Servoy Mobile Unit Testing

Starting with Servoy 7.2 (in developer) and Servoy 7.3 (for automated testing) you have the ability to run jsunit tests for mobile solutions. The test will run browser-side in an mobile 'test' client (which is actually a normal mobile client with small changes to allow unit tests - for automatic authentication, test-suite reporting, ...).

Mobile jsunit tests can be run - similar to Smart Client jsunit tests - both from Servoy Developer (using a browser installed in your OS) or as automated tests that can run as part of an automated build system (Jenkins, Cruise-Control, ...).

Creating a Test Solution

In order to write tests for your mobile solution - let's name it 'x' - you can create a new solution 'test_x' solution that has 'x' as a module.

Then you write the unit tests in the usual way. Methods prefixed with 'test' in global/form scopes will be considered to be test-cases and will run wrapped between 'setUp' and 'tearDown' methods of the same scope. Use 'jsunit.assert...' methods to check that everything happens as expected.

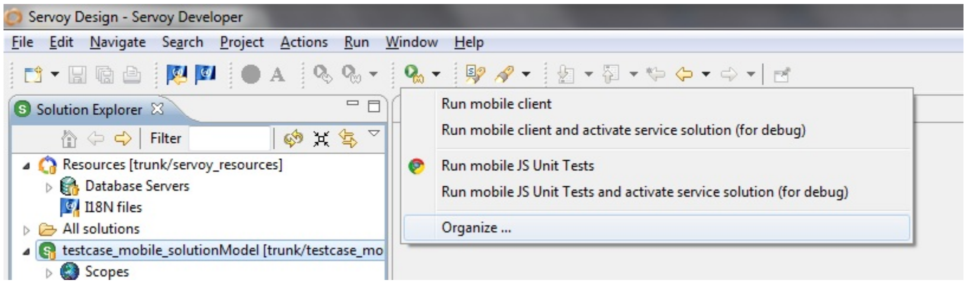
Create a 'test_x_service' solution that will feed data and table information as needed to your mobile client test solution - just like you would do with a normal mobile test solution.

A very basic sample mobile test solution setup is available for download at building a software factory tutorial.

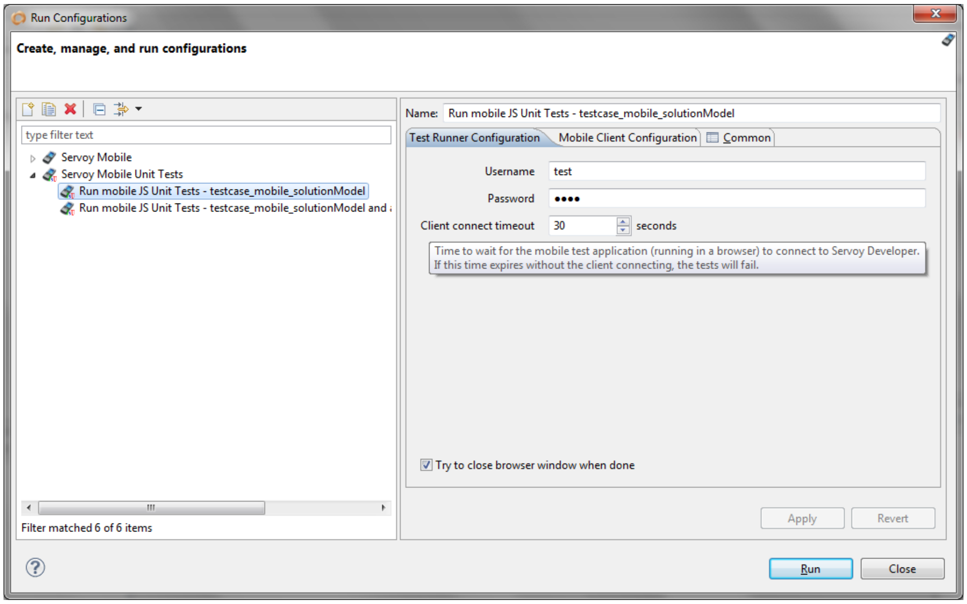
Running Tests in Servoy Developer

In order to run mobile jsunit tests in Servoy Developer you can use:

- the **Start Mobile Client in Browser** toolbar drop-down button. If you expand the drop-down, you will see two extra-entries for testing: **Run mobile JS Unit Tests** and **Run mobile JS Unit Tests and activate service solution (for debug)**:


Servoy Design - Servoy Developer
File Edit Navigate Search Project Actions Run Window Help
Solution Explorer
Filter
Resources [trunk/servoy_resources]
Database Servers
I18N files
All solutions
testcase_mobile_solutionModel [trunk/testcase_mo
Scopes
Run mobile client
Run mobile client and activate service solution (for debug)
Run mobile JS Unit Tests
Run mobile JS Unit Tests and activate service solution (for debug)
Organize ...

- Click on **Organize** and you will get a launch configuration window where you can see/modify the details used for running mobile client tests (a new tab besides the settings used by the usual mobile client launchers):


Run Configurations
Create, manage, and run configurations
type filter text
Servoy Mobile
Servoy Mobile Unit Tests
Run mobile JS Unit Tests - testcase_mobile_solutionModel
Run mobile JS Unit Tests - testcase_mobile_solutionModel and a
Filter matched 6 of 6 items
Test Runner Configuration
Mobile Client Configuration
Common
Apply
Revert
Run
Close

Name: Run mobile JS Unit Tests - testcase_mobile_solutionModel

Username: test

Password: ●●●●

Client connect timeout: 30 seconds

Time to wait for the mobile test application (running in a browser) to connect to Servoy Developer. If this time expires without the client connecting, the tests will fail.

☑ Try to close browser window when done

There you can for example change the username / password to match those that should be recieved in the mobile test service solution and alter the test client connect timeout (if developer doesn't receive a connection from the browser mobile test client in that interval it declares the tests as failed).
These two launchers mentioned above **Run mobile JS Unit Tests** and **Run mobile JS Unit Tests and activate service solution (for debug)** - even when modified - will be used by the context menu actions below. If you want to configure multiple launchers you can create new ones or duplicate an existing one. For example you might want to run tests when authenticated with one user or another - and you can define one launch configuration for each.

- the context menu in SolutionExplorer (on a solution, form, test method), similar to Smart Client: **right click** > **Run JS Unit Test(s)** > **Run tests** or **Run tests (activate service solution)**. You have two launchers here in case you want to debug the service solution while runnning tests - in which case it will automatically activate the service solution when running tests and switch back to the mobile solution when tests are done.
- once you already ran tests once, you can use the **Script Unit Test** view's context menu / toolbar to run tests again - similar to Smart Client testing.

For those interested in what's going on behind the scenes during a test-run:

- export mobile test client .war file - customized by launcher settings
- deploy mobile test client .war file in Servoy Developer's embedded container (Tomcat)
- activate service solution if launch required this
- prepare for testing in developer and open a browser to execute the browser-side jsunit tests
- try to close the browser when tests are done (this will work only for some browsers and some configurations - for example it will probably not work if the URL is open in a new tab of an already open browser)
- re-activate mobile testing solution if launch activated service solution

Creating Automated Servoy Mobile Unit Tests

You can automate this testing process by using the command-line/ant mobile .war workspace exporter, the .servoy solution workspace exporter and the mobile import test client as detailed here.

These, in conjunction with a Selenium based (local or cloud) setup allow you to run tests automated in all popular browsers/platforms/devices. For example a nice integration can be achieved with Servoy - Jenkins (build automation software) - Sauce Labs (cloud based selenium test automation provider).