

Como obter o histórico de dividendos e outros proventos com Python direto do site da B3

Escrito por Fábio Minutti Teixeira em 01 Dezembro 2022

De forma bem simples, sem o uso de recursos avançados e em poucas linhas de código é possível extrair o histórico de proventos de companhias abertas direto do site da B3 em formato **json**.

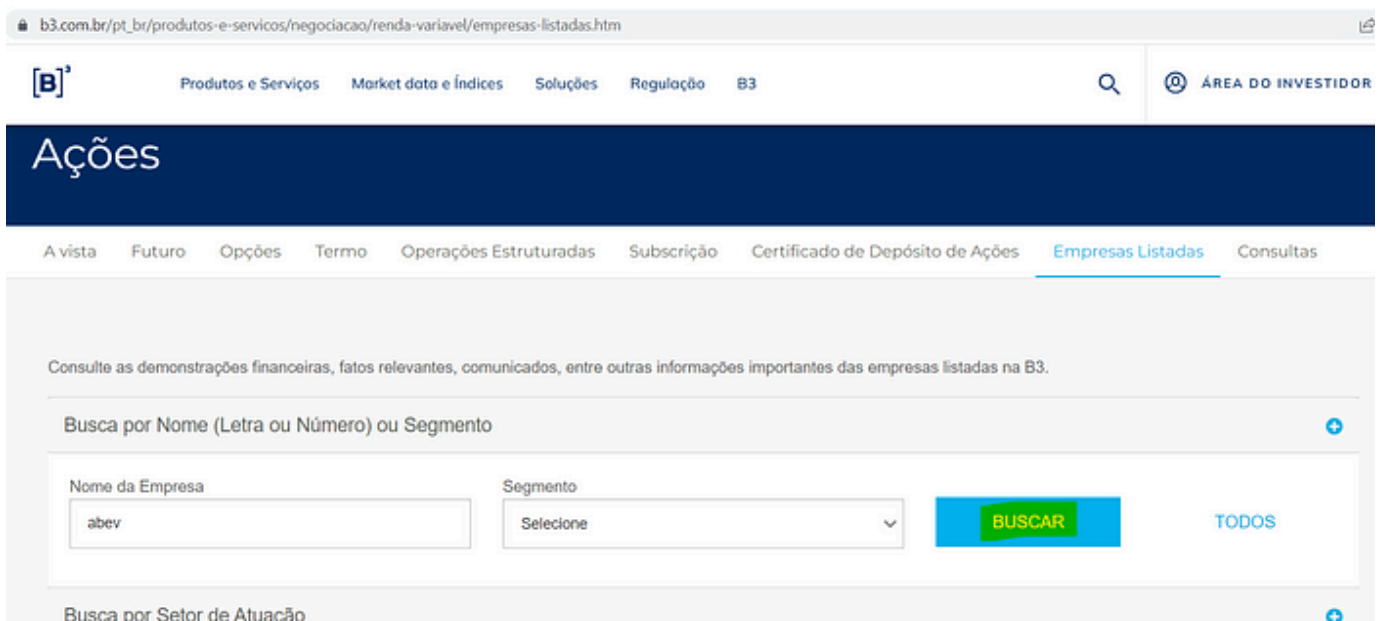
Se preferir só ver o código completo pode pular para o final, mas recomendo ler tudo para ver como cada API usada foi encontrada e passo a passo para o desenvolvimento, **pois o mesmo método pode ser usado para obter outras informações no site da B3 e em alguns outros sites.**

