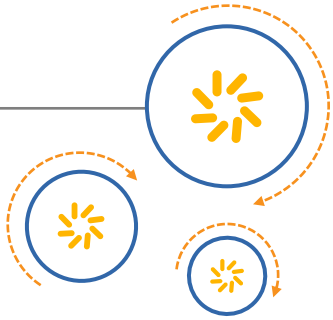
Qualcomm Technologies, Inc.



Genuine Device Detection Subsystem 1.0.0

SPA Installation Guide

GDDS-Installation-Guide-SPA-1.0.0

November 11, 2020

Revision history

Revision	Date	Description
A	November 2020	Initial Release

Contents

Tables

1. Introduction

1.1 Purpose & Scope

This document provides:

- Installation instructions for Single Page Application of Genuine Device Detection Subsystem

1.2 Definitions, Acronyms & Abbreviations

Table 1 - Definitions, Acronyms & Abbreviations

Term	Explanation
DIRBS	Device Identification, Registration & Blocking System
GDDS	Genuine Device Detection 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
Yarn	A package manager that replaces the existing workflows for the npm client or other package managers while remaining compatible with the npm registry

1.3 References

Table 2 - References

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

1.4 Getting Started

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

2. Installation

NOTE: 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.

2.1 System Requirements

2.1.1 Software Requirements

- Node v8.9.4 or greater
- Yarn 1.16.0
- NPM v5.6.0 or greater
- Nginx 1.14.X

2.1.2 Hardware Requirement

Minimum hardware requirements are:

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

2.1.3 Operating System

This system will be installed and configured with Ubuntu 16.04. Refer to the Ubuntu Installation Guide for additional installation help.

2.1.4 Supported Desktop Browsers

Table 3 - Supported Desktop Browsers

Name	Version
Internet Explorer	11.0
Firefox	82.0.3
Chrome	86.0
Safari	13.1.2
Edge	86.0

2.1.5 Supported Mobile Browsers

Table 4 - Supported Mobile Browsers

Name	Version
Chrome	86.0.4
UC Browser	13.3.3
Opera (Android)	60.3
Opera Touch (iOS)	2.5.1
Safari	13.1.2
Samsung Internet	13.0.1
Android	10

2.2 Extracting Software Release

The GDDS SPA release can be downloaded via one of the following two methods. To extract the contents of the distribution, run either:

Method 1: Download and unzip from GitHub

```
unzip Genuine-Device-Detection-Subsystem-Frontend.zip
```

Method 2: Clone the repository from GitHub

```
git clone https://github.com/CACF/Genuine-Device-Detection-Subsystem-Frontend.git
```

2.3 Manual Installation

- Go to `Genuine-Device-Detection-Subsystem-Frontend`
  `cd Genuine-Device-Detection-Subsystem-Frontend`
- Install dependencies by running below mentioned command
  `npm (use yarn instead of npm) install`

3. Configuration

3.1 Apiman, Keycloak and API Configuration

For Keycloak, Apiman and API Configurations, make a copy of settings-template.json in src directory and name it settings.json.

cp src/settings-template.json src/settings.json

nano src/settings.json

```
{
  "appDetails": {
      "appName": "gdds", // configure Application name, make sure that this appName must be same as mentioned in Keycloak roles, e.g. gdds_authority
      "supportEmail": "support@example.com", // configure this email as it will be visible for unauthorized user
      "supportNumber": "PHONE_NUMBER", // configure this contact number as it will be visible for unauthorized user
      "defaultLanguage": "en"
  },
  "api": {
      "host": "HOST API", // Configure API Host e.g. http://www.api-example.com
      "port": "API PORT", // Configure API Port e.g. 3000
      "version": "API VERSION", // Configure API Version e.g. /api/v1/
      "use": false // for directly hitting API, make it *True*
  },
  "apiman": {
      "host": "HOST APIMAN", // configure Apiman Host e.g. http://www.apiman-example.com
      "port": "APIMAN PORT", // Configure Apiman Port e.g. 8000
      "clientId": "APIMAN CLIENT ID", // configure clientID e.g. /apiman-gateway/example/appname/1.0
      "use": true  // for hitting Apiman Gateway directly, make it *True*
  },
  "keycloak": {
      "host": "http://SERVER_IP", // keycloak url
      "port": "PORT_NUMBER", // keycloak port
      "version": "VERSION_OR_SUBPATH",
      "clientId": "CLIENT_ID",
      "realm": "REALM",
      "use": true
  }
}
```


3.2 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. "gdds_admin_user" in here "gdds" part refers to appName
and

// second part "admin" refers to user’s type and last part "user" refers to
user

// role level configured in Keycloak.

export const AUTHORITY = 'authority';

export const OEM = 'oem';


//Configure country code

export const COUNTRY_CODE = '92';
```

Note: To configure SPA in IAM, see IAM User Guide document section 3 and then perform following steps:

- 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`

4. Testing

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

