Phase 5: Apex Programming (Developer)

Project: ***Sports Equipment Rental Management System***

🧩 Objective

In this phase, we implemented **Apex programming** to automate backend logic for managing equipment availability, rental validation, and integration between related objects using **triggers, handlers, and best practices**.

We also connected our **VS Code project** with the **Salesforce Developer Org** to write, test, and deploy Apex classes and triggers efficiently.

🧱 Implementation Steps

STEP 1: Apex Class — EquipmentHandler

Purpose: Automatically update equipment availability status based on available quantity.

Procedure:

1. Go to **Setup → Developer Console**
2. Click **File → New → Apex Class**
3. Enter name: EquipmentHandler
4. Paste the following code:

```
public class EquipmentHandler {
   public static void updateAvailabilityStatus(List<Equipment__c> equipmentList) {
      for(Equipment__c eq : equipmentList) {
         if(eq.Available_Quantity__c != null) {
            if(eq.Available_Quantity__c == 0) {
               eq.Availability_Status__c = 'Rented';
            } else if(eq.Available_Quantity__c > 0) {
               eq.Availability_Status__c = 'Available';
            }
         }
      }
   }
}
```

5. **File → Save All**

✅ This class ensures every equipment's **Availability_Status__c** updates automatically when its quantity changes.

STEP 2: Apex Trigger — EquipmentTrigger

Purpose: Call the handler when Equipment records are inserted or updated.

Procedure:

1. In Developer Console → **File → New → Apex Trigger**
2. Trigger Name: EquipmentTrigger
   sObject: Equipment__c
3. Paste:

```
trigger EquipmentTrigger on Equipment__c (before insert, before update) {
  if(Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)){
    EquipmentHandler.updateAvailabilityStatus(Trigger.new);
  }
}
```

4. **Save All**

✅ Keeps equipment status in sync automatically.

STEP 3: Apex Class — RentalAgreementHandler

Purpose: Validate that rented quantity does not exceed available equipment.

Procedure:

1. Developer Console → New Apex Class → Name: RentalAgreementHandler
2. Paste:

```
public class RentalAgreementHandler {

  public static void validateEquipmentAvailability(List<Rental_Agreement__c> newAgreements) {
    Set<Id> equipmentIds = new Set<Id>();
    for (Rental_Agreement__c ra : newAgreements) {
      if (ra.Equipment__c != null) {
        equipmentIds.add(ra.Equipment__c);
      }
    }
```

```
    Map<Id, Equipment__c> equipmentMap = new Map<Id, Equipment__c>(
        [SELECT Id, Name, Available_Quantity__c
         FROM Equipment__c
         WHERE Id IN :equipmentIds]
    );

    for (Rental_Agreement__c ra : newAgreements) {
        if (ra.Equipment__c != null && ra.Quantity_Rented__c != null) {
            Equipment__c eq = equipmentMap.get(ra.Equipment__c);

            if (eq != null) {
                if (eq.Available_Quantity__c <= 0) {
                    ra.addError('Equipment "' + eq.Name + '" is currently unavailable for rent.');
                } else if (ra.Quantity_Rented__c > eq.Available_Quantity__c) {
                    ra.addError(
                        'Only ' + eq.Available_Quantity__c +
                        ' units of "' + eq.Name + '" are available. You requested ' +
                        ra.Quantity_Rented__c + '.'
                    );
                }
            }
        }
    }
}
```

✅ Prevents overbooking or renting unavailable equipment.

STEP 4: Apex Trigger — RentalAgreementTrigger

Purpose: Run the validation before saving Rental Agreement records.

```
trigger RentalAgreementTrigger on Rental_Agreement__c (before insert, before update) {
```

```
    if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
        RentalAgreementHandler.validateEquipmentAvailability(Trigger.new);
    }
}
```

✅ Adds real-time validation when saving Rental Agreements.

STEP 5: Setup Salesforce Project in VS Code

Procedure:

1. **Install Prerequisites**
    - Visual Studio Code
    - Salesforce CLI (sfdx)
    - Salesforce Extension Pack
2. **Create Project**
3. sfdx force:project:create -n SportsRentalProject
4. cd SportsRentalProject
5. code .
6. **Authorize Salesforce Org**
7. sfdx force:auth:web:login -a MyOrgAlias

Login to Salesforce → Org connected.

