

A primeiro momento, tenho as seguintes observações:

```
33 @app.get('/ObterDebitos')
34 def get_retorna_debitos(renavam: str, placa: str, cpf: str):
35     renavam = renavam
36     plac = placa
37     cp = cpf
38     driver = webdriver.Chrome(executable_path='C:\chromedriver\chromedriver.exe')
39     driver.get("http://www.sefaz2.to.gov.br/ipva/dare.php")
40     driver.maximize_window()
41     time.sleep(5)#esperar a página carregar
42     renavam = driver.find_element_by_xpath("//input[@name='renavam']")
43     renavam.send_keys(renavam)
44     placa = driver.find_element_by_xpath("//input[@name='placa']")
45     placa.send_keys(placa)
46     time.sleep(4)
47     driver.find_element_by_xpath("/html/body/div/div[3]/div[2]/div/div/form/div[3]/div/input").click()
48     time.sleep(4)
49     cpf = driver.find_element_by_xpath("//input[@name='IPVINRDO']")
50     cpf.send_keys(cp)
51     driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[2]/tbody/tr/td[1]/p[3]/font/a/img").click()
```

- As linhas destacadas podem executar o comando “sen.keys()” na mesma linha e sem a necessidade de armazenar numa variável, pois o código não usa essa variável posteriormente e deixa espaço sendo ocupado, da seguinte forma:

```
33 @app.get('/ObterDebitos')
34 def get_retorna_debitos(renavam: str, placa: str, cpf: str):
35     renavam = renavam
36     plac = placa
37     cp = cpf
38     driver = webdriver.Chrome(executable_path='C:\chromedriver\chromedriver.exe')
39     driver.get("http://www.sefaz2.to.gov.br/ipva/dare.php")
40     driver.maximize_window()
41     time.sleep(5)#esperar a página carregar
42     driver.find_element_by_xpath("//input[@name='renavam']").send_keys(renavam)
43     driver.find_element_by_xpath("//input[@name='placa']").send_keys(placa)
44     time.sleep(4)
45     driver.find_element_by_xpath("/html/body/div/div[3]/div[2]/div/div/form/div[3]/div/input").click()
46     time.sleep(4)
47     driver.find_element_by_xpath("//input[@name='IPVINRDO']").send_keys(cp)
48     driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[2]/tbody/tr/td[1]/p[3]/font/a/img").click()
```

- Outro detalhe importante são os “time.sleep(t)”. O comando força o código a esperar todo o tempo informado como parâmetro na função, enquanto, se usarmos o comando “driver.implicitly_wait(t)”, o código irá, dentro do intervalo de tempo informado como parâmetro, esperar que a página carregue. Assim, se o elemento carregar antes do tempo informado, ele irá executar o comando, poupando o restante do tempo que seria gasto se esperasse o intervalo de tempo inteiro.

```
50 Nome = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/span").text
51 Endereco = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/font/").text
52 Bairro = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/font/sp").text
53 Municipio = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/font/sp").text
54 MarcaModelo = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/font/sp").text
55 Fabricacao = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/font/sp").text
56 ValorDare = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[7]/tbody/tr/td[2]/strong/font/p[2]/font").text
57 Observacao = driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[7]/tbody/tr/td[2]/strong/font/p[1]/font").text
```

- O código não buscava corretamente o valor DARE e também não buscava as informações sobre o código de barras do pagamento: a variável ValorDare busca um caminho que não tem conteúdo:

```
<td valign="top" align="LEFT">
  <input type="hidden" name value>
  <span id="span_" style="background:#ffffff;font-family:'Arial'; font-size:12pt; font-weight:normal; font-style:normal"></span>
```

A caixa span está sem texto algum, o que deixa a variável vazia. No entanto, o valor DARE pode ser encontrado dentro do PDF que pode ser baixado na “table[8]”, referente ao ano de 2022 e com todos os dados necessários que o código precisa. Além disso, não existe uma variável referente ao código de barras.

- Certa atenção ao uso da função “find_element_by_xpath”, que foi depreciada:

```
ProjetoAPI\main.py:134: DeprecationWarning: find_element_by_xpath is deprecated.  
ValorDare = driver.find_element_by_xpath(f"{table_7}p[2]/table/tbody/tr[2]/td/span").text  
f:ProjetoAPI\main.py:136: DeprecationWarning: find_element_by_xpath is deprecated.  
Observacao = driver.find_element_by_xpath(f"{table_7}p[1]/span").text  
f:ProjetoAPI\main.py:141: DeprecationWarning: find_element_by_xpath is deprecated.  
driver.find_element_by_xpath("/html/body/form/table/tbody/tr[1]/td/table[6]/tbody/tr/td/p/a").click()  
Boleto com desconto: 85650000001-8 84000173202-2 20117804767-0 22206389130-3  
Data de vencimento com desconto: 17/01/2022  
Deseja fechar?
```

