

# Salesforce Sandbox Debugging & Testing

An Overview of Logs, Flows, Apex Testing, and Best Practices



Tathagat Saha, QA Configuration manager

# Where to Find Important Logs in Sandbox

- **Debug Logs**
  - Setup → Debug Logs
  - Add your user to start capturing logs
  - View detailed Apex execution, SOQL queries, DML operations in the below log table
- **Flow Error Logs**
  - Setup → Paused and Failed Flow Interviews
  - See exactly where Flows failed and why
- **Apex Test Logs**
  - Developer Console → Test → New Run
  - After running, view execution and assertion details
- **Email Logs** (for issues with emails)
  - Setup → Email Log Files
  - Request and review email communication history
- **Login History**
  - Setup → Login History
  - Monitor authentication issues, IP blocks, and session problems

The logo for 'fennia' is displayed in a green, lowercase, sans-serif font. The letter 'f' is stylized with a curved top. The background of the slide features abstract green geometric shapes on the right side.



# Triggered Flows: Overview

- ▶ Triggered when a record is **created/updated/deleted**.
- ▶ React to data changes, no manual start.
- ▶ **Where Are They Triggered From?**
  - Object level events (e.g., Account Created, Opportunity Updated)
  - Examples: After Save, Before Save Flows
- ▶ Found via Setup → Flows → Record-Triggered Flows.
- ▶ how many are there in Fender???? 300+ active flows



## Non triggered flows

- ▶ "Autolaunched flows are like background scripts — they sit quietly until something else tells them to start running."
- ▶ It must be called manually — by:
  - Apex code
  - Another Flow



# Apex Unit Tests

- ▶ **Where Are Apex Unit Tests Located?**
  - Developer Console → Test Classes Tab
  - OR Setup → Apex Classes → Search "@isTest" classes
- ▶ Viewable via Developer Console → Test Classes.
- ▶ Organized and isolated testing.



# Running Apex Unit Tests

- ▶ Developer Console → Test → New Run.
- ▶ Alternatively: Setup → Apex Test Execution.
- ▶ Important: Use **Run Asynchronously** for large classes to avoid timeout



# What to Check When Encountering an Error

- ▶ Check Debug Logs (Identify where exactly the failure happened (Apex Class, Trigger, Flow Element))
- ▶ Failed Flows (Find flow element failures)
- ▶ User Permissions (Ensure proper field-level security, object permissions)
- ▶ Record Data Integrity (Look for missing/invalid fields)



# What Data to Include in a Bug Report

- ▶ Full error message (copy + paste it)
- ▶ Steps to reproduce
- ▶ User/Record IDs
- ▶ Logs/screenshots
- ▶ Sandbox environment name (e.g., INS, Staging)
- ▶ Timestamps
- ▶ **Goal:** Make it easy for a developer to reproduce and fix it.





# How to Get More Debug Info - I

- ▶ Add Finer Debug Level
- ▶ Go to Setup → Debug Logs.
- ▶ Click "New" or "Edit" next to your user. Go to "Debug Level". Click "New Debug Level" to create a custom debug level.
- ▶ While creating a new Debug Level, you can set:
- ▶ FINE or FINER
- ▶ **Important:** Turn down debug levels after diagnosing to prevent excess log usage.

# How to Get More Debug Info - II

- ▶ Add more System.debug() lines in your Apex class, triggers and flows.
- ▶ Example:

```
public static void updateAccountName(Id accountId) {  
    System.debug('Starting updateAccountName with ID: ' + accountId);  
    Account acc = [SELECT Name FROM Account WHERE Id = :accountId];  
    System.debug('Current Account Name: ' + acc.Name);  
    acc.Name = acc.Name + ' - Updated';  
    update acc;  
    System.debug('Account update complete');  
}
```

- ▶ This way, if something fails, the debug logs tell you exactly where and why!



## How to Get More Debug Info - III

- ▶ Flow Fault Path
- ▶ In Flow Builder, when you select an action (like Update Records):
  - Drag the **red fault line** to a **Log Error** element.
  - In that "Log Error" subflow, create a record in a **Custom Error Log** object (example: Flow\_Error\_Log\_\_c).
- ▶ You can thus **catch the error gracefully**, log it, or notify someone by triggering some other flow.



# How to Get More Debug Info - IV

- ▶ Capture and Log Flow Errors into a Custom Object
  - Create a custom object like `Flow_Error_Log__c` with fields:
    - Error Message
    - Flow Name
    - Record ID
    - Timestamp
    - User ID
- ▶ **When a Flow fails:**
  - The Flow writes an entry into this table.
- ▶ You get a full **error history** without needing to check debug logs manually.



## How to Get More Debug Info - V

- ▶ Enable Email Notifications for Flow Errors
- ▶ Go to **Setup** → **Process Automation Settings**.
- ▶ Set “Send Process or Flow Error Email” to “Apex Exception Email Recipients”
- ▶ Go to **Setup** → **Apex Exception Email**
- ▶ **Add email addresses** (multiple, comma-separated)

# Thank You!

Questions?

Let's make Fender better together!

The logo for Fennia, featuring the word "fennia" in a green, lowercase, sans-serif font. The letter "f" has a distinctive curved top. The logo is positioned on the right side of the slide, which has a white background with a green diagonal stripe at the bottom right.