8. **Retrieve Metadata**
9. sf project retrieve start --metadata ApexClass --metadata ApexTrigger --target-org MyOrgAlias
10. **Deploy Local Code to Org**
11. sfdx force:source:deploy -p force-app

✅ Your local VS Code project is now connected to Salesforce for seamless development.

🧪 **Testing Procedure**

Test Case	Input	Expected Result
Case 1	Equipment with Available_Quantity__c = 0, Rent 1	❌ Error → “Equipment unavailable for rent.”

Test Case	Input	Expected Result
Case 2	Equipment with Available_Quantity__c = 3, Rent 5	❌ Error → “Only 3 units available…”
Case 3	Equipment with Available_Quantity__c = 3, Rent 2	✅ Record saves successfully
Case 4	Equipment updated to Available_Quantity__c = 0	✅ Status automatically changes to “Rented”

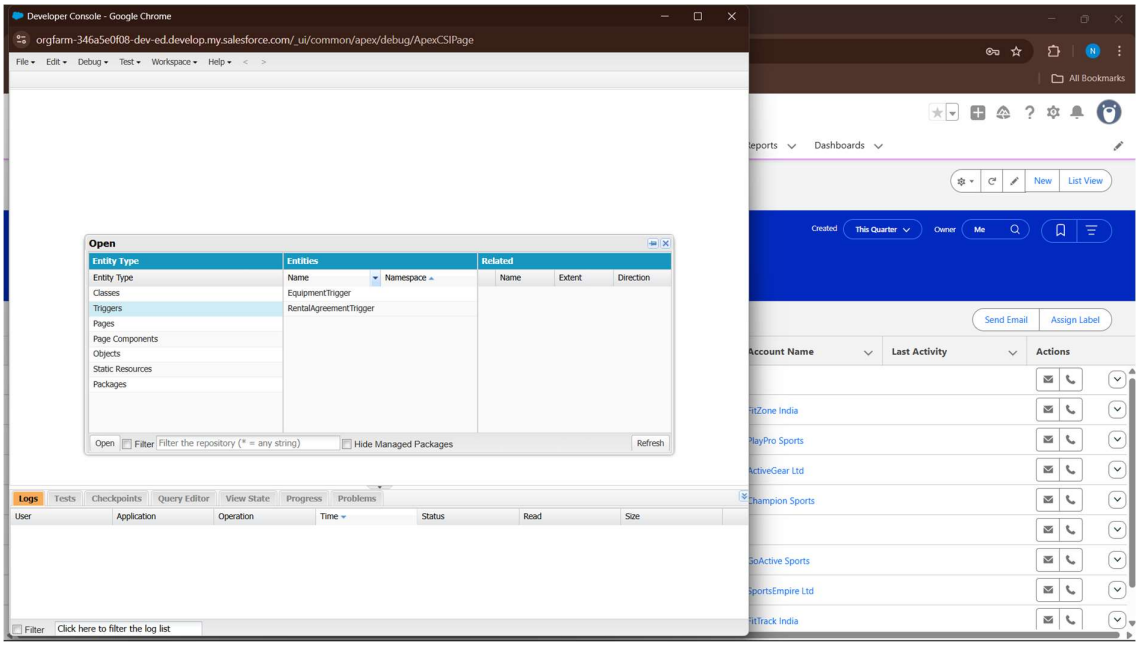
🧠 **Learning Outcomes**

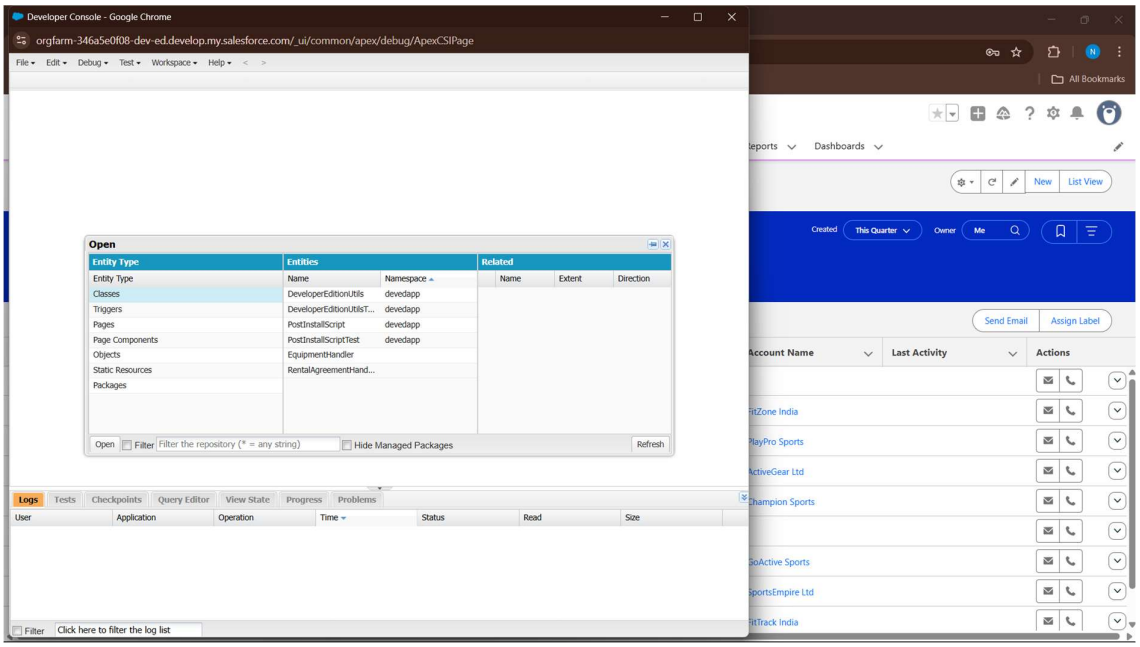
- Understood **Apex Class-Trigger pattern** for clean, modular logic.
- Practiced **SOQL queries** and **collections** to handle related data efficiently.
- Learned to connect Salesforce with **VS Code** using **Salesforce CLI**.
- Applied **real-time validation** and **error handling**.
- Automated data flow between **Equipment** and **Rental Agreement** objects.

