

# What is Batch Apex Salesforce? How to execute Batch Apex Class

## What is Batch Apex Salesforce?

Batch as the name suggests, is used when a large data (bulk) volume is involved. **Batch Apex** Salesforce allows you to define a single job that can be broken up into manageable chunks, Where Query chunk can be processed separately.

**Example** :- If we need to make a field update of every record of Standard Account object in your salesforce organization, then governor limits would restrict from field updating of every records of Account object because in a single transaction, we can process only 10,000 records. In above case, if we have more than 10,000 records we can not perform field update.



## How Batch Apex in Salesforce works?

*Batch Apex Salesforce* will fetch all the records on which you want to perform the field update and divide them into list of 200 records and on every 200 records operation is performed separately.

- Batch Apex Salesforce helps to execute on more than 10,000 records as, it won't perform an operation on all the records in a single transaction instead it divides them in to no of sub tasks, where each subtask may contain the records upto 2000.

## What is Database.Batchable Interface ?

To use the Batch Apex, we should write an Apex class that implement `Database.Batchable` Interface. `Database.Batchable` Interface consists of three methods, that must be implemented, they are

1. Start method.
2. execute method.
3. Finish method.

**Start method**: Start method is automatically called at the beginning of the batch apex job. This method will collect the records or objects on which the operation should be performed.

- The records are broken down into sub tasks and given to execute method.

**Start method syntax** :

```
global
(Database.QueryLocator
```

```
global (Database.QueryLocator | Iterable<Object>) start(Database.BatchableContext ***bc***) {}
```

- The return type of the start method can be Database.QueryLocator or Iterable <Object>.

**Database.QueryLocator:** Use Database.QueryLocator as return type to the start method when you are fetching the records using a simple select Query.

- Total number of records retrieved by SOLQ queries are 50,000.
- Total number of records retrieved by Database.SetQueryLocator are 10,000.

But, in the Batch Apex the governor limits for SOQL query are bypassed and we can fetch upto 50 million record using Database.QueryLocator.

**Example :-**

```
global
Database.QueryLocator
```

```
global Database.QueryLocator Start(Database.BatchableContext bc)
{
return Database.getQueryLocator('Select id, name FROM Account');
}
```

**Iterable :** Use Iterable as a return type when you want to use complex logic or scope to fetch the records for the batch job.

**Example :**

```
global class MyTest
implements Iterable
```

```

global class MyTest implements Iterable <Account>

{

}

global Iterable<Account> start(Database.BatchableContext bc)
{

return new Mytext();

}

```

**Execute Method**: The records which are fetched from the Start method are divided into batches of 200 records each. Now every batch of 200 records are separately pass to the execute method separately and the operation what we want to perform on the records written in the execute method.

**Execute Method Syntax:**

```

global void
execute(Database.ExecutableContext Bc, List<p> {}

```

```

global void execute(Database.ExecutableContext Bc, List<p> {})

```

- This method takes two parameters.
- A reference to the Database.BatchableContext object.
- A list of Subjects such as List<Subject>, or a list of parameterized types.

**Finish Method** :- When the start method has fetched 1000 records on which we want to perform the operation they are divided into 5 batch/Groups. Each batch/Group of size 200 records.

- On every batch/Group the execute method is called (totally called 5 times).
- After executing every Group/batch the governing limits are reset for the execute method.
- Once all the batches are executed then finish method will be called to send Email notification for post execution work.

**Finish Method syntax:**

```

global void
finish(Database.BatchableContext BC) {}

```

```

global void finish(Database.BatchableContext BC) {}

```

## Salesforce Apex

‡ What is Salesforce Apex?

## Apex Basics

‡ How to Enable Developing Mode in Salesforce?

‡ How to use Salesforce developer Console.

‡ Apex Data types.

‡ Apex - Variables

‡ Apex - Class

‡ Apex - Methods

‡ Apex - Constructors

‡ Apex - Scheduler

‡ What is Salesfore Batch Apex

‡ Batch Apex - Governor Limits

‡ Triggers in Salesforce

‡ Email Messages - Inbound, Outbound

‡ Apex - Interface

‡ Apex - DML statements