Article
Muhammad Waseem · Sep 13, 2022 5m read

Open Exchange

Sustainable energy data by calling API from the major Independent System Operators (ISOs) in the United States with the help of PEX and Embedded Python

Hi Community,

In this article I will demonstrate the functionality of my app **iris-energy-isodata** .
Application is accessing energy data (production, demand and supply) from the major Independent System Operators (ISOs) in the United States to ensure sustainable consumption and production patterns (SDG's 12)

Application is using python library **isodata** , Production EXtension **PEX** along with **Embedded Python.** Special Thanks to @Guillaume Rongier for the **template** template for guidance

Below is the list of Independent System Operators(ISOs)

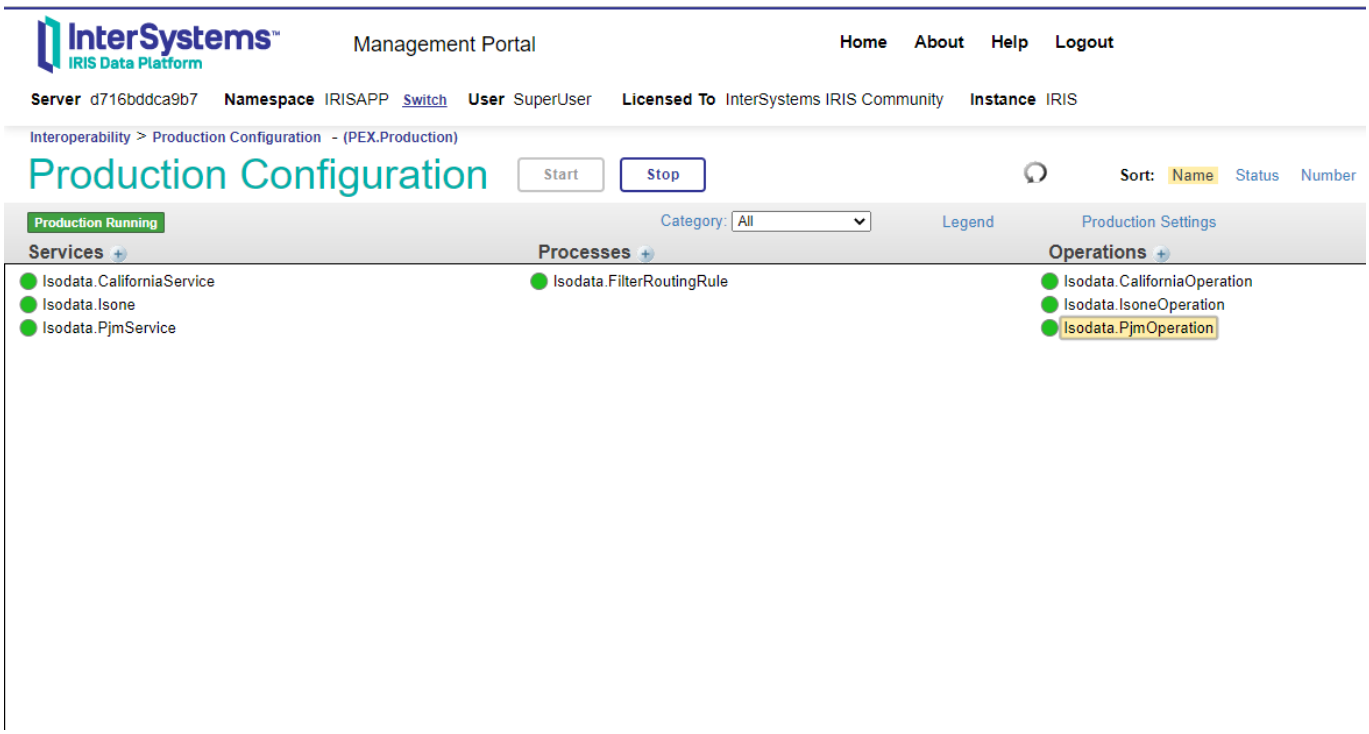
1. **California ISO (caiso)**
2. **PJM (pjm)**
3. **ISO New England (isone)**

And below are the details of energies

1. **Natural Gas**
2. **Solar**
3. **Imports**
4. **Wind**
5. **Large Hydro**
6. **Nuclear**
7. **Batteries**
8. **Geothermal**
9. **Biomass**
10. **Small hydro**
11. **Biogas**
12. **Coal**

So let's start

First of all we have to start the production.



