

Examples

Best practices to use the Structure.Testy plugin:

- [Create some universal checklists](#)
- [Planning tests for a release](#)

Create some universal checklists

You might have sanity tests, usability checklists and other lists of tasks that you run through regardless of the release scope. In this case you'll find that it's easiest to [create a single structure](#) and do multiple uniquely-named test runs in it.

This is easy to do:

- Create your hierarchical list of issues and a unique View for displaying columns
- Add a test run with a unique name that describes the test version and environment.
- Set the result of each test for each issue. You can apply the same result to an entire group of issues by setting that result in the group's parent issue.

Planning tests for a release

Displaying test progress alongside development progress makes it easier to understand overall progress and plan releases.

- Create a structure
- Add a [Filter Synchronizer](#) to pull in development issues for the release:

Filter Saved filter JQL query
project="My Project" and fixVersion="0.1" ✓

Add If an issue passes the filter, add it to the structure.
 Place added issue at the top level
 Place added issue as a sub-issue of
 Allow move: if the matching issue is already in the structure, but it's not under this "parent" issue, move it.

Remove Remove from structure issues that don't pass the filter.
 Remove only from where added issues are placed

Create Cancel

- Add your test cases as sub-issues of the development issues. Test case issues can be of any type and from any project:

Key	Issue	Summary	v.0.05, Android	Fix Version/s
MP-1	+	Create mobile application	2 1 1 1	0.1
TE-125	☑	downloading, fresh installation	PASSED	
TE-126	☑	OS updates	PASSED	
TE-127	☑	data saving: SD card?	SKIPPED	
TE-130	☑	same data should be available from mobile	1	
TE-128	☑	login with existent account	FAILED	
TE-129	☑	create new account		

