

Table of Contents

Installation	3
1. Installation Option.....	3
2. Initialize App.....	3
3. Manage Access	5
Application at a Glance.....	6
1. APEX Resources.....	6
2. Settings Overview	6
➤ ReSync.....	6
➤ Settings	6
➤ App Info.....	6
3. Code Coverage	6
4. Metrics & Tasks Tabs.....	6
Settings Overview	7
1. ReSync.....	7
2. Settings	7
3. App Info.....	7
Continued... ..	7
➤ Total.....	8
➤ Available.....	8
➤ Used	8
➤ Rollover Period:	8
Generate Unit Tests.....	9
1. Multiple selection	9
2. One Test at a time	9
Code coverage view with color code.....	10
Troubleshoot "View Issues"	12
Few items to remember.....	13
Do's.....	13
Don'ts.....	13

Introduction

“Test Gen” the lightning application assists Salesforce developers with code coverage and test-data generation to speed up production deployments. It can quickly create in-depth test cases by analyzing APEX code syntax, SOQL, SOSL statements and Web Service Callouts.

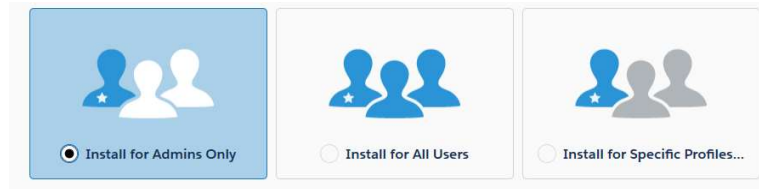
App guides developers to revisit code for issues (e.g restructure) to maximize code coverage. Developers can write code more efficiently, while businesses can significantly reduce project overhead and timelines.

Installation

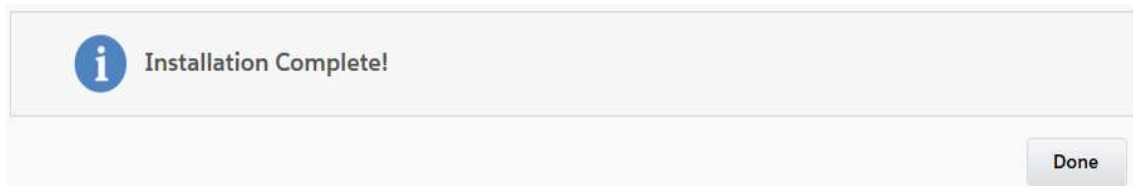
1. Installation Option

Please Note: Install only in Dev or Sandbox orgs.

- Select "Install for Admins Only"



- After installation is complete, Click "Done" button.

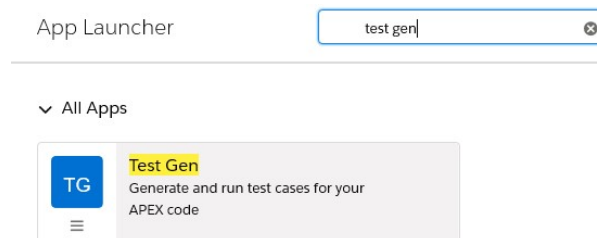


2. Initialize App

- From Home page go to "App Launcher" (Top Left corner).



- In App Launcher Search by "Test Gen" and click on App to initialize.



Please Note: It may take few minutes to initialize. After initialized, you will see page showing APEX classes, triggers, and code coverage.

If Initialize takes longer than 5 minutes then please check below.

1. Make sure "My Domain" is set and deployed.
In order to run Lightning App "My Domain" in Company Settings section is required.
2. Make sure debug mode is turned off.

3. Modify Custom Settings

Go To Setup > Search "Custom Settings" in Quick Find.
 You will see below record for Test Gen. Click "Manage" link.

Action	Label ↑	Visibility	Settings Type	Namespace Prefix
Manage	Test Gen	Public	List	tgn

If there is any record shown as below, simply "Delete" it and wait until "Test Gen" initializes.

Action	Name ↑
Edit Del	org

Delete - Record 1 - org

If you still have any issues, please contact us via testgen@webinteractiveinc.com.

Once you see similar page as below then App is loaded successfully.