E as linhas que usavam essa função foram corrigidas:

```
Nome = driver.find_element(by=By.XPATH, value=f"{table_4}span").text  
Endereco = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[1]").text  
Bairro = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[2]").text  
Município = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[3]").text  
MarcaModelo = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[4]").text  
Fabricacao = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[5]").text  
ValorDare = driver.find_element(by=By.XPATH, value=f"{table_7}p[2]/table/tbody/tr[2]/td/span").text  
try:  
    Observacao = driver.find_element(by=By.XPATH, value=f"{table_7}p[1]/span").text  
except NoSuchElementException:  
    Observacao = not_info
```

- A condicional “try - except” foi usada devido a observação
- O código não identifica local e nem baixa o arquivo PDF, e assim não consegue visualizar seu conteúdo, falhando em coletar informações que deveriam ser retornadas.
- Como o código não era competente em pegar o CPF, foi adicionado ao código linhas que baixa o PDF e buscam as informações dentro dele:

```
33  
34 def obterNomeDoDownload(waitTime, driver):  
35     driver.execute_script("window.open()")  
36     # switch to new tab  
37     driver.switch_to.window(driver.window_handles[-1])  
38     # navigate to chrome downloads  
39     driver.get('chrome://downloads')  
40     # define the endTime  
41     endTime = time.time()+waitTime  
42     while True:  
43         try:  
44             # get downloaded percentage  
45             downloadPercentage = driver.execute_script(  
46                 "return document.querySelector('downloads-manager').shadowRoot.querySelector('#downloadsList downloads-it")  
47             # check if downloadPercentage is 100 (otherwise the script will keep waiting)  
48             if downloadPercentage == 100:  
49                 # return the file name once the download is completed  
50                 return driver.execute_script("return document.querySelector('downloads-manager').shadowRoot.querySelector(  
51             except:  
52                 pass  
53                 time.sleep(1)  
54                 if time.time() > endTime:  
55                     break
```

- Essa função servirá de auxiliar para o código, e atuará para encontrar o nome do arquivo que foi baixado pelo código anteriormente.

```
83 ▼ def extrairDadosPdf(nome_pdf):
84     pdfFileObj = open(f'C:\\Boletos_Dare\\{nome_pdf}', 'rb')
85     # Cria um objeto de leitura do PDF
86     pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
87     # Cria uma página do objeto
88     pageObj = pdfReader.getPage(0)
89     # Extrai o texto da página
90     texto = pageObj.extractText()
91     index = texto.find("DescontoParcela")
92     valor_boleto_com_desconto, boleto_com_desconto, data_vencimento_com_desconto = acharDadosBoleto(index, texto)
93     index2 = texto.find("DescontoParcela", index+1)
94     valor_boleto_sem_desconto, boleto_sem_desconto, data_vencimento_sem_desconto = acharDadosBoleto(index2, texto)
95     pdfFileObj.close()
96
97     return boleto_com_desconto, valor_boleto_com_desconto, data_vencimento_com_desconto, boleto_sem_desconto, valor_boleto
```

- Já essa função extrai os dados do PDF baixado pelo bot. Os dados extraídos do boleto são os seguintes: Número do código de barras de cada parcela, valor de cada parcela e data de vencimento de cada parcela. Os demais dados que são retornados pelo código são extraídos do próprio código fonte da página do Sefaz.

```
56 def acharDadosBoleto(index, texto):
57     tamanho_texto = len(texto)
58     boleto = ""
59     data_vencimento = ""
60     numeros = ""
61     achei_num = False
62     index_valor = 0
63     valor_boleto = ""
64     for k in range(index, tamanho_texto):
65         if texto[k].isdigit() or (achei_num == True and ( texto[k] == " " or texto[k] == "-" ) ):
66             numeros += texto[k]
67             achei_num = True
68         elif texto[k] == "/":
69             boleto = numeros[0: -2]
70             numeros = numeros[-2:]
71             data_vencimento += numeros + texto[k:k+8]
72             index_valor = k+8
73             break
74     for l in range(index_valor, tamanho_texto):
75         if texto[l] == ',':
76             valor_boleto += texto[l:l+3]
77             break
78         else:
79             valor_boleto += texto[l]
80     return valor_boleto, boleto, data_vencimento
```

