

Python + Google Sheets

Manipulando planilhas do Google usando Python

**Renan Pessoa** | [Follow](#)

Senior DevOps Engineer at Host...

88

18

0

O Google Sheets é muito utilizado hoje em dia por sua facilidade de utilização, basta acessar a planilha através do navegador e pronto! Uma outra vantagem é a sua API para gerenciar através de código, é possível manipular os dados usando diversas linguagens de programação, hoje vamos abordar em Python.

Esse recurso é muito interessante pois podemos automatizar muita coisa, como por exemplo preencher uma planilha a partir dos dados de um arquivo local.

Preparando o ambiente

Eu estou utilizando o sistema operacional Linux, mas pode ser feito também no Windows sem problemas.

É necessário ter instalado o **Python 3** e o gerenciador de pacotes **pip**, caso não tenha realize a instalação antes de continuar.

Criando um ambiente isolado

Crie um diretório para salvar o código e em seguida execute os comandos abaixo para criar um ambiente isolado, com isso os módulos que serão instalados não irão afetar as outras instalações.

```
renanpessoa@renan-Notebook:~/linkedin$ python3 -m venv venv
```

```
renanpessoa@renan-Notebook:~/linkedin$
```

Após a instalação vamos ativar o ambiente criado

```
renanpessoa@renan-Notebook:~/linkedin$ . venv/bin/activate
```

```
(venv) renanpessoa@renan-Notebook:~/linkedin$
```

Repare que após executar o comando no início da linha foi adicionado "(venv)", isso significa que estamos dentro de um ambiente isolado, podemos continuar!

Instalando dependências

Agora vamos instalar todos os módulos necessários, será utilizado o **gspread** para gerenciar a API do Google Sheets, essa biblioteca de código aberto facilita muito o gerenciamento, recomendo a leitura da sua documentação(basta clicar em gspread), pois vamos utilizar pouca coisa da sua imensa quantidade de funções.

```
pip install gspread oauth2client
```

Exemplo

```
Requirement already satisfied: requests>=2.2.1 in ./venv/lib/python3.6/site-packages (from gspread)
Requirement already satisfied: six>=1.6.1 in ./venv/lib/python3.6/site-packages (from oauth2client)
Requirement already satisfied: pyasn1-modules>=0.0.5 in ./venv/lib/python3.6/site-packages (from oa
Requirement already satisfied: pyasn1>=0.1.7 in ./venv/lib/python3.6/site-packages (from oauth2clie
Requirement already satisfied: rsa>=3.1.4 in ./venv/lib/python3.6/site-packages (from oauth2client)
Requirement already satisfied: httplib2>=0.9.1 in ./venv/lib/python3.6/site-packages (from oauth2cl
Requirement already satisfied: idna<2.8,>=2.5 in ./venv/lib/python3.6/site-packages (from requests>
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in ./venv/lib/python3.6/site-packages (from re
Requirement already satisfied: certifi>=2017.4.17 in ./venv/lib/python3.6/site-packages (from requ
Requirement already satisfied: urllib3<1.24,>=1.21.1 in ./venv/lib/python3.6/site-packages (from re
(venv) renanpessoa@renan-Notebook:~/linkedin$
```

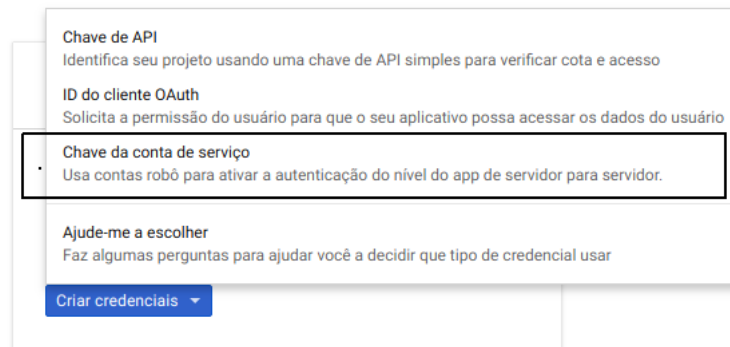
Criando credenciais

Após instalar todas as bibliotecas necessárias vamos criar uma credencial no Google para gerenciar as planilhas.

Acesse o endereço <https://console.developers.google.com/project> e clique em 'Criar Projeto', preencha com o nome do projeto, eu coloquei 'py-sheets' e clique em criar. Se o projeto não for listado aperte a tecla F5 para atualizar a página.

Clique em cima do nome do projeto, em seguida no canto superior esquerdo clique no menu de navegação -> 'API's e serviços' -> 'Credenciais'

Em 'Criar credenciais' clique na seta para baixo e em seguida em 'Chave da conta de serviço'



4. Preencha o campo 'Nome da conta de serviço' e em seguida o papel para aquela conta, eu selecionei 'Projeto -> Proprietário' logo abaixo na opção 'Tipo de chave' selecione JSON e clique em 'Criar', vai aparecer um arquivo para ser baixado, salve esse arquivo no diretório que vai ficar o script.