The screenshot displays the Test Gen application interface. On the left, the 'APEX Resources' section shows a table of 13 resources, including 'BasicTest' and various 'Employee' classes. The table columns are Name, Type, Modified, Size, and Coverage. On the right, the 'Code Coverage' section shows a circular progress indicator at 77%, indicating the overall code coverage for the organization. Below this, the 'Metrics' section provides a breakdown of APEX Classes (Total: 13, Covered: 2, Uncovered: 11, Coverage: 67%) and APEX Triggers (Total: 0, Covered: 0, Uncovered: 0, Coverage: 0%).

3. Manage Access

- Go to “Set Up > Apps > Packaging > Installed Packages”.
- Click on “Manage Licenses”.

The screenshot shows the 'Installed Packages' section with a table containing one package, 'TestGen'. Below this is a summary table showing 'Allowed Licenses' and 'Used Licenses' both at 1. At the bottom is the 'Licensed Users' section, which includes a table with one user, 'System Administrator', who is active and has the role of 'System Administrator'.

Installed Packages	
Action	Package
Uninstall Manage Licenses	TestGen

Status	Active	Allowed Licenses	1
Expiration Date	Does not Expire	Used Licenses	1

Licensed Users			
Action	Full Name	Role	Active
Remove	testgen@webinteractiveinc.com		✓

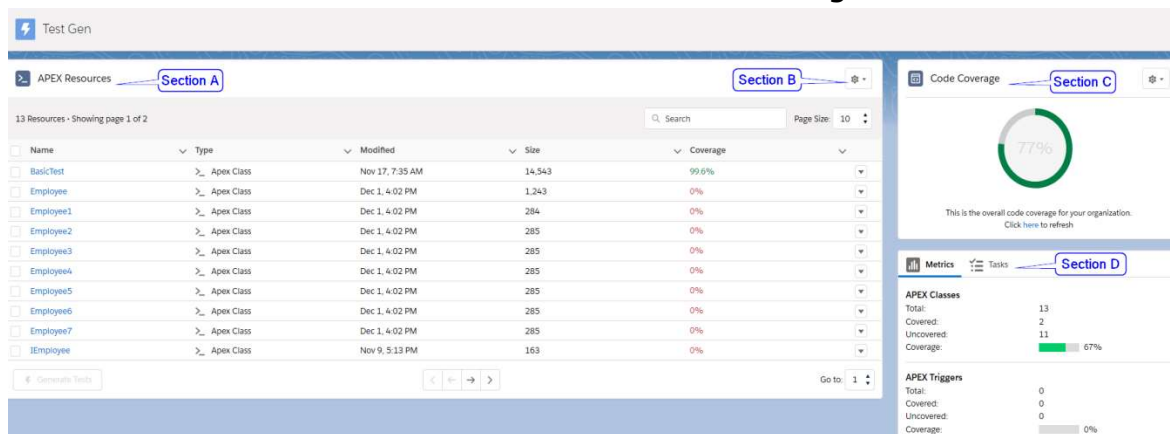
- To change please remove assigned user and add appropriate developer having “System Administrator” profile.

Please note: Free version only allows 1 user license and 5 test generations/month. There is no limit for number of lines. We would like to provide power to users to generate tests for the biggest class.

For License related queries please contact us via testgen@webinteractiveinc.com.

Application at a Glance

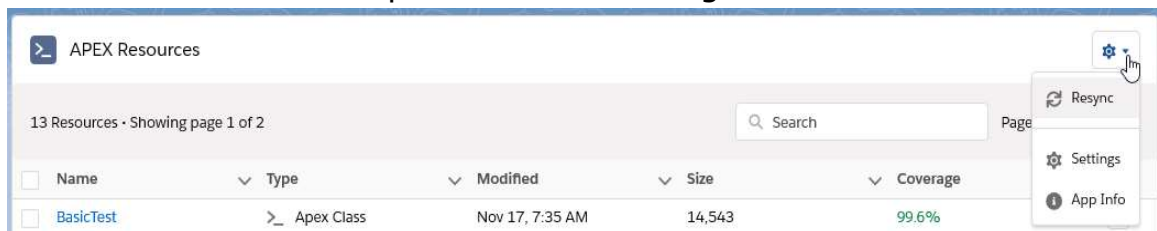
- Please refer to 4 main sections marked in below image.



