeile mit dem Eintrag „iris-symposium:DEMO“ und klicken Sie auf das Symbol „Neuer Ordner“.



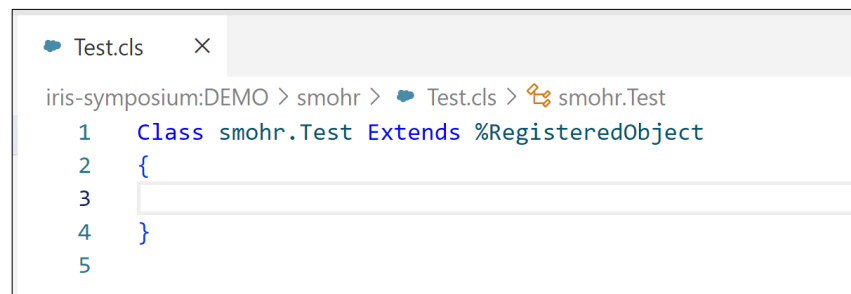
Verwenden Sie als Ordernamen den Anfangsbuchstaben Ihres Vornamens gefolgt vom Nachnamen, z.B. „smohr“ für Stephan Mohr und drücken Sie die Enter-Taste.



Während Ihr Ordner markiert ist, tippen Sie auf das Symbol „Neue Datei“ und vergeben den Namen Test.cls.



Daraufhin erscheint im Editorfenster Ihre neu angelegte Klasse. Speichern Sie diese mit STRG-S.



7 Python Methoden

Erstellen Sie folgende Klassenmethode innerhalb Ihrer Test Klasse. Durch die Angabe von [Language = python] wird innerhalb der Methode Python-Code erwartet. Achten Sie dabei auf eine gleichmäßige Einrückung der Codezeilen. Um Tippzeit zu sparen können Sie den Text der Methode auch aus dem geteilten Ordner „symposium/code“ kopieren. Codeänderungen jeweils mit STRG-S speichern.

```
ClassMethod ExportToExcel(pFilePath As %String) As %Integer [ Language = python ]
{
    import iris
    rs = iris.sql.exec("SELECT Name, Age, Home_Zip, Home_City FROM Sample.Person")
    df = rs.dataframe()
    df.to_excel(pFilePath)
    return df.index.size
}
```

Dieses Beispiel verwendet die dataframe-Klasse aus dem bekannten Pandas-Package. Wie Sie sehen, können wir mit der dataframe()-Methode unser Result Set in ein Dataframe-Objekt umwandeln. Dadurch stehen uns ein Vielzahl von Analyse- und anderer Werkzeuge zur Verfügung. Siehe

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

Zum Testen rufen Sie diese Methode aus Ihrem VSCode InterSystems Lite Terminal im DEMO Namespace auf und verwenden Sie dabei Ihren Namenskürzel als Packagenamen und im Pfad (statt smohr), den Sie im Abschnitt 2.2.2 verwendet haben:

```
write ##class(smohr.Test).ExportToExcel("C:/symposium/aufgaben/smohr/test.xlsx")
```

Sofern Sie Excel auf Ihrem Rechner installiert haben, öffnen Sie Datei, um das Ergebnis zu überprüfen.

8 ObjectScript Methoden

Wie Sie im Abschnitt 5.2 bereits gesehen haben, können wir die umfangreiche Python-Bibliothek auch direkt mit ObjectScript verwenden, also ohne Python-Programmierung. Auf die Angabe von [Language = objectscript] können wir hierbei verzichten, da es die Standardsprache ist.

In diesem Beispiel verwenden Sie das zip-Package, das bereits mit Python mitgeliefert wird.

Erstellen Sie folgende Klassenmethode innerhalb Ihrer Test Klasse. Um Tippzeit zu sparen können Sie den Text der Methode auch aus dem geteilten Ordner „symposium/code“ kopieren.

```
ClassMethod ExtractZipFile(pZipFile As %String, pTargetFolder As %String) As %Status
{
    set zipLib = ##class(%SYS.Python).Import("zipfile")
    set zipRef = zipLib.ZipFile(pZipFile,"r")
    do zipRef.extractall(pTargetFolder)
    do zipRef.close()
    return $$$OK
}
```

Zum Testen rufen Sie diese Methode nach dem Speichern aus Ihrem VSCode InterSystems Lite Terminal im DEMO Namespace auf und verwenden Sie dabei Ihren Namenskürzel im Zielordner (statt smohr), den Sie im Abschnitt 2.2.2 angegeben haben:

```
do ##class(smohr.Test).ExtractZipFile("C:/symposium/files.zip",
"C:/symposium/aufgaben/smohr")
```

Überprüfen Sie, ob die Dateien aus dem ZIP-Archiv in Ihrem Zielordner ausgepackt wurden.

