

文章

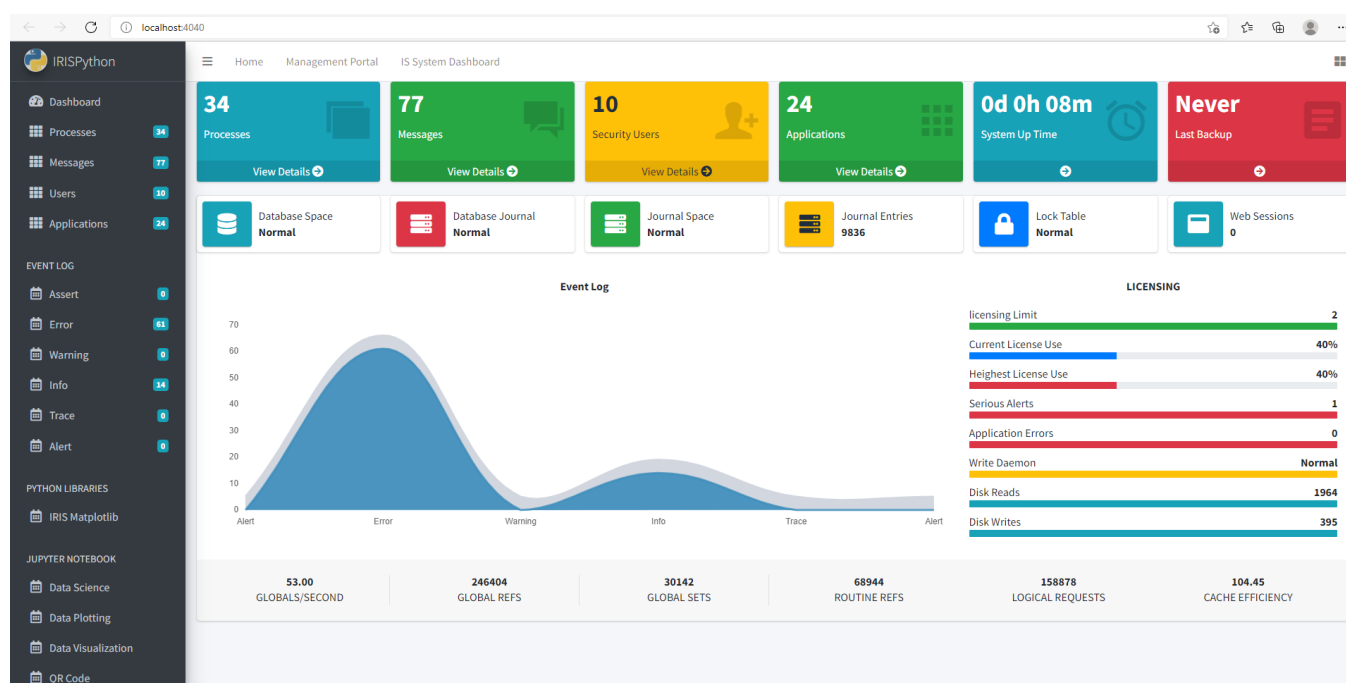
[Johnny Wang](#) · 二月 25, 2022 阅读大约需 6 分钟

让我们来看看本次开发者大赛外国参赛选手Muhammad Waseem的作品：使用 Python Flask Web 框架构建 IRIS 响应式仪表板

Hello 社区成员们，

这篇文章是对Muhammad Waseem的 OpenExchange IRIS-Python-Apps 应用程序的介绍。使用嵌入式 Python 和 Python Flask Web 框架构建。

应用程序还演示了一些 Python 功能，如数据科学、数据绘图、数据可视化和 QR 码生成。来看看最终效果：



特征：

- 响应式引导 IRIS 仪表板；
- 查看仪表板详细信息以及互操作性事件日志和消息；
- 使用 IRIS 中的 Python 绘图；
- 使用Jupyter Notebook；
- 数据科学、数据绘图和数据可视化简介；
- 来自python的二维码生成器。

使用嵌入式 python 的响应式引导 IRIS 仪表板

下面是用户定义函数 `getdashboardstats()` 的代码，用于使用嵌入式 python 从 `SYS.Stats.Dashboard IRIS` 类中获取数据：

代码：getdashboardstats ()

