

Training Guide for Administrators



Table Of Content:

- [Projects](#)
- [Adding users](#)
- [PractiTest terminology](#)
- [Bug tracker integration](#)
- [Categorizing tests](#)
- [Importing/Creating tests](#)
- [Organizing tests by categories](#)
- [Grouping tests into sets and organizing test sets](#)
- [Requirements and User Stories management](#)
- [Running your first cycle](#)
- [Issues management](#)
- [Dashboards setup](#)
- [Reports](#)
- [Starting a new cycle](#)
- [Automation integration](#)
- [Additional resources](#)

Setting up PractiTest projects

PractiTest projects are independent entities. By default, your account is set with one project. Using fields and filters, you can organize your data in a convenient way to display different types of product components, different versions. projects.

Reports and dashboards are on a project level.

There are a number of reasons why you may want to have more than one project in PractiTest:

- When you have multiple independent teams working in separate projects.
- When your company has multiple development projects with different external customers and you don't want them to see each other's information.
- You are testing 2 or more completely separate products that have no tests and issues overlapping whatsoever.

Adding New Projects:

You can add new projects from the Project Management tab of the Account Settings. When adding a new project, you will have 2 options:

Clone from project

If you choose to clone from an existing project, you will have the option to choose from which project you would like to clone and what configuration settings of this project you want in your new project.

You can independently choose to copy your fields (custom and system), your user & permissions groups and workflow, and/or your users and their groups' assignments.

Choose from template

When selecting the "choose from template" option, you will have the option to choose the method according to which you want your project structure to be.

The following options are available:

Traditional - supports Waterfall methodology or similar

Traditional + Automation - same as Traditional with a few additions to support automation

Agile - supports Agile, DevOps and Agile-like methodologies.

Agile + Automation - same as Agile with a few additions to support automation.

Start from scratch - clean and empty project.

Demo data - you can also decide to add demo data to your project, and thus see an example setup for a project that includes data. To add demo data, tick the 'Add demo data to my project' checkbox at the bottom of the page.

Choose your project template

 <p>Traditional Waterfall, V shape and similar</p>	 <p>Agile Agile, Agile like, DevOps</p>
 <p>Traditional + Automation Traditional methodology with automation CI/CD integration</p>	 <p>Agile + Automation Agile methodology with automation CI/CD integration</p>
 <p>Start from scratch Start with a clean project and build it yourself</p>	

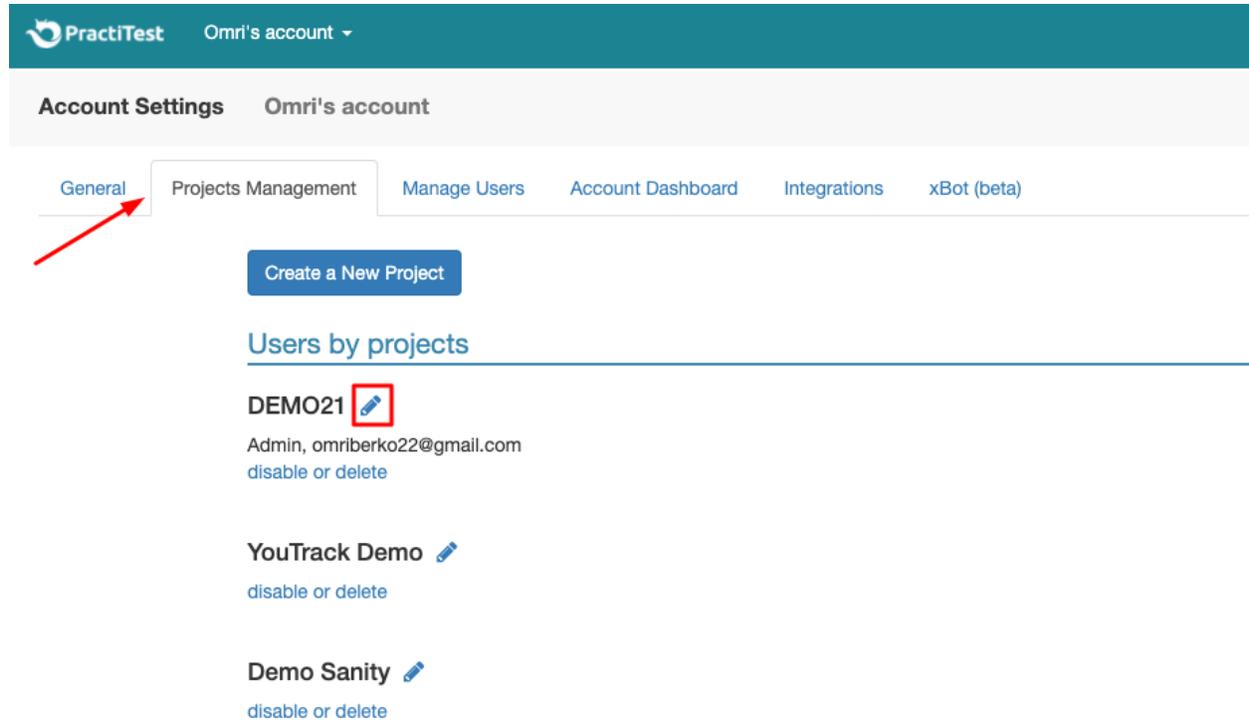
[What are the differences?](#) Add demo data to my project

