

Phase 7: Integration & External Access

Project: Debug Log Analysis and Bug Fixing in Salesforce Apex

1. Objective of Phase 7

The purpose of this phase is to **integrate Salesforce DX project components** and provide controlled **external access** to support testing and debugging.

Although the project primarily focuses on Apex code and debugging, integration ensures:

- Salesforce DX project communicates correctly with the org.
- Apex classes and test classes are deployable and executable.
- Replay Debugger can work seamlessly with generated debug logs.

2. Integration within Salesforce DX

Key integration steps include:

1. Apex Class & Test Class Integration

- AccountService.cls (logic) integrated with AccountServiceTest.cls (test).
- Test class triggers execution of Apex logic and generates debug logs.

2. Salesforce DX Project Structure Integration

- All classes, metadata, and configuration are part of the DX project:
 - force-app/main/default/classes
 - sfscratch-project.json for org connections

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER:** Shows files like AccountService.cls, AccountServiceTest.cls, and AccountService.cls-meta.xml.
- OUTPUT:** Displays log messages from the SFDX command-line interface:
 - 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 objects
 - Processed 4 Custom objects
 - 02:36:53.693 sf sobject definitions refresh
 - ended with exit code 0
- TERMINAL:** Displays log messages from the SFDX command-line interface:
 - 02:38:40.695 Starting SFDX: Create Apex Class
 - target dir = c:\Users\CRONAL\Downloads\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
- PROBLEMS:** Shows a warning about the Red Hat XML extension.
- Salesforce CLI:** Shows the status "Signed out".

```

1  public with sharing class AccountService {
2
3      public Account createAccount( String accountName, String accountNumber, String tickerSymbol ) {
4          Account newAcct = new Account(
5              Name = accountName,
6              AccountNumber = accountNumber,
7              TickerSymbol = accountNumber
8          );
9
10     return newAcct;
}

```

3. UI Integration

- The minimal Lightning Web Component (from Phase 6) connects with Apex class.
- Allows manual triggering of methods to produce debug logs.

4. Replay Debugger Integration

- Debug logs generated during Apex execution are linked with breakpoints and checkpoints.
- Ensures Replay Debugger correctly maps execution flow to code lines.

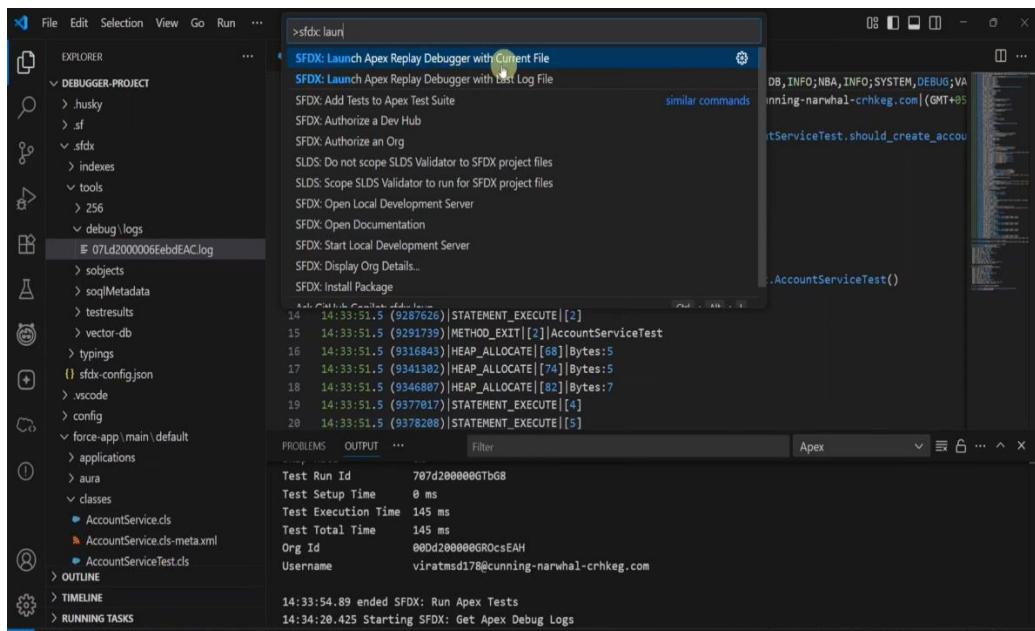
3. External Access Considerations

While the project does not require public external access, external integration can involve:

