

Bus Status Batch Apex Documentation

Apex Classes

BusCapacityExceededException

- Custom exception class to signal when a bus has more than 20 passengers.
- Usage: Thrown in `BusStatusBatch.execute()` when passenger count exceeds 20.

```
public class BusCapacityExceededException extends Exception {}
```

BusStatusBatch

- **Purpose:** Batch class to update `Bus_Status__c` based on `Number_of_Passengers__c`.
- Implements `Database.Batchable<sObject>` and `Database.Stateful`.

Constructor

```
public BusStatusBatch(List<Id> busIds)
```

- Accepts a list of Bus Ids to process.

Methods

start

```
Database.QueryLocator start(Database.BatchableContext context)
```

- Queries Bus records by Ids.
- Returns a `QueryLocator` for the batch.

execute

```
void execute(Database.BatchableContext context, List<Bus__c> buses)
```

- Processes each batch of buses (up to 200 records per batch).
- Logic:
 - `passengerCount < 10` → `Bus_Status__c = 'Available'`
 - `10 < passengerCount < 20` → `Bus_Status__c = 'Limited Seats'`
 - `passengerCount == 20` → `Bus_Status__c = 'Full'`
 - `passengerCount > 20` → throw `BusCapacityExceededException` (logged via `System.debug`, process continues)
- Updates buses that pass validation.

finish

```
void finish(Database.BatchableContext context)
```

- Logs completion of batch process.
-

Test Class – BusStatusBatchTest

Test Data Setup

- Creates 5 Bus records.
- Populates Passenger__c records to generate rollup summary Number_of_Passengers__c:
 - Bus 1: 5 passengers → Available
 - Bus 2: 15 passengers → Limited Seats
 - Bus 3: 20 passengers → Full
 - Bus 4: 25 passengers → Exception (status not updated)
 - Bus 5: 0 passengers → Available

Test Method

```
@isTest static void testBusStatusBatch()
```

- Executes the batch with all Bus Ids.
 - Asserts updated Bus_Status__c values and Number_of_Passengers__c.
 - Confirms that exception does not stop batch execution.
-