

Project Name: **${Project.Name}**

Issue: **@{title=${Key}|href=${BaseURL}/browse/${Key}}**

Prepared By: ${fullname:Reporter}

 **CONFIDENTIAL**

!{http://api.qrserver.com/v1/create-qr-code/?color=000000&bgcolor=FFFFFF&data=${BaseURL}/browse/${Key}&qzone=1&margin=0&size=100x100&ecc=L}

**TEST PLAN**

${Summary}

**TABLE OF CONTENTS**

1. Introduction 3

1.1. Document Overview 3

1.2. Test Plan ${Key} Details 3

1.3. Overall Execution Status 3

1.4. Tests Summary 4

1.5. Test Runs Summary 7

1.6. Test Executions Summary 10

Introduction

Document Overview

This Test Report provides a summary of a test plan. This document has been generated automatically from the test management platform.

Test Plan ${Key} Details

|  |  |  |  |
| --- | --- | --- | --- |
| Begin Date | ${dateformat("dd-MM-yyyy HH:mm:ss"):Begin Date} | End Date | ${dateformat("dd-MM-yyyy HH:mm:ss"):End Date} |
| Summary | ${Summary} | Description | ${wiki:Description} |

Overall Execution Status



Of the ${TestsCount} Tests contained on ${Key}:

|  |  |
| --- | --- |
| **TODO** | **${Overall Execution Status.TODO} %** |
| **EXECUTING** | **${Overall Execution Status.EXECUTING} %** |
| **PASS** | **${Overall Execution Status.PASS} %** |
| **FAIL** | **${Overall Execution Status.FAIL} %** |
| **ABORTED** | **${Overall Execution Status.ABORTED} %** |

Tests Summary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Summary** | **Issue Assignee** | **Requirements** | **#TestExecutions** | **Latest Status** |

#{for tests}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| @{title=${Tests[n].Key}|href=${BaseURL}/browse/${Tests[n].Key}**}** | ${wiki:Tests[n].Summary} | #{if (%{'${Tests[n].Assignee}'.equals('Unassigned')})} Unassigned #{end} #{if (%{!'${Tests[n].Assignee}'.equals('')})} ${fullname:Tests[n].Assignee} #{end} | #{for a=JQLIssuesCount|clause=key in testRequirements('${Tests[n].Key}')}  ${JQLIssues[a].Key}  #{end} | ${jqlcount:issue in testTestExecutions("${Tests[n].Key}") and "Test Plan" = ${Key}} | #{if (%{'${Tests[n].Latest Status}'.equals('FAIL')})}  **${Tests[n].Latest Status}**  #{end}  #{if (%{'${Tests[n].Latest Status}'.equals('PASS')})}  **${Tests[n].Latest Status}**  #{end}  #{if (%{'${Tests[n].Latest Status}'.equals('TODO')})}  **${Tests[n].Latest Status}**  #{end}  #{if (%{'${Tests[n].Latest Status}'.equals('EXECUTING')})}  **${Tests[n].Latest Status}**  #{end}  #{if (%{'${Tests[n].Latest Status}'.equals('ABORTED')})}  **${Tests[n].Latest Status}**  #{end} |

#{if (%{'${jqlcount:issue in testTestExecutions(${Tests[n].Key}) AND "Test Plan" = ${Key}}' > 0})}

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Summary** | **Environment** | **Status** |

#{end}

#{for c=JQLIssuesCount|clause=issue in testTestExecutions(${Tests[n].Key}) AND "Test Plan" = ${Key} order by Key desc}

|  |  |  |  |
| --- | --- | --- | --- |
| @{title=${JQLIssues[c].Key}|href=${BaseURL}/browse/${JQLIssues[c].Key}**}** | ${wiki:JQLIssues[c].Summary} | #{if (%{'${JQLIssues[c].Test Environments}'.equals('')})}  -  #{end}  #{if (%{!'${JQLIssues[c].Test Environments}'.equals('')})}  %{'${JQLIssues[c].Test Environments}'.replace("[", "").replace("]", "")}  #{end} | #{for b=TestExecutionsCount}  #{if (%{‘${TestExecutions[b].Key}’.equals(’${JQLIssues[c].Key}’)})}  #{for a=TestExecutions[b].TestRunsCount}  #{if (%{‘${TestExecutions[b].TestRuns[a].Key}’.equals(’${Tests[n].Key}’)})}  #{if (%{'${TestExecutions[b].TestRuns[a].Execution Status}'.equals('FAIL')})}  **${TestExecutions[b].TestRuns[a].Execution Status}**  #{end}  #{if (%{'${TestExecutions[b].TestRuns[a].Execution Status}'.equals('PASS')})}  **${TestExecutions[b].TestRuns[a].Execution Status}**  #{end}  #{if (%{'${TestExecutions[b].TestRuns[a].Execution Status}'.equals('TODO')})}  **${TestExecutions[b].TestRuns[a].Execution Status}**  #{end}  #{if (%{'${TestExecutions[b].TestRuns[a].Execution Status}'.equals('EXECUTING')})}  **${TestExecutions[b].TestRuns[a].Execution Status}**  #{end}  #{if (%{'${TestExecutions[b].TestRuns[a].Execution Status}'.equals('ABORTED')})}  **${TestExecutions[b].TestRuns[a].Execution Status}**  #{end}  #{end}  #{end}  #{end}  #{end} |

