Compass Learning App

Localization Guide

November 2024

Contents

[1. Introduction 8](#_Toc183437915)

[2. Before You Start 8](#_Toc183437916)

[3. Overall Localization Process 9](#_Toc183437917)

[4. Downloading the Assets 9](#_Toc183437918)

[4.1. GitHub Repositories 9](#_Toc183437919)

[4.1.1. Main Repository 9](#_Toc183437920)

[4.1.2. Offline Localization Repositories 9](#_Toc183437921)

[4.1.3. Asset Repositories 9](#_Toc183437922)

[4.1.4. Source Code Repositories 10](#_Toc183437923)

[5. Localizing and Translating the Assets Offline 10](#_Toc183437924)

[6. Updating the App 12](#_Toc183437925)

[6.1. Download the Files from GitHub 12](#_Toc183437926)

[6.2. Running the App Locally 12](#_Toc183437927)

[6.3. Software Setup 12](#_Toc183437928)

[6.4. Folder Structure 12](#_Toc183437929)

[6.5. JSON File Setup 13](#_Toc183437930)

[6.6. Editing JSON Content 14](#_Toc183437931)

[6.7. JSON Editing Tips and Tricks 16](#_Toc183437932)

[6.7.1. Figures 16](#_Toc183437933)

[6.7.2. Callouts 16](#_Toc183437934)

[6.7.3. Making Body Content Easier to View/Edit for Significant Edits 17](#_Toc183437935)

[6.8. Viewing Updates on the App 19](#_Toc183437936)

[6.9. Overwriting Images If Required 20](#_Toc183437937)

[7. Compiling the Compass Learning App 21](#_Toc183437938)

[8. Deploying and Maintaining the Compass Learning App 21](#_Toc183437939)

[9. More Information 22](#_Toc183437940)

# Introduction

The **Compass Learning App** is a PWA (progressive web application) that provides educational content to facilitate the development of diabetes-related clinical capacity among primary care healthcare professionals (i.e. medical officers, nurses and community health workers).

This document provides partner countries with guidance on how to localize and build the Compass Learning App from a technical perspective only. Guidance on the localization of scientific and medical content is not given in this document.

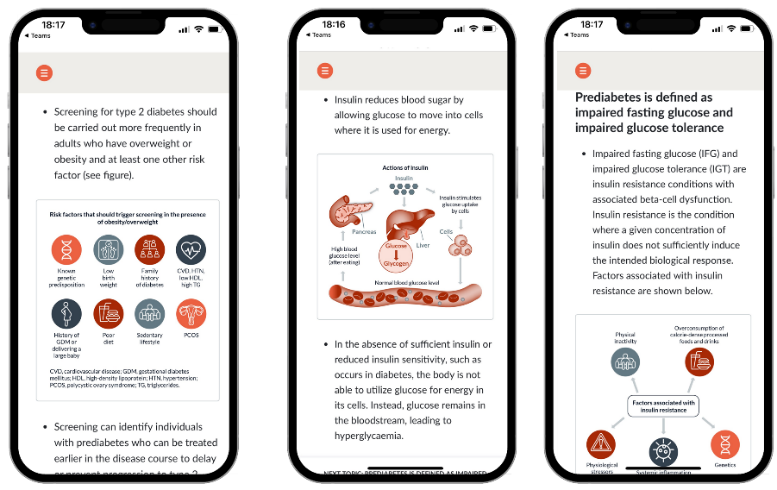
# Before You Start

To be able to localize the Compass Learning App, it is important that you have team members with the following experience available:

* Content Editor
  + **Therapy area experience** and an **understanding of local and regional treatment recommendations and options** to enable review and localization of scientific and medical content
* Developer
  + Intermediate experience with building **PWAs** using React, a popular library
  + Accompanying skills to understand and edit **JSON files, SCSS/CSS, Bootstrap CSS** and **HTML**
* Designer
  + Basic to intermediate experience with using **photo-editing software** (e.g. Adobe Photoshop, GIMP) and **image-editing software** (e.g. Adobe Illustrator, Inkscape)
* Video Editor
  + Basic to intermediate experience with using **video-editing software** (e.g. Adobe Premiere Pro, CapCut)

The roles listed above are examples; specific team structure/roles will vary from country to country. The above skills are recommended base-level requirements to localize and deploy the Compass Learning App.

Please note that this document provides an overview of the localization process, along with tips and tricks, and assumes a good understanding of photo, video and image editing, and web app development.



# Overall Localization Process

The overall process for localizing the Compass Learning App is as follows:

Deploy and maintain the app

Download the assets

Localize and translate the assets offline

Update the app

Compile the app

# Downloading the Assets

## GitHub Repositories

### Main Repository

This repository is the starter location that lists all the URLs required for accessing assets, documentation and the source code for both Doctor and Nurse versions of the Compass Learning App.

<https://github.com/SynaptikDigitalUK/WdfCompassElearningApp>

### Offline Localization Repositories

The following two repositories contain all the content (as Word documents and videos) from both the Doctor and Nurse versions of the Compass Learning App for offline review.