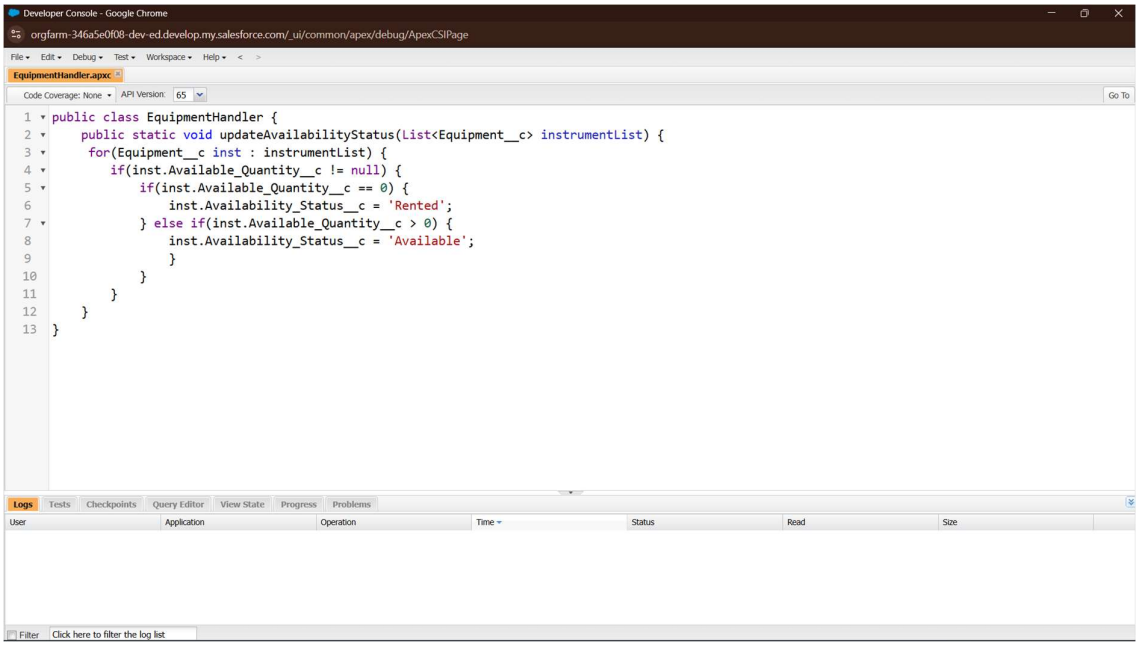
✅ **Result**

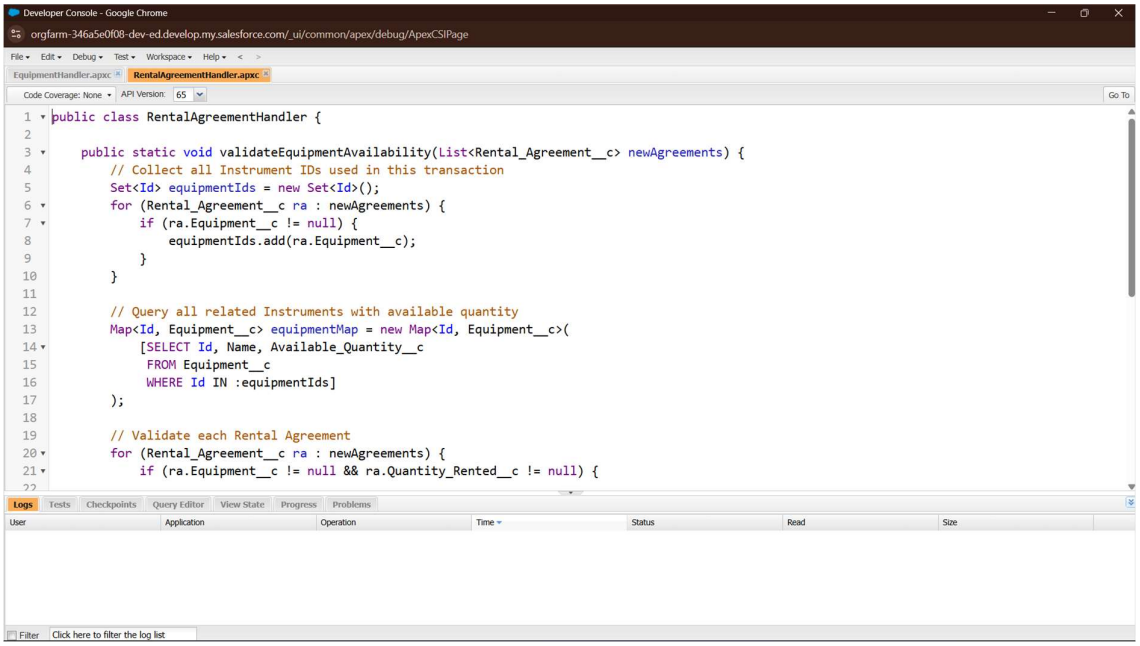
Successfully implemented a **backend automation system** using **Apex Classes and Triggers** that:

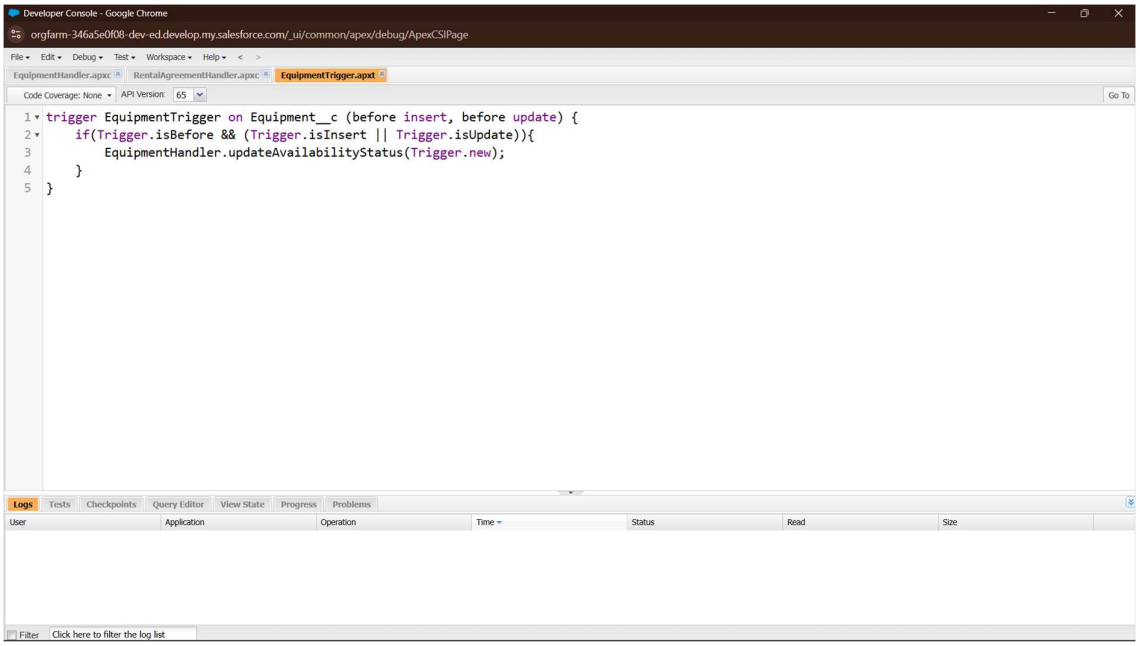
- Prevents renting unavailable equipment.
- Updates equipment status dynamically.
- Enhances system accuracy and efficiency.
- Demonstrates **developer-level Apex concepts** for Salesforce project execution.