[< Back](#) [Create my new project](#)

You will be able to create another project with demo data from the project settings option in your account

Renaming your project

Once your new project is created based on the selected template, make sure you edit the default name assigned to your project to reflect the project goals, and avoid unnecessary confusion in the future. You can edit project names from the project management tab of account settings, by clicking the pencil icon next to the project name.



PractiTest Omri's account ▾

Account Settings Omri's account

General Projects Management Manage Users Account Dashboard Integrations xBot (beta)

Create a New Project

Users by projects

DEMO21 
Admin, omriberko22@gmail.com
[disable](#) or [delete](#)

YouTrack Demo 
[disable](#) or [delete](#)

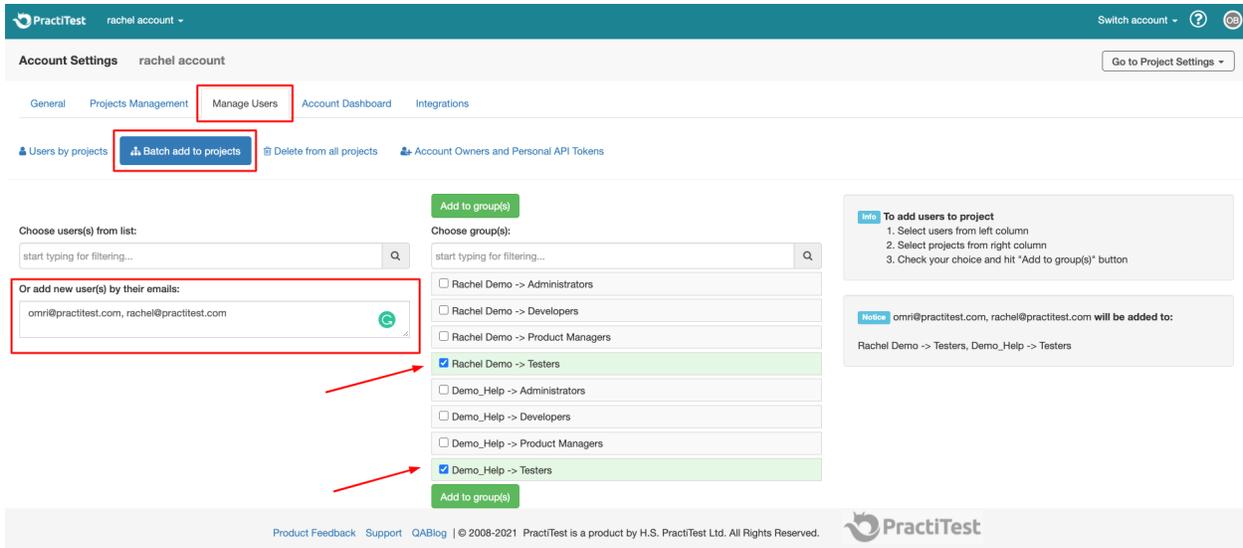
Demo Sanity 
[disable](#) or [delete](#)

Adding Users

There are two different methods to add new users.

