PHASE 5: Apex Programming

Debug Log Analysis and Bug Fixing in Salesforce Apex

1. Objective of Phase 5

The main focus of this project is **writing Apex code that contains intentional bugs and debugging it using the Apex Replay Debugger**.
This phase involves building the Apex classes and test classes that you will later deploy, test, and debug.

2. Creating Apex Classes

In this project, the primary Apex class created is:

AccountService.cls

This class is responsible for creating an Account record with the inputs:

- Name
- Account Number
- Ticker Symbol

The method:

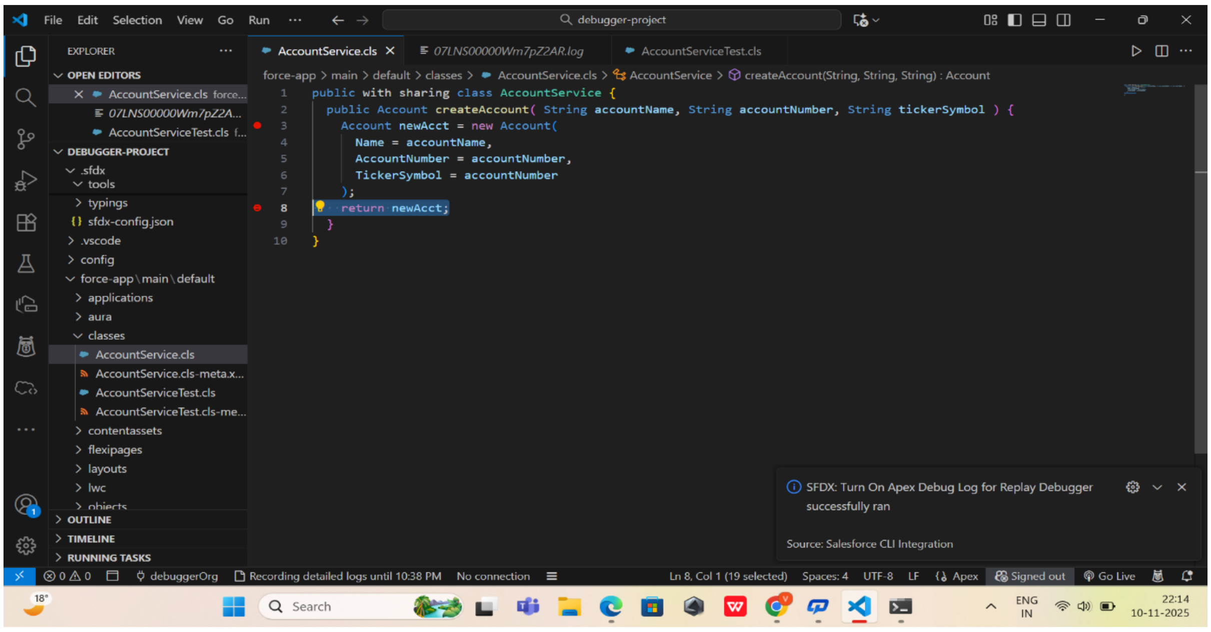
public Account createAccount(String accountName, String accountNumber, String tickerSymbol)

Intentional Bug for Debugging

To support the debugging use case, the code includes a **deliberate bug**:

TickerSymbol = accountNumber;

This bug is later detected and fixed using the Apex Replay Debugger.



```
public with sharing class AccountService {
  public Account createAccount( String accountName, String accountNumber, String tickerSymbol ) {
    Account newAcct = new Account(
      Name = accountName,
      AccountNumber = accountNumber,
      TickerSymbol = accountNumber
    );
    return newAcct;
  }
}
```

SFDX: Turn On Apex Debug Log for Replay Debugger successfully ran

Source: Salesforce CLI Integration

debugger-project
AccountService.cls
07LNS00000Wm7pZ2AR.log
AccountServiceTest.cls
force-app > main > default > classes > AccountService.cls > AccountService > createAccount(String, String, String) : Account
EXPLORER
OPEN EDITORS
DEBUGGER-PROJECT
Recording detailed logs until 10:38 PM
Ln 8, Col 1 (19 selected)