9 Python Klassen

Mit Embedded Python können Sie nicht nur öffentliche Python Packages verwenden, sondern auch Ihre eigene Bibliothek anlegen, also Klassen und Methoden einbinden, die Ihr Entwicklungsteam mit reinem Python-Code entwickelt hat. Wie bereits in Abschnitt erwähnt, sucht IRIS standardmäßig im Unterordner mgr/python des Installationsverzeichnis nach eigenen installierten Python Packages. Diesen Ordner haben wir unter dem Namen „python“ mit Ihnen geteilt (siehe Abschnitt 2.2.2).

9.1 Python-Klasse anlegen

Legen Sie im geteilten Ordner ein Unterverzeichnis mit Ihrem Namenskürzel als Namen an, also z.B. smohr. Öffnen Sie Ihren neuen Ordner in VSCode und legen Sie darin eine Datei mit dem Namen geometrie.py an. Achten Sie beim Eingeben des Codes auf die richtigen Einrückungen. Zur Veranschaulichung beschränken wir uns auf ein einfaches Beispiel. Für eine Zeitersparnis können Sie die fertige Datei auch aus dem geteilten Ordner „symposium/code“ in Ihren Python-Ordner kopieren.

```
1  def info():
2      return "Geometrische Berechnungen"
3
4  class Rechteck():
5      def __init__(self, breite, hoehe):
6          self.b = breite
7          self.h = hoehe
8
9      def flaeche(self):
10         return self.b * self.h
11
12     def umfang(self):
13         return 2 * (self.b + self.h)
```

9.2 Python-Klasse verwenden

Testen Sie Ihre eigene Python-Klasse im IRIS-Terminal (DEMO> Prompt). Starten Sie nach Ihren Codeänderungen das IRIS-Terminal jeweils erneut (siehe Abschnitt 2.3), damit Ihr neues bzw. aktualisiertes Package vom Terminalprozess erkannt wird. Verwenden Sie zum Testen folgende Kommandos, aber benutzen Sie dabei Ihren eigenen Ordner- bzw. Packagenamen (nicht smohr).

```
set geometrie = $system.Python.Import("smohr.geometrie")
zwrite geometrie
```

Welchen Datentyp hast dieses Objekt? _____

Um auf die help-Funktion oder andere in Python eingebauten Datentypklassen, Konstanten und Funktionen zuzugreifen, können Sie das builtin-Package importieren.

```
set builtins = $system.Python.Import("builtins")
do builtins.help(geometrie)
```

Welche beiden Instanzmethoden der Klasse Rechteck werden genannt? _____

```
write geometrie.info()
set rechteck1 = geometrie.Rechteck(7,12)
write rechteck1.umfang()
write rechteck1.flaeche()
```

Wie groß sind Umfang und Fläche? _____

10 Abschluss

Vielen Dank für die Mitarbeit bei dieser Hands-on Session. Wir hoffen, es hat Ihnen Spaß gemacht und neue Ideen geliefert. Falls Sie vorzeitig fertig geworden sind, schauen Sie sich noch einmal die verschiedenen Abschnitte an oder experimentieren Sie gerne selbst noch etwas, indem Sie den vorhandenen Code anpassen oder erweitern.

Zur weiteren Vertiefung empfehlen wir Ihnen unsere aktuelle Dokumentation.

https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=AFL_epython

<https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GEPYTHON>

11 Musterlösungen

Zu 5.1.2

```
from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="Symposium smohr")
location = geolocator.reverse("48.85844, 2.294599")
print(location.address)
```

Zu 5.2.2

```
set geopy = ##class(%SYS.Python).Import("geopy")
set args = { "user_agent": "Symposium smohr" }
set geolocator = geopy.Nominatim(args...)
set location = geolocator.reverse("40.6894, -74.0445")
write location.address
```