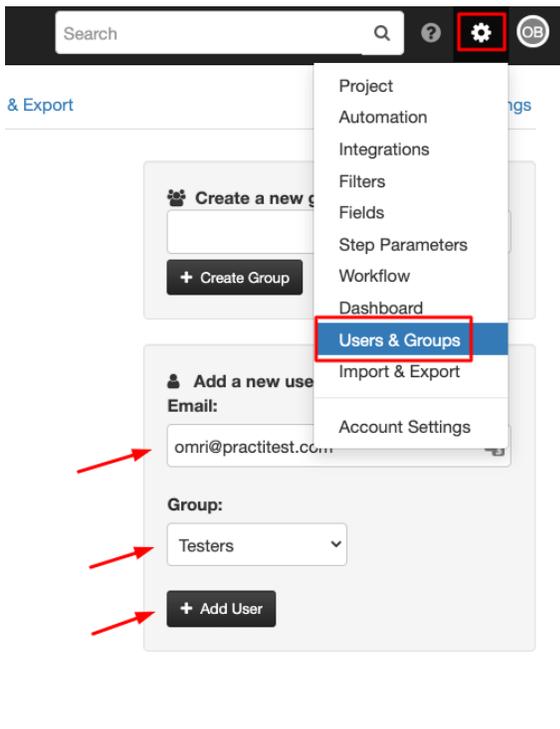
Account Settings

The first method allows you to batch add users to any project in your account. Access the account settings - manage users tab - select the 'Batch add to projects' option. Paste the email addresses of the users you want to add in the 'add new user(s) by their email' field, then select the permission group you want to add the users to, next to the project name. This method is only available for account owners. See example below:



Users & Groups

The second method focuses on the project level, and allows you to add users to a specific project. Navigate to Users & Groups from settings. In the 'Add a new user' box, paste the email of the user, then select the permission group you want to add the user to. This option is available for project administrators as well as account owners.



Permission groups

By default, you will have three permissions groups in your project: Administrators, Testers and Developers.

These default permissions can't be edited, but you can create custom permission groups, and assign users to them, thus granting them or removing permissions that are not relevant to the group of users assigned to them.

Note: from a billing perspective, only users added to the 'Developers' group will be billed as Developers, and the rest will be billed as testers.

PractiTest Terminology

This section explains the differences between Tests, Test Sets, Instances, and Runs. These terms are crucial to understand how the system works, and to avoid unnecessary confusions when working with the different features within the system.

Tests in the Test Library

The test library is the master repository of your tests (or test cases), where you keep and manage the master versions of your tests. In the test library, you write your tests or import existing tests from spreadsheets. When you edit a test in the test library, all related instances will be affected. Linking tests to requirements is also done here using the traceability tab of your tests.

Test Sets and Test Instances

A Test Set (sometimes called test suite) is a group of tests that you want to run together. When you add a test to a test set, the system creates an instance of this test within the test set. An Instance is a dynamic copy of a test from the test library and it allows you to run the test as many times as you need to as part of multiple test sets.

When you run your instance, the test (including all the steps it contains) is copied to a new Test Run. This ensures that you are running the latest version of the test from the test library. You can have multiple instances of the same test in a different, or a single test set. In addition, you can run instances as many times as you like.