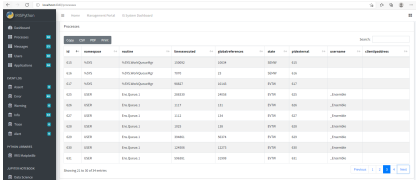
以下是调用 getdashboardstats() 用户定义函数以获取 deta、渲染 index.html 页面并将内容数据传递给它的 主要 python 代码 (app.py)：

主要 python 代码 (app.py)

index.html 使用内容字典变量来显示相关数据。 例如 {{ content.CSPSessions }} 用于显示 CSP 会话：

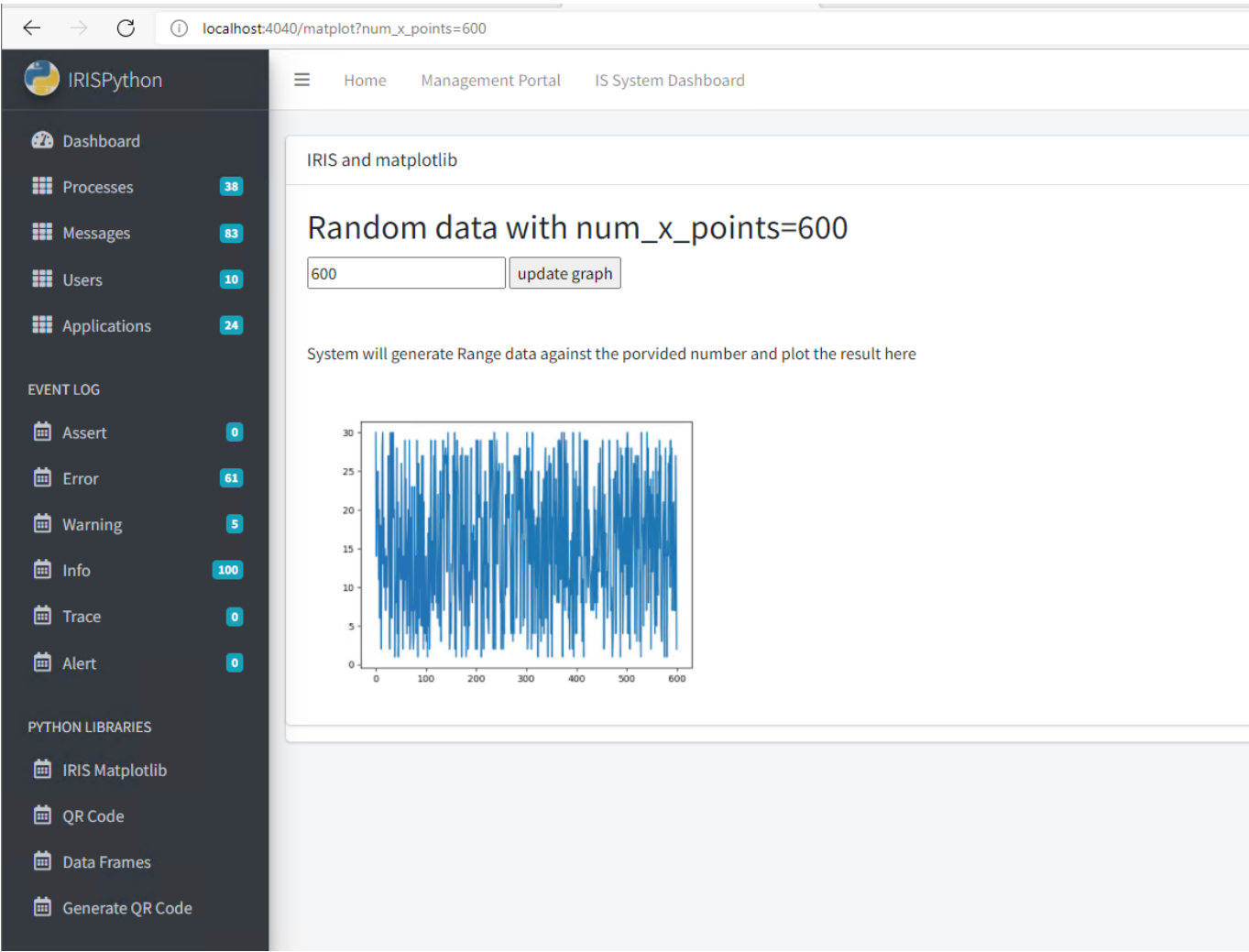
{{ content.CSPSessions }}

在仪表板中，我们还可以使用 iris.sql.exec() 功能查看正在运行的进程、消息、安全用户、应用程序和事件日志的详细信息。



ID	Session	Host	SessionID	IP	Port	Protocol	Language
101	1001	10.10.10.10	1001	10.10.10.10	8080	HTTP	JavaScript
102	1002	10.10.10.10	1002	10.10.10.10	8080	HTTP	JavaScript
103	1003	10.10.10.10	1003	10.10.10.10	8080	HTTP	JavaScript
104	1004	10.10.10.10	1004	10.10.10.10	8080	HTTP	JavaScript
105	1005	10.10.10.10	1005	10.10.10.10	8080	HTTP	JavaScript
106	1006	10.10.10.10	1006	10.10.10.10	8080	HTTP	JavaScript
107	1007	10.10.10.10	1007	10.10.10.10	8080	HTTP	JavaScript
108	1008	10.10.10.10	1008	10.10.10.10	8080	HTTP	JavaScript