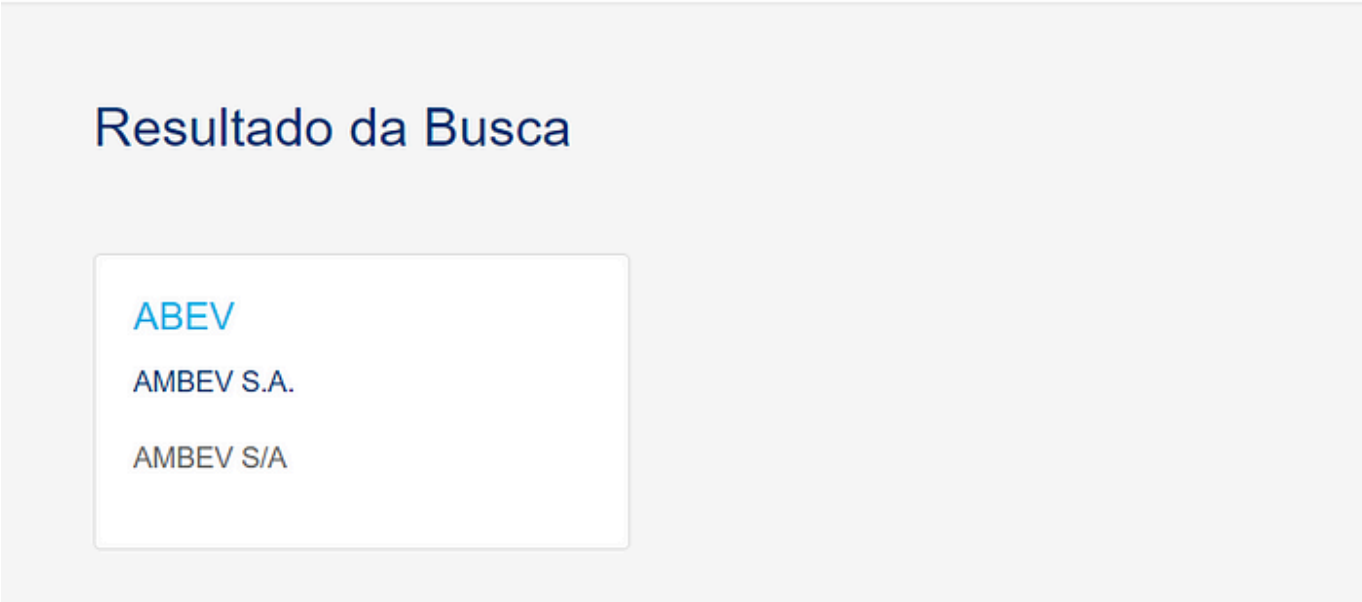


Origem das informações no site da B3

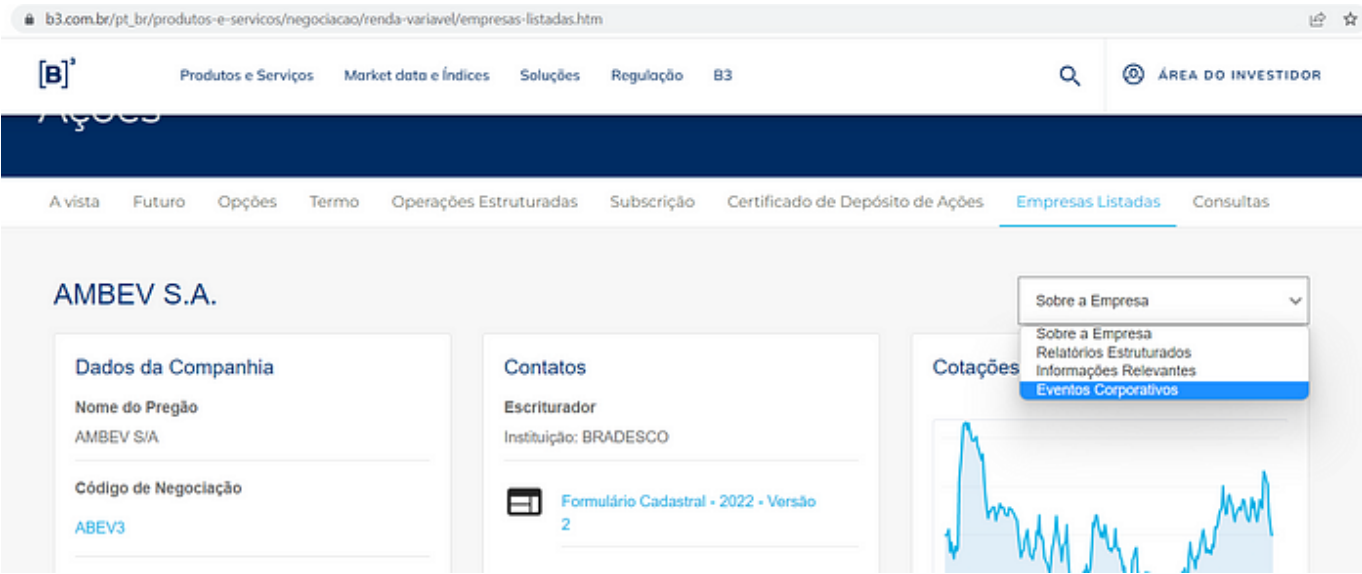
Primeiro é importante para saber como encontrar as APIs no site da B3 que serão usadas no script.

