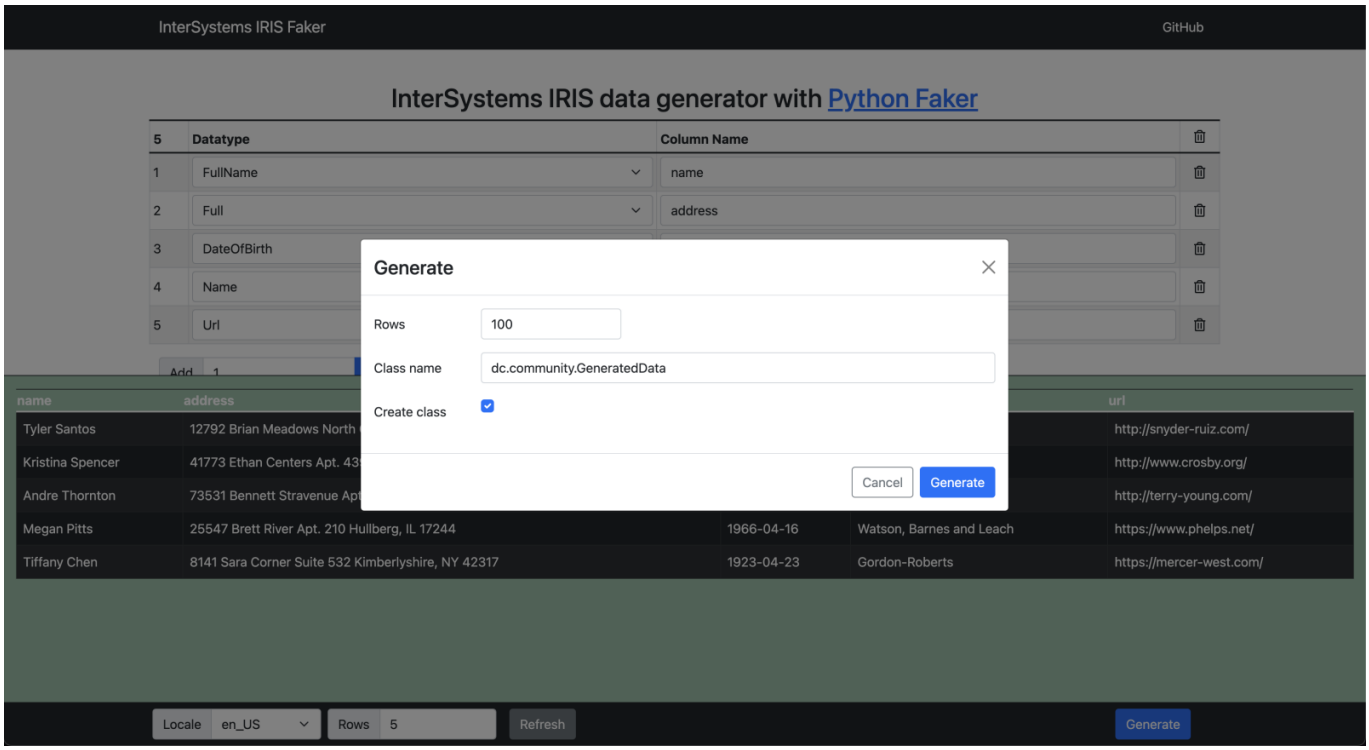
Announcement
Dmitry Maslennikov · Jan 13

Open Exchange

Populate random data that makes sense

I think it's a known fact that Populate Utility has very limited functionality. It supports only one language and one country. The list of possible values does not have so many options.

There is a kind of tool that now can help with it, named Faker. It has implementations in different languages, including Python. Since IRIS has now had the Embedded Python feature, Python faker can be implemented in IRIS.



Python Faker has

- 80+ locales, so, generated data, can be almost any language English, Russian, Chinese. Person names, Addresses, and many other values can be localized
- 20+ providers for many different types of data.
- community providers for some other data generations, such as music, posts, vehicles, and so on.

Implementation of this project iris-python-faker in IRIS can be found on OpenExchange. Online demo available, in preview mode only.

Installation

To test it locally, possible with Docker or with zpm. To run it with Docker from the repository it requires BuildKit

enabled. ZPM-way requires IRIS version with Python Embedded support (2021.2+) and python-pip installed (official Docker image with IRIS does not have it).

