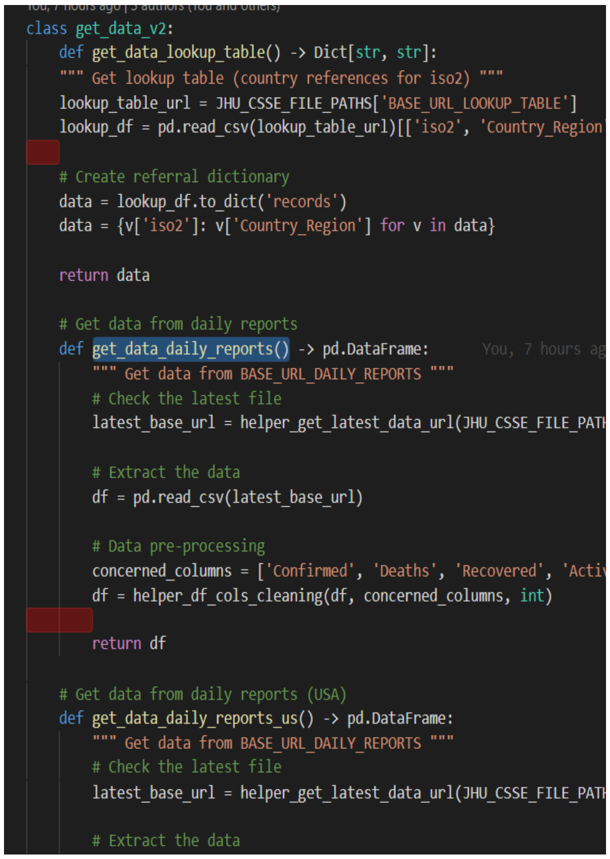
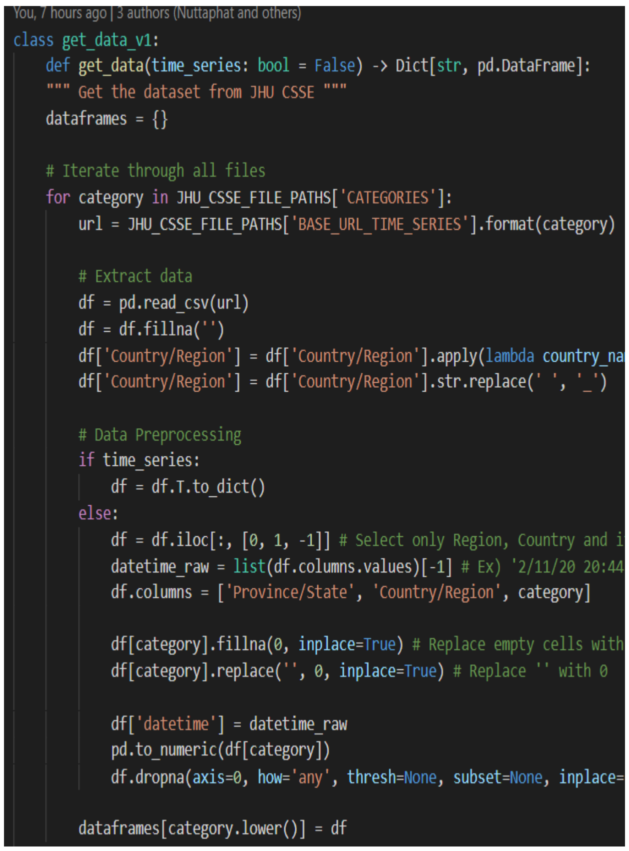
Student Name	Kanisha Patel
Student ID	216006595
GitHub User ID	Kanisha14
GitHub Email	Kanishapatel14@gmail.com
Chosen Project	Proj_2: COVID19 API
Project Repository url	https://github.com/Kanisha14/covid19-api
First Pull Request Number	#143, branchName = "Versions_intoClass"
Second Pull Request Number	#146 branchName = "Integrator_docFixes"

Aggregate pattern in Domain-Driven Design usually refers the several similar objects grouped together that can be treated as one unit for the purpose of the data changes.