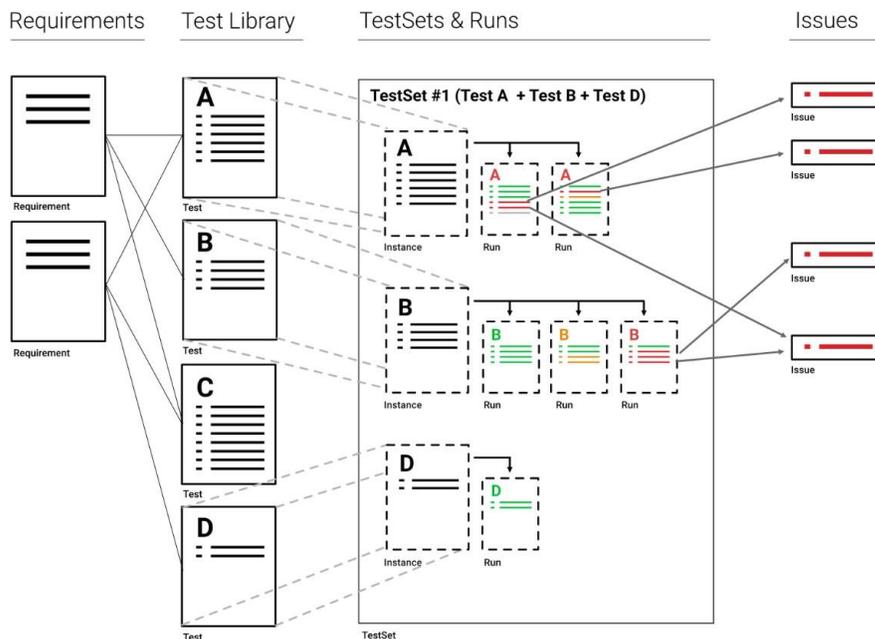
Test Runs

Each time you run an instance, you create a new run under it. You can see all the runs related to an instance by clicking on the instance name from a test set.

If you edit a step during a test run, the original test in the test library will not be updated unless you click 'update original test'. You can access the original test from the top right-hand corner of an instance.

Keep in mind that when you are creating reports and dashboard items to showcase test execution data, you should focus on the Instance entity or run entity (mostly for test automation).

The below diagram shows the correlation between the different terms:



Setting up an integration with Bug Trackers

If you are working with an independent bug tracker, start by configuring the integration between PractiTest and the bug tracker you are using.

You can find guides for configuring your integration [here](#).

Tests categorization using fields

Before you start importing or creating your tests in PractiTest, we highly recommend that you take the time to consider how you want to categorize them, and set up your fields and filters based on the categories. Here are a few examples for categories you can use in your project:

Module, Component, Feature, Test Level (sanity, regression, etc..).

Once you have the structure in mind, start by setting up fields based on your categories.