- Essa função auxilia a função “extrairDadosPDF”. Essa função será usada para retirar do texto do PDF dados específicos, que são: número do código de barras e data de vencimento. Essa função aproveita um padrão do conteúdo do PDF para, no texto, retirar as informações almeçadas.

```

98 def get_retorna_debitos(renavam: str, placa: str, cpf: str):
99     download_dir = "C:\\Boletos_Dare" # Para o linux/*nix, download_dir="/usr/Public"
100     options = webdriver.ChromeOptions()
101     caps = DesiredCapabilities().CHROME
102     caps["pageLoadStrategy"] = "eager"
103     profile = {"plugins.plugins_list": [{"enabled": False, "name": "Chrome PDF Viewer"}], # Disable Chrome's PDF Viewer
104             "download.default_directory": download_dir, "download.extensions_to_open": "applications/pdf",
105             "directory_upgrade": True, "download.prompt_for_download": False, #To auto download the file
106             "plugins.always_open_pdf_externally": True }
107     options.add_experimental_option("prefs", profile)
108     options.add_argument("user-agent=Mozilla/5.0 (iPhone; CPU iPhone OS 12_3_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML,
109
110     renav = renavam
111     plac = placa
112     cp = cpf
113     driver = webdriver.Chrome(executable_path='F:\\chromedriver_win32\\chromedriver.exe', chrome_options=options)
114     driver.get("http://www.sefaz2.to.gov.br/ipva/dare.php")
115     driver.maximize_window()
116     driver.find_element(by=By.XPATH, value="//input[@name='renavam']").send_keys(renav)
117     driver.find_element(by=By.XPATH, value="//input[@name='placa']").send_keys(plac)
118     driver.implicitly_wait(8) #Espera até 8 segundos para um elemento da páginas carregar. Se ele carregar antes, ele prosseguir
119
120     driver.find_element(by=By.XPATH, value="/html/body/div/div[3]/div[2]/div/div/form/div[3]/div/input").click()
121     driver.find_element(by=By.XPATH, value="//input[@name='IPVINRDO']").send_keys(cp)
122     driver.find_element(by=By.XPATH, value="/html/body/form/table/tbody/tr[1]/td/table[2]/tbody/tr/td[1]/p[3]/font/a/img").
123
124     #Criação para redução da repetição de texto
125     table_4 = '/html/body/form/table/tbody/tr[1]/td/table[4]/tbody/tr/td[2]/dl/dt[1]/div/'
126     table_7 = '/html/body/form/table/tbody/tr[1]/td/table[7]/tbody/tr/td[2]/strong/font/'
127     not_info = "Sem informações"
128
129     Nome = driver.find_element(by=By.XPATH, value=f"{table_4}span").text
130     Endereco = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[1]").text
131     Bairro = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[2]").text
132     Municipio = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[3]").text
133     MarcaModelo = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[4]").text
134     Fabricacao = driver.find_element(by=By.XPATH, value=f"{table_4}font/span[5]").text
135     ValorDare = driver.find_element(by=By.XPATH, value=f"{table_7}p[2]/table/tbody/tr[2]/td/span").text
136     try:
137         Observacao = driver.find_element(by=By.XPATH, value=f"{table_7}p[1]/span").text
138     except NoSuchElementException:
139         Observacao = not_info
140
141     driver.find_element(by=By.XPATH, value="/html/body/form/table/tbody/tr[1]/td/table[6]/tbody/tr/td/p/a").click()
142     ultimo_pdf = obterNomeDoDownload(180, driver) #Espera 3 minutos para baixar o boleto
143
144     b_c_d, v_c_d, d_c_d, b_s_d, v_s_d, d_s_d = extrairDadosPdf(ultimo_pdf)
145
146     """
147     b_c_d = Código do Boleto com Desconto
148     v_c_d = Valor do Boleto com Desconto
149     d_c_d = Data de Vencimento do Boleto com Desconto
150     b_s_d = Código do Boleto sem Desconto
151     v_s_d = Valor do Boleto sem Desconto
152     d_s_d = Data de Vencimento do Boleto sem Desconto
153     """
154     return {"CPF": cp, "Renavam": renav, "Placa": plac, "Nome": Nome, "Endereco": Endereco, "Bairro": Bairro,
155           "Municipio": Municipio, "MarcaModelo": MarcaModelo, "Fabricacao": Fabricacao, "ValorDare": ValorDare,
156           "Observacao": Observacao, "Valor Boleto com Desconto": v_c_d, "Valor Boleto sem Desconto": v_s_d,
157           "Data de Vencimento com Desconto": d_c_d, "Data de Vencimento sem Desconto": d_s_d,
158           "Codigo do Boleto com Desconto": b_c_d, "Codigo do Boleto sem Desconto": b_s_d}
159

```

- Essa função é a função principal do projeto. Ela faz a chamada para todas as outras funções mostradas anteriormente. A função deverá retornar, em um JSON com todas as informações coletadas pelo código. A função define as configurações do bot usado para acessar a página, efetivamente usa o bot para entrar na página, coleta os dados que são mais fáceis de tirar da página: Nome, endereço, Bairro, Municipio, marca e o modelo e a fabricação do carro. Em seguida, o bot baixa o PDF e investiga o PDF da forma demonstrada anteriormente pelas funções já explicadas. Ao designar os dados encontrados às suas respectivas variáveis, então ela retorna os dados.

- Em suma, o código configura um bot otimizado para visitar a página do site, coletar certos dados dessa página e baixa, posteriormente, um PDF com outros dados a serem coletados. Após baixar, o bot lê o PDF para extrair os dados faltando e guarda em variáveis. Com isso, retorna tudo o que foi coletado.