In [2]:
```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:80% !important; }</
import pandas as pd
import numpy as np
import math
import datetime as dt
import glob
import os

# pre setting
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

In [3]:
```
## read all files in a folder

print(os.getcwd())

# find all roads with idle/s
def find_idle_count_for_roads(df):

    df_of_zeros_rows = df.loc[df['speed'] == 0] # return a
    list_road = np.unique(df_of_zeros_rows['road_name'],re
    output_df = pd.DataFrame(list_road[0].tolist(), column
    output_df['idle_count'] = list_road[1].tolist() # seco

    return output_df

# output = find_idle_count_for_roads(df)

def pick_road_with_long_idle(output_df, length=5):
    df["bus_name"] = ""
    return output_df[output_df['idle_count']>length] # ret

# pick_road_with_long_idle(output)


```

```
X:\506Final
```

In [18]:

```
# def process_one_day(path):
for file in os.listdir("API_dataset/"):
    print(file)
    all_files = glob.glob("API_dataset/"+ file + "/*.csv")
    frames = []

    for filename in all_files:
        bus_name = filename.replace(("API_dataset/"+file+'
        df = pd.read_csv(filename, index_col=None, header=
        output = find_idle_count_for_roads(df)
        output = pick_road_with_long_idle(output)
        output["bus_name"] = bus_name.replace('API_dataset/
        frames.append(output)
        result = pd.concat(frames)

        if(filename==file+"\WB958.csv"):
            break
    # print(result)
    print("done for files in", file)
    print(result)
    result.to_csv("_processed"+file+".csv")
```

```
7     B306
23                              Industrial Dr          1
8     B306
38                                   River St
8     B306
46                                Townsend St
8     B306
4                                   Bowdoin St
6     B308
11                              Industrial Dr          1
0     B308
18                                   River St
8     B308
27                              Washington St
6     B308
16                              Industrial Dr
6     B310
24                    RT-145 / Bennington St
8     B310
```

In [ ]:

```
## return final result
all_processed_files = glob.glob("*.csv") # read all proces

li = []
for filename in all_processed_files: # put all processed f
    df_temp = pd.read_csv(filename)
    li.append(df_temp)
final = pd.concat(li, axis=0, ignore_index=True)
final = final.sort_values(by=['road_name'])

final = final.groupby('road_name')['idle_count'].sum().res
final = final[final['idle_count'] >10000]

# ax = final.plot.bar(x=final['road_name'],rot=0)
# final
final.groupby("road_name")['idle_count'].mean().plot(kind=
```

In [16]:
```
## find the highest idling time route
list_idlenum = []
list_bus = []
for file in os.listdir("X:/506Final/dataset/"):
    all_files = glob.glob("X:/506Final/dataset/"+ file + "/
    for filename in all_files:
        bus_name = filename.replace(("X:/506Final/dataset/
        df = pd.read_csv(filename, index_col=None, header=0
        output = find_idle_count_for_roads(df)
        busName = bus_name.replace(file,'')
        list_idlenum.append((output['idle_count']).sum())
        list_bus.append(busName.replace(file,''))
        #print(len(list_idlenum),len(list_bus))
total_idle = pd.DataFrame(list_idlenum,columns=['idle_num']
total_idle['bus_name'] = list_bus

total_idle = total_idle.groupby('bus_name')['idle_num'].sum
total_idle = total_idle.sort_values(by=['idle_num'])
total_idle.tail(10)

```

Out[16]:

	bus_name	idle_num
107	B425	2696
687	WB922	2698
351	HS294	2699
228	B546	2742
368	HS311	2753
427	HS371	2761
378	HS321	2791
298	HS241	2807
510	HS454	3101
363	HS306	3152

In [4]:
```
1
```

B297