3. Creating Apex Test Classes

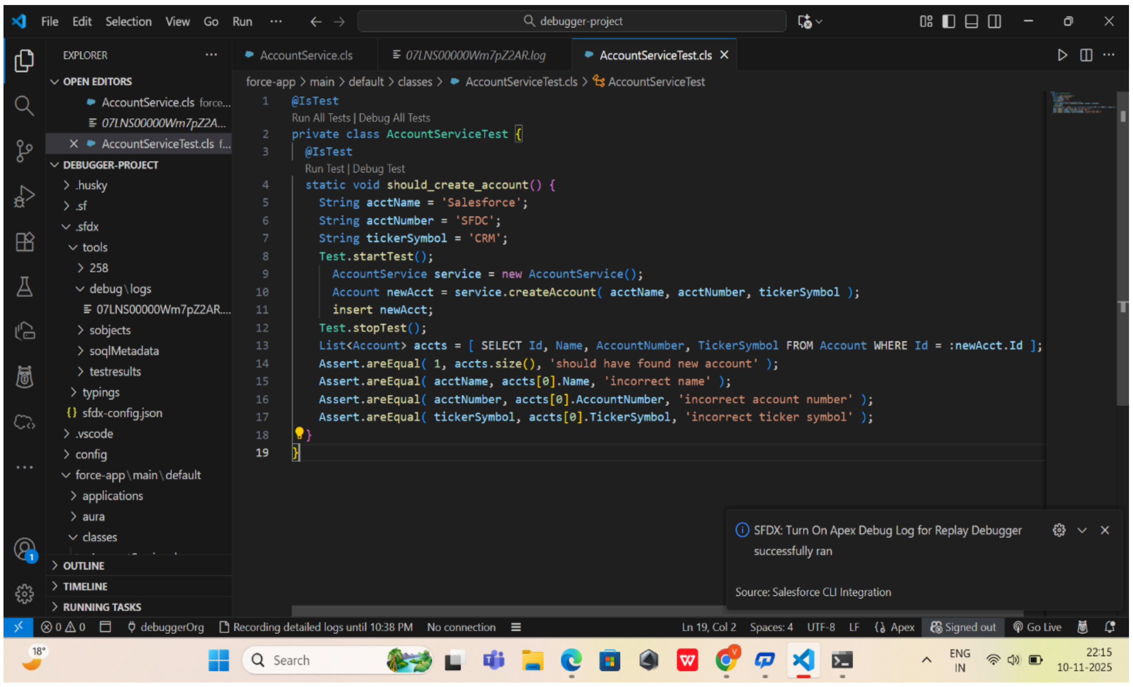
To verify and debug the Apex logic, you create the test class:

AccountServiceTest.cls

This class:

- Calls AccountService.createAccount()
- Inserts the Account
- Validates:
  - Name
  - Account Number
  - Ticker Symbol

Role of the Test Class

- Forces the Apex code to execute
- Generates debug logs
- Helps locate the bug during debugging
- Ensures your fix works after debugging
- 

```
@IsTest
private class AccountServiceTest {
  @IsTest
  static void should_create_account() {
    String acctName = 'Salesforce';
    String acctNumber = 'SFDC';
    String tickerSymbol = 'CRM';
    Test.startTest();
      AccountService service = new AccountService();
      Account newAcct = service.createAccount( acctName, acctNumber, tickerSymbol );
      insert newAcct;
    Test.stopTest();
    List<Account> accts = [ SELECT Id, Name, AccountNumber, TickerSymbol FROM Account WHERE Id = :newAcct.Id ];
    Assert.areEqual( 1, accts.size(), 'should have found new account' );
    Assert.areEqual( acctName, accts[0].Name, 'incorrect name' );
    Assert.areEqual( acctNumber, accts[0].AccountNumber, 'incorrect account number' );
    Assert.areEqual( tickerSymbol, accts[0].TickerSymbol, 'incorrect ticker symbol' );
  }
}
```