Entrando nessa [página](#) podemos acessar informações sobre qualquer empresa listada pesquisando pelo código da ação ou nome:

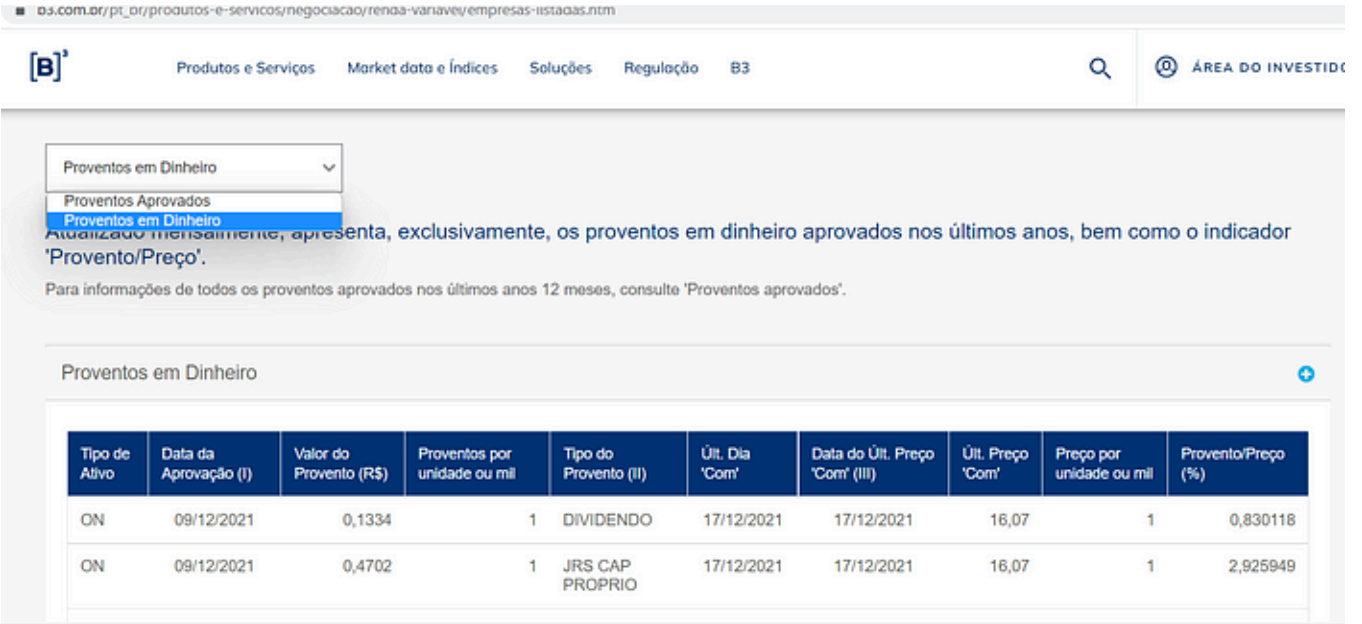




Entrando na página da empresa temos várias informações, como no exemplo:



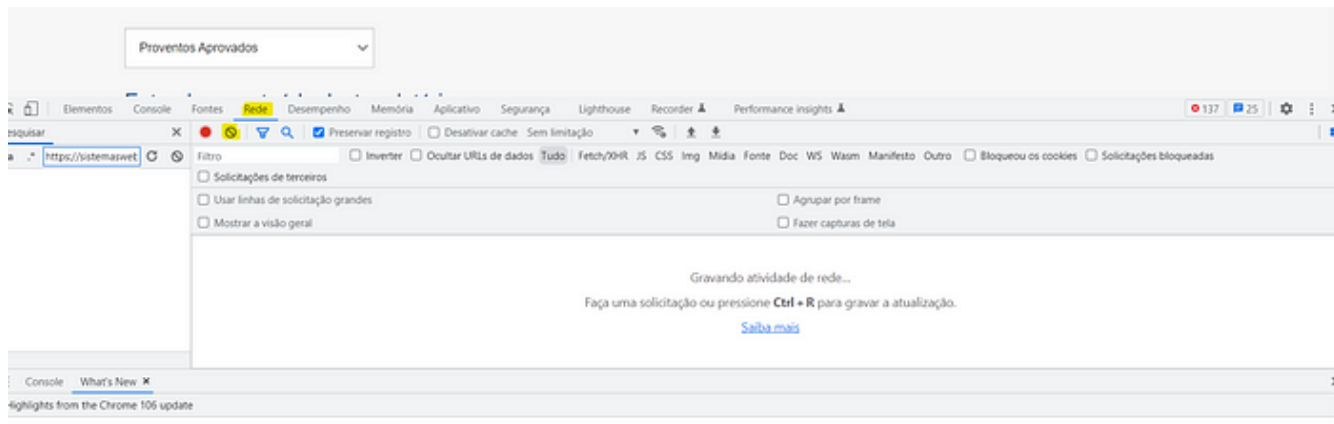
Clicando em *Eventos Corporativos*, entramos na página de proventos. Selecionando *Proventos em Dinheiro*, obtemos a lista de proventos conforme mostrado abaixo.



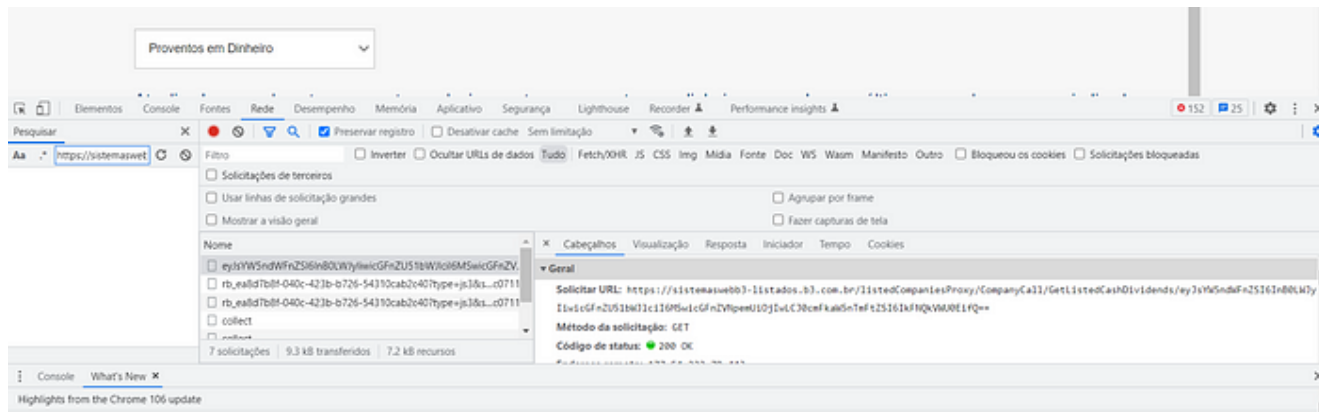
Encontrando a API de importação dos dados

Para encontrar a API, precisamos ver as requisições que a página faz para obter os dados. Para isso, na página de proventos, pressione **CTRL+ALT+I** para inspecionar a página.

