Constructor

```
publicfunction__construct()
```

- Initializes the service.
- Currently only has placeholder comments for session code.

---

1. getAllData($data)

- Runs a **complex SQL query** to fetch containment/building/owner info that needs desludging.
- Converts results to a Laravel Collection.
- Adds helper fields (`display_name`, `display_contact`, etc.).
- Applies optional filters: by owner name, containment ID, holding number, or BIN.
- Returns the data formatted for **Yajra DataTables** , with dynamic **action buttons** (Confirm, Reschedule, Delete) depending on user permissions.

---

2. getContainmentData()

- Fetches **containment IDs** that:
  - Have not been emptied.
  - Don’t pay WASA bill.
  - Status = `0`, `4`, or `NULL`.
- Orders them by **priority** and **distance to FSTP** .
- Returns the list of containment models.

---

3. fetchSiteSettings()

- Gets site-wide settings from `sdm_sitesettings` table.
- Returns them as a collection (e.g., daily trip capacity, weekends, holidays, etc.).

---

4. tripsAllocated($date)

- Calculates **how many desludging trips can still be allocated** for a given date.
- Counts confirmed + auto-scheduled applications.
- Checks **trip capacity per day** (from site settings).
- If the date is a weekend or holiday → returns 0 trips.
- Otherwise, returns **remaining trips** available that day.

---

5. fetchContainmentsInRange($start$,end, $containments)

- Picks a slice of containments from a larger list, between `$start` and `$end`.
- Helps distribute containments into daily schedules.

---

6. setEmptyingDate()

- **Generates emptying schedules** for containments:

    1. Gets settings + containment data.
    2. Decides a start date.
    3. Skips holidays/weekends.
    4. Assigns containments to available trips (capacity-based).
    5. Updates DB (`next_emptying_date`) in chunks of 500 for performance.
- Returns success or error JSON.

---

7. tripsAllocatedRange($request)

- Like `tripsAllocated()`, but works for a **date range** .
- Loops day by day from `start_date` to `end_date`.
- Marks each day with available trips + whether it’s a holiday/weekend.
- Returns JSON.

---

8. disagreeEmptying($bin)

- Lets a user **disagree with a scheduled desludging date** .
- Fetches the containment linked to a BIN.
- Updates containment status:
    - First disagreement → status = `4`.
    - Second disagreement → status = `5` (permanent removal).
- Returns JSON response with the appropriate message.

---

9. redirectApplication($request)

- Stores building/containment details in **session flash data** .
- Redirects to the application creation route, passing along the action type.
- Basically prepares session state for the next form.

---

10. download($data)

- Exports containment/desludging schedule data as a **CSV file** .

- Runs SQL to fetch records.
- Applies filters (owner name, containment ID, holding num, bin).
- Uses **Box\Spout** to generate CSV with styled header row.
- Streams CSV download to the browser.

---

11. setPriority()

- **Recalculates containment priority** for scheduling:
  - Priority 1 = older than 3 years since emptied/constructed.
  - Priority 2 = between 1–3 years.
  - Priority 3 = emptied/constructed within last year.
  - If no date → defaults to Priority 1.
- Processes containments in **chunks of 10,000** for efficiency.
- Only updates records where the priority actually changes.

Helper functions (private):

normalizeDateOrNull($value,$anchor)

- Converts input date → Carbon object.
- Invalid or **future dates** → returns NULL.

decidePriorityByDate($emptied,$constructed, $anchor)

- Decides which date (last emptied > construction) is relevant.
- Compares it with cutoffs (`today-3y`, `today-1y`) to assign Priority 1, 2, or 3.