109	1009	10.10.10.10	1009	10.10.10.10	8080	HTTP	JavaScript
110	1010	10.10.10.10	1010	10.10.10.10	8080	HTTP	JavaScript

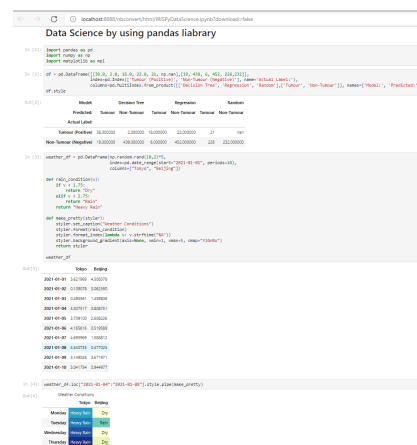
应用程序还使用 python matplotlib 库以 HTML 格式显示绘图：



下面是在 html 中显示绘图的 python 代码：

显示绘图的 python 代码

数据科学简介、数据绘图、数据可视化和使用 Jupyter Notebook 生成二维码：



localhost:8888/nbconvert/html/IRISPyMatplotlib.ipynb?download=false

Data plotting by using matplotlib Library

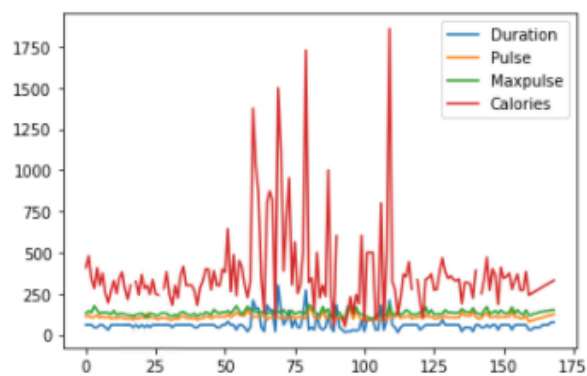
```
In [11]: %matplotlib inline
import sys
import matplotlib
matplotlib.use('Agg')

import pandas as pd
import matplotlib.pyplot as plt

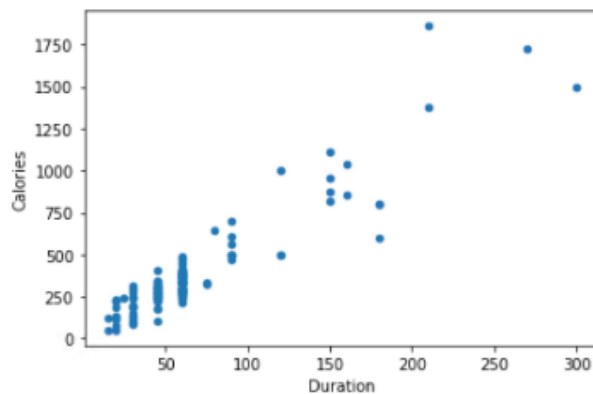
df = pd.read_csv('/opt/irisapp/misc/data.csv')

df.plot()

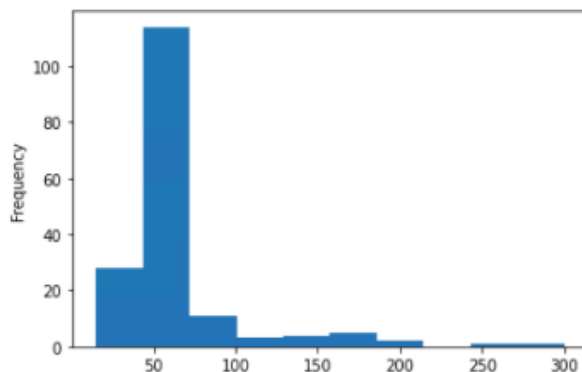
plt.show()
```



```
In [13]: df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')
plt.show()
```



```
In [14]: df["Duration"].plot(kind = 'hist')
plt.show()
```