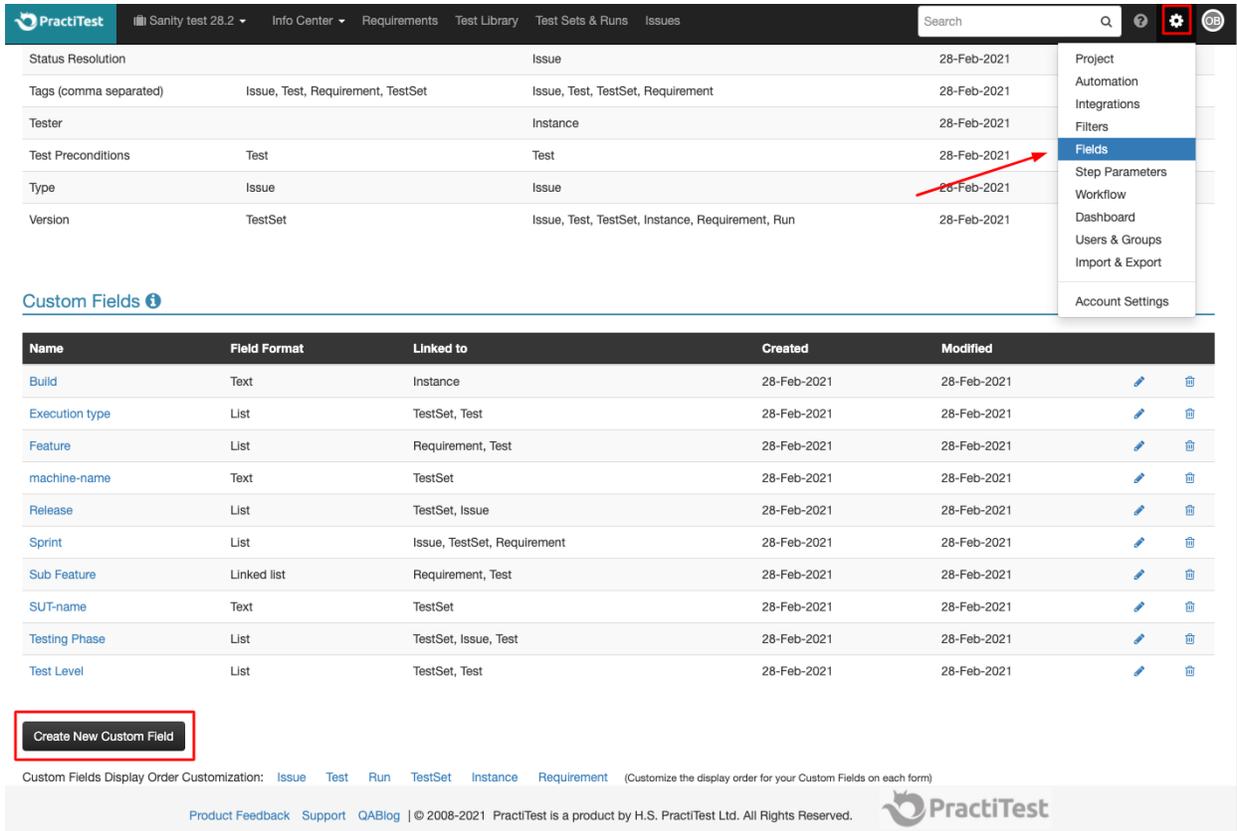
System Fields

System fields are fields that will be available in every PractiTest project by default. You can edit your system fields, and decide where you want to display them from the fields section of settings.

Depending on the project template you selected, a few custom fields will be created for your project by default. Please note that you can edit and remove those fields as well, from the custom fields section of the fields page.

Custom Fields setup

To create a new custom field, navigate to settings - fields. Scroll down the page, and click 'Create new custom field'. We highly recommend using list type custom fields in most cases. Using list fields, you can pre-define a set of values that can be selected to populate the field.



The screenshot shows the PractiTest interface with the 'Custom Fields' configuration page. A table lists existing custom fields with columns for Name, Field Format, Linked to, Created, and Modified. A red box highlights the 'Create New Custom Field' button. A red arrow points to the 'Fields' option in the settings menu.

Name	Field Format	Linked to	Created	Modified
Build	Text	Instance	28-Feb-2021	28-Feb-2021
Execution type	List	TestSet, Test	28-Feb-2021	28-Feb-2021
Feature	List	Requirement, Test	28-Feb-2021	28-Feb-2021
machine-name	Text	TestSet	28-Feb-2021	28-Feb-2021
Release	List	TestSet, Issue	28-Feb-2021	28-Feb-2021
Sprint	List	Issue, TestSet, Requirement	28-Feb-2021	28-Feb-2021
Sub Feature	Linked list	Requirement, Test	28-Feb-2021	28-Feb-2021
SUT-name	Text	TestSet	28-Feb-2021	28-Feb-2021
Testing Phase	List	TestSet, Issue, Test	28-Feb-2021	28-Feb-2021
Test Level	List	TestSet, Test	28-Feb-2021	28-Feb-2021

After adding the values, make sure you link the field to the 'Test' entity, which applies for tests in the Test Library. After saving the field, you will see it in the screen of your tests, and could populate it for every new test you create or import.

If your categorization structure requires sub-lists that are dependent on a parent list, you can use the linked list field type. When creating a linked list field, you select a parent list field already created in the system, and define a set of possible values for every value of the parent field.

For example, my setup requires categorization based on components. But, my tests can relate to different sub components within the main component. To set this categorization up, I will first create a list type field for 'Component'. Then, I will create a linked list field for 'Sub-Component' making the 'Component' field the 'parent' field.

You can find a detailed step-by-step guide for setting up your custom fields [here](#).

Importing/ Creating Tests

Importing Tests

When you are done with the initial setup for your fields, you can start importing your tests from Excel/spreadsheet. To access the import page, click on Settings - Import & Export. Then click on

'Import tests'. In the tests import screen, you will see all the fields you created for your project. Instead of populating them for each test individually, you can populate the fields directly from the import process. In the import page, you need to map the columns of your spreadsheet, to the fields shown on the page.

For example, if in my spreadsheet test names are listed in column 'C', I will delete the 'A' letter that is filled in there by default, and will replace it with a 'C'.