1. **APEX Resources** (Marked as Section A)
 - Lists Apex classes & triggers with code coverage.
 - Generate tests for selected resources.
 - Go to view tests generated by app.
 - Shows issues if any.
2. **Settings Overview** (Marked as Section B)
 - **ReSync**: Refreshes latest information for APEX Resources, Code Coverage, and Metrics sections.
 - **Settings**: You can control output Test Class filename.
 - **App Info**: Shows App Name, Version, Licensee, Test Generation Usage, Limits, and Rollover period.
3. **Code Coverage** (Marked as Section C)
 - Shows overall code coverage for your organization. You can refresh view to see the latest code coverage at any time.
4. **Metrics & Tasks Tabs** (Marked as Section D)
 - Metrics tab shows count of Total, Covered, Uncovered APEX classes & triggers. Displayed percentage is only based on resource count.
 - Task tab shows currently running tasks (if any). E.g. If you start generating test then 1 running task will be displayed.

Settings Overview

- Please refer to available options in below image.

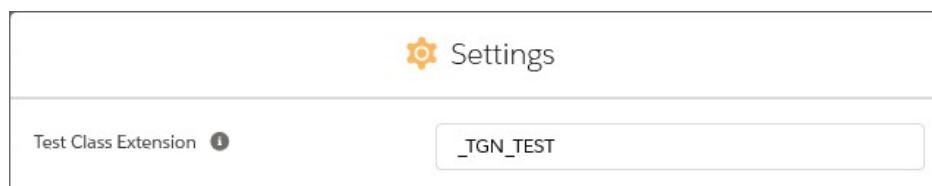


1. ReSync

- At any time you can click on "ReSync" to refresh data on entire page.

2. Settings

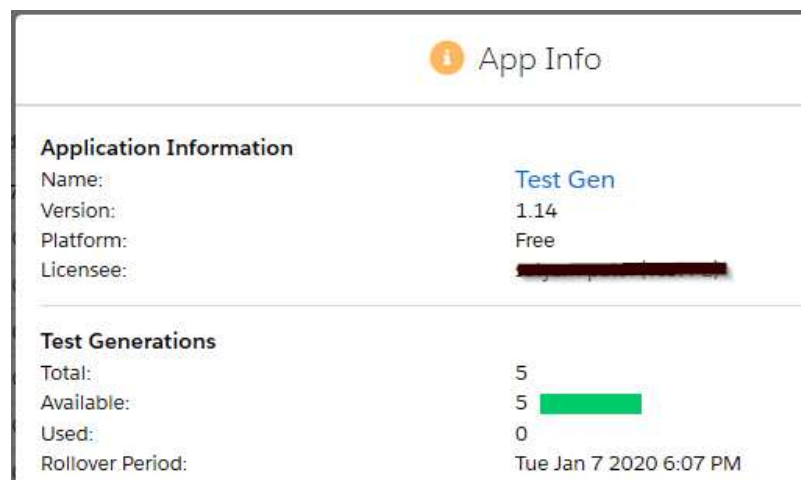
- Before generating Unit Tests, you can control Unit Test output filename generated by App.
- Go to Settings and use any unique name you would like.



Example: Keeping default settings, "Employee" class will have unit test class filename "Employee_TGN_TEST".

3. App Info

- You can see Application Information and Test Generations as below.



Continued...

- **Total:** Free version has total 5 test generations per month.
- **Available:** Shows available test generations. This value will decrease only when you perform "Generate Tests" for classes not covered by "Test Gen" app.
- **Used:** Shows count of attempted test generations. This value will increase only when you perform "Generate Tests" for classes not covered by "Test Gen" app.
- **Rollover Period:** When Rollover Period Ends Available and Used count will be reset to 5, 0 respectively.

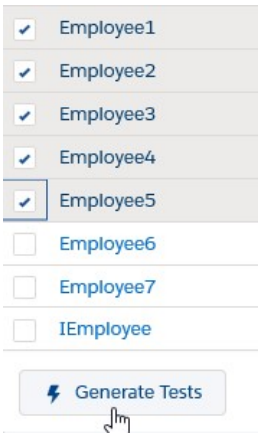
Please refer **Do's & Don'ts** section for more details.