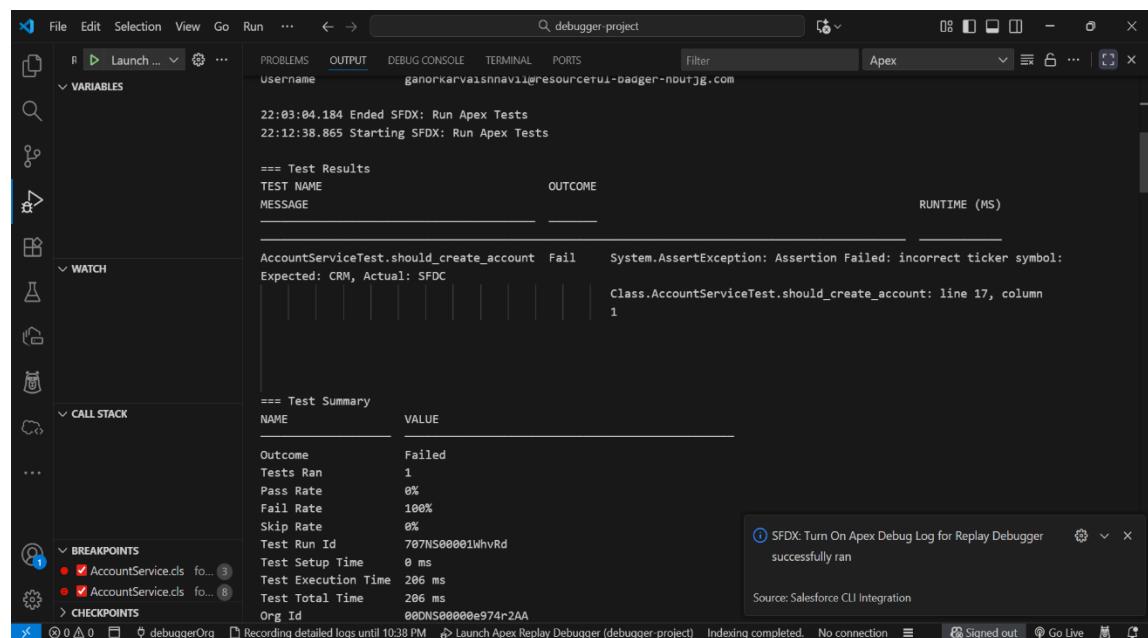
- **Trailhead Playground / Scratch Org Access**
 - Users (developers) authorize org via Salesforce CLI.
 - Access is required to deploy, run tests, and retrieve logs.
- **API / Tool Access**
 - CLI commands (sfdx force:apex:test:run, sfdx force:apex:log:get) act as external tools interacting with the org.
- **Security & Authorization**
 - Only authorized Salesforce users can execute Apex methods or view logs.
 - Protects sensitive data while allowing controlled debugging.

4. Benefits of Integration & Controlled External Access

- Centralized control of Apex code, test classes, and logs
- Simplified testing process across multiple orgs or environments
- Replay Debugger works consistently because code, logs, and breakpoints are aligned
- Developers can test, debug, and deploy safely without affecting production



●



5. Deliverables of Phase 7

By the end of this phase:

- ✓ Apex classes, test classes, and UI fully integrated in DX project
- ✓ External access for authorized users via CLI or sandbox is configured
- ✓ Replay Debugger fully functional with integrated breakpoints and checkpoints
- ✓ Debugging workflow ready for real execution scenarios

6. Outcome

- Salesforce DX project is **fully integrated** and functional.
- Authorized developers can trigger Apex methods, run tests, and replay debug logs.
- Replay Debugger now works in a complete, end-to-end workflow from code → org → logs → debugging.