Enhanced querying with JQL

- JQL FunctionsCustom Fields

JQL Functions

The following JQL functions are available for querying Xray issues in the Issue Search Page. They enable you to query the relationships between Xray issue types.

testExecutionTests

JQL Function	Parameters	Description	Example
testTestSet	P1 - Test Issue Key	Returns a list of Test Set issues associated with the input Test issue key.	<pre>issuetype = 'Test Set' and key in testTestSet('DEMO-1')</pre>
testSetTests	P1 - Test Set Issue Key /Filter of Test Sets	Returns a list of Test issues associated with the input Test Set issue key.	<pre>(1) issuetype = 'Test' and key in testSetTests ('DEMO-5') (2) issuetype = 'Test' and key in testSetTests('Test sets saved filter')</pre>
testsWithN oTestSet	P1 - Saved filter Name/ID	Returns a list of Test issues not associated with a Test Set.	<pre>(1) issue in testsWithNoTestSet() (2) issue in testsWithNoTestSet("saved_filter")</pre>
testPreCon ditions	P1 - Test Issue Key	Returns the Pre-Condition issues associated with the input Test issue key.	<pre>issuetype = 'Pre-Condition' and key in testPreConditions('DEMO-1')</pre>
preConditi onTests	P1 - Pre-Condition Issue Key	Returns the Test issues associated with the input Pre- Condition issue key.	<pre>issuetype = 'Test' and key in preConditionTests('DEMO-1')</pre>
testRequire ments	P1 - Test Issue Key/Filter name of Tests	Returns a list of Requirement issues associated with the input Test issue key/Filter of tests.	<pre>(1) issuetype = 'Feature' and key in testRequirements('DEMO-1') (2) issuetype = 'Feature' and key in testRequirements('Tests saved filter')</pre>
requiremen tTests	P1 - Requirement Issue Key/Filter of Requirement Issues	Returns a list of Test issues associated with the input Requirement issue key or saved filter with Requirements.	<pre>(1) issuetype = 'Test' and key in requirementTests('DEMO-10') (2) issuetype = 'Test' and key in requirementTests('Requirements saved filter')</pre>