No Chrome vai aparecer a seguinte tela:



Clique em *Rede (ou Network)* e no botão indicado embaixo para limpar as requisições. Ao selecionar novamente *Proventos em Dinheiro*, será exibido nessa tela as requisições que o site faz para importar os dados.



No caso desse página, a requisição que precisamos é a primeira, que está selecionada na imagem. A URL exibida ao lado é a API. Ao fazermos a requisição temos os proventos em formato json, conforme a imagem abaixo:

```

1 import requests
2 r = requests.get('https://sistemaswebb3-listados.b3.com.br/listedCompaniesProxy/CompanyCall/GetListedCashDividends/'+
3                 'eyJ3YW5ndWFnZSI6InB0LWJyIiwicGFnZU51bWJlciI6MSwicGFnZVpemUiOjIwLCJ0cmFkaW5nTmFtZSI6IkFNQkVWU0EifQ==')
1 r.json()

```

```

{'page': {'pageNumber': 1,
'pageSize': 20,
'totalRecords': 29,
'totalPages': 2},
'results': [{'typeStock': 'ON',
'dateApproval': '09/12/2021',
'valueCash': '0,1334',
'ratio': '1',
'corporateAction': 'DIVIDENDO',
'lastDatePriorEx': '17/12/2021',
'dateClosingPricePriorExDate': '17/12/2021',
'closingPricePriorExDate': '16,07',
'quotedPerShares': '1',
'corporateActionPrice': '0,830118',
'lastDateTimePriorEx': '2021-12-17T00:00:00'},
{'typeStock': 'ON',
'dateApproval': '09/12/2021',
'valueCash': '0,4702',
'ratio': '1',
'corporateAction': 'JRS CAP PROPRIO',
'lastDatePriorEx': '17/12/2021',
'dateClosingPricePriorExDate': '17/12/2021',
'closingPricePriorExDate': '16,07',
'quotedPerShares': '1',
'corporateActionPrice': '2,925949',
'lastDateTimePriorEx': '2021-12-17T00:00:00'},
{'typeStock': 'ON',
'dateApproval': '21/12/2020',
'valueCash': '0,0767',
'ratio': '1'.

```

O seguinte texto na última parte da

API: [eyJ3YW5ndWFnZSI6InB0LWJyIiwicGFnZU51bWJlciI6MSwicGFnZVpemUiOjIwLCJ0cmFkaW5nTmFtZSI6IkFNQkVWU0EifQ==](#), são os parâmetros da busca codificados base64. Para ver os parâmetros podemos decodificar usando a biblioteca [base64](#) para ver quais são os parâmetros.

```

]: 1 from base64 import b64decode
2
3 b64decode('eyJ3YW5ndWFnZSI6InB0LWJyIiwicGFnZU51bWJlciI6MSwicGFnZVpemUiOjIwLCJ0cmFkaW5nTmFtZSI6IkFNQkVWU0EifQ==')
]: b'{"language": "pt-br", "pageNumber": 1, "pageSize": 20, "tradingName": "AMBEVA"}'

```

Sabendo os parâmetros, podemos usar a função [b64encode](#) da mesma biblioteca para criar os parâmetros e codificar. Por exemplo só traz os 20 últimos registros, mas mudando o parâmetro é possível trazer todos, como nesse exemplo que traz todos os proventos da Petrobras desde 1996:

```

1 from base64 import b64encode
2
3 string = b64encode(b'{"language": "pt-br", "pageNumber": 1, "pageSize": 99999, "tradingName": "PETROBRAS"}').decode()
4 print(string)

```

eyJ3YW5ndWFnZSI6InB0LWJyIiwicGFnZU51bWJlciI6MSwicGFnZVpemUiOjIwLCJ0cmFkaW5nTmFtZSI6IlBFVFJPQlJBUyJ9

```

1 r = requests.get('https://sistemaswebb3-listados.b3.com.br/listedCompaniesProxy/CompanyCall/GetListedCashDividends/'+
2                 string)
3
4 print(r.json()['results'][-1])

