Test Plan

New OpenWrt Web interface

Les informations d'identification du document et les éléments de vérification du document			
Référence du document :	T1	**Validé par :**	Baptiste Jonglez
Version du document :	1	**Validé le :**	18/06/2020
Date du document :	10/06/2020	**Soumis le :**	
Auteur(s) :	Biyun WANG	**Type de diffusion :**	Document électronique (.pdf)
Mots clés : Plan de test			

Table of Contents

1. Introduction

This document presents all the tests which will allow validating the good functional behavior of the LuCI interface from the router's user perspective.

1.1. Objectives and methods

The objective is to declare its acceptance criteria and test methods so that the user of the router validates the functionalities.

The router user can switch to a browser and connect to the OpenWrt system on a real router or Docker environment to manipulate tests.

2. Basic concepts

It is preferable to read the User Manual and the Installation manual to prepare the environment of testing.

3. Functional tests

3.1. Scenario 1

Identification

TF_wifi

Description

The objective of this test is to configure a name (SSID) and a password for the wifi network.

Constraints

It must be tested under an OpenWrt system which carries the new interface in a real router or a Docker environment.

Dependencies

A router, which carries the new interface of the OpenWrt system or the Docker environment, must function correctly and configure without problems.

Test procedure

1. The tester or user of the router logs in to the OpenWrt system.
2. Access the page of Quick setup.

3. Input a new SSID and a new password, then save them.
4. Find out if there is a wifi network with the new SSID on your device.
5. Use the new password to connect this wifi network.

3.2. Scenario 2

Identification

TF_administrator

Description

The objective of this test is to configure the administrator password for accessing the router.

Constraints

It must be tested under an OpenWrt system which carries the new interface in a real router or a Docker environment.

Dependencies

A router, which carries the new interface of the OpenWrt system or the Docker environment, must function correctly and configure without problems.

Test procedure

1. The tester or user of the router logs in to the OpenWrt system.
2. Access the page of Quick setup.
3. Input a new password and confirm it, then save it.
4. Log out.
5. Relog in with the new administrator password.

3.3. Scenario 3

Identification

TF_wan

Description

The objective of this test is to change the connection type of internet.

Constraints

It must be tested under an OpenWrt system which carries the new interface in a real router or a Docker environment.

Dependencies

A router, which carries the new interface of the OpenWrt system or the Docker environment, must function correctly and configure without problems.

Test procedure

1. The tester or user of the router logs in to the OpenWrt system.
2. Access the page of Quick setup.
3. Choose the right connection type of internet.
4. Fill all the information asked.
5. Save and apply.
6. Verify the status of internet in the page Network -> interfaces.

3.4. Scenario 4

Identification

TF_dashboard

Description

The objective of this test is to check the network connection status (public IP, DNS, etc.).

Constraints

Users can quickly find the information they are looking for.

Dependencies

1. An Internet connection
2. A router, which carries the new interface of the OpenWrt system or the Docker environment, must function correctly and configure without problems.

Test procedure

1. The tester or router user logs into the OpenWrt system.
2. Find the information on internet network on the dashboard page.

3.5. Scenario 5

Identification

TF_static_leases

Description

The objective of this test is to assign fixed IP addresses and symbolic hostnames to DHCP clients.

Constraints

Users can choose the information in a list.

Dependencies

A device connected with this router.

A router, which carries the new interface of the OpenWrt system or the Docker environment, must function correctly and configure without problems.

Test procedure

1. The tester or router user logs into the OpenWrt system.
2. Access the page of Static IP.
3. Click at the button "add".
4. Indicate the hostname.
5. Choose the right MAC address from the list.
6. Choose the IPv4 address associated with the MAC address.
7. Save and apply.
8. A line describing the status of this lease will be created in case of success.

3.6. Scenario 6

Identification

TF_port_redirects

Description

The objective of this test is to allow remote computers on the internet to connect to a specific computer.

Constraints

Users can choose the information in a list.

Dependencies

A device connected with this router.

A router, which carries the new interface of the OpenWrt system or the Docker environment, must function correctly and configure without problems.

Test procedure

1. The tester or router user logs into the OpenWrt system.
2. Access the page of Port redirects.
3. Click at the button "add".
4. Give a name to this server.
5. Choose the right protocol from the list.
6. Input the external port.
7. Choose the right internal address from the list.
8. Input the internal port.
9. Save and apply.
10. A line describing the status of this server will be created in case of success.

4. Unit testing

During the coding phase, we didn't create any new functions, all the functions we used are already written in LuCI's documentation. Therefore, unit testing is not necessary in our case.

5. Integration testing

Our project is to create a new Web interface of LuCI, we should send a merge request to the developper group of LuCI. By this way, integration testing will be done.

6. Glossary

OpenWrt (OPEN Wireless RouTer): an open-source project for operating system embedded Linux based, mainly used on embedded devices with a web interface (LuCI). All components have been optimized to be small enough to fit in the limited storage and memory available in home routers.

SSID: The name of wifi.

DHCP: a network management protocol which allows a device to obtain an IP address and associated information automatically.

Docker: a free software that easily initializes the development environment.

MAC address: a unique identifier assigned to a network interface controller (NIC) for use as a network address in communications within a network segment.

IPv4 address: the fourth version of the Internet Protocol (IP).