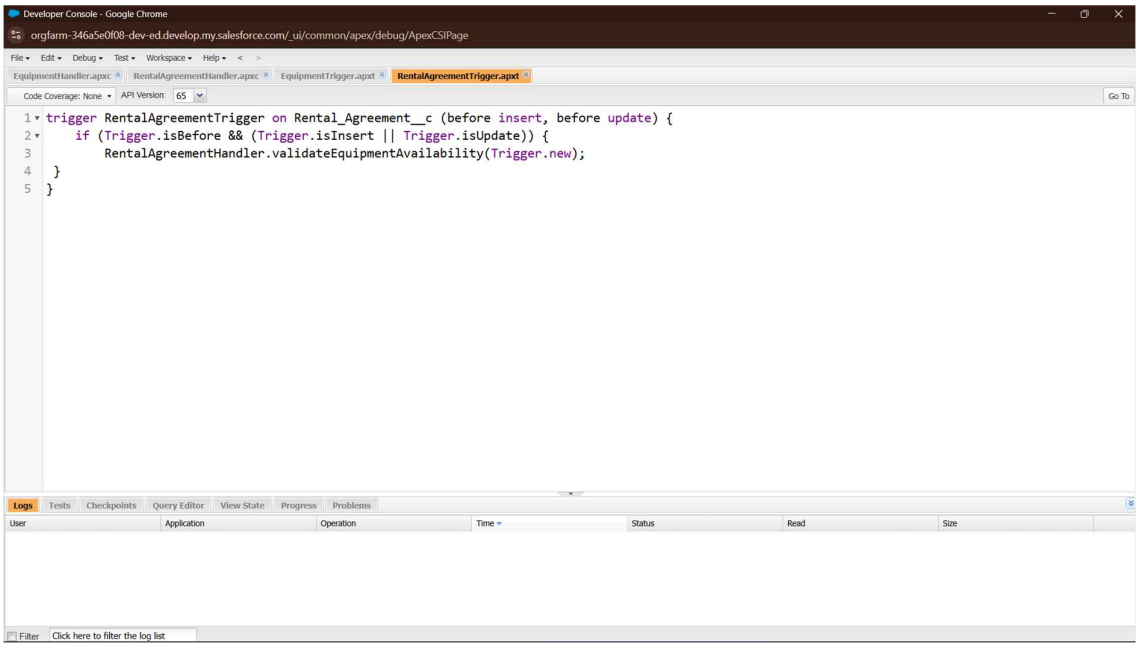

Developer Console - Google Chrome
orgfarm-346a5e0f08-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
Open
Entity Type
Entities
Related
Entity Type
Name
Namespace
Name
Extent
Direction
Classes
Triggers
Pages
Page Components
Objects
Static Resources
Packages
EquipmentTrigger
RentalAgreementTrigger
Open
Filter
Filter the repository (* = any string)
Hide Managed Packages
Refresh
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list
All Bookmarks
Reports
Dashboards
New
List View
Created
This Quarter
Owner
Me
Send Email
Assign Label
Account Name
Last Activity
Actions
FitZone India
PlayPro Sports
ActiveGear Ltd
Champion Sports
GoActive Sports
SportsEmpire Ltd
FitTrack India


Developer Console - Google Chrome
orgfarm-346a5e0f08-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
Open
Entity Type
Entities
Related
Entity Type
Name
Namespace
Name
Extent
Direction
Classes
Triggers
Pages
Page Components
Objects
Static Resources
Packages
DeveloperEditionUtils
DeveloperEditionUtilsT...
PostInstallScript
PostInstallScriptTest
EquipmentHandler
RentalAgreementHand...
devedapp
Open
Filter
Filter the repository (* = any string)
Hide Managed Packages
Refresh
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list
All Bookmarks
Reports
Dashboards
New
List View
Created
This Quarter
Owner
Me
Send Email
Assign Label
Account Name
Last Activity
Actions
FitZone India
PlayPro Sports
ActiveGear Ltd
Champion Sports
GoActive Sports
SportsEmpire Ltd
FitTrack India


Developer Console - Google Chrome
orgfarm-346a5e0f08-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
EquipmentHandler.apxc
Code Coverage: None
API Version: 65
Go To
public class EquipmentHandler {
public static void updateAvailabilityStatus(List<Equipment__c> instrumentList) {
for(Equipment__c inst : instrumentList) {
if(inst.Available_Quantity__c != null) {
if(inst.Available_Quantity__c == 0) {
inst.Availability_Status__c = 'Rented';
} else if(inst.Available_Quantity__c > 0) {
inst.Availability_Status__c = 'Available';
}
}
}
}
}
