**WARD DATA EXTRACTION:**

**Step 1: Download PDF:**

PDF containing ward data is downloaded from <https://stopcoronavirus.mcgm.gov.in/key-updates-trends> in the code file: **/tasks/districts/FetchMumbaiWardsTask.py** (the generic download file function is available in **/tasks/districts/DownloadFileTask.py**), and stored in a local directory

**Step 2: Identify relevant page and extract data, and save raw-data to AWS**

The function for extracting data from the PDF downloaded above is called in **/tasks/districts/FetchMumbaiWardsTask.py** (scrape\_mumbai\_pdf()). All the required functions are defined and stored in the file: **/backend/data/extract\_mumbai\_wards\_pdf.py**.

The scrape\_mumbai\_pdf() function in turn calls two other functions in succession:

1. find\_ward\_wise\_breakdown\_page() : This function tries to identify the page titled “Ward-wise breakdown of positive cases” in the PDF.
2. \_extract\_wards\_data\_from\_page(): This function tries to identify the bounding-box containing the cumulative active/recovered/deceased counts in the ward-wise breakdown page identified above, and then extracts the data and returns a pandas dataframe containing the ward-wise cumulative counts

The data extracted above is then saved to the AWS S3 Storage bucket in **/tasks/fetch\_ward\_data.py**. This code tries to check if an older version of the ward-data (“raw\_ward\_data.csv”) is available in the S3 bucket. If it exists, then the new data is appended to it and saved in the S3 bucket. If it doesn’t exist, the code creates a new file in the S3 bucket with the data extracted above.

**Step 3: Interpolate values for missing dates, and create delta variables, and save processed ward-data to AWS**

The raw ward data extracted above could have a few dates missing due to code-failures. To deal with this, the code in **/tasks/fetch\_ward\_data.py** also tries to interpolate the cumulative active/recovered/deceased counts for the missing dates using a linear-interpolation algorithm. The function used to interpolate (interpolate\_values()) is available at **/backend/data/utility.py.**

After the interpolation, the daily new cases/deaths/recoveries are calculated using a function (create\_delta\_cols()) available in **/backend/data/utility.py.** The final dataset is saved as **“Phase 2 – Wards.csv”** in the S3 bucket (RT/DT values are calculated later and appended to this CSV file itself)

**COMMONLY ENCOUNTERED ISSUES:**

**The ward page on the dashboard could encounter issues such as:**

1. **Erroneous ward-names showing up on the list:** This could mean that the format of the page in the PDF containing the ward-data has changed. To fix this, modify the function identify\_wardnames\_top\_left() in the /backend/data/extract\_mumbai\_wards\_pdf.py, which tries to identify the coordinates of the top-left corner of the table containing the active/deceased/recovered numbers
2. **RT/DT values not showing up properly:** This could be due to many reasons. The recommendation would be to check the github-actions associated with the repository <https://github.com/swb-ief/etl-pipeline/actions> and examine the log of the latest R\_proc action, to see if the data has been updated. If R\_proc has successfully run, check if the step where the RT/DT values are merged to the main csv is working as expected (the merge is performed in **/tasks/updateEpiStatsRepo\_wards\_task.py**