Please note: if your spreadsheet contains a row of headers, you will need to tick the 'ignore first row' box.

You can find the full import guide [here](#).

Creating Tests

There are three types of tests you can create in PractiTest - Scripted, Exploratory and Automated.

Scripted Tests

Scripted tests contain predefined steps for the tester to verify.

The general tab of the test contains the test name and description, all system and custom fields related to the test entity, and a comments section.

In the steps tab, the steps information is defined.

In the traceability tab, linkage to requirements and issues is defined and shown.

In the history tab, all historical actions related to the test are listed.

Exploratory Tests

Exploratory tests contain a charter, mission, for the exploratory session, and guide points for the executing tester. During the session, the tester can add annotation on the fly based on their findings.

The general tab of the test contains the test name, charter and guide points, all system and custom fields related to the test entity, and a comments section.

The traceability and history tabs are identical to scripted tests.

Automation Tests

There are a few different types of automated tests you can create, based on the methods you use for test automation. This guide will cover the different types in a later chapter.

You can find the complete test library guide [here](#).

Organizing Tests by categories using filters

Working with filters allows you to have a flexible structure for your tests. The concept is simple - you define a criteria for a filter. Once a test meets the criteria, it will appear under it. This allows you to have the same tests in several different filter trees, under different categories. For example, one of my tests relates to the 'Login' feature. It is also a sanity test. By assigning the 'Login' value for the 'Feature' field, and the 'Sanity' value for the 'Test Level' field, I can make sure the test will appear under the 'Feature' filter tree AND under the 'Test Level' filter tree.

To create a new filter - navigate to the test library and click '+ New Filter'. Add a name for your filter, then add criteria in the filter criteria section. Finally, select which fields you want to add to the view of your filter from the custom fields section of this page.

Auto Filters - The auto filter feature allows you to create filter trees automatically based on any of your list type fields. To enable this option, tick the 'Auto filter' checkbox in the filter creation page. Then, select a list type field in the filter criteria section. Your filter will be named automatically based on the selected field.

You can find a detailed step-by-step guide for setting up your filters [here](#).

Organizing Test Sets

Test Sets allow you to organize your tests in the same way that you organize your work into cycles or tasks.

A Test Set is a group of tests that belong to specific tasks or features, or have some other reason to be run together.

Good candidates for Test Sets can be, for example:

- Tests that focus on a certain part of the system, such as GUI or Database
- Tests that belong to a certain task, such as Regression or Sanity
- A set of tests that need to be run by a single tester during a day or calendar week.

Test sets are created in the test sets & runs module. When creating a new test set, you will first need to name it. Then, after the test set is created you can add test instances to it.

Test Set fields and filters

In the Test Sets module, we recommend using time based items to organize your sets. For example - sprints, releases, cycles, and versions. You can also use different categories such as features and test levels (regression, sanity, etc). This, of course, depends on the methodology you use and your overall process.

As mentioned in the Tests section, we highly recommend using list type fields in most cases.

You can find the full Test Sets & Runs guide [here](#).

Requirements and User stories management

The requirements module is where you create and manage your system requirements or user stories. You can either create and define your requirements/user stories directly in PractiTest, or import them from an Excel spreadsheet. The import process for requirements is similar to the test import process. Please note, that when you import requirements, you can import the linkage to tests as well using the traceability field. To access requirements import, navigate to settings - import & export. Then select 'Import Requirements'.

Creating requirements in PractiTest

When creating a new requirement in the system, you will first need to fill in the information of the requirement in the Name and Description fields. After you filled in the requirements information, you can proceed by linking it with tests to cover the requirement, from the traceability tab.

Organizing your requirements

Like every other module in PractiTest, the most recommended method to organize your requirements is first using fields to categorize your requirements, then create filter views to sort your requirements based on the categories.

Requirements status field

The requirements status field will be populated automatically based on the status of the tests you link to it.

Below you can find the criteria for each status:

NOT COVERED – There are no tests linked to this Requirement.

NO RUN – There is at least one test linked to the requirement. None of the linked tests ran.