 Lestevino P. P Pojeck Namekayda Martine all car of Test issues associated with the Requeres of Lesse 1. Test Version (Control of the specified paper.) Test Version (Control of the specified p				1	
P2. Fx Version (Optional) P2. Fx Version (Control of the fx Tool Status (Optional) P2. Fx Version (Control of the fx Tool Status (FX FX F		P1 - Project Name/Key/Id		issuetype = 'Test'	
(c)divides) (c)divides) (c)divides)	equeision	P2 - Fix Version		and issue in	
Image: Second J 10 - Car (Vacion / Car (Value)) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description J P - For Vacion / Car (Value) Description Value / For (Value) P - For Vacion / Car (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (Value) P - For Vacion / For (Value) Description Value / For (V				testsWithReqVersi	on('DEMO',
Pro-Field basis Pro-Field		(Optional)			'v1.0', 'v1.1')
(Dopone) (Dopone) escaptions P-Dipole Normatively II Returns a List of Test issues associated with the Test State of the inpol Fix Versions of the specified project. Issues 1 = (1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 1 + 0 + 0					
ediServer n edistry e					
on P2-Fx Version (Quironal) and issue in totatskithTestSetVersion(: 'DENO'. (Quironal) refErent (Quironal)		P1 - Project Name/Key/Id		issuetype = 'Test'	
image: Control in the control in th		P2 - Fix Version	issues of the input Fix Versions of the specified project.	and issue in	
Interference Interference<				testsWithTestSetV	Version('DEMO',
Image: Control of the control base Returns a List of Test Base associated with he pour feature structure s					'v1.0', 'v1.1')
testExecuti onTests P1 - Test Execution issue Keyfol or Filer D 2, Jger essigned in separated by Tripley (Veg Pacebox P2 - User viols scale from (Optional). Returns a List of Test issues associated with the input Test Execution issues from P7 optionally filered by the current for monitors to reach Test files with separated by Tripley (Veg Pacebox Test R, Status be cent to at subset (P2 - User viols scale from (Optional). (1) issuetype = 'Test' and issue in testExecutionTests('DEMO-9') (2) P3 - User viols scale from (Optional). Pachet Test files with subset the or Table from (Optional). Pachet Test files with values are: PASS FAL EXECUTING. ABORTED, TODO and all custom statuses. (1) issuetype = 'Test' and issue in testExecutionTests('DEMO-9') (2) P4 - Defects Flog with values from a particular soft of Test Executions where no comments (Optional). Pachet Test from status states are: PASS FAL EXECUTING. ABORTED, TODO and all custom statuses. (1) issuetype = 'Test' and issue in testExecutionTests('DEMO-9') (3) P5 - Existence of drom (Spiper, mond/smiller or pack(Symaller or pack()), respectively, respectively, the = sign or the tablesmes of the absense of digins is read as exactly If the absense is read as exactly If the absense is read as exactly P9 - Finished not (Spiper, mond/smiller or pack(smiller or pack(smiller or pack(smiller or pack(smiller om as exa		Pn - Fix Version			
onTests Keyld or Film ID Execution issues from Pr optionally filtered by the current first run status for orch Test is subset. issuettype = 'Test': P2-Test Run Status liter Terminetr P tem sine to a single Test Execution issues from Pr optionally filter containing multiple Test Execution issues from Pr optionally filter containing multiple Test Execution issues from Pr optionally filter containing multiple Test Execution issues from Pr optional all customs statuses. issuettype = 'Test': and issue in testExecutionTests('DEXO-9', (); P3-Defects Flag with on the run of side (red from Run in out as assigned to execute the Test Run Status et al. The Ten analyzing the joint values of all Test Run Assignees, in the Test Run Status et al. The Ten analyzing the joint values of all Test Run Assignees, in the Test Run Status et al. The Ten analyzing the joint values of all Test Run Assignees, in the Test Run Status et al. The Ten analyzing the joint values of all Test Run Assignees, in the Test Run Run Assignees, in the "Test": and issue in testExecutionTests('DEXO-9', (); P6-Existence of executer is the able of test Run Run Assignees, in a particular set of Test Executions where no Defects were created. ''''''''''''''''''''''''''''''''''''		(Optional)			
<pre>test run status for each Test issue is quartiets by "Test ?" segurated by "Toppi (op nows)" results for each Test issue by in 10 or saved filter containing mulpipe Test Execution issue by in 10 or saved filter containing mulpipe Test Execution issue by in 10 or saved filter containing mulpipe Test Execution issue by in 10 or saved filter containing mulpipe Test Execution issue by in the save in transfilter containing mulpipe Test Execution issue by in the save in transfilter containing mulpipe Test Execution issue by in the save in transfilter containing mulpipe Test Execution issue by in the save in transfilter containing mulpipe Test Execution issue by in the save in transfilter containing mulpipe Test Execution issue by in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save in the save of save in the save in the save in the save of save in the save of the full absence of save in the save is read as execution resubly smaller or execution is read as exe</pre>				(1)	
separated by "T(pipe) (Or P3 - User assigned to execute Test Run (Optional).and issue in testExecutionTests(`DENO-5'.)P3 - User assigned to execute Test Run (Optional).Possible Test Run Status values are: PASS, FAIL, EXECUTING, ABONTED. TODO and all austion statuses.C()P4 - Defects Flag with without any Lassignee, then "NULL"should be used.Possible Test Run Status values are: PASS, FAIL, EXECUTING, ABONTED. TODO and all austion statuses.C()P4 - Defects Flag with without any Assignee, then "NULL"should be used.Possible Test Run Status values are: PASS, FAIL, EXECUTING, ABONTED. TODO and all austion statuses.C()P5 - User who executed the Test Run Option and all austion statuses.Possible Test Run Status values are: PASS, FAIL, EXECUTING, ABONTED. TODO and all austion statuses.C()P6 - Existence of comments (Optional)Privation aparticular set of Test Executions where noPrivation aparticular set of Test Executions where no(a)P7 - Existence of evidences (Optional)Privation aparticular set of Test Executions where no'saved Test ExecutionTests('DENO-5', 'user A')'and issue in testExecutionTests('DENO-5', 'and issue in testExecutionTests('DENO-5', 'user A')P7 - Existence of evidences (Optional)'saved Test Execution'saved Test ExecutionTests('DENO-5', 'user A')'saved Test ExecutionTests('DENO-5', 'user A')P7 - Existence of evidences (Optional)'saved Test ExecutionTests('DENO-5', 'user A')'saved Test ExecutionTests('DENO-5', 'user A')P7 - Existence of evidences (Optional)'saved Test ExecutionTests('DENO-5', 'user A', 'true')'saved Test Execut	onrooto	-		issuetype = 'Test'	
 P3-User assigned to execute Test Run (Optional). P4-Defects Ray with reactions. ABORTED. TODO and all custom statuses. P5-User who executed the Test Run (Sizue values are readed to execute the Test Run Sizues. Should be used. For status in a count the Test Run (Sizue values of all Test Run Assigner, then "		separated by " "(pipe) (Op		and issue in test	ExecutionTests('DEMO-9')
execute Test Run (Optional). Possible Test Run Status values arc PASS, FALL, EXECUTING, ABORTED, TODO and all custom statuses. issuetype = 'Test': P4 - Defects Flag with value in the or hise (new ds to be true or fails (new the test Run (optional). P3 corresponds to the user assigned to execute the Test Run, while P5 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to executions where no Defects were oreated. P3 corresponds to the user assigned to execute the Test Run (optional). P3 corresponds to the user assigned to executions where no Defects were oreated. P3 corresponds to the correspond to the run of task passed to execution Test Run of task passed test Execution Test Run of task passed test Execution Test Run of				(2)	
<pre>P3 corresponds to the user assigned to execute the Test Run, while P3 corresponds to the one who actually executed the test Run (optional) P5 - User who executed the Test Run (optional) P6 - Existence of execution rests (Optional) P7 - Existence of execution rests (Optional) P7 - Existence of execution rests (Optional) P8 - Started from (symbols >/c are read as bygger, exactly, respectively, the = sign or the full absense of signs is read as exactly) P9 - Thished on (symbols >/c are read as bygger, execution/rest of signs is read as exactly)</pre>			Possible Test Run Status values are: PASS, FAIL,	issuetype = 'Test'	
