

Apex Trigger

Apex triggers-Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- 1.insert
- 2.update
- 3.delete
- 4.merge
- 5.upsert
- 6.undelete

Create An Apex Trigger

Use Case- when we try to create the account with the same name i.e. Preventing the users to create Duplicate Accounts.

- 1.Click on setup gear
- 2.Below the setup gear you can find developer console click on that
- 3.click on file and select new and select Apex trigger.

Copy this code

```
trigger PreventDuplicateAccounts on Account (before insert) {  
    Set<String> accountNames = new Set<String>();  
    for (Account acc : Trigger.new) {  
        if (accountNames.contains(acc.Name)) {
```

```
        acc.addError('A duplicate account with this name  
already exists.');
```

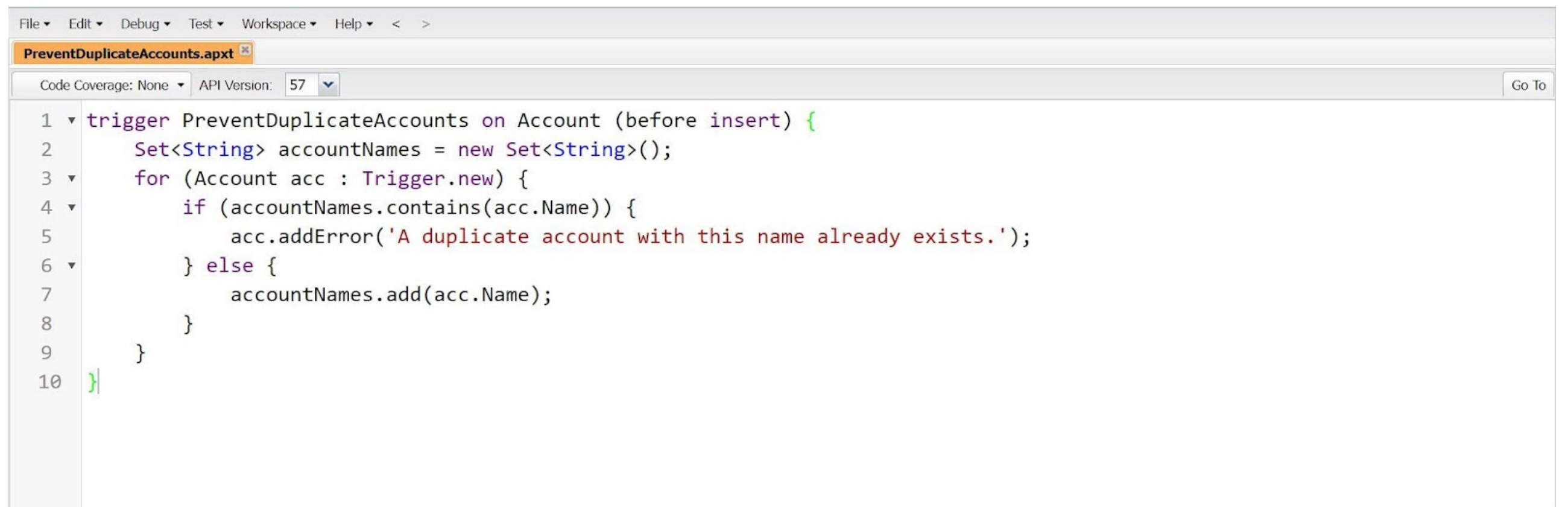
```
    } else {
```

```
        accountNames.add(acc.Name);
```

```
    }
```

```
}
```

```
}
```



The screenshot shows the Salesforce IDE interface. The top menu bar includes File, Edit, Debug, Test, Workspace, and Help. The active file is 'PreventDuplicateAccounts.apxt'. Below the menu bar, there are tabs for 'Code Coverage: None' and 'API Version: 57'. The main editor area displays the following Apex code:

```
1  trigger PreventDuplicateAccounts on Account (before insert) {  
2      Set<String> accountNames = new Set<String>();  
3      for (Account acc : Trigger.new) {  
4          if (accountNames.contains(acc.Name)) {  
5              acc.addError('A duplicate account with this name already exists.');6          } else {  
7              accountNames.add(acc.Name);  
8          }  
9      }  
10 }
```