```

```

{'typeStock': 'PN', 'dateApproval': '21/03/1996', 'valueCash': '5,1471', 'ratio': '1000', 'corporateAction': 'DIVIDENDO', 'lastDatePriorEx': '21/03/1996', 'dateClosingPricePriorExDate': '20/03/1996', 'closingPricePriorExDate': '114,99', 'quotedPerShare s': '1000', 'corporateActionPrice': '4,476128', 'lastDateTimePriorEx': '1996-03-21T00:00:00'}

```

```

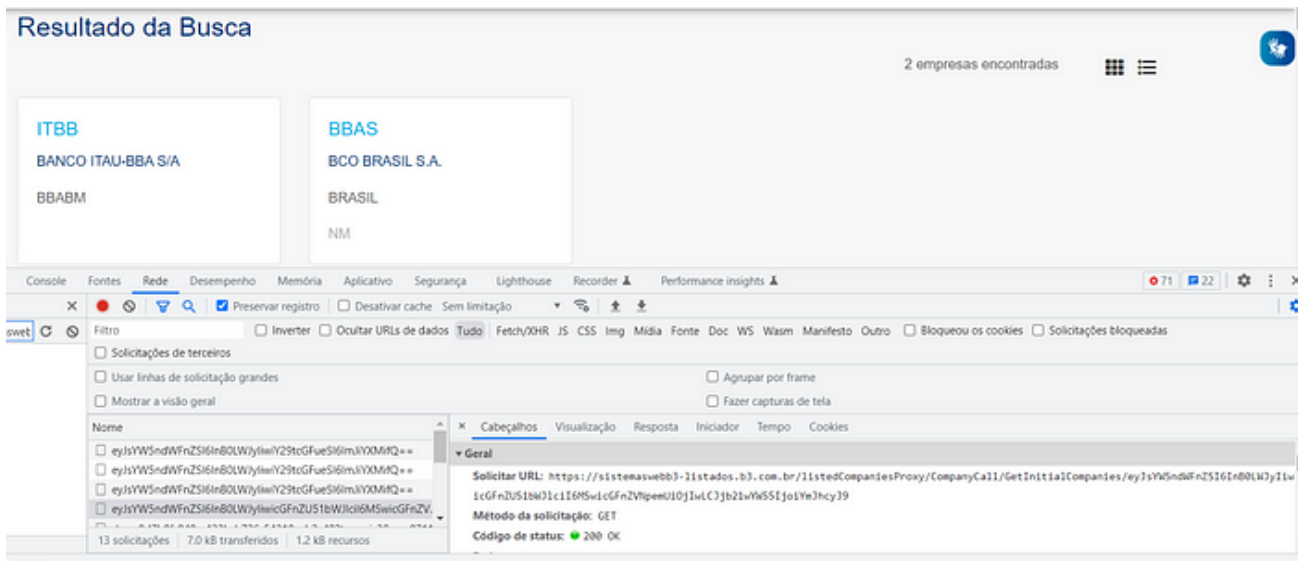
1 r.json()
{'page': {'pageNumber': 1,
'pageSize': 99999,
'totalRecords': 225,
'totalPages': 1},
'results': [{'typeStock': 'ON',
'dateApproval': '28/07/2022',
'valueCash': '2,938861',
'ratio': '1',
'corporateAction': 'DIVIDENDO',
'lastDatePriorEx': '11/08/2022',
'dateClosingPricePriorExDate': '11/08/2022',
'closingPricePriorExDate': '38,96',
'quotedPerShares': '1',
'corporateActionPrice': '7,543278',
'lastDateTimePriorEx': '2022-08-11T00:00:00'},
{'typeStock': 'ON',
'dateApproval': '28/07/2022',
'valueCash': '3,366001'.

