

In [82]:

```
from selenium import webdriver
import json
import time
import pandas as pd
```

In [2]:

```
driver = webdriver.Chrome()
driver.get('https://auto.ru/catalog/cars/all/?view_type=list&year_from=2019')
```

In [5]:

```
models_links = []
next_button = driver.find_element_by_class_name('pager__next')

while True:
    time.sleep(1)
    elem = driver.find_element_by_class_name('catalog-all-text-list')
    models_links += [x.get_attribute('href') for x in elem.find_elements_by_tag_name('a')]
    time.sleep(1)
    if next_button.is_enabled():
        next_button.click()
    else:
        break

with open('models_links.json', 'w') as out_file:
    json.dump({'links': models_links}, out_file)
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-5-00866a633364> in <module>
    16
    17 with open('models_links.json', 'w') as out_file:
--> 18     json.dump({'links': links})
```

**TypeError:** dump() missing 1 required positional argument: 'fp'

In [38]:

```
with open('models_links.json', 'r') as models_links_file:
    models_links = json.load(models_links_file)['links']
```

In [18]:

```
driver.get(links[110])
```

In [21]:

```
elem = driver.find_elements_by_class_name('catalog-all-text-list__desc')
```

In [36]:

```
elem[0].find_elements_by_class_name('mosaic__title')[0].get_attribute('href')
```

Out[36]:

```
'https://auto.ru/catalog/cars/buick/excelle/21221217/21221252/?year_from=2019'
```

In [39]:

```
all_models_links = []
for models_link in models_links:
    driver.get(models_link)
    elem = driver.find_elements_by_class_name('catalog-all-text-list__desc')
    for e in elem:
        models_types = e.find_elements_by_class_name('mosaic__title')
        for model_type in models_types:
            all_models_links.append(model_type.get_attribute('href'))
with open('all_models_links.json', 'w') as out_file:
    json.dump({'links': all_models_links}, out_file)
```

In [81]:

```
for link in all_models_links:
    end_index = link.index('?')
    driver.get(link[:end_index]+'specifications/?year_from=2019')
    models_repres = driver.find_element_by_class_name('catalog-table__inner').find_elements
    for repres in models_repres:
        repres.click()
        #считываем характеристики
        comp = driver.find_elements_by_link_text('Комплектации')
        if len(comp) != 0:
            comp[0].click()
            # считываем комплектацию
            driver.find_element_by_link_text('Характеристики').click()
```

In [ ]: