

# Runinng SQL script from JUnit

Asked 16 years ago   Modified 8 years, 7 months ago   Viewed 5k times



1

In the setup method to a JUnit test case I'm working on I need it to run a sql script on my database before each test case, and then a rollback afterwards.



I have tried using a tokenizer, which added each SQL command to a batch and then executing them. But I can't get working. So my question is if there is some standard method in JUnit to perform this action?



sql

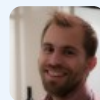
junit

Share

Improve this question

Follow

asked Dec 3, 2008 at 10:12



[Henrik Bierbum Bacher](#)

80 ● 2 ● 10

2 Answers

Sorted by:

Highest score (default)



2

You can try [DbUnit](#)



DbUnit is a JUnit extension (also usable with Ant) targeted at database-driven projects that, among other things, puts your database into a known state between test runs. This is an excellent way



to avoid the myriad of problems that can occur when one test case corrupts the database and causes subsequent tests to fail or exacerbate the damage.

DbUnit has the ability to export and import your database data to and from XML datasets. Since version 2.0, DbUnit can also work with very large datasets when used in streaming mode. DbUnit can also help you to verify that your database data match an expected set of values.

Share Improve this answer

Follow

edited Jun 20, 2020 at 9:12



Community Bot

1 • 1

answered Dec 3, 2008 at 10:14



Aaron Digulla

328k • 110 • 622 • 834



1



It's not the task of JUnit to test SQL statements.

You should create a Mocker ([EasyMock](#) e.g.) and isolate the connection. So the mocker can imitate the sql connection and its results. With this mocking object, you can check, if your sql connector class called the right statements.

If you want to test the SQL statement, its results and so on, you should use [DBUnit](#), as Aaron said.

Share Improve this answer

answered Dec 4, 2008 at 7:49

Follow



[guerda](#)

24k ● 28 ● 100 ● 150

---