<pre>value in true or fase (new dist to be true or fase (new dist to be true or fase). P5 - User who executed the Test Run (optional) P6 - Existence of comments (Optional) P7 - Existence of evidences (Optional) P3 - Started from (symbols >k - are read as biggor, exactly, mealer or exactly,) P9 - Finished on (symbols >k - are read as biggor, exactly, respectively; the sign or the full absense of signs is read as exactly) P9 - Finished on (symbols >k - are read as biggor, the full absense a sign or the full absense of signs is read as exactly) P9 - Finished on (symbols >k - are read as biggor, the full absense a sign or the full absense of signs is read as exactly) (6) issuetype = 'Test' and issue in testExecutionTests(issuetype = 'Test' and issue in testExecutionTests(issue in testExecutionTes</pre>		(Optional).	EXECUTING, ABORTED, TODO and all custom statuses.	and issue in test	ExecutionTests('DEMO-9',
<pre>ds to be true or fase). B. F. Or analyzing the joint values of all Test Run Assignees. " if. For analyzing the joint values of all Test Run Assignees. " if. For analyzing the used. For taking (into account the Test Run without any Assignee, then "NULL" should be used. Test Run (foint) P6 - Existence of comments (Optiona) P7 - Existence of comments (Optiona) P7 - Existence of comments (Optiona) P8 - Sharde from gogne: sare fad as by acutly or specify: respectively: the sing is read as exactly or rhe full absense of signe is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectively: the sign of the full absense of signs is read as exactly respectiv</pre>			P3 corresponds to the user assigned to execute the Test Run, while P5 corresponds to the one who actually executed		'PASS')
P5-User who executed the Test Run (optional). without any Assignce, then 'NULL_' should be used. isauetype = 'Test' P6-Existence of comments (Optional) P7-Existence of other assignce, then 'NULL_' should be used. isauetype = 'Test' P7-Existence of other assignce, then 'NULL_' should be used. ''PASS'. ''PASS'. P7-Existence of other assignce, then 'NULL_' should be used. ''PASS'. ''PASS'. P8-Started from (optional) ''PASS'. ''user A'') P8-Started from (optional) ''PASS'. ''user A'') P9-Finished on (symbols) ''Saved Test Execution ''Saved Test Execution P9-Finished on (symbols) ''Saved Test Execution ''Saved Test Execution P9-Finished on (symbols) ''Saved Test Execution ''Saved Test Execution Filter', ''Saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''saved Test Execution ''sav			it. For analyzing the joint values of all Test Run Assignees, ""	(3)	
Implementation If you pass the as the value for P4, the query returns all Tests from a particular set of Test Executions where no Delects were created. and insue in testExecutionTests('DEMO-9', 'PASS', 'user A') P7 - Existence of evidences (Optional) P7 - Existence of evidences (Optional) ''DASS', 'user A') P8 - Started from (symbols >/c are read as bigger, exactly respectively; the = sign or fs will absense of signs is read as exactly) ''Saved Test ExecutionTests(P9 - Finished on (symbols >/c are read as bigger, exactly respectively; the = sign or the full absense of signs is read as exactly) ''Saved Test ExecutionTests((6) issue the testExecutionTests(is read as exactly ''saved Test Execution Pilter', ''saved Test Ex				issuetype = 'Test'	
P6-Existence of evidences (Optional) Tests from a particular set of Test Executions where no Delects were created. 'LASS', 'Less'					ExecutionTests('DEMO-9',
<pre>evidences (Optional) P3 - Started from (symbols ></pre>					
<pre>(4) (symbols >/< are read as bigger, exactly implementations is read as bigger, exactly implementations and is a sectly) P9 - Finished on (symbols >/< are read as bigger, exactly implementations is read as exactly) P9 - Finished on (symbols >/< are read as bigger, exactly implementations is read as exactly) (5) (6) issuetype = 'Test' and issue in testExecutionTests(and issue in testExecutionTests('Saved Test Execution Filter', 'Saved Test Execution Filter', 'user A') (6) issue in testExecutionTests('Saved Test Execution Filter', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution Filter', 'Saved Test Execution F</pre>					'user A')
<pre>(symbols >/< are read as bigger, exactly/smaller or exactly, respectively; the = sign or the full absense of signs is read as exactly) P9 - Finished on (symbols >/< are read as bigger, exactly/smaller or exactly, if the = sign o r the full absense of signs is read as exactly) (5) respectively; the = sign o r the full absense of signs is read as exactly (6) respectively; the = sign o r the full absense of signs is read as exactly (6) respectively; the = sign o r the full absense of signs is read as exactly (6) respectively; the = sign o r the full absense of signs is read as exactly (6) respectively; the = sign o r the full absense of signs is read as exactly (6) respectively; the = sign o r the full absense of signs is read as exactly (7) respectively; the = sign o r, user A', 'true') (7) result in testExecutionTests(r'saved Test ExecutionTests(r'saved Test Executio</pre>				(4)	
<pre>and issue in testExecutionTests(</pre>		(symbols >/< are read as		issuetype = 'Test'	
of signs is read as exactly P9 - Finished on (symbols >/s are read as bigger, exactly/smaller or exactly, respectively, the = sign o r the full absense of signs is read as exactly) (5) (6) issue in testExecutionTests(Filter', 'user A') (6) issue in testExecutionTests(Filter', 'user A') (6) issue in testExecutionTests(Filter', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution Filter', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution Filter', 'user A', 'true')		exactly, respectively; the =		and issue in test	ExecutionTests(
<pre>(5) * are read as bigger, exactly/smaller or exactly, respectively; the = sign o r the full absense of signs is read as exactly)</pre> (5) issuetype = 'Test' and issue in testExecutionTests(of signs is read as exactly		Filter',	'Saved Test Execution
<pre>>< are read as bigger, exactly/smaller or exactly, respectively; the = sign o r the full absense of signs is read as exactly) (5) issuetype = 'Test' and issue in testExecutionTests('Saved Test Execution Filter', '', 'user A') (6) issue in testExecutionTests(Saved Test Execution Filter', '', 'user A', 'true') (7) issue in testExecutionTests(Saved Test Execution Filter', '', '', '', '', '', '', '', '', '',</pre>		P9 - Finished on (symbols			'PASS')
respectively; the = sign o r the full absense of signs is read as exactly) (6) (6) issue in testExecutionTests('Saved Test Execution ''., ''user A') (6) issue in testExecutionTests('Saved Test Execution Filter', '', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution		>/< are read as <i>bigger</i> ,		(5)	
and issue in testExecutionTests('Saved Test Execution Filter', 'user A') (6) issue in testExecutionTests('Saved Test Execution Filter', '', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution Filter', '', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution		respectively; the = sign o			
<pre>'Saved Test Execution Filter', '', 'user A') (6) issue in testExecutionTests(Filter', '', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution</pre>				and issue in test	:ExecutionTests(
<pre>Filter',</pre>					'Saved Test Execution
<pre>'user A') (6) issue in testExecutionTests(</pre>				Filter',	
(6) issue in testExecutionTests('Saved Test Execution Filter', '', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution					· · · ,
<pre>Saved Test ExecutionTests(</pre>					'user A')
'Saved Test Execution Filter', '', 'user A', 'true') (7) issue in testExecutionTests('Saved Test Execution				(6)	
<pre>Filter',</pre>				issue in testExecuti	onTests(
<pre>'', 'user A', 'true') (7) issue in testExecutionTests(</pre>					'Saved Test Execution
<pre>'user A', 'true') (7) issue in testExecutionTests(</pre>				Filter',	
(7) issue in testExecutionTests('Saved Test Execution					
issue in testExecutionTests('Saved Test Execution					'user A', 'true')
'Saved Test Execution				(7)	
				issue in testExecuti	lonTests(
Filler,				Filter'	'Saved Test Execution
				, ,	
i I					

				",", 'false', 'admin')
			(8)	
			issuetype = 'Test'	
			and issue in testExecutionTest	sts(
				'CALC-397',
				",
				",
				'false',
				",
				'true',
				'false',
				'>2016-05-31',
				'<2016-06-30')
testsWitho utTestExec	P1 - Saved filter Name/ID	Returns a list of Tests that are not associated with a Test Execution to be executed	(1)	
ution			<pre>issuetype = Test and issue testsWithoutTestExecution()</pre>	in
			(2)	
			<pre>issuetype = Test and issue testsWithoutTestExecution("</pre>	

requiremen ts	P1 - Status list separated by " "(pipe)	Returns a list of Requirement Issues with the provided coverage status.	
	P2 - Project (Optional)	Please provide <i>Project</i> parameter (P2) to restrict the requirements to the specified project.	<pre>issue in requirements('OK','Calculator')</pre>
	P3 - Version to calculate requirement status (Optional) P4 - Test Environment (Optional)	If analyzing on a specific version, then the <i>Project and Versio n</i> parameters must be filled.	Please note Although optional, it is highly recommended to specify the <i>Project</i> parameter as means to define the project having the requirments and thus
	P5 - Flat (Optional) P6 - ToDate (Optional) P7 - Saved Filter	<u>Optional</u> filters include: <i>Test Environment</i> , for taking into account the Test Executions made for that environment. For analyzing the joint	reduce the amount of issues that will be processed/returned. Otherwise, requirements from <i>all</i> JIRA projects will be processed, which possibly is something that you don't want or need at all.
	(Optional)	values of all environments, "" should be used. For taking into account the Test Executions without any Test Environment assigned, then "NULL" should be used.	
		Flat that indicates whether all Requirements (not only parents) should be searched. If "Flat" is not provided, the default value is 'false'.	(2) priority = Major and fixVersion <= 'v3.0' and
		<i>ToDate</i> considers only those requirements executions before a specific date/time (the date literal must follow the ISO8601 format).	<pre>issue in requirements('NOK', 'Calculator', 'V4.0')</pre>
		Saved Filters considers only requirements from that specific filter.	(3) issue in
			requirements('NOK', '', '', '', '', '2014- 01-01')
			(4)
			issue in
			requirements('OK',
			'Calculator',
			'v1.0',
			'chrome'
			'false'
			'2014-08-30')
			(5)
			issue in
			requirements('NOK',
			'Calculator',
			'v2.0',
			'', 'true')
			ciue)
			<pre>(6) issue in requirements('NOK', 'Calculator', 'v2.0', 'chrome', 'false', '', 'Requirements saved filter')</pre>