Esse arquivo contém as credenciais para acessar o projeto, guarde em segredo.

5. No canto esquerdo clique em 'Biblioteca' e no campo de busca digite 'sheets', em seguida clique em 'Google Sheets API' e após isso em 'Ativar'.

A credencial foi criada com sucesso, agora vamos criar uma nova planilha e permitir o acesso.

Criando planilha

Crie uma nova planilha no endereço <https://docs.google.com/spreadsheets/u/0/> e de um título qualquer, eu coloquei 'Teste Python'.

Em seguida clique em 'Compartilhar', será solicitado um e-mail, abra o arquivo que você baixou e procure pela linha 'client_email', coloque o e-mail que está nesse arquivo e clique em 'Concluído'.

Testando o código

Após todas as etapas vamos enfim começar a colocar a mão na massa!

Crie um arquivo com o nome 'spreads.py' no mesmo diretório em que o arquivo com a autenticação foi salva com o conteúdo abaixo:

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials

scope = ['https://spreadsheets.google.com/feeds']

credentials = ServiceAccountCredentials.from_json_keyfile_name('credentials.json', scope)

gc = gspread.authorize(credentials)
```

Na quarta linha altere o nome de 'credentials.json' para o nome do arquivo baixado com as credenciais.

Acesse a planilha que foi criada, no link que está no seu navegador vai estar algo do tipo:

https://docs.google.com/spreadsheets/d/1R1_MJKisuXQc-7o-Uonjz--sDKM2x-2iKsEanSqXewBYmI/edit#gid=0

Precisamos pegar o ID da sua planilha, no exemplo acima o código é o:

1R1_MJKisuXQc-7o-Uonjz--sDKM2x-2iKsEanSqXewBYmI

Na linha seis altere 'KEY' para o ID da sua planilha e salve, agora vamos executar o script para testar se está funcionando:

```
(env) renanpessoa@renan-Notebook:~$ python spreads.py
```

```
(env) renanpessoa@renan-Notebook:~/linkedin$
```

Se nenhuma mensagem de erro for retornada está funcionando!

Manipulando planilha

Vamos agora executar algumas operações simples na planilha.

Adicionando valores

Vamos adicionar uma mensagem na primeira célula, veja o código abaixo:

```
import gspread
```

```
from oauth2client.service_account import ServiceAccountCredentials
```

```
#Escopo utilizado
```

```
scope = [ 'https://spreadsheets.google.com/feeds' ]
```

```
#Dados de autenticação
```

```
credentials = ServiceAccountCredentials.from_json_keyfile_name(
    'credentials.json', scope)
```

```
#Se autentica
```

```
gc = gspread.authorize(credentials)
```

```
#Abre a planilha
```

```
wks = gc.open_by_key('1R1_MJUKiXQc-7o-Uonjz--sDKM2x-2iNMflvXewBYmI')
```

```
#Para selecionar a planilha pelo o nome use o código abaixo
```

```
#wks = gc.open('Teste Python')
```

```
#Seleciona a primeira página da planilha
```

```
worksheet = wks.get_worksheet(0)
```

```
#Atualiza célula
```

```
worksheet.update_cell('A1', 'Funciona :D')
```

Foi adicionado uma nova linha no código, ele diz para atualizar a célula 'A1' com o texto 'Funciona :D', salve o código e execute, em seguida verifique a sua planilha.



Para adicionar múltiplos valores basta fazer um laço de repetição, vamos fazer um exemplo bem básico, digamos que você precise preencher a planilha com os estados do Brasil e a sua respectiva capital, existem diversas maneiras de fazer isso, eu vou mostrar no exemplo uma maneira simples

```
worksheet.update_cell('A1', 'Estado')
```

```
worksheet.update_cell('B1', 'Capital')
```

```
#Dicionario com os estados e as capitais
```

```
capitais = {'Paraíba': 'João Pessoa', 'Santa Catarina': 'Florianópolis', 'São Paulo' : 'São Paulo'}
```

```
#Contador de colunas e células
```

```
columns = 1
```

```
cel = 2
```

```
for estado, capital in capitais.items():
```

```
    #Atualiza a célula 2 da coluna 1 com o nome do estado
```

```
    worksheet.update_cell(cel, columns, estado)
```

```
    #A coluna agora é a B
```

```
#A coluna agora é a A
columns = 1
#Acréscimo mais um valor no número da célula
cel += 1
```

Execute o código acima e veja como ficou a planilha



Pegando valor de uma célula

Para não ficar repetindo código, eu vou mostrar apenas a linha que foi adicionada, agora vamos 'pegar' os valores de uma célula

```
val = worksheet.cell(1,1).value
```

```
print(val)
```

Ao invés de utilizar as letras da planilha para selecionar a célula podemos utilizar coordenadas como no exemplo acima, eu estou selecionando a primeira célula da primeira coluna.