Our production contains 3 Business Services (Every ISO's has its own service)
Business service importing BusinessService class which is inherited from Ens.BusinessService and isodata python library.
Below python code of business service Isodata.CaliforniaService is calling getlatestfuelmix() function to get detail productions of energies, getdemandtoday() function to get today demand and getsupplytoday() function to get today supply.

```
from grongier.pex import BusinessService
import isodata

#Event to get data and pass to process
def on_task(self) -> PostClass:
     #selecting ISO's
     iso = isodata.get_iso('caiso')
     caiso = iso()
     #Get production details of the entergies
     getdata = str(caiso.get_latest_fuel_mix())
     #Get today Demand
     demandpd = caiso.get_demand_today()
     demand = str(demandpd['Demand'].iloc[0])
     #Get today Supply
     supplypd = caiso.get_supply_today()
     supply = str(supplypd['Supply'].iloc[0])
     post = PostClass.from_dict(value['data'])
     post.fuel_mix = getdata
     post.title="caiso"
     post.demand = demand+" NW"
     post.supply = supply+" NW"
     self.log_info(post)
     return post
```

Business Process

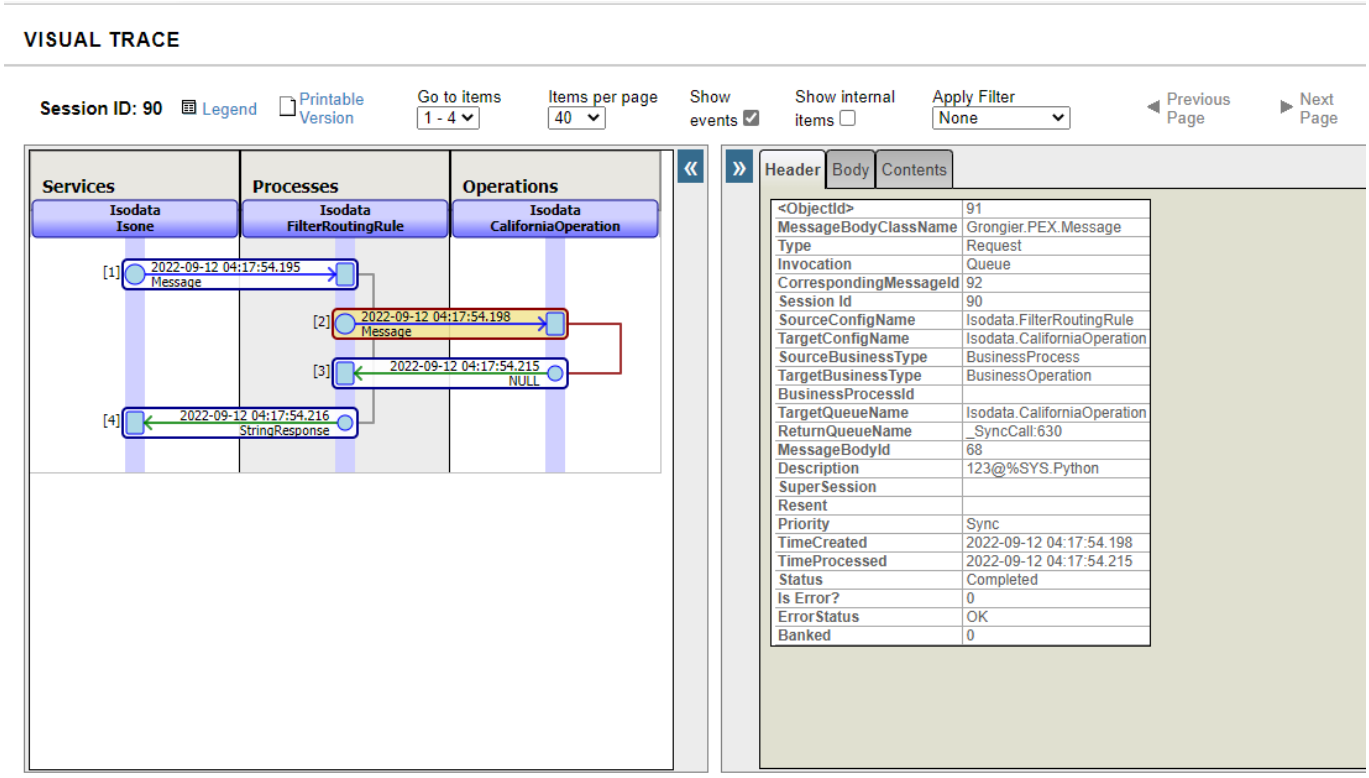
Business process importing BusinessProcess class which is inherited from Ens.BusinessProcess. Getting message from service and based on the title passing to the operation.
Below is the python code of Business Process (Isodata.FilterRoutingRule)

```
from grongier.pex import BusinessProcess
from message import PostMessage
from obj import PostClass
import iris

class FilterPostRoutingRule(BusinessProcess):
    """
    This process receive a PostMessage and send to operation based on the title
    """
    def on_init(self):
        if not hasattr(self,'target'):
            self.target = "Isodata.CaliforniaOperation"
        return
    def iris_to_python(self, request:'iris.dc.Demo.PostMessage'):
        request = PostMessage(post=PostClass(title=request.Post.Title,

                                             created_utc=request.Post.CreatedUTC,
                                             fuel_mix=request.Post.OriginalJSON))
        return self.on_python_message(request)

    def on_python_message(self, request: PostMessage):
        #based on the title from message, routing to the operation
        if 'caiso'.upper() in request.post.title.upper():
            target = "Isodata.CaliforniaOperation"
        elif 'ercot'.upper() in request.post.title.upper():
            target = "Isodata.TexasOperation"
        elif 'nyiso'.upper() in request.post.title.upper():
            target = "Isodata.NewYorkOperation"
        elif 'spp'.upper() in request.post.title.upper():
            target = "Isodata.SouthWestOperation"
        elif 'pjm'.upper() in request.post.title.upper():
            target = "Isodata.PjmOperation"
        elif 'miso'.upper() in request.post.title.upper():
            target = "Isodata.MidcontinentOperation"
        elif 'isone'.upper() in request.post.title.upper():
            target = "Isodata.IsoneOperation"
        if target is not None:
            self.send_request_sync(target,request)
            rsp = iris.cls('Ens.StringResponse')._New(f"{request.post.title}")
            return rsp
        else:
            return
```