```

Obtendo o TradingName

O parâmetro *TradingName* da API não é um nome padronizado, mas é possível obtê-lo através do código da ação.

Novamente na [página de empresas listadas](#), ao pesquisar a empresa, o site faz a requisição de uma API onde tem informações da empresa, incluindo o TradingName, como no exemplo:



Decodificando vemos que a API de busca usa o código da empresa no parâmetro “company”. E no resultado conseguimos o TradingName.

Nesse exemplo foi pesquisado o código da ação do Banco do Brasil (BBAS), mas a pesquisa também retornou o Itaú BBA por também conter “BBAS” no nome. Isso pode acontecer em alguns casos. Para pegar o TradingName certo é só pesquisar pelo código da ação no json como mostrado abaixo.

```

1 b64decode('eyJ3Y5ndWFnZSI6InB0LWJyIiwicGFnZU51bWJlciI6MSwicGFnZVpemUiojIwLCJjb21wYW55IjoimJhcyJ39')
2 b'{"language": "pt-br", "pageNumber": 1, "pageSize": 20, "company": "bbas"}'

1 r = requests.get(r'https://sistemaswebb3-listados.b3.com.br/listedCompaniesProxy/CompanyCall/GetInitialCompanies/' +
2 'eyJ3Y5ndWFnZSI6InB0LWJyIiwicGFnZU51bWJlciI6MSwicGFnZVpemUiojIwLCJjb21wYW55IjoimJhcyJ39')
3
4 r.json()

{
  'page': {
    'pageNumber': 1,
    'pageSize': 20,
    'totalRecords': 2,
    'totalPages': 1
  },
  'results': [
    {
      'codeCVM': '904516',
      'issuingCompany': 'ITBB',
      'companyName': 'BANCO ITAU-BBA S/A',
      'tradingName': 'BBABM',
      'cnpj': '17298092000130',
      'marketIndicator': '1',
      'typeBDR': '',
      'dateListing': '31/12/9999',
      'status': 'A',
      'segment': 'Não Classificados',
      'segmentEng': 'Not Classified',
      'type': '7',
      'market': ''
    },
    {
      'codeCVM': '1023',
      'issuingCompany': 'BBAS',
      'companyName': 'BCO BRASIL S.A.',
      'tradingName': 'BRASIL',
      'cnpj': '191'
    }
  ]
}

1 tradingName = [i['tradingName'] for i in r.json()['results'] if i['issuingCompany'] == 'BBAS']
2 print(tradingName)

['BRASIL']

```

Script final do Python

Temos todas as APIs necessárias para criar o código para obter os proventos de uma empresa pelo código da ação. Segue abaixo o código:


```
import requests
from base64 import b64encode

# Função para buscar o trading name
def get_trading_name(ticker):
    # cria os parâmetros
    params = {"language": "pt-br", "pageNumber": 1, "pageSize": 20, "company": ticker}
    #params = '{"language": "pt-br", "pageNumber": 1, "pageSize": 20, "company": "' + ticker + "'"

    # para codificar em base64 é preciso fazer a conversão dos parâmetros em bytes
    params = bytes(str(params), encoding="ascii")

    # faz a codificação em formato bytes
    string = b64encode(params)

    # decodifica para transformar bytes em string
    string = string.decode()

    # faz a requisição com os parâmetros
    r = requests.get(r'https://sistemaswebb3-listados.b3.com.br/listedCompaniesProxy/CompanyCall/GetInitialCompanies/'+
                    string)

    # retorna o trading name da empresa, (é necessário remover pontos e barras)
    for i in r.json()['results']:
        if i['issuingCompany'].lower() == ticker.lower():
            return i['tradingName'].replace('/', '').replace('.', '')

    # Se a empresa não for encontrada, retorna a mensagem de erro
    raise ValueError('Empresa não encontrada')

# Função para obter os proventos
def proventos(ticker):
    # encontra o trading name da empresa
    tradingName = get_trading_name(ticker)

    # cria os parâmetros com o trading name
    params = {"language": "pt-br", "pageNumber": 1, "pageSize": 99999, "tradingName": tradingName}

    # codifica os parâmetros em base64
    params = bytes(str(params), encoding="ascii")
    string = b64encode(params)
    string = string.decode()

    r = requests.get('https://sistemaswebb3-listados.b3.com.br/listedCompaniesProxy/CompanyCall/GetListedCashDividends/'+
                    string)

    return r.json()
```