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list


Developer Console - Google Chrome
orgfarm-346a5e0f08-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
EquipmentHandler.apxc
RentalAgreementHandler.apxc
Code Coverage: None
API Version: 65
Go To
public class RentalAgreementHandler {
public static void validateEquipmentAvailability(List<Rental_Agreement__c> newAgreements) {
// Collect all Instrument IDs used in this transaction
Set<Id> equipmentIds = new Set<Id>();
for (Rental_Agreement__c ra : newAgreements) {
if (ra.Equipment__c != null) {
equipmentIds.add(ra.Equipment__c);
}
}
// Query all related Instruments with available quantity
Map<Id, Equipment__c> equipmentMap = new Map<Id, Equipment__c>(
[SELECT Id, Name, Available_Quantity__c
FROM Equipment__c
WHERE Id IN :equipmentIds]
);
// Validate each Rental Agreement
for (Rental_Agreement__c ra : newAgreements) {
if (ra.Equipment__c != null && ra.Quantity_Rented__c != null) {
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list


Developer Console - Google Chrome
orgfarm-346a5e0f08-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
EquipmentHandler.apxc
RentalAgreementHandler.apxc
EquipmentTrigger.apxt
Code Coverage: None
API Version: 65
Go To
trigger EquipmentTrigger on Equipment__c (before insert, before update) {
if(Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)){
EquipmentHandler.updateAvailabilityStatus(Trigger.new);
}
}
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list


Developer Console - Google Chrome
orgfarm-346a5e0f08-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
EquipmentHandler.apxc
RentalAgreementHandler.apxc
EquipmentTrigger.apxt
RentalAgreementTrigger.apxt
Code Coverage: None
API Version: 65
Go To
trigger RentalAgreementTrigger on Rental_Agreement__c (before insert, before update) {
if (Trigger.isBefore && (Trigger.isInsert || Trigger.isUpdate)) {
RentalAgreementHandler.validateEquipmentAvailability(Trigger.new);
}
}
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
User
Application
Operation
Time
Status
Read
Size
Filter
Click here to filter the log list