```
(venv) renanpessoa@renan-Notebook:~/linkedin$ python spreads.py
```

Estado

```
(venv) renanpessoa@renan-Notebook:~/linkedin$
```

Pegando dados de uma coluna

Podemos pegar todos os dados que estão em uma determinada coluna, para isso basta adicionar a linha abaixo no código

```
val = worksheet.col_values(1)
```

```
print(val)
```

Foi retornado os dados da coluna

```
(venv) renanpessoa@renan-Notebook:~/linkedin$ python spreads.py
```

```
['Estado', 'Paraíba', 'Santa Catarina', 'São Paulo']
```

```
(venv) renanpessoa@renan-Notebook:~/linkedin$
```

Pegando dados de uma linha

Também podemos pegar todos os dados de uma linha em específico

```
val = worksheet.row_values(1)
```

```
print(val)
```

Dados da primeira linha

```
(venv) renanpessoa@renan-Notebook:~/linkedin$ python spreads.py
```

```
['Estado', 'Capital']
```

```
(venv) renanpessoa@renan-Notebook:~/linkedin$
```

Procurando dados na coluna

E para finalizar vamos procurar uma palavra na planilha e conseguir as coordenadas, no exemplo abaixo vamos procurar em qual célula e coluna está a capital 'Florianópolis'

```
cell = worksheet.find("Florianópolis")
```

```
print(f'Encontrado na célula {cell.row} coluna {cell.col}')
```

Os valores foram retornados com sucesso

```
(venv) renanpessoa@renan-Notebook:~/linkedin$ python spreads.py
```

```
Encontrado na célula 3 coluna 2
```

```
(venv) renanpessoa@renan-Notebook:~/linkedin$
```

Conclusão

Vimos como executar as operações mais comuns neste artigo, é possível realizar muito mais operações, como por exemplo criar novas planilhas, remover, compartilhar...

Recomendo a leitura da [documentação](#) e também da página do projeto no Github ->

<https://github.com/burnash/gspread>

Se você tiver alguma dúvida, sugestão ou problema me mande uma mensagem.

Publicado por



Rena

Renan Pessoa

Senior DevOps Engineer at HostGat...

n

Pessoa

Você fica preenchendo planilha manualmente? Então vem aprender a automatizar isso com Python de maneira simples! #Python #Sheets #Python+Sheets #GoogleSheets

Seguir



Entre para deixar seu comentário

**Leandro Moreira** (👤 Ram Uttar)

3 m

Leandro Gerente de projetos na bycoders_

Moreira



Olá Renan Pessoa, a biblioteca oauth2client será descontinuada, sendo recomendado o uso da nova biblioteca (Google Auth Library for Python). Fonte: <https://google-auth.readthedocs.io/en/latest/oauth2client-deprecation.html>

Gostei Responder | 1 gostou

**Geysson Aguiar de Melo**

3 m

Geysson Python | PHP | Javascript

Aguiar

de Muito obrigado!

Melo Gostei Responder | 1 gostou

**Francisco Matos**

4 m

Francisco Analista Desenvolvedor | Delphi | Python | SQL | Automação | ACBR

Matos

Cara, sensacional. Funcionou perfeitamente. A parte de criar a chave mudou um pouquinho, mas é só ter atenção que a pessoa consegue fazer. Valeu demais.

Gostei Responder | 1 gostou

**Wilson Nunes de Souza Jr.**

5 m

Wilson Empreendedor focado em ampliar a visão operacional, tática e estratégica do negócio. Experiência em prospecção de dados.

Renan Pessoa, parabéns pelo artigo!!! Bem escrito, fácil de entender e muito prático. Respondeu algumas dúvidas de minha pesquisa.

Jr. Gostei Responder | 1 gostou 1 resposta

**Renan Pessoa**

5 m

n Senior DevOps Engineer at HostGator América Latina

Pessoa

Obrigado Wilson Nunes de Souza Jr. fico feliz que o artigo tenha ajudado



Gostei Responder | 1 gostou

**Leandro Moreira** (👤 Ram Uttar)

6 m

Leandro Gerente de projetos na bycoders_

Moreira



(👤) Simples e útil, muito obrigado!

Ram Gostei Responder | 1 gostou

Uttar)

**Pedro Simon**

1 a

Pedro Estudante de Engenharia Civil | UFSC

Simon

Muito bom, ótimo conteúdo!

Gostei Responder | 1 gostou

[Exibir mais comentários.](#)

More from Renan Pessoa [3 articles](#)



Executando AWS cli em múltiplas contas de maneira fácil

[Executando AWS cli em múltiplas contas de...](#)


29 de janeiro de 2021



Validando Jenkinsfile no VSCode

[Validando Jenkinsfile no VSCode](#)

5 de agosto de 2020



Cadastre-se agora

Entrar

Acessibilidade

Política de Privacidade do LinkedIn

Política de Direitos Autorais

Controles de visitantes

Idioma

Contrato do Usuário

Política de Cookies

Política da Marca

Diretrizes da Comunidade