Business Operation

Business Operation importing BusinessOperation class which is inherited from Ens.BusinessOperation.
Business Operation is receiving message from Business process and writing the message details to the file based on the message title.
Below is the python code of Business Operation (Isodata.CaliforniaOperation)

```
from grongier.pex import BusinessOperation, Utils
import iris
import os
import datetime
import smtplib
from email.mime.text import MIMEText

class FileOperation(BusinessOperation):
    """
    This operation receive a PostMessage and write down in the right company
    .txt all the important information and the time of the operation
    """
    def on_init(self):
        if hasattr(self,'path'):
            os.chdir(self.path)

    def on_message(self, request):
        ts = title = fuel_mix =  demand = supply = ""
        fuel_mix = request.post.fuel_mix
        demand = request.post.demand
        supply = request.post.supply

        if (request.post is not None):
            title = request.post.title
            ts = datetime.datetime.fromtimestamp(request.post.created_utc).__str__()

        line = ts+" : "+title
        filename = title+".txt"
        self.put_line(filename, line)
        self.put_line(filename, "")
        self.put_line(filename, fuel_mix)
        self.put_line(filename, "")
        self.put_line(filename, demand)
        self.put_line(filename, "")
        self.put_line(filename, supply)
        self.put_line(filename,
" * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *")

        return
    def put_line(self,filename,string):
        try:
            with open(filename, "a",encoding="utf-8") as outfile:
                outfile.write(string)
        except Exception as e:
            raise e
```



Below is the message details which contains ISO name, Total production, details of energies, today demand and supply

message.PostMessage
{
"post": {
"title": "caiso",
"created_utc": 1647879710,
"fuel_mix": "ISO: ISO New England
Total Production: 12036 MW
Time: 2022-09-12 00:11:46-04:00
+----------+------+----------+
| Fuel | MW | Percent |
|----------+------+----------|
Natural Gas	6265	52.1
Nuclear	3323	27.6
Net Imports	1581	13.1
Refuse	324	2.7
Hydro	287	2.4
Wood	169	1.4
Wind	51	0.4
Landfill Gas	35	0.3
Other	1	0
+----------+------+----------+",
"demand": "11765.939 NW",
"supply": "10482 NW"
}
}

Thanks

#Contest #Embedded Python #Interoperability #Ensemble #InterSystems IRIS for Health
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/sustainable-energy-data-calling-api-major-independent-system-operators-isos-united-states-help