|  |  |  |  |
| --- | --- | --- | --- |
| **Enterprise Business Process** | Optum Connect | **Last Updated Date** | 03/07/2023 |
| **Revision #** | 1.0 | **Process Owner** | Charles Colstrom |
| **Written/Revised By** | Tejinder Singh | | |

**Table of Contents**

[1. Purpose 2](#_Toc108286765)

[2. Scope 2](#_Toc108286766)

[3. Roles & Responsibilities 2](#_Toc108286767)

[Role 2](#_Toc108286768)

[Responsibilities 2](#_Toc108286769)

[4. Overview 2](#_Toc108286770)

[5. Project Code Structure 2](#_Toc108286771)

[6. Major Code Files 5](#_Toc108286772)

[7. Build/Deploy 6](#_Toc108286773)

[8. Glossary 7](#_Toc108286774)

[Figure 1: OCAP Project Code Structure 3](#Figure1)

[Figure 2: Dist Folder 3](#Figure2)

[Figure 3: Node Modules Folder 4](#Figure3)

[Figure 4: SRC Folder 4](#Figure4)

[Figure 5: Head Section Index. html 5](#Figure5)

[Figure 6: NG Directives in Body Section 5](#Figure6)

[Figure 7: NG Build Command 6](#Figure7)

[Figure 8: NG Serve Command 7](#Figure8)

[Figure 9: Command NG Deploy 7](#Figure9)

# Purpose

The purpose of this document is to serve as a guide for the Optum Connect Cloud (OCC) team to follow for services related to Optum Cloud Automation Platform (OCAP) application development, understanding, associated procedures, and technical aspects.

# Scope

Use this document as a reference guide to understand the internal functioning and development code structure of the OCAP application.

# Roles & Responsibilities

Refer to the following table for OCC Engineering Team responsibilities relating to OCAP.

|  |  |
| --- | --- |
| Role | Responsibilities |
| OCC Engineering Team | Develop and maintain the OCAP application |

# Overview

OCAP is a web-based, GitHub pages, hosted application. This tool provides users with the ability to deploy landing zone resources in an automated fashion, through an intake form, without the need to write actual code.

As the application requires to be deployed/hosted on GitHub pages, hence developed using components, and majorly focused on the client-side development. Refer to the following list of the programming components leveraged:

* Angular
* Hyper Text Markup Language (HTML)
* Cascading Style Sheets (CSS)/Bootstrap
* Node JavaScript (JS)
* Application Programming Interface (API) integrations
* JavaScript Object Notation (JSON)

# Project Code Structure

The application is developed on a NodeJS platform, leveraging AngularJS, and the project code structure is like any NodeJS application as shown in Figure 1.

A picture containing table

Description automatically generated  
Figure 1: OCAP Project Code Structure

The root directory includes the following list of folders and files:

* **Dist**

The Dist folder, short for distribution folder, is dynamically generated when compiling or executing a build on the project code and includes the production ready HTML files and assets. Refer to Figure 2.

Graphical user interface, text, application

Description automatically generated

Figure 2: Dist Folder

* **Node\_Modules**

The directory includes libraries and dependencies for JS packages, used by the Node Package Manager (NPM). Refer to Figure 3.

Graphical user interface, application

Description automatically generated with medium confidence

Figure 3: Node\_Modules Folder

* **SRC**

The SRC folder, short for source, includes the source files to build and develop the project. This location is where the original source code files are located, before being compiled. Refer to

Figure 4.

A picture containing graphical user interface

Description automatically generated

Figure 4: SRC Folder

* **Root files**

Major files at the root of the project source code are:

* + **Package.json**: Defines libraries and dependencies for JS packages, used by NPM
  + **Package-lock.json**: Specific version lock for dependencies installed from package.json, used by NPM
  + **Readme.md**: Readme/help file for the application repository
  + **Optumfile.yml**: Config file for the Abilities, Skills, Knowledge (ASK)/AIDE application
  + **Angular.json**: Provides project-specific configuration defaults for build and development tools provided by the Angular Command Line Interface (CLI)

# Major Code Files

Below is a list of major files involved into OCAP application development:

1. **Index.html**

The index.html file is located at the root under the SRC folder (i.e., src/index.html, and comprises of the intake form html and angular module code).

The header section comprises of Uniform Resource Locator (URLs) of the Bootstrap/CSS, AngularJS, and other JavaScript files being called within the HTML tags. Refer to Figure 5.

Text

Description automatically generated

Figure 5: Head Section Index.html

The body section of the code calls the AngularJS ng-app and ng-controller directives and defines the whole page structure under various div tags. Refer to Figure 6.



Figure 6: NG Directives in Body Section

1. **Services.js**

Services.js is the JavaScript file located under the Assets/JS folder in the SRC folder (i.e., at the project path - **src/assets/js/services.js)**. The file comprises of all the backend logic related to:

* Data read/write operations
* Intake form
* Page/Div load
* GitHub API calls
* Terraform/Jenkins code

1. **WinVM.js**

WinVM.js is the JavaScript file located under the assets/js/resourceModules folder in SRC folder (i.e., at the project path - **src/assets/js/resourceModules/winVM.js)**. The file comprises of all the backend logic related to Virtual Machine Deployments for:

* Data read/write operations
* Intake form
* Page/Div load
* Github API calls
* Terraform/Jenkins code

# Build/Deploy

The Angular (ng) commands used for builds and deployment of the application are as follows:

* **Build**:

The command ng build is used for the compilation of the application and generates the compiled code and binaries in the Dist folder. Refer to Figure 7.

Table

Description automatically generated with low confidence

Figure 7: Ng Build Command

The above figure demonstrates an example of the ng build run on the OCAP application; the output presents the logs related to build compilation.

* **Run**:

The command ng serve is used to run/serve the application locally on your dev machine; it rebuilds the application in case any changes are present from the last build. Refer to Figure 8.

Graphical user interface, text, application

Description automatically generated

Figure 8: Ng Serve Command

Figure 8 demonstrates an example of the ng serve run on the OCAP application; the output presents the logs related to the compilation, and the localhost URL of the application.

* **Deploy**:

The command ng deploy invokes the deploy builder for the application and deploys the build to the GitHub pages. Refer to Figure 9.

**ng deploy --base-href=/pages/<*ORG*>/<*Repository*>/**

Figure 9: Command NG Deploy

<***ORG***> -> Identifies the Organization of the GitHub repository

<***Repository***> -> Is the repository name of the application in GitHub

# Glossary

Refer to the following acronyms and definitions.

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| API | Application Programming Interface |
| ASK | Abilities, Skills, Knowledge |
| CLI | Command Line Interface |
| CSS | Cascading Style Sheet |
| DIST | Distribution |
| HTML | Hyper Text Markup Language |
| JS | JavaScript |
| JSON | JavaScript Object Notation |
| NG | Angular Command Line |
| NPM | Node Package Manager |
| OCAP | Optum Cloud Automation Platform |
| ORG | Organization |
| SRC | Source |
| URL | Uniform Resource Locator |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **The following are approved changes incorporated into the revision numbers indicated below** | | | |
| **Version No.** | **Date** | **Description of Change** | **Modified By** |
| 1.0 | 06/10/2022 | *Initial Draft* | Tejinder Singh |
| 1.1 | 07/08/2022 | Standardized and formatted content and sent to SME for review, approval, and acceptance | Melanie Parker |
| 1.2 | 03/07/2023 | Standardized & revised the document | Sneha |