requiremen tsWithStat usByTestPI an	P1 - Status list separated by " "(pipe) P2 - Test Plan Issue Key P3 - Test Environment (Optional) P4 - Flat (Optional) P5 - ToDate (Optional) P6 - Project (Optional) P7 - Saved Filter (Optional)	Returns a list of Requirement Issues with the coverage status calculated for the given Test Plan issue. Optional filters include: Test Environment, for taking into account the Test Executions made for that environment. For analyzing the joint values of all environments, "" should be used. For taking into account the Test Executions without any Test Environment assigned, then "NULL" should be used. Flat that indicates whether all Requirements (not only parents) should be searched. If "Flat" is not provided, the default value is 'false'. ToDate considers only those requirements executions before a specific date/time (the date literal must follow the ISO8601 format). Project and Saved Filters considers only requirements from that specific project or filter.	<pre>(1) issue in requirementsWithStatusByTestPlan('OK', 'TP-123') (2) issue in requirementsWithStatusByTestPlan('NOK',</pre>
defectsCre atedDuring Testing defectsCre atedDuring TestExecut ion	P1 - Test Issue Key/Filter of Test Issues P1 - Test Execution issue Key or Test Execution based Filter P2 - List of users separated by " " (pipe). (Optional)	Return a list of defects created during the execution of the specified Tests. Returns a list of Defects created during the execution of the specified Test Executions; can optionally be filtered by the Defect Issue Assignee username.	<pre>(1) issue in defectsCreatedDuringTesting() (2) issue in defectsCreatedDuringTesting("TEST- 123") (3) issue in defectsCreatedDuringTesting ("saved_filter") (1) issue in defectsCreatedDuringTestExecution(TEST-123) (2) issue in defectsCreatedDuringTestExecution (saved_filter) (3) issue in defectsCreatedDuringTestExecution (saved_filter, 'userl user2') (4) issue in defectsCreatedDuringTestExecution(TEST- 123, 'userl user2')</pre>