SFDX: Turn On Apex Debug Log for Replay Debugger successfully ran

Source: Salesforce CLI Integration

-

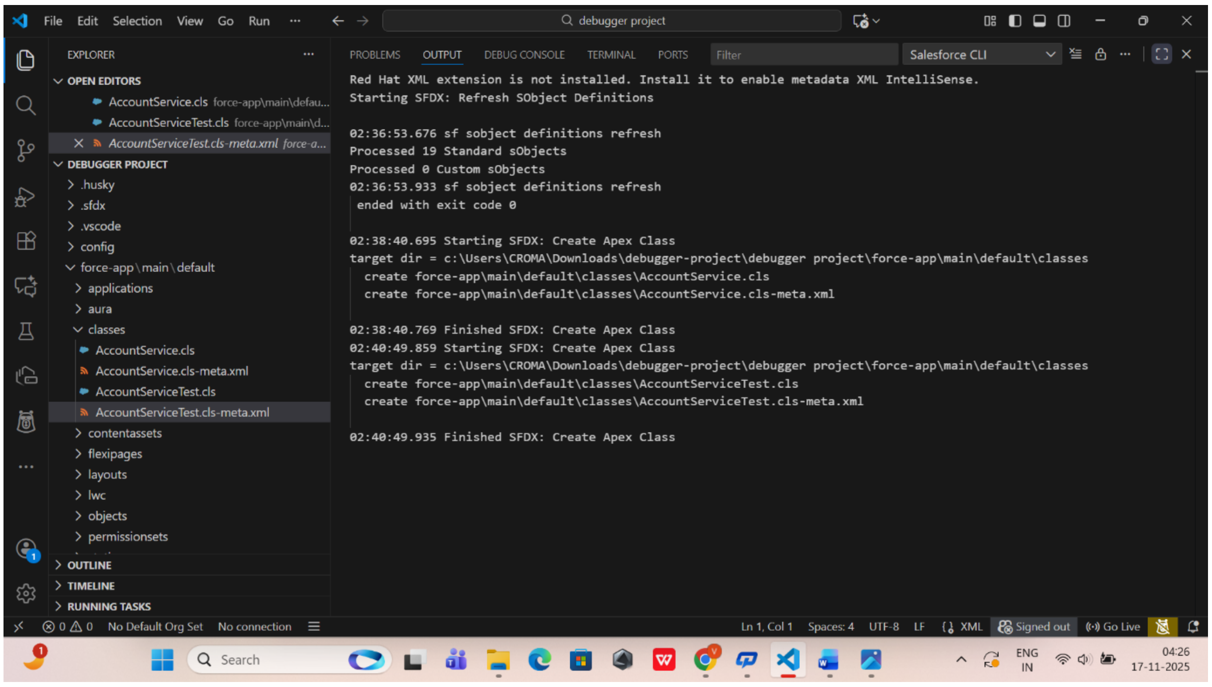
4. Writing Code for Debugger Compatibility

Code Requirements for Replay Debugger

To ensure the debugger works correctly:

- Apex Code should be deployed from the Salesforce DX project
- Code must match exactly what generated the debug log
- Test class must run with detailed log levels
- Logs must include **FINER/FINEST** granularity

These rules ensure breakpoints, checkpoints, and replay logs sync correctly.



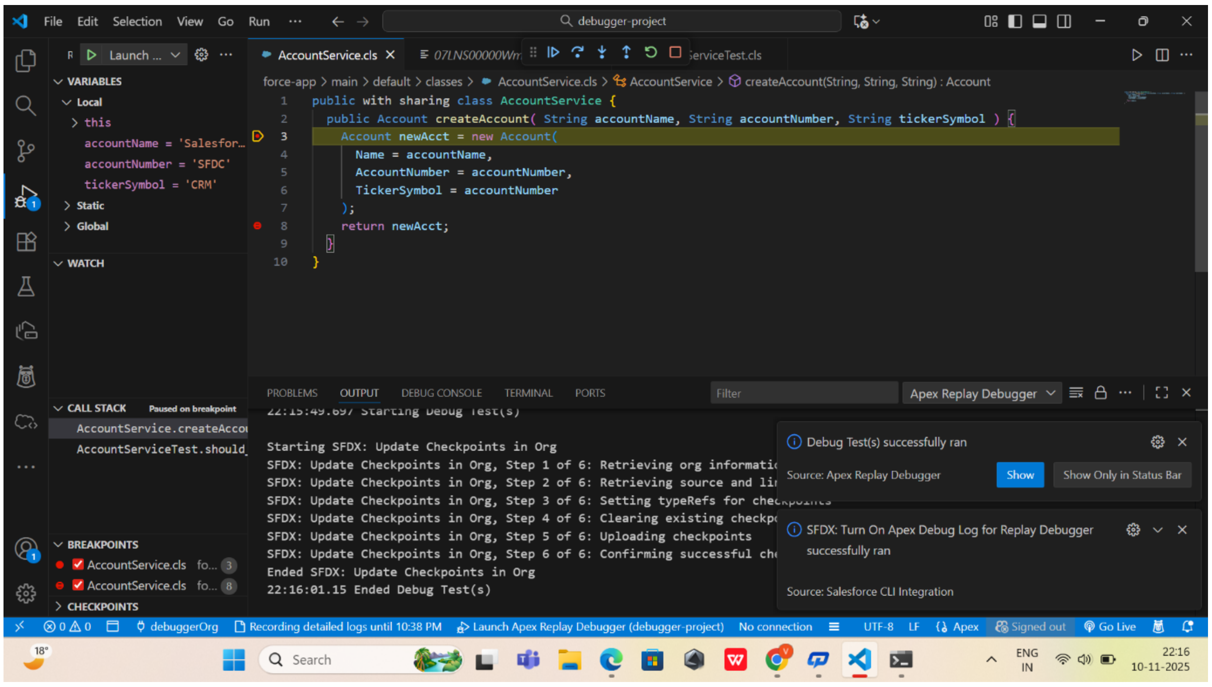
```
Red Hat XML extension is not installed. Install it to enable metadata XML IntelliSense.
Starting SFDX: Refresh SObject Definitions

02:36:53.676 sf sobject definitions refresh
Processed 19 Standard sObjects
Processed 0 Custom sObjects
02:36:53.933 sf sobject definitions refresh
 ended with exit code 0

02:38:40.695 Starting SFDX: Create Apex Class
target dir = c:\Users\CROMA\Downloads\debugger-project\debugger project\force-app\main\default\classes
  create force-app\main\default\classes\AccountService.cls
  create force-app\main\default\classes\AccountService.cls-meta.xml

02:38:40.769 Finished SFDX: Create Apex Class
02:40:49.859 Starting SFDX: Create Apex Class
target dir = c:\Users\CROMA\Downloads\debugger-project\debugger project\force-app\main\default\classes
  create force-app\main\default\classes\AccountServiceTest.cls
  create force-app\main\default\classes\AccountServiceTest.cls-meta.xml

02:40:49.935 Finished SFDX: Create Apex Class
```