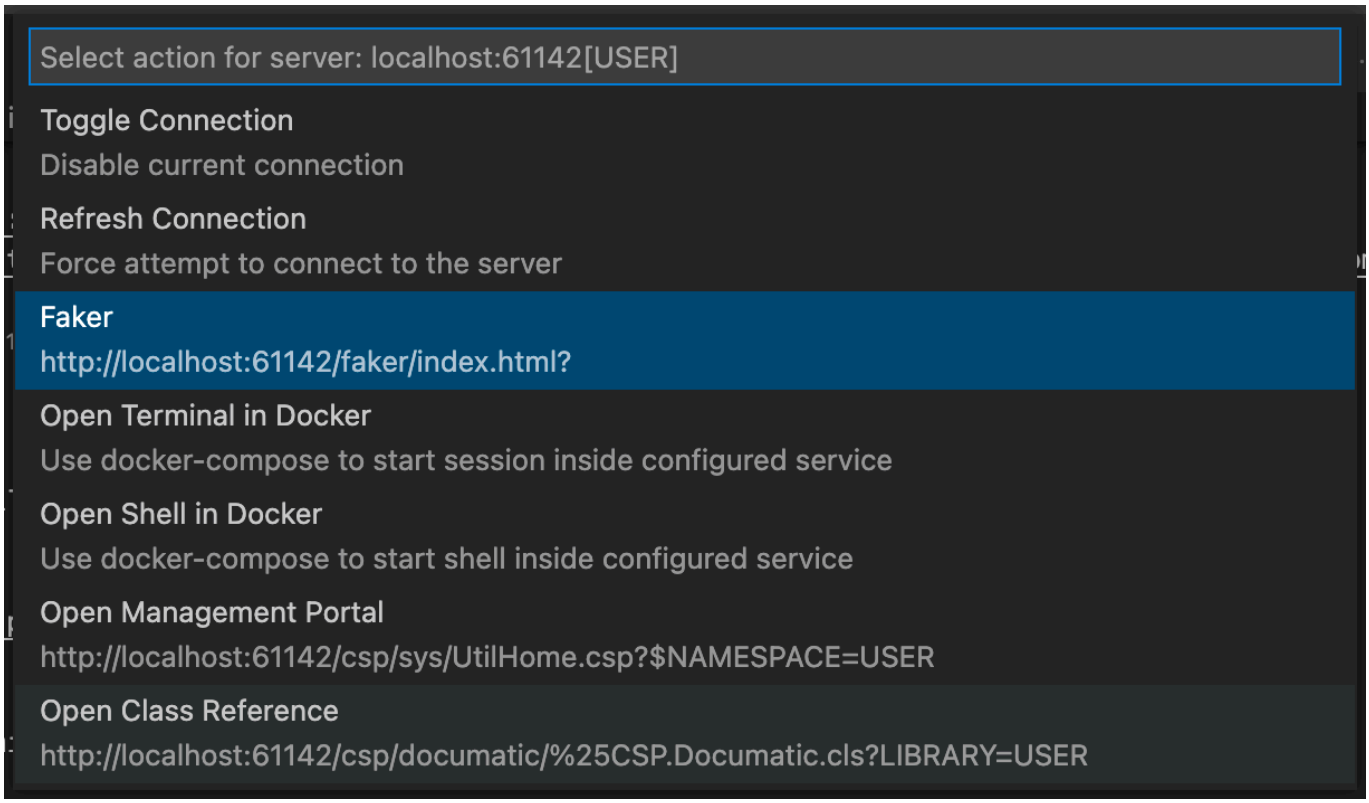
```
git clone https://github.com/caretdev/iris-python-faker.git
cd iris-python-faker
DOCKER_BUILDKIT=1 docker-compose up --build -d
```

