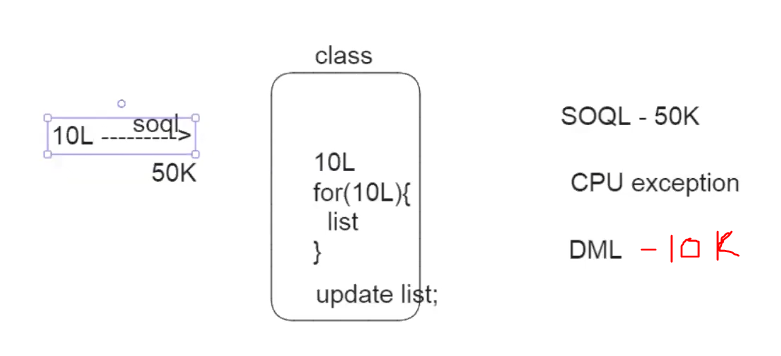
In previous classes we have seen about list and set now can see about map we know that in maps the data can be stored in the form of

key -value pairs

Okay, we can see later

In **Apex**, when you use **SOQL (Salesforce Object Query Language)**, you can **retrieve a maximum of 50,000 records** **per transaction**.

If we have 1 lakh record, then how to handle



For example, let us consider one bank having more than 10L records

In apex Salesforce can store 50K records by using SOQL retrieving the data, if for example it can take 10L records if we want to add interest to that record we can use for loop and we can do it but for 10K records it may takes more than 10second so here we can face CPU time limit exception but for example it is also managed to do push the record we need DML operation which can push 10K records to database so this is also not possible

A screenshot of a computer

AI-generated content may be incorrect.

For example, from the above case each developer using their own method and decimal limit point in this we are getting inconsistent results

To avoid that we have an architect in our company who gives some standard things that need to be followed by all developers.

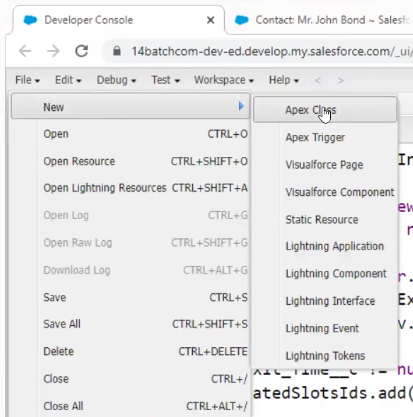
The architect will create one interface which has some rules

A white square with black text

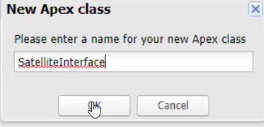
AI-generated content may be incorrect.

Interface which contains method declarations and constants

How to create an interface

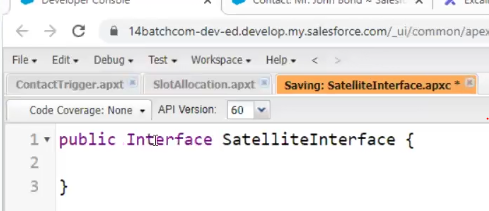


Here we don’t have the option to create an interface so create a class first



A screenshot of a computer

AI-generated content may be incorrect.



**🔹 Access Modifiers for Apex Interface Methods**

**❗ Apex Rule:**

🔸 **Interface methods in Apex must NOT have any access modifiers.**

This is **strictly enforced** by Apex.

**❌ Not Allowed:**

apex

CopyEdit

public interface MyInterface {

public void myMethod(); // ❌ Error: Interface methods cannot have modifiers

}

**✅ Correct Way:**

apex

CopyEdit

public interface MyInterface {

void myMethod(); // No access modifier

}

**🔹 Why No Access Modifier?**

* In Apex, **all interface methods are implicitly public**.
* You cannot use private, protected, or even public in the method signature.

**🔹 Implementing the Interface**

When a class implements the interface:

* The methods must be **public**
* Because you are implementing a public interface method

**✅ Example:**

apex

CopyEdit

public interface Vehicle {

void start(); // No modifier here

}

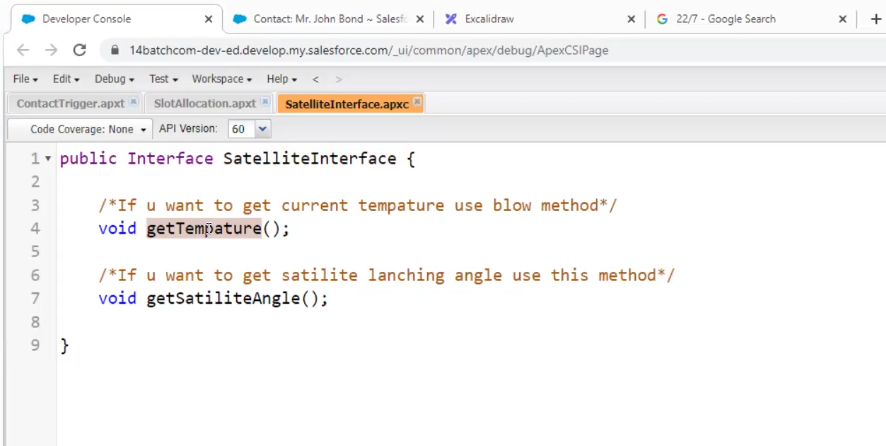
public class Car implements Vehicle {

public void start() { // Must be public here

System.debug('Car started');

}

}



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

If we don’t want to implement that method leave that

A close-up of a computer code

AI-generated content may be incorrect.

Abstract means hiding

A diagram of a diagram

AI-generated content may be incorrect.

In the above x person is dividing the data into small parts (chunks) and giving to workers

A, B, C, D …. etc. are called as workers

Y is the supervisor who checks whether the task is completed or not

Start() calling the other method execute()

A number of numbers on a white background

AI-generated content may be incorrect.

Here X ------ start()----the chunk capacity 200 (default batch size)

A, B, C, …. ----execute()