5. Deploying Apex Code to Org

Apex code is deployed using:

✓ SFDX: Deploy Source to Org
✓ Or right-click → **Deploy this Source to Org**

This makes the class available for execution and debugging in the connected Salesforce org.



```
public with sharing class AccountService {
  public Account createAccount( String accountName, String accountNumber, String tickerSymbol ) {
    Account newAcct = new Account(
      Name = accountName,
      AccountNumber = accountNumber,
      TickerSymbol = accountNumber
    );
    return newAcct;
  }
}
```

VARIABLES
Local
accountName = 'Salesfor…
accountNumber = 'SFDC'
tickerSymbol = 'CRM'

CALL STACK Paused on breakpoint
AccountService.createAccou
AccountServiceTest.should_

BREAKPOINTS
AccountService.cls fo... 3
AccountService.cls fo... 8

```
22:15:49.697 Starting Debug Test(s)

Starting SFDX: Update Checkpoints in Org
SFDX: Update Checkpoints in Org, Step 1 of 6: Retrieving org informatic
SFDX: Update Checkpoints in Org, Step 2 of 6: Retrieving source and lir
SFDX: Update Checkpoints in Org, Step 3 of 6: Setting typeRefs for checkpoints
SFDX: Update Checkpoints in Org, Step 4 of 6: Clearing existing checkpc
SFDX: Update Checkpoints in Org, Step 5 of 6: Uploading checkpoints
SFDX: Update Checkpoints in Org, Step 6 of 6: Confirming successful che
Ended SFDX: Update Checkpoints in Org
22:16:01.15 Ended Debug Test(s)
```

Debug Test(s) successfully ran
Source: Apex Replay Debugger

SFDX: Turn On Apex Debug Log for Replay Debugger successfully ran
Source: Salesforce CLI Integration

debuggerOrg | Recording detailed logs until 10:38 PM | Launch Apex Replay Debugger (debugger-project) | No connection

File Edit Selection View Go Run
debugger-project
Launch ...
AccountService.cls
07LNS00000Wm
erviceTest.cls
force-app > main > default > classes > AccountService.cls > AccountService > createAccount(String, String, String) : Account
this
Static
Global
WATCH
CHECKPOINTS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Filter
Apex Replay Debugger
Show
Show Only in Status Bar
UTF-8 LF Apex Signed out Go Live
Search
ENG IN
22:16 10-11-2025

6. Purpose of Apex Programming in This Project

Although the project is debugging-focused, Apex programming is essential because:

✓ It provides real logic for testing

✓ It contains the bug that the Replay Debugger will detect

✓ It enables generation of heap dumps and breakpoints

✓ It demonstrates proper Apex development within Salesforce DX

7. Outcome of Phase 5

At the end of this phase, you will have:

✓ An Apex class (AccountService.cls) with intentional errors
✓ An Apex test class (AccountServiceTest.cls) to validate logic
✓ Successfully deployed Apex code in the org
✓ Setup ready for debugging, breakpoint setting, and log replay