Check the port

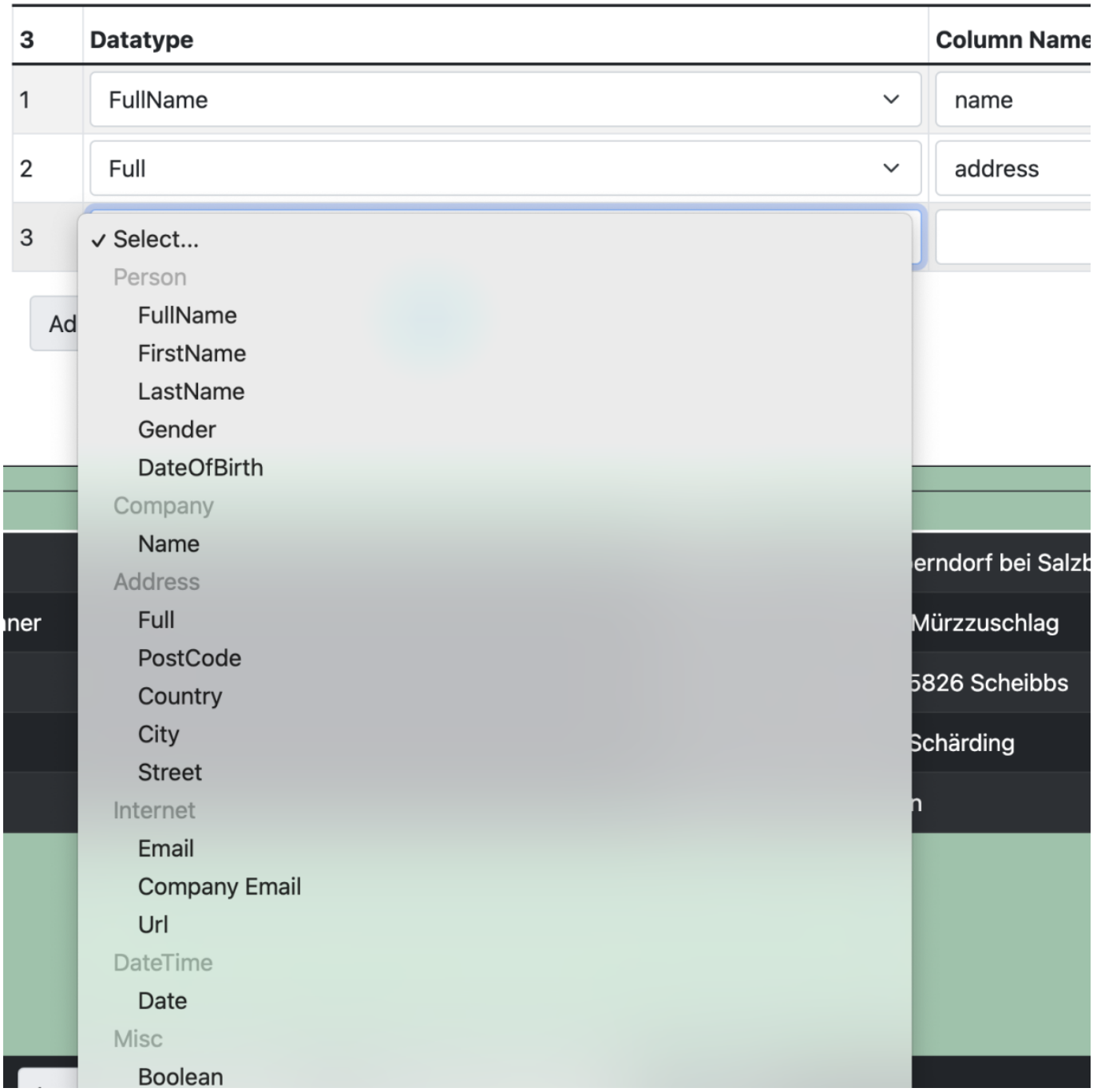
```
$ docker-compose ps
          Name                        Command                       State
                                     Ports
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------
iris-python-faker-iris-1   /tini -- /iris-main -a iri ...   Up (health: starting)   0
.0.0.0:50628->1972/tcp, 2188/tcp, 0.0.0.0:50629->52773/tcp, 53773/tcp, 54773/tcp
```

Usage

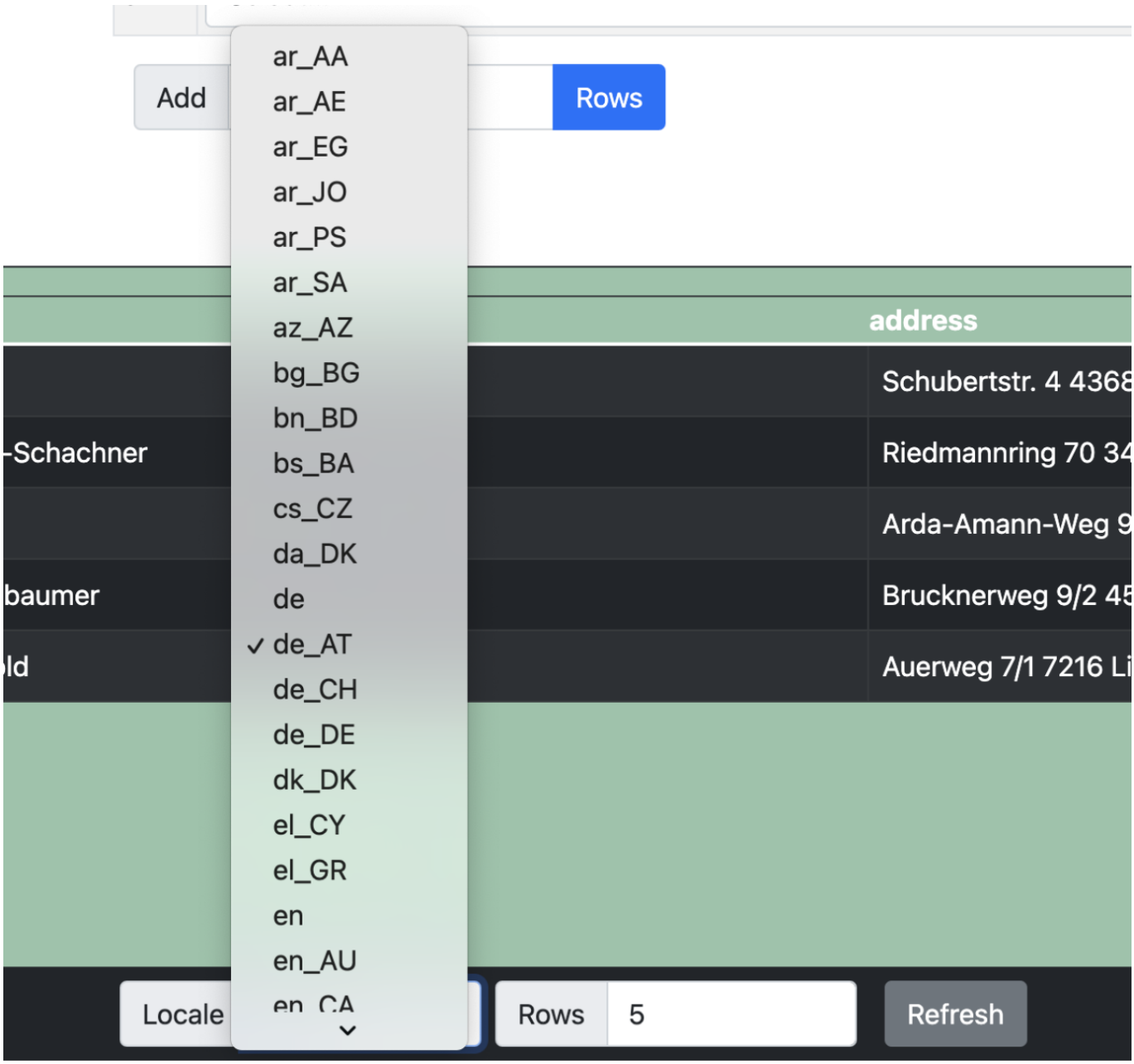
In this case web port is 50629, and UI can be opened by URL http://localhost:50629/faker/index.html. Or if you work from VSCode, it can be opened from the menu.



The table rows represents columns in target table. You can select type from the list of implemented data types.



On the bottom toolbar, you can choose one of the locales, select how many rows, you would like to see in preview. Any changes will refresh the preview table, but you can forcefully refresh it with button Refresh.



Generate button will show a dialog, where you will be asked for the target Class and amount of rows to generate. If you wish to add more rows to an existing class, just uncheck the Create Class.

In case of any error happened on the server side, it will show the error message.

It is also possible to generate new data programmatically

```
Set params = {
  "rows": 3,                // How many rows generate
  "className": (className), // Where to store generated data
  "locale": "en",           // Select Locale for generated data
  "columns": [{
    "type": "person_name",
    "name": "name"
  }, {
    "type": "date_of_birth",
```

```
    "name": "dob"
  }, {
    "type": "boolean",
    "name": "flag"
  }]
}
Set faker = ##class("caretdev.Faker.Main").%New()
Do faker.%JSONImport(params)
Quit:'$$$AssertStatusOK(faker.GenerateClass())
Do $ClassMethod(className, "%KillExtent")

Set result = faker.Generate()
If $Extract(result)=0 Do $system.OBJ.DisplayError(result)
Write !,"Rows created",result.created
```

Or just generate without saving to class

```
Set params = {
  "previewRows": 3,          // How many items generate
  "locale": "en",            // Select Locale for generated data
  "columns": [{
    "type": "person_name",
    "name": "name"
  }, {
    "type": "date_of_birth",
    "name": "dob"
  }, {
    "type": "boolean",
    "name": "flag"
  }]
}
Set faker = ##class("caretdev.Faker.Main").%New()
Do faker.%JSONImport(params)

Set result = faker.Generate()
If $Extract(result)=0 Do $system.OBJ.DisplayError(result)
Set items = result."__getitem__"("items")
Set count = items."__len__"()
For i=0:1:count-1 {
  Set item = items."__getitem__"(i)
  Set name = item."__getitem__"("name")
  Set dob = item."__getitem__"("dob")
  Set flag = item."__getitem__"("flag")
}
```

You can vote for the project on the current Datasets contest. If you would like to support my contribution to OpenSource projects, you can donate here.

#Embedded Python #Python #InterSystems IRIS
Check the related application on InterSystems Open Exchange

Source URL: https://community.intersystems.com/post/populate-random-data-makes-sense