The **first pull request**, "Wrapper Class for V1 & V2" makes the changes in the file *get_data.py*. It have several methods like get_data_daily_reports(), get_data_time_series(), get_data() for our API version 1 and version 2. Two wrapper class get_data_v1 and get_data_v2 which groups the methods of version 1 in one class and version two in different classes, respectively. The aggregates are formed when classes form a "Has-a" relationship with its method.



```
class get_data_v2:
    def get_data_lookup_table() -> Dict[str, str]:
    """ Get lookup table (country references for iso2) """
    lookup_table_url = JHU_CSSE_FILE_PATHS['BASE_URL_LOOKUP_TABLE']
    lookup_df = pd.read_csv(lookup_table_url)[['iso2', 'Country_Region

    # Create referral dictionary
    data = lookup_df.to_dict('records')
    data = {v['iso2']: v['Country_Region'] for v in data}

    return data

    # Get data from daily reports
    def get_data_daily_reports() -> pd.DataFrame:      You, 7 hours ag
        """ Get data from BASE_URL_DAILY_REPORTS """
        # Check the latest file
        latest_base_url = helper_get_latest_data_url(JHU_CSSE_FILE_PATH

        # Extract the data
        df = pd.read_csv(latest_base_url)

        # Data pre-processing
        concerned_columns = ['Confirmed', 'Deaths', 'Recovered', 'Acti
        df = helper_df_cols_cleaning(df, concerned_columns, int)

        return df

    # Get data from daily reports (USA)
    def get_data_daily_reports_us() -> pd.DataFrame:
        """ Get data from BASE_URL_DAILY_REPORTS """
        # Check the latest file
        latest_base_url = helper_get_latest_data_url(JHU_CSSE_FILE_PATH

        # Extract the data
```

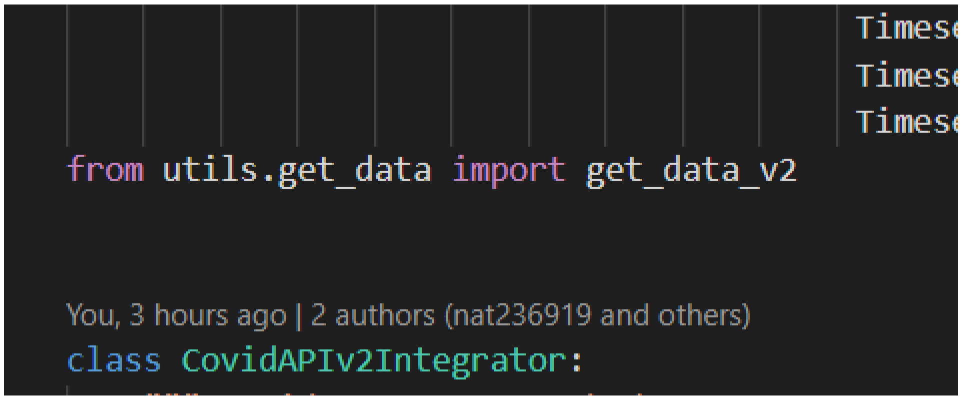


```
You, 7 hours ago | 3 authors (Nuttaphat and others)
class get_data_v1:
    def get_data(time_series: bool = False) -> Dict[str, pd.DataFrame]:
    """ Get the dataset from JHU CSSE """
    dataframes = {}

    # Iterate through all files
    for category in JHU_CSSE_FILE_PATHS['CATEGORIES']:
        url = JHU_CSSE_FILE_PATHS['BASE_URL_TIME_SERIES'].format(category)

        # Extract data
        df = pd.read_csv(url)
        df = df.fillna('')
        df['Country/Region'] = df['Country/Region'].apply(lambda country_na
        df['Country/Region'] = df['Country/Region'].str.replace(' ', '_')

        # Data Preprocessing
        if time_series:
            df = df.T.to_dict()
        else:
            df = df.iloc[:, [0, 1, -1]] # Select only Region, Country and i
            datetime_raw = list(df.columns.values)[-1] # Ex) '2/11/20 20:44
            df.columns = ['Province/State', 'Country/Region', category]

            df[category].fillna(0, inplace=True) # Replace empty cells with
            df[category].replace('', 0, inplace=True) # Replace '' with 0

            df['datetime'] = datetime_raw
            pd.to_numeric(df[category])
            df.dropna(axis=0, how='any', thresh=None, subset=None, inplace=

        dataframes[category.lower()] = df
```