→ ↺ ⓘ localhost:8888/nbconvert/html/IRISPyPandas.ipynb?download=false

IRIS Data Visualization by using python pandas library

```
In [6]: import pandas as pd
pd.options.display.max_rows = 9

import iris
statement = iris.sql.exec('SELECT * FROM Titanic_Table.Passenger')
df = statement.dataframe()
print(df.info())
df
df.duplicated()
df.corr()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   id           891 non-null    int64
1   survived     891 non-null    int64
2   pclass       891 non-null    int64
3   name         891 non-null    object
4   sex          891 non-null    object
5   age          891 non-null    int64
6   sibsp        891 non-null    int64
7   parch        891 non-null    int64
8   ticket       891 non-null    object
9   fare         891 non-null    float64
10  cabin        891 non-null    object
11  embarked     891 non-null    object
dtypes: float64(1), int64(6), object(5)
memory usage: 83.7+ KB
None
```

```
Out[6]:
```

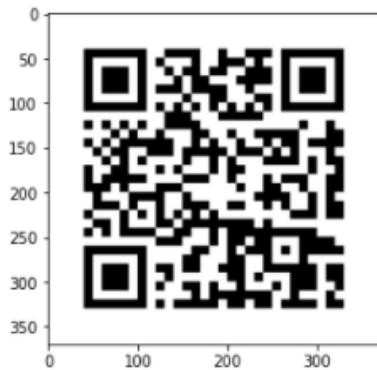
	id	survived	pclass	age	sibsp	parch	fare
id	1.000000	-0.005007	-0.035144	0.038220	-0.057527	-0.001652	0.012658
survived	-0.005007	1.000000	-0.338481	0.010508	-0.035322	0.081629	0.257307
pclass	-0.035144	-0.338481	1.000000	-0.361566	0.083081	0.018443	-0.549500
age	0.038220	0.010508	-0.361566	1.000000	-0.184593	-0.049061	0.135663
sibsp	-0.057527	-0.035322	0.083081	-0.184593	1.000000	0.414838	0.159651
parch	-0.001652	0.081629	0.018443	-0.049061	0.414838	1.000000	0.216225
fare	0.012658	0.257307	-0.549500	0.135663	0.159651	0.216225	1.000000

QR Code generation by using Python qrcode library

```
In [13]: import qrcode  
import matplotlib.pyplot as plt
```

```
In [15]: data = "InterSystems Python QR CODE generator"  
filename = "IRISQRCode.png"  
  
image = qrcode.make(data)  
  
image.save(filename)  
plt.imshow(image,cmap='gray')
```

```
Out[15]: <matplotlib.image.AxesImage at 0x7f9754316760>
```



[#Python](#) [#InterSystems](#) [IRIS](#) [#Open Exchange](#)

源

URL:

<https://cn.community.intersystems.com/post/%E8%AE%A9%E6%88%91%E4%BB%AC%E6%9D%A5%E7%9C%8B%E7%9C%8B%E6%9C%AC%E6%AC%A1%E5%BC%80%E5%8F%91%E8%80%85%E5%A4%A7%E8%B5%9B%E5%A4%96%E5%9B%BD%E5%8F%82%E8%B5%9B%E9%80%89%E6%89%8Bmuhammad-waseem%E7%9A%84%E4%BD%9C%E5%93%81%E5%BC%9A%E4%BD%BF%E7%94%A8-python-flask-web-%E6%A1%86%E6%9E%B6%E6%9E%84%E5%BB%BA-iris-%E5%93%8D%E5%BA%94%E5%BC%8F%E4%BB%AA%E8%A1%A8%E6%9D%BF>