STARTED – There is at least one test linked to the requirement that started running. None of them Failed or was set to Blocked

PASSED – All linked tests ran and passed.

BLOCKED – At least one of the linked tests was set to blocked. None of the linked Tests was Failed.

FAILED – At least one of the linked tests failed.

N/A – All tests linked to this requirement marked as N/A

You can find the full Requirements guide [here](#).

Running your first cycle

Once you are done with the initial setup, you are ready to go live and start running your first test cycle. To make sure your test cycle is documented correctly, verify that the test sets executed for this cycle are populated with the relevant cycle value for the cycle field. If you use 'Release', or 'Sprint' instead, make sure those fields are populated properly.

Assigning individual instances to users

If your workflow requires a few testers to execute a single test set, you can also assign individual instances to different testers. To do that, navigate to settings - fields, edit the 'Assigned to' field, and activate it for 'Instances'.

Running instances and reporting issues

To run an entire test set, click the 'Run Now' button from the test set screen. This will initiate test execution for all test instances residing in the test set in the order they were added. You can also run an individual instance from the test set, by clicking on 'Run' next to it.

In both cases, a new run will be created for your instance/s. In the run screen, when running a scripted test, you will see all steps information including the name, description, expected results and a field for the actual result. You can assign each individual step with a status. If the testers find an issue during the test run, they can report the issue directly from the test run using the Fail & Issue option. Or, if the issue already exists in the system, use the 'Link existing issue' option to link the issue to the step.

The Tester field

The tester field is a system field which is deactivated by default. By enabling it, you will be able to see which tester ran each one of the instances from the test instances grid. To enable the tester field, navigate to settings - fields. Edit the 'Tester' field from the system fields section, and activate it for instances. Once enabled, you can add the field as a column to your test instances grid using the 'Columns' option.

See the full test sets and runs guide [here](#).

Issues management

PractiTest Issues Module

Note: If you are using an external bug tracker integration, this section is not relevant for you.

Issues that were reported from test runs will be added directly to the PractiTest issues module. To create an organizational structure for your Issues modules, we recommend using fields and filters.

Issues Workflow

The workflow editor, which you can access from settings, allows you to customize the Life-Cycle of your issues to match your needs. You can add statuses to issues, and also decide which statuses an issue can be changed to, based on its current status.

Defect Age

The defect age field will be added to every issue created in your project. This field allows you to keep track of the time (in days) that the issue was opened. You can also generate a dashboard graph to reflect the data from this field.

Traceability

In the traceability tab of your issues, you can see the tests and requirements that the issue is linked to.

Deleting Issues

Issues deletion is disabled by default. To enable issue deletion for a project, navigate to the account settings - project management tab. Then, edit the relevant project and tick the 'Enable issue deletion' box. Only account owners have the permission to enable the deletion.

See the full issues module guide [here](#).

Bug tracker integration

If you are using a bug tracker integration, when you report issues from test runs the issues will be reported to the integrated bug tracker directly. If you are using a two-way integration, a copy of the issue will also be created in PractiTest, and the name, description and status fields will be synced between the PractiTest issue to the bugtracer's issue.

Dashboards

PractiTest dashboards are fully customizable, and we highly recommend adding new dashboard items and tabs and customizing the default dashboard to tailor it for your needs and your specific projects. Every dashboard tab can contain up to 8 items, and you can create an unlimited number of tabs for each PractiTest project.

See the full guide for dashboards [here](#).

Dashboard items

When creating a new dashboard item, first, you need to select the entity you want to focus on: Tests, Test Sets, Instances, Runs, Issues or Requirements. To focus on test execution data, make sure you select the Instances entity or Runs entity if you want to show test automation data. After you selected the focused entity, select the item type you would like to create, and press on continue. Next, select the filter you want to focus on from the selected entity. For example, if I want to create an item that will display test execution data from my last cycle (cycle 3) - I will create an instance based item and the 'Cycle 3' filter.

By ticking the box next to 'After saving stay on this page to preview the Dashboard Item', you can preview your item in this same page, and make changes to it until you are satisfied with the result.

