

Test your application

In order to write integration tests for your application, you need ability to spin up a geode cluster from your test and then run tests for your application that work against the cluster. Geode now ships with a JUnit Rule to help.

You can use this GfshRule like so:

Integration Test

```
public class LibraryTest {  
    @Rule  
    public GfshRule gfshRule = new GfshRule();  
  
    @Test  
    public void testSomeLibraryMethod() {  
        Library classUnderTest = new Library();  
        GfshScript.of("start locator --name=loc",  
            "start server --name=serv1",  
            "create region --name=test --type=REPLICATE").execute(gfshRule);  
        classUnderTest.doPut();  
        assertEquals("one", classUnderTest.doGet());  
    }  
}
```

While running this test, a locator and a server JVM will be spun-up by GfshRule.

Step-by-step guide

You need to follow the following steps to get this to work:

1. Specify geode-junit as a compile dependency for your tests by adding the following in your build.gradle

```
testCompile 'org.apache.geode:geode-junit:geodeVersion'
```

2. Specify the gradle plugin that will download and install the Geode distribution to be used by GfshRule by adding the following at the top of your build.gradle file

```
plugins {  
    id "io.pivotal.GeodeIntegrationTestPlugin" version "1.0"  
}
```

This will download the latest version of Apache Geode. If you want to specify a version use the following in your build.gradle

```
geodeIntegration {  
    version = "1.2.0"  
}
```

3. Write your integration tests by spinning up a Geode cluster using gfsh commands as shown above.

A very simple application can be found at: <https://github.com/sbawaska/geode-integration-test-example>