#{end}

#{end}

Test Runs Summary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Summary** | **Test Type** | **#Req** | **#Def** | **Test Sets** | **Assignee** | **Status** |

#{for testExecutions}

#{for a=TestExecutions[n].TestRunsCount}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| @{title=${TestExecutions[n].TestRuns[a].Key}|href=${BaseURL}/browse/${TestExecutions[n].TestRuns[a].Key}**}** | ${wiki:TestExecutions[n].TestRuns[a].Summary} | ${TestExecutions[n].TestRuns[a].Test Type} | ${TestExecutions[n].TestRuns[a].RequirementsCount} | %{var total=${TestExecutions[n].TestRuns[a].ExecutionDefectsCount} + ${TestExecutions[n].TestRuns[a].TestStepsDefectsCount}; var total2=’’+total;var totalParts= total2.split(‘.’); totalParts[0];} | #{for ts=TestExecutions[n].TestRuns[a].TestSetsCount}  @{title=${TestExecutions[n].TestRuns[a].TestSets[ts].Key}|href=${BaseURL}/browse/${TestExecutions[n].TestRuns[a].TestSets[ts].Key}}  #{end} | #{if (%{'${TestExecutions[n].TestRuns[a].Assignee}'.equals('Unassigned')})}  *Unassigned*  #{end}  #{if (%{!'${TestExecutions[n].TestRuns[a].Assignee}'.equals('')})}  ${fullname:TestExecutions[n].TestRuns[a].Assignee}  #{end} | **#{if (%{'${TestExecutions[n].TestRuns[a].Execution Status}'.equals('FAIL')})}**  **${TestExecutions[n].TestRuns[a].Execution Status}**  **#{end}**  **#{if (%{'${TestExecutions[n].TestRuns[a].Execution Status}'.equals('PASS')})}**  **${TestExecutions[n].TestRuns[a].Execution Status}**  **#{end}**  **#{if (%{'${TestExecutions[n].TestRuns[a].Execution Status}'.equals('TODO')})}**  **${TestExecutions[n].TestRuns[a].Execution Status}**  **#{end}**  **#{if (%{'${TestExecutions[n].TestRuns[a].Execution Status}'.equals('EXECUTING')})}**  **${TestExecutions[n].TestRuns[a].Execution Status}**  **#{end}**  **#{if (%{'${TestExecutions[n].TestRuns[a].Execution Status}'.equals('ABORTED')})}**  **${TestExecutions[n].TestRuns[a].Execution Status}**  **#{end}** |

#{end}

#{end}

Test Executions Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key** | **Summary** | **#Tests** | **Assignee** | **Status** |

#{for testExecutions}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| @{title=${TestExecutions[n].Key}|href=${BaseURL}/browse/${TestExecutions[n].Key}} | ${wiki:TestExecutions[n].Summary} | ${jqlcount:issue in testExecutionTests("${TestExecutions[n].Key}")} | #{if (%{'${TestExecutions[n].Assignee}'.equals('Unassigned')})}  *Unassigned*  #{end}  #{if (%{!'${TestExecutions[n].Assignee}'.equals('')})}  ${fullname:TestExecutions[n].Assignee}  #{end} | TODO: **${TestExecutions[n].Overall Execution Status.TODO} %**  EXECUTING: **${TestExecutions[n].Overall Execution Status.EXECUTING} %**  PASS: **${TestExecutions[n].Overall Execution Status.PASS} %**  FAIL: **${TestExecutions[n].Overall Execution Status.FAIL} %**  ABORTED: **${TestExecutions[n].Overall Execution Status.ABORTED} %** |

#{end}

Appendix A: Approval

The undersigned acknowledge they have reviewed the **Test Plan** and agree with the approach it presents. Changes to this **Test Plan** will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: |  |  |  |
| Title: |  |  |  |
| Role: |  |  |  |