Test, Test Set, and LastRun field data in Instances based items

When creating items based on instances, you can select to show data from fields linked to tests, test sets, and to run in your dashboard item. In the drop down list of fields, fields starting with an entity name, followed by a colon, are fields from one of the different levels. Fields with no entity name at the beginning, are fields from the instance level.

For example, if I want to show information from a 'Feature' field linked to my tests in the test library - I will select the field 'Tests: Feature' as my X field.

See the full dashboard items guide [here](#).

External dashboards

The external dashboard feature allows you to share your dashboard tabs as an external URL, with your stakeholders and non-PractiTest users.

See more information about external dashboards [here](#).

Reports

While dashboards provide you with a high level view of your project, you can utilize reports to get a deep level look into your data. There are two main types of reports available for all modules, and a few additional types depending on the module you want to create the report for. Like with dashboards, test execution reports should be created based on the instance entity, and run reports should be created based on test automation execution.

When creating a new report, make sure you select a filter from the relevant module to display information for a specific slice of your information. For example, to create a report based on information from 'Cycle 3', I will create an instance report with the 'Cycle 3' filter selected.

Tabular summary reports - Excel based reports. Using tabular summary reports, you can fully customize the data you display in the report, and can decide which fields you want to display, and which charts you want to display. Tabular with steps reports are identical to tabular summary reports, but also includes steps data, and issues data.

Detailed reports - PDF based reports that provide detailed information about the selected entity and scope.

Scheduling reports

When creating a report, you can decide to schedule it to run on a daily, weekly or monthly basis. The report will be regenerated with the settings you define for it based on the selected interval, and will be sent by email to users you select from the reports settings.

See more information about the reports module [here](#), and report samples [here](#).

Starting a new Test Cycle

Cloning Test Sets and Test Set Permutations

When you are ready to proceed to the next test cycle, you can reuse the test sets you used over the previous cycle by cloning the test sets, and assign them with the new value for the cycle field. There are two different options for cloning test sets. First, you can use the 'Clone Test Sets' option. Second, is the 'Permutations' option. Permutations allow you to create multiple clones for selected test sets, and assign them to different environments.

See the full guide for Test Set Permutations [here](#).

Cloning dashboard tabs

You can also clone the dashboard tab you created for your previous cycle, and simply edit the filters that the dashboard items are focused on to the new cycle's filters.

Test Automation Integration

With PractiTest, you can run, control, view and manage your automation testing alongside your manual testing. Thus, have complete visibility for your entire process in one place. Below you can find the details about the different methods available for integrating your automated testing

into PractiTest.

API

We have a fully featured API that can help you in integrating Practitest with automated testing frameworks. Our API allows you to fetch details of test sets from the system and then push back the results of your automated runs into Practitest. Our API documentation is [here](#).

You can find a step-by-step guide of how to use our API to integrate your automation testing [here](#).

FireCracker

Firecracker is a PractiTest developed tool that allows you to integrate any CI/CD framework with your PractiTest project. FireCracker allows you to take XML report files from any CI/CD tool and upload them into PractiTest easily and automatically. See the step-by-step guide for using FireCracker [here](#).

xBot

xBot is a PractiTest developed tool that allows users to run (or initiate a run by scheduling) automated test scripts on a remote machine from the PractiTest UI. See the step-by-step guide for using xBot [here](#).

If you would like to discuss how to implement Automation, please let us know and we will be happy to assist with a personal meeting.

Additional Resources

Call to test

When creating new tests or editing existing tests, you can use the call to test feature to add test steps from other existing tests in the test library to your test. This feature can save time by reusing already well-defined test steps.

See full guide about call to test [here](#).

Task Board (Kanban)

PractiTest's task board allows you to define entities in a way that will help you prioritize, share progress, and have a cross modules view of your entities. Using the task board, you can assign entities to lanes, and thus manage them in a board as tasks.

See full guide about task board [here](#).

Time management

The time management feature allows you to set an approximate time for test executions, and measure the actual time it takes to execute your tests. This will allow you to better manage the project timelines team productivity.

See full guide about time management [here](#).

Personal settings

You can customise your personal account from the personal settings section. From this section, you can change your password, edit the default dashboard settings and email notifications. To access the personal settings, click the person icon on the right of the screen in the Navigation Bar and select Personal Settings.