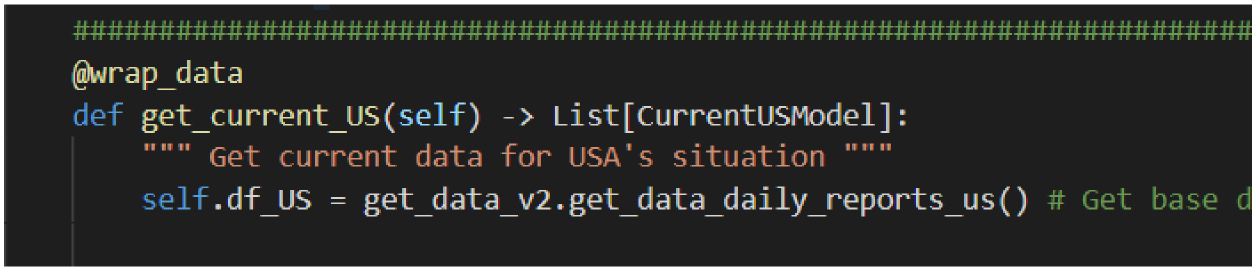
Hence, get_data_v1 has-a method get_data() that functions for the API version 1. Similarly, get_data_v2 has-a methods like get_data_lookup_table(), get_data_daily_reports() etc.

Each Aggregates has a root, i.e in our case will be our two classes, get_data_v1 and get_data_v2 and it has a boundary i.e all the methods inside each class.

Furthermore, the pull request also makes changes in the file *covid_api_v2_Integrator.py* that initially imported few of the methods for version 2, but now import the class get_data_v2. Hence it also further makes changes to all the place where the method is called. Now it calls the same method but via our class for version 2.



```
                                                  Timese
                                                  Timese
                                                  Timese
from utils.get_data import get_data_v2


You, 3 hours ago | 2 authors (nat236919 and others)
class CovidAPIv2Integrator:
```



```
##########################################################################
@wrap_data
def get_current_US(self) -> List[CurrentUSModel]:
    """ Get current data for USA's situation """
    self.df_US = get_data_v2.get_data_daily_reports_us() # Get base d
```

Similar changes need to be made in *covid_api_v1_Integrator.py* file for version 1.

The name get_data_v1 and get_data_v2 comes from the filename get_data.py and v1, v2 represents version 1 and version 2 respectively. Since its easier to understand what the class i.e., our root represents and what its functions are.

The **Second pull request**, “Separate Wrapper Classes for method in Integrator file” makes changes in the file *covid_api_v2_Integrator.py*. this file initially had several methods like get_country(), get_current(), get_current_US(), get_confirmed(), get_deaths() and get_time_series() and many more. The pull request made group all the method with similar functionality into one class, hence forming a cluster of objects that can be used as single unit. Hence the classes being the root of the aggregate pattern formed and its respective methods being the boundary of that aggregate. Below table shows all the classes and its subsequent methods.

Roots	Boundary	Reason for the Name

CovidAPIv2Integrator	def __init__(self)
	def wrap_data(func)
	get_country()
		Class name is similar to the file name as it is the main class of the file.

CovidAPIv2Integrator_current	get_current()
	get_current_US()
		This class groups all the methods that pulls the current data. Hence the current name appears on the end.

CovidAPIv2Integrator_Status	get_confirmed()
	get_deaths()
	get_recovered()
	get_active()
	get_total()
		Similarly, this class get the total STATUS by summing all the cases together for respective fields. Hence status is added at the end.

CovidAPIv2Integrator_timeseries	get_time_series()
	__extract_time_series()
	__extract_time_series_global()
	get_US_time_series()
	__extract_US_time_series()
		This class retrieves the timeseries for different regions and hence the name.