Device Pairing Subsystem 1.0.0

SPA Installation Guide

DPS-SPA-Installation-Guide-1.0.0

Oct 15, 2018

**Revision history**

| **Revision** | **Date** | **Description** |
| --- | --- | --- |
| A |  | Initial release |

Contents

[1 Introduction 4](#_Toc529540464)

[1.1 Purpose & Scope 4](#_Toc529540465)

[1.2 Definitions, Acronyms & Abbreviations 4](#_Toc529540466)

[1.3 References 4](#_Toc529540467)

[1.4 Getting Started 4](#_Toc529540468)

[2 Installation 5](#_Toc529540469)

[2.1 System Requirements 5](#_Toc529540470)

[2.1.1 Software Requirements 5](#_Toc529540471)

[2.1.2 Hardware Requirement 5](#_Toc529540472)

[2.1.3 Operating System 5](#_Toc529540473)

[2.1.4 Supported Desktop Browsers 5](#_Toc529540474)

[2.1.5 Supported Mobile Browsers 5](#_Toc529540475)

[2.2 Extracting Software Release 6](#_Toc529540476)

[2.3 Manual Installation 6](#_Toc529540477)

[3 Configuration 7](#_Toc529540478)

[3.1 Apiman, Keycloak and API Configurations 7](#_Toc529540479)

[3.2 User Role & Frontend Configurations 8](#_Toc529540480)

[4 Testing 10](#_Toc529540481)

**Tables**

[Table 1- Definitions, Acronyms & Abbreviations 4](#_Toc527736812)

[Table 2- References 4](#_Toc527736813)

# Introduction

## Purpose & Scope

This document provides:

* Installation instructions for Single Page Application of Device Pairing Subsystem

## Definitions, Acronyms & Abbreviations

**Table 1- Definitions, Acronyms & Abbreviations**

| Term | Explanation |
| --- | --- |
| DIRBS | Device Identification, Registration & Blocking System |
| DPS | Device Pairing Subsystem |
| OS | Operating System |
| Nginx | An open source, lightweight, high-performance web server or proxy server. |
| Apache | An open source web server |
| API | Application Program Interface |
| Gateway | A Gateway route requests from clients to services |
| SPA | Single Page Application |
| IAM | Identity Access Management |

## References

**Table 2- References**

| Ref no. | Document |
| --- | --- |
| REF-1 | IAM User Guide (section 3) |

## Getting Started

The instructions provided in this document assumes that the required equipment (hardware, software) has been installed and configured.

# Installation

1. The reader acknowledges and agrees that he is entirely and solely responsible for the selection and use of all third-party software modules downloaded and installed by this installation method, including securing all appropriate and proper rights of use to any of such third-party software modules and to comply fully with any terms of use that may apply to or accompany any such third-party software modules.

Qualcomm Technologies, Inc. does not undertake any obligations, duties, or other responsibilities in connection with the selection or use by the reader of any of such third-party software modules.

## System Requirements

### Software Requirements

* Node v8.9.4 or greater
* NPM v5.6.X or greater
* Nginx 1.14.X

### Hardware Requirement

Minimum hardware requirements are:

* At least 512 MB of RAM
* At least 1G of disk space

### Operating System

This system will be installed and configured with Ubuntu 16.04. Refer to the [Ubuntu Installation Guide](https://help.ubuntu.com/lts/installation-guide/i386/install.en.pdf) for additional installation help.

### Supported Desktop Browsers

* Chrome 63.0 (recommended)
* Internet Explorer 11.0
* Firefox 52.5 ESR, 57.0
* Safari 11.0
* Edge 41.16299

### Supported Mobile Browsers

* Chrome 63.0 (recommended)
* Safari 11.1
* UC Browser 11.5
* Opera (Android) 44.1
* Opera Mini (iOS) 16.0.7
* Samsung Internet 6.2
* Android (included with Android OS4.1.X-4.3)

## Extracting Software Release

The DPS software release is distributed as a tar.gz file. To extract the contents of the  
distribution, run:

tar xvzf dirbs-dps-spa-1.0.0.tar.gz

## Manual Installation

* Go to dirbs-dps-spa-1.0.0

pushd dirbs-dps-spa-1.0.0

* Install dependencies by running below mentioned command

npm install

# Configuration

## Apiman, Keycloak and API Configurations

To align Apiman and API Configurations go to *src/settings.json* file and make configurations accordingly.

nano src/settings.json

|  |
| --- |
| {  "appDetails": {  //Configure Application name, make sure that this appName must be the first //part of keycloak user’s role e.g. “dps\_mno\_operator1” in here “dps” part //refers to appName and “mno” part refers to user’s type and last part “operator1” refers to user level configured in Keycloak.  "appName": "APP\_NAME",  //configure your support email  "supportEmail": "example@company.com",  //configure your support number  "supportNumber": "+923001234567" },  //Note: Configure to directly hit APIs or API Gateway (Apiman). In order to //use any of the configuration, change its "use" value to "True". If both //configurations are enabled, preference will be given to first //configuration.  "api": {  //Configure API Host e.g. http://www.api-example.com  "host": "http://SERVER\_IP",  //Configure API Port e.g. 3000  "port": "PORT\_NUMBER",  //Configure API Version e.g. /api/v1/  "version": "VERSION\_OR\_SUBPATH",  //for directly hitting API, make it \*True\*  "use": true  },  "apiman": {  //configure Apiman Host e.g. http://www.apiman-example.com  "host": "http://SERVER\_IP",  //Configure Apiman Port e.g. 8000  "port": "PORT\_NUMBER",  //configure clientID e.g. /apiman-gateway/example/appname/1.0  "clientId": "CLIENT\_ID\_OR\_SUBPATH",  //for hitting Apiman Gateway directly, make it \*True\*  "use": false  }  } |

**Note**: To configure SPA in IAM see IAM User Guide document (section 3)

* Copy the downloaded Keycloak.json file into public directory.

## User Role & Frontend Configurations

To configure User Roles go to *src/utilities/constants.js* file and make configurations accordingly.

nano src/utilities/constants.js

// configure pagination limit for Search module  
export const PAGE\_LIMIT = 10;

//configure User role, make sure that this user role must

//be the last part of keycloak user’s role e.g. //“dps\_authority\_authority” in here “dps” part   
//refers to appName and second part “authority” refers to

// user’s type and last part “authority” refers to user role level //configured in Keycloak.

export const AUTHORITY = 'authority';

export const MNO = 'mno';

//Configure country code

export const COUNTRY\_CODE = '92';

* Build the code for production

npm run build

* It will create a build directory, copy the content of the directory to web root directory e.g. /var/www/html (default Nginx web root directory)

cp -r build/\* /var/www/html

# Testing

To verify the installation, visit server web IP on browser i.e. <http://your_server_IP_address>