defectsCre atedForRe	P1 - Requirement key or saved filter	Returns a list of defects created during the execution of Tests covering the specified requirements.	(1) issue in defectsCreatedForRequirement("REQ-
quirement			123")
			(2)
			<pre>issue in defectsCreatedForRequirement ("saved_filter")</pre>
manualTes	P1 - Saved filter Name/ID	Returns a list of manual tests that have no test steps.	(1)
tsWithoutS teps			issue in manualTestsWithoutSteps()
			(2)
			<pre>issue in manualTestsWithoutSteps ("saved_filter")</pre>
testTestEx ecutions	P1 - Test Issue Key/Id or Filter Name/Id	Returns a list of test executions associated with the input	(1)
ecutions		Test Issues from <i>P1</i> optionally filtered by the current Test status in each Test Execution issue.	issuetype = 'Test Execution'
	P2 - Test Run Status list separated by " "(pipe) (Op	Parameter P1 can either be a single Test issue key or Id or a	and issue in testTestExecutions('DEMO-9')
	tional)	saved filter name or id containing multiple Test issues.	(2)
		Possible Test Run Status values are: PASS, FAIL, EXECUTING, ABORTED, TODO and all custom statuses.	issuetype = 'Test Execution'
			and issue in testTestExecutions('DEMO-9',
			'PASS')
			(3)
			issuetype = 'Test Execution'
			and issue in testTestExecutions(
			'Saved Test Filter',
			'PASS')
testExecWi	P1 - Username (Optional)	Returns a list of test executions where a user has at least	(1)
thTestRuns AssignedT oUser	P2 - Status (Optional) Username is required in	one test run assigned to him. You can optionally specify a user with P1, or if the user is omitted the current user will be used. Note that if you are not logged in to JIRA, a user must	issuetype = 'Test Execution' and
ouser	case we use this parameter	be specified.	issue in testExecWithTestRunsAssignedToUser()
		If you use status parameter then user is required	(2)
			issuetype = 'Test Execution' and
			<pre>issue in testExecWithTestRunsAssignedToUser ('userDPC')</pre>
			(3)
			issuetype = 'Test Execution' and
			<pre>issue in testExecWithTestRunsAssignedToUser ('userDPC', "FAIL")</pre>
	P1 - Test Execution Issue	Return a list of Test Sets that have at least one test in P1.	(1)
testSetPa rtiallyIn	Key/Test Plan Issue Key /Id or Filter Id		issuetype = 'Test Set' and
- · · ·			
-			issue in testSetPartiallyIn('DEMO-15')
•			<pre>issue in testSetPartiallyIn('DEMO-15') (2)</pre>
•			(2)
			(2) issuetype = 'Test Set' and
			<pre>(2) issuetype = 'Test Set' and issue in testSetPartiallyIn('testExecList')</pre>
			<pre>(2) issuetype = 'Test Set' and issue in testSetPartiallyIn('testExecList') (3)</pre>