Generate Unit Tests

- There are 2 ways to generate Unit Tests.

1. Multiple selection

- Select one or more classes and click “Generate Tests” button.



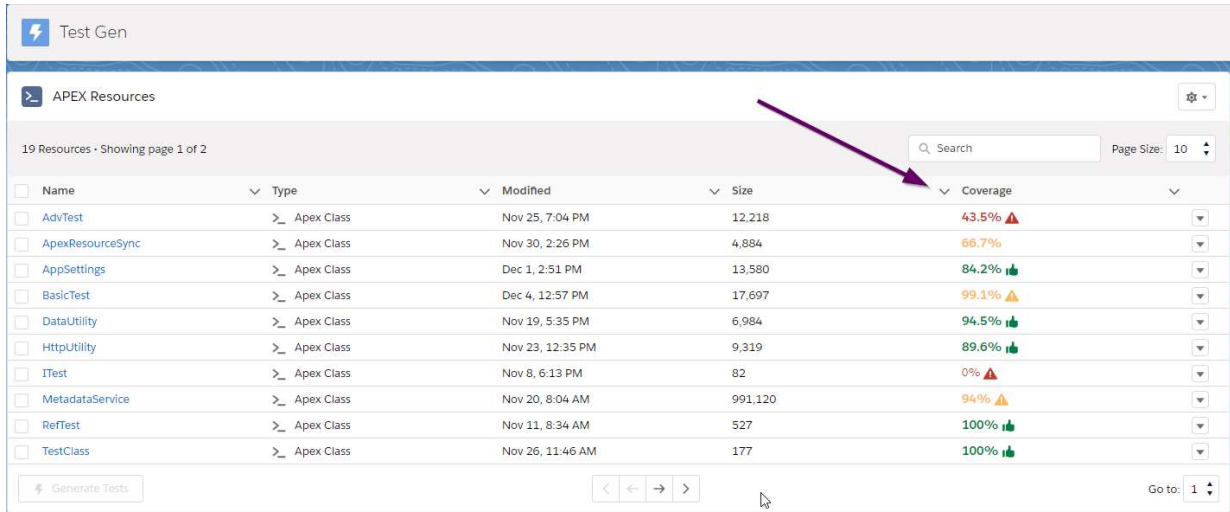
2. One Test at a time

- Locate the class you want to generate unit test for. Click menu item (last column) and select “Generate Tests”.



Code coverage view with color code

- Table will update view after “Running Task” is completed. If view is not refreshed please “ReSync”, wait until sync completes and reload page via clicking “Test Gen” App tab on top.
- Coverage column will have diff color and text based on below conditions.



Test Gen

APEX Resources

19 Resources • Showing page 1 of 2

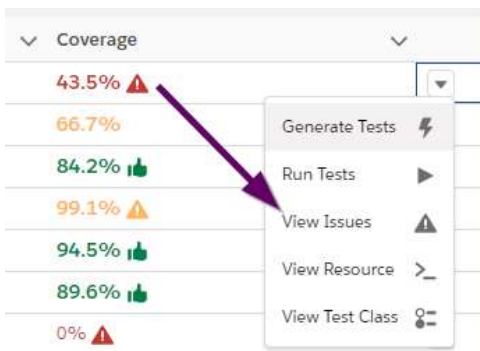
Search Page Size: 10

Name	Type	Modified	Size	Coverage
AdvTest	Apex Class	Nov 25, 7:04 PM	12,218	43.5% ⚠
ApexResourceSync	Apex Class	Nov 30, 2:26 PM	4,884	66.7% 🟡
AppSettings	Apex Class	Dec 1, 2:51 PM	13,580	84.2% 👍
BasicTest	Apex Class	Dec 4, 12:57 PM	17,697	99.1% 🟡
DataUtility	Apex Class	Nov 19, 5:35 PM	6,984	94.5% 👍
HttpUtility	Apex Class	Nov 23, 12:35 PM	9,319	89.6% 👍
ITest	Apex Class	Nov 8, 6:13 PM	82	0% ⚠
MetadataService	Apex Class	Nov 20, 8:04 AM	991,120	94.4% 🟡
RefTest	Apex Class	Nov 11, 8:34 AM	527	100% 👍
TestClass	Apex Class	Nov 26, 11:46 AM	177	100% 👍

Generate Tests

Go to: 1

- Coverage shows **BOLDED** text only for “Test Gen” generated tests.
- 75% or above code coverage without any issues will be **green**.
- 75% or above code coverage with any issues will be **Orange**.
- Below 75% code coverage will be **Red**.
- If there are any issues, you can see 🟡 or ⚠ based on code coverage %. Go to Menu item and select “View Issues” for details.



