SandboxPostCopy Interface

To make your sandbox environment business ready, automate data manipulation or business logic tasks. Extend this interface and add methods to perform post-copy tasks, then specify the class during sandbox creation.

Namespace

System

Usage

Create an Apex class that implements this interface. Specify your class during sandbox creation. After your sandbox is created, the `runApexClass(context)` method in your class runs using the automated process user's permissions.

Important

The SandboxPostCopy Apex class is executed at the end of the sandbox copy using a special Automated Process user that isn't visible within the org. This user doesn't have access to all object and features; therefore, the Apex script cannot access all objects and features. If the script fails, run the script after sandbox activation as a user with appropriate permissions.

- **SandboxPostCopy Methods**
- **SandboxPostCopy Example Implementation**
  These examples show a simple implementation of the SandboxPostCopy interface and a test for that implementation. To test your SandboxPostCopy implementation, use the `System.Test.testSandboxPostCopyScript()` method.

See Also

- *Tooling API*: SandboxInfo
- *Tooling API*: SandboxProcess

SandboxPostCopy Methods

The following method is for `SandboxPostCopy`.

- **runApexClass(context)**
  Executes actions in a new sandbox to prepare it for use. For example, add logic to this method to create users, run sanitizing code on records, and perform other setup tasks.

runApexClass(context)

Executes actions in a new sandbox to prepare it for use. For example, add logic to this method to create users, run sanitizing code on records, and perform other setup tasks.

context

Type: System.SandboxContext

The org ID, sandbox ID, and sandbox name for your sandbox. To work with these values, reference `context.organizationId()`, `context.sandboxId()`, and `context.sandboxName()` in your code.

Return Value

Type: void

SandboxPostCopy Example Implementation

These examples show a simple implementation of the SandboxPostCopy interface and a test for that implementation. To test your SandboxPostCopy implementation, use the `System.Test.testSandboxPostCopyScript()` method.

Important

The SandboxPostCopy Apex class is executed at the end of the sandbox copy using a special Automated Process user that isn't visible within the org. This user doesn't have access to all objects and features; therefore, the Apex script can't access all objects and features. If the script fails, run the script after sandbox activation as a user with appropriate permissions.

This example implements the `System.SandboxPostCopy` interface.

```
global class PrepareMySandbox implements SandboxPostCopy {

    global PrepareMySandbox() {
        // Implementations of SandboxPostCopy must have a no-arg constructor.
        // This constructor is used during the sandbox copy process.
        // You can also implement constructors with arguments, but be aware that
        // they won't be used by the sandbox copy process (unless as part of the
        // no-arg constructor).
        this(some_args);
    }

    global PrepareMySandbox(String some_args) {
        // Logic for constructor.
    }

    global void runApexClass(SandboxContext context) {
        System.debug('Org ID: ' + context.organizationId());
        System.debug('Sandbox ID: ' + context.sandboxId());
        System.debug('Sandbox Name: ' + context.sandboxName());

        // Insert logic here to prepare the sandbox for use.
    }
}
```

The following example tests the implementation using the `System.Test.testSandboxPostCopyScript()` method. This method takes four parameters: a reference to a class that implements the SandboxPostCopy interface, and the three fields on the context object that you pass to the `runApexClass(context)` method. An overload on the method takes an optional Boolean parameter to indicate if the test must be performed as the Automated Process user.

```
            // Insert logic here to create records of the objects that the class you're testir
            // manipulates.

            Test.startTest();

            // Replace '00D000000000000' with your sandboxId and
            // execute test script with RunAsAutoProcUser set to true.

            Test.testSandboxPostCopyScript(
                new PrepareMySandbox(), UserInfo.getOrganizationId(),
                    '00D000000000000', UserInfo.getOrganizationName(), true);

            Test.stopTest();

            // Insert assert statements here to check that the records you created above have
            // the values you expect.
        }
    }
```

For more information on testing, see Testing Apex.

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

Share your feedback