	P1 Toot Execution leaves	Poture a list of Tast Sate that have all its tasts in D4	(1)
testSetFu llyIn	P1 - Test Execution Issue Key/Test Plan Issue Key	Return a list of Test Sets that have all its tests in P1.	(1)
	/Id or Filter Id		<pre>issuetype = 'Test Set' and issue in testSetFullyIn('DEMO-15')</pre>
			issuetype = 'Test Set' and
			<pre>issue in testSetFullyIn('testExecList')</pre>
			(3)
			issuetype = 'Test Set' and
			issue in testSetFullyIn('testPlanList')
estPlanTe sts	P1 - Test Plan Key/ Filter of Test Plans	Returns a list of tests that are associated with the test plan.	(1)
115		The "status" parameter is optional and allows to filter Test	issue in testPlanTests("DEMO-10")
	P2 - Status (Optional)	issues in a specific Plan with the specified execution status. If the "status" parameter is present, users might also pass the	(2)
	P3 - Environment (Optional)	"environment" parameter. If this parameter is filled, Xray will provide all Tests in a Test Plan that are in the specified "status" and for the specified "environment".	issue in testPlanTests("Test Plans saved filter","TODO")
			(3)
			issue in testPlanTests("DEMO-10","TODO")
			(4)
			issue in testPlanTests("DEMO-10","TODO","IOS")
			When searching for Tests with a certain status inside a Test Plan, we recommend you to use the custom field search instead.
			Xray has created a new way of searching with big improvements when filtering by test status, using the Custom Fields: (3) issuetype = Test and TestRunStatus = "DEMO-
			10 - TODO" (4) issuetype = Test and TestRunStatus = "DEMO- 10 - TODO environment:IOS"
stExecutio	P1 - Test Plan Key/Filter of Test Plans	Returns a list of test executions that are associated with a Test Plan or a saved filter of Test Plans.	10 - TODO" (4) issuetype = Test and TestRunStatus = "DEMO-
estPlanTe stExecutio ns			<pre>10 - TODO" (4) issuetype = Test and TestRunStatus = "DEMO- 10 - TODO environment:IOS" (1) issue in testPlanTestExecutions("DEMO-10")</pre>
stExecutio			<pre>10 - TODO" (4) issuetype = Test and TestRunStatus = "DEMO- 10 - TODO environment:IOS" (1) issue in testPlanTestExecutions("DEMO-10") (2) issue in testPlanTestExecutions("Test Plans</pre>
estPlanRe	of Test Plans P1 - Test Plan Key/Filter	Test Plan or a saved filter of Test Plans. Returns the Requirement issues that are indirectly associated, through Test issues, with a Test Plan or a saved	<pre>10 - TODO" (4) issuetype = Test and TestRunStatus = "DEMO- 10 - TODO environment:IOS" (1) issue in testPlanTestExecutions("DEMO-10") (2) issue in testPlanTestExecutions("Test Plans saved filter") (1) issue in testPlanRequirements("DEMO-20")</pre>
estPlanRe quirements	of Test Plans P1 - Test Plan Key/Filter	Test Plan or a saved filter of Test Plans. Returns the Requirement issues that are indirectly associated, through Test issues, with a Test Plan or a saved	<pre>10 - TODO" (4) issuetype = Test and TestRunStatus = "DEMO- 10 - TODO environment:IOS" (1) issue in testPlanTestExecutions("DEMO-10") (2) issue in testPlanTestExecutions("Test Plans saved filter") (1) issue in testPlanRequirements("DEMO-20") (2) issue in testPlanRequirements("Test Plans</pre>

testReposit oryFolderT ests	P1 - Project Key P2 - Folder Path <i>P3 - Flatten (Optional)</i>	Returns the list of Tests contained in a folder (P2) of the Test Repository of a Project (P1) May optionally include the Tests in sub-folders by setting Flatten (P3) to "true".	 (1) issue in testRepositoryFolderTests("CALC", 'Parent/Child') (2) issue in testRepositoryFolderTests("CALC", 'Parent/Child', "true")
testPlanFol derTests	 P1 - Test Plan Key P2 - Folder Path P3 - Flatten (Optional) P4 - Test Run Status (Optional) P5 - Test Environment (Optional) 	Returns the list of Tests contained in a folder (P2) of a Test Plan (P1). May optionally include the Tests in sub-folders by setting Flatten (P3) to "true". Can also filter by Tests Run Status (P4) for a given Test Environment (P5). To analyze the joint values of all Test Environments, "" should be used. To analyze the Test Executions without any Test Environment assigned, then "NULL" should be used.	 (1) issue in testPlanFolderTests(CALC-10, 'Parent/Child') (2) issue in testPlanFolderTests(CALC-10, 'Parent/Child', "true") (3) issue in testPlanFolderTests(CALC-10, 'Parent/Child', "true", "TODOJFAIL", "windows")
projectPare ntRequire ments	P1 - Project Key	Returns the list of Requirement issues, from a given Project, which are not Sub-requirements	(1) issue in projectParentRequirements("CALC")
testExecuti onsWithCo mpletedTe stRunsSince	P1 - Date P2 - Filter ID/Name (Optional)	Return the list of Test execution (belonging to the given filter) that have Test Runs finished since the given date.	 (1) issue in testExecutionsWithCompletedTestRunsSince (2022-01-01) (2) issue in testExecutionsWithCompletedTestRunsSince (2022-01-01 12:00, "Current Sprint TestExecs") (3) issue in testExecutionsWithCompletedTestRunsSince(-3 d, 10101)
testPlansW ithComplet edTestRun sSince	P1 - Date P2 - Filter ID/Name (Optional)	Return the list of Test Plans (belonging to the given filter) that have Test Runs finished since the given date.	 (1) issue in testPlansWithCompletedTestRunsSince(2022-01-01) (2) issue in testPlansWithCompletedTestRunsSince(2022-01-01 12:00, "Current Sprint TestPlans") (3) issue in testPlansWithCompletedTestRunsSince(-3d, 10101)