Coverage
43.5% ⚠
66.7% 🟡
84.2% 👍
99.1% 🟡
94.5% 👍
89.6% 👍
0% ⚠

- Generate Tests ⚡
- Run Tests ▶
- View Issues ⚠
- View Resource >
- View Test Class 🛠

“View Issues” page will list method(s) having issues. You can select each method for details. You can browse to your class and generated test class.

EmpWrapper Issues

Type	Description	Class	Method	Line ↑	Ratio
☐	Error	Test Exception	Employee	getByFirstname	41
☐	Error	Test Exception	Employee	add	49
☒	Error	Test Exception	Employee	remove	64
☐	Error	Test Exception	Employee	print	77

System.ListException: Invalid id at index 0: null

Class:	Employee	Line Number:	64
Namespace:	EmpWrapper	Column:	1
Method:	remove		

 class.EmpWrapper.Employee.remove: line 64, column 1
class.EmpWrapper_TGN_TEST222.Employee_remove_TGN_TEST222: line 64, column 1

 View Test Class

 Close

Troubleshoot “View Issues”

➤ Message:

Internal Error (107) - Unable to overwrite test class

Reason: This is due to new version can not overwrite test class generated by previous version. This happens when package is upgraded manually or via Push Upgrades.

Solution: Simply delete the associated test class and regenerate.

Please note: Delete/Generate test class will not increase used count.

➤ Message:

System.NullPointerException: Attempt to de-reference a null object

Reason: Required Null checks.

- Add null check on variables within method displayed in table above the message.
- Add null checks in constructor for input parameters.

E.g.

```
if (param != null)
{
    //initialize class variables
    //other code
}
```

- In constructor, initialize class variables with null or associated datatype values.

E.g.

```
List<string> employeenames = null;
String employeename = "";
Integer employeecount = 0;
```

Few items to remember...

Do's

1. Install "Test Gen" app only in Dev, Sandbox, or Test org and assign access to only developer(s) as shown in installation section.
2. Select biggest class with 0% code coverage. "Available" count will be reduced only when "Generate Tests" action produces a file.
3. You can generate tests for a class already having tests written outside of "Test Gen" app. In this case app will keep existing tests as-is and will create a new test file based on "Test Class Extension" setting.
4. "Resync" any time, if code coverage in developer console is diff then in app. "ReSync" and "Run Tests" actions will not affect "Available" count.
5. Please note that "Test Gen" app ignores to write tests for interfaces.
6. Every month you can generate tests for the same as previous or 5 new classes/triggers. "Available" and "Used" count resets every month after rollover period ends.
7. We understand there will be bug fixes and minor changes in code. You can re-generate tests for the same class/trigger as many times as you want. However, "Available" count reduces only when updated code has more than 20% changes.
8. To cover various conditions, you can write your own custom code-coverage in new file leveraging "Test Gen" generated tests.
9. If you select multiple classes to generate tests and "View Issues" show this message: *Process could not finish. Exceeded Salesforce resource limitations*. This is due to Salesforce limitations. Please re-generate test only for specific class again.

Don'ts

1. Don't install "Test Gen" app in production org.
2. Don't update or add your custom code-coverage in "Test Gen" generated files. These files will be overwritten and you will lose your changes.
3. Don't uninstall/reinstall app to reset "Available" & "Used" counter. "Test Gen" app will show same count even after reinstall.
4. Don't delete "Test Gen" generated tests to cover new code. Even such test(s) are removed; "Available" & "Used" count will not change within current rollover period.
5. Don't delete code file to cover new code. Even such code file(s) covered by "Test Gen" app is removed; "Available" & "Used" count will not change within current rollover period.