Exemplos de execução:

```

1 proventos('bbas')

{'page': {'pageNumber': 1,
'pageSize': 99999,
'totalRecords': 225,
'totalPages': 1},
'results': [{'typeStock': 'ON',
'dateApproval': '16/08/2022',
'valueCash': '0,2737355124',
'ratio': '1',
'corporateAction': 'JRS CAP PROPRIO',
'lastDatePriorEx': '12/09/2022',
'dateClosingPricePriorExDate': '12/09/2022',
'closingPricePriorExDate': '41,06',
'quotedPerShares': '1',
'corporateActionPrice': '0,666672',
'lastDateTimePriorEx': '2022-09-12T00:00:00'},
{'typeStock': 'ON',
'dateApproval': '08/08/2022',
'valueCash': '0,20018899819',
'ratio': '1',
'corporateAction': 'JRS CAP PROPRIO'}]}

```

```
1
```

```
1 proventos('viia')
```

```

{'page': {'pageNumber': 1,
'pageSize': 99999,
'totalRecords': 46,
'totalPages': 1},
'results': [{'typeStock': 'ON',
'dateApproval': '26/04/2018',
'valueCash': '0,0119353731',
'ratio': '1',
'corporateAction': 'DIVIDENDO',
'lastDatePriorEx': '26/04/2018',
'dateClosingPricePriorExDate': '26/04/2018',
'closingPricePriorExDate': '8,08',
'quotedPerShares': '1',
'corporateActionPrice': '0,147715',
'lastDateTimePriorEx': '2018-04-26T00:00:00'},
{'typeStock': 'ON',
'dateApproval': '12/12/2017',
'valueCash': '0,0270351356',
'ratio': '1',
'corporateAction': 'JRS CAP PROPRIO'}]}

```

Exemplo em dataframe do Pandas:

1

import pandas as pd

2

resultados = proventos('embr')

3

df=pd.DataFrame(resultados['results'])

4

df

	typeStock	dateApproval	valueCash	ratio	corporateAction	lastDatePriorEx	dateClosingPricePriorExDate	closingPricePriorExDate	quotedPerShares	corpc
0	ON	14/12/2018	0,01	1	DIVIDENDO	26/12/2018	26/12/2018	21,47	1	
1	ON	13/09/2018	0,01	1	DIVIDENDO	24/09/2018	24/09/2018	19,02	1	
2	ON	14/06/2018	0,02	1	JRS CAP PROPRIO	25/06/2018	25/06/2018	23,99	1	
3	ON	12/04/2018	0,07201612	1	DIVIDENDO	23/04/2018	23/04/2018	23,70	1	
4	ON	05/03/2018	0,02	1	JRS CAP PROPRIO	15/03/2018	15/03/2018	20,82	1	
...
115	PN	24/03/2000	0,042	1	JRS CAP PROPRIO	24/03/2000	24/03/2000	8,39	1	
116	PN	31/01/2000	0,1869	1	DIVIDENDO	31/01/2000	31/01/2000	8,10	1	
117	PN	28/09/1999	0,08	1	DIVIDENDO	28/09/1999	28/09/1999	5,58	1	
118	PNA	30/03/1999	0,734	1000	DIVIDENDO	30/03/1999	25/03/1999	17,90	1000	
119	PNA	18/09/1998	0,461	1000	DIVIDENDO	18/09/1998	18/09/1998	14,99	1000	

120 rows x 11 columns

É importante saber que existe uma infinidade de formas que cada site pode fazer as requisições das informações. Em alguns podemos obter os dados dessa forma simples, mas em muitos isso não é possível.