Custom Fields

Xray also provides custom fields to allow more refined queries when searching for issues.

JQL Function	lssue Type	Description	Example
Test Type	Test	The Test type: Manual; Cucumber; Generic	issuetype = 'Test' and "Test Type" = "Manual"

TestRunSt atus	Test	This is a calculated custom field that provides the latest Test Run status based on the current "Test Run Status Version Strategy" option configured in the Xray administration.	issuetype = 'Test'
			and TestRunStatus in ("FAIL", "ABORTED")
		Syntax: TestRunStatus = "[Group (version or TestPlan)] - [Status] environment:[environment]"	-
		Only the Status is mandatory; if only the status is provided, Xray will assume you are searching for the	issuetype = 'Test'
		latest execution	and TestRunStatus = "PASS"
		Xray will lookup for all Tests with Status in that particular version and environment .	-
		Read more about Status and environments.	issue in testPlanTests ("TESTPLAN-123")
		Test Run Status The latest Test Run Status is calculated based on the latest final Test Run (i.e., latest finish	and TestRunStatus = "TESTPLAN-123 - PASS"
		date) or, in case there are no final Test Runs, the latest non-final Test Run (i.e., latest start date). Please see the custom fields preferences page.	_
			issue in testPlanTests ("TESTPLAN-123")
		For consistent results when searching for TestPlan results, filter with the testPlanTests function.	and TestRunStatus = "TESTPLAN-123 - TODO"
			-
			issuetype = 'Test'
			and TestRunStatus = "FAIL environment:Android"
			- issuetype = 'Test'
			and TestRunStatus = "v3.0 - PASS environment: Android"
			-
			issue in testPlanTests ("TESTPLAN-123")
			and TestRunStatus = "TESTPLAN-123 - PASS environment:Android"

Requireme nt Status	Require ment	This is a calculated custom field that provides the requirement coverage status . Possible status values are:	<pre>issuetype = 'New Feature' and "Requirement</pre>
		OK - All tests passed the requirement	Status" = "OK"
			_
		NOK - At least one test failed	issuetype = 'New Feature'
		NOTRUN - At least one test did not run	and "Requirement Status" in ("NOTRUN",
		UNCOVERED - The requirements is not associated with tests	"UNCOVERED")
		testTestExecutions	_
			issuetype = 'New Feature'
		Syntax: "Requirement Status" = "[Group (version or TestPlan)] - [Status] environment:[environment]"	and "Requirement Status"
		Only the Status is mandatory; if only the status is provided, Xray will assume you are searching for the	= "v1.0 - OK" -
		latest execution	issuetype = 'New Feature'
		Xray will lookup for all Requirements with Status in that particular version and environment .	and "Requirement
		Read more about Status and environments.	Status" = "v1.0 - OK environment:Android"
		Requirement Coverage	
		For more information, please check out our page dedicated to requirements coverage.	
		If the Requirements Coverage Strategy depends on the version, then you must also include	
		the actual version name and the status when you do the search. The syntax: "[version name] - [status]"	
Steps	Test	The number of Steps in a Manual Test	issuetype = 'Test'
Count			and "Steps Count" = 3
Manual	Test	Find Tests by text present in the Manual Test Steps fields	issuetype = 'Test'
Test Steps			and "Manual Test Steps" ~ "Login with user administrator"

The Test Set Status and Test Plan Status custom fields, mentioned in Custom Fields, are not queryable.