Contents

[Introduction 2](#_Toc525826740)

[Document overview 2](#_Toc525826741)

[Abbreviations 2](#_Toc525826742)

[Glossary 2](#_Toc525826743)

[Document references 2](#_Toc525826744)

[The results 3](#_Toc525826745)

[Test case results notation 3](#_Toc525826746)

[Scenario summary notation 3](#_Toc525826747)

[Test items 4](#_Toc525826748)

[Scenarios to be tested 4](#_Toc525826749)

[Features to be tested 4](#_Toc525826750)

[dfwnet 4](#_Toc525826751)

[Features not to be tested 6](#_Toc525826752)

[Environmental needs 7](#_Toc525826753)

[Linux environment 7](#_Toc525826754)

[Windows environment 7](#_Toc525826755)

[Responsibilities 7](#_Toc525826756)

[Staffing and training needs 8](#_Toc525826757)

[Approvals 8](#_Toc525826758)

# Introduction

## Document overview

This document is the software test plan report of the devon4net testing phase of the devonfw software platform project. It contains the final results of tests, which were executed during the testing phase according to IEEE 829 standard.

## Abbreviations

## Glossary

|  |  |
| --- | --- |
| Term | Description |
| Nuget | NuGet is a free and open-source package repository designed for the Microsoft development platform. |
| API | Application Programming Interface is a set of definitions, communication protocols, and tools for building software. It is used to share information between client applications (frontend) and backend. |
| devonfw | An Open Source solution to help and improve the software life cycle. |
| devon4net | The .NET technology stack for the devonfw platform. |

## Document references

For this release no external documentation references has not been added.

# The results

## Test case results notation

After executing a test, the decision is defined according to the following rules:

|  |  |
| --- | --- |
| Result | Description |
| **OK** | The test sheet is set to "OK" state when all steps are in "OK" state. The real result is compliant to the expected result. |
| **Fail** | The test sheet is set to "Fail" state when all steps of the test are set to "Fail" state or when the result of a step differs from the expected result. |
| Not Run | Default state of a test sheet not yet executed. |
| **Not Completed** | The test sheet is set to "Not Completed" state when at least one step of the test is set "Not Run" state. |

Also, the test priority is defined according to:

|  |  |
| --- | --- |
| Priority | Description |
| Low | All the suggestions, and small changes or product improvements will be included. They will not affect the software in anyway and can be avoided if there is tight deadline. |
| Normal | All the suggestions, and small changes or product improvements will be included. Cannot be avoided and must be part of the release. |
| High | This test cases are related to all the major functionalities performed by the software keeping in mind how end users are going to use it. These functionalities are very important to the end users and if these fail then customer will trash the software so they also need to be fixed ASAP. |
| Critical | If any of the feature stops working, then this will block the use of the software or feature. |

## Scenario summary notation

For each scenario, the result will be defined according to the following rules:

|  |  |
| --- | --- |
| Result | Description |
| **OK** | The scenario is set to "OK" state when all test cases are in "OK" state. |
| **Fail** | The scenario is set to "Fail" state when all test cases of the scenario are set to "Fail" state or when the result of a test case differs from the expected result. |
| **Not Completed** | The scenario is set to "Not Completed" state when at least one step of the test case is set "Not Run" state or “Not Completed” state |

# Test items

## Scenarios to be tested

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Scenario ID** | **Description** | **Tests Passed** | **Tests Fail** | **Comments** | **Date** | **State** |
| dfwnet | .NET stack for the devonfw platform | 17 | 0 | The test cases cover from download and install to be ready for use on Linux and Windows systems. | 2018-09-26 | **OK** |

## Features to be tested

### dfwnet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Remarks** | **External Links** | **Result** |
| dfwnet-tc-000 | Main API template from NuGet is downloaded and installed on machine context |  | dfwnet-tc-000.docx | **OK** |
|
| dfwnet -tc-001 | Each component has a NuGet package with correct version |  | dfwnet -tc-001.docx | **OK** |
|
| dfwnet -tc-002 | Each component has been uploaded to NuGet with correct version |  | dfwnet -tc-002.docx | **OK** |
|
| dfwnet -tc-003 | The API template downloaded from GitHub compiles and works |  | dfwnet -tc-003.docx | **OK** |
|
| dfwnet -tc-004 | Each component downloaded from GitHub compiles and works |  | dfwnet -tc-004.docx | **OK** |
|
| dfwnet -tc-005 | Each sample downloaded from GitHub compiles and works |  | dfwnet -tc-005.docx | **OK** |
|
| dfwnet -tc-006 | The API template downloaded from GitHub compiles and works on Linux |  | dfwnet -tc-006.docx | **OK** |
|
| dfwnet -tc-007 | Each component downloaded from GitHub compiles and works on Linux |  | dfwnet -tc-007.docx | **OK** |
|
| dfwnet -tc-008 | Each sample downloaded from GitHub compiles and works on Linux |  | dfwnet -tc-008.docx | **OK** |
|
| dfwnet -tc-009 | Swagger is auto generated when a new project is created from template |  | dfwnet -tc-009.docx | **OK** |
|
| dfwnet -tc-010 | The API Template integrates with Graylog |  | dfwnet -tc-010.docx | **OK** |
|
| dfwnet -tc-011 | The AOP trace can be set on/off |  | dfwnet -tc-011.docx | **OK** |
|
| dfwnet -tc-012 | The API Template compiles on Visual Studio Code on Linux |  | dfwnet -tc-012.docx | **OK** |
|
| dfwnet -tc-013 | The API Template compiles on Visual Studio Code on Windows |  | dfwnet -tc-013.docx | **OK** |
|
| dfwnet -tc-014 |  |  | dfwnet -tc-014.docx | **OK** |
|
| dfwnet -tc-015 | The API Template compiles on Visual Studio on Windows |  | dfwnet -tc-015.docx | **OK** |
|
| dfwnet -tc-016 | Docker-compose for My Thai Star |  | dfwnet -tc-016.docx | **OK** |
|
| dfwnet -tc-017 | Docker-compose for Graylog |  | dfwnet -tc-017.docx | **OK** |
|

## Features not to be tested

As this is the first release for the devon4net stack for the devonfw platform, there are no features excluded on the test cases.

# Environmental needs

### Linux environment

* Visual studio code with omnisharp plugin
* Dotnet core 2.1+ SDK
* Internet connection
* Docker service up and running
* User with admin grants to run docker commands

### Windows environment

* Visual studio code with omnisharp plugin
* Visual Studio Community|Professional|Enterprise 2017
* Dotnet core 2.1+ SDK
* Internet connection
* Docker service up and running

# Responsibilities

The testing for the project will consist of Unit, System/Integration (combined) and Acceptance test levels.

*UNIT Testing* will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test person.

*SYSTEM/INTEGRATION* Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. No specific test tools are set for this project.

*ACCEPTANCE* Testing will be performed by the actual end users with the assistance of the test manager and development team leader. The acceptance test will be done in parallel with System/Integration test process.

# Staffing and training needs

For this release the team does not need any training to perform the test cases.

# Approvals

|  |  |
| --- | --- |
| Role | Name |
| Product Owner | Maurice Driessen |
| Team Leader | Santos Jiménez |
| Scenario Leader | Santos Jiménez |
| Testers/Developers | José Manuel Sánchez |