**Doctor**  
<https://github.com/SynaptikDigitalUK/WdfCompassElearningAppDoctorDocumentation>

**Nurse**

<https://github.com/SynaptikDigitalUK/WdfCompassElearningAppNurseDocumentation>

### Asset Repositories

The following two repositories contain all the editable image/figure in EPS and PSD formats, so they can be edited using commonly supported image-editing software.

**Doctor**  
<https://github.com/SynaptikDigitalUK/WdfCompassElearningAppDoctorAssets>

**Nurse**

<https://github.com/SynaptikDigitalUK/WdfCompassElearningAppNurseAssets>

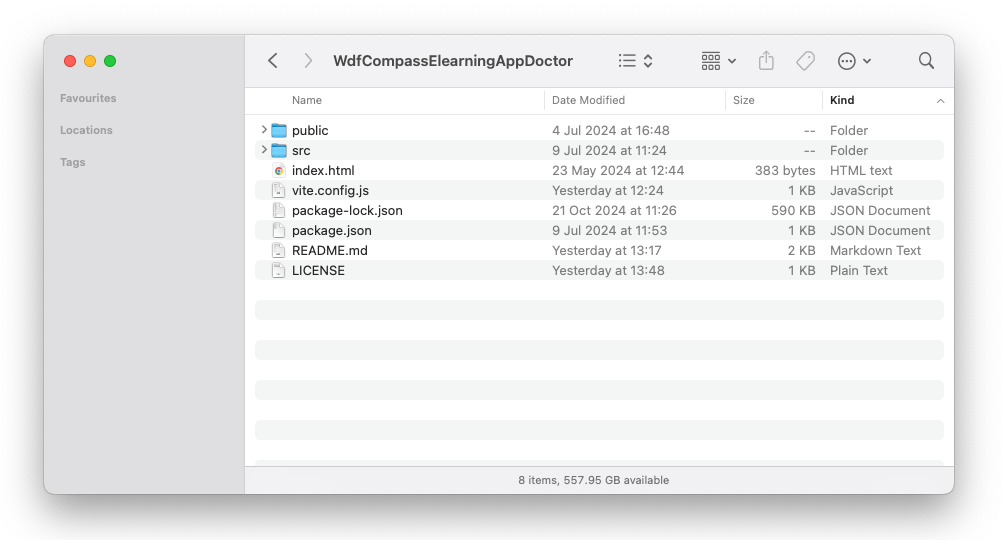
### Source Code Repositories

The following two repositories contain all the working source files that are used to build the PWAs. Each repository contains a **README.md** file (2kB) that instructs the developer on how to set up a working folder locally on their machine and provide a quick start guide.

**Doctor**  
https://github.com/SynaptikDigitalUK/WdfCompassElearningAppDoctor

**Nurse**

<https://github.com/SynaptikDigitalUK/WdfCompassElearningAppNurse>



# Localizing and Translating the Assets Offline

The Content Editor or equivalent team member should download the Word and video files from the Offline Localization Repositories in GitHub. These files can be reviewed, localized (including translations as needed) and used as briefs for the Designers, Video Editors and Programmers, and other relevant team members.

If any of the images or figures need to be updated, the source files can be downloaded from the GitHub Assets Repository and edited using image-editing software that can open .eps files.

Please note that the width of figures in the .eps file should fit within the artboard area of 670px width and there is no height restriction. Please keep in mind that the artboard area must sit as close as it can to the top and bottom of the graphic for when you export them as PNG files.

A graph of diabetes type one

Description automatically generated with medium confidence

When updating scientific content, consider the length of each screen. Whilst there is no strict recommendation for word limit per screen, where viable it is recommended to include 4–5 short paragraphs per screen plus a combination of 1–2 images and, where applicable, 1 video. Avoid continuing the same sub-topic over two separate screens. For maintenance, see   
Section 8.

# Updating the App

## Download the Files from GitHub

Download the working source files from the GitHub source code repositories.

## Running the App Locally

To run the source code for each app (Nurse or Doctor) locally, please follow the quick guide below:

* Ensure you have Node and GIT installed onto your machine
* Clone the repository from GitHub using your preferred GIT-based version control system (gitKraken, sourcetree, terminal etc.)
* Open theterminal app or command line and navigate to the app folder using the ‘**cd**’ command
* Within the same terminal or command line window, type ‘**npm install**’. This will install all third-party dependencies required for the app to run locally on your machine. You will notice a new folder appear in the root of the app folder called ‘**node\_modules**’; this is normal.
* Type ‘**npm run dev**’ to run the app locally on your machine
  + This will set up a local webserver on your machine and the app can be viewed from your browser using ‘**http://localhost:5173**’
  + From this point when you edit and save files, the browser will update those changes automatically in the open browser

## Software Setup

Please download and install VS Code and the VSCode JSON Editor plugin from the following links to be able to edit the body text of the app:

* VS Code: <https://code.visualstudio.com/>
* VSCode JSON Editor plugin: <https://marketplace.visualstudio.com/items?itemName=nickdemayo.vscode-json-editor>

## Folder Structure

To change the content, you will need to edit the ‘**s1.json, s2.json, s3.json, s4.json, s5.json** and **references.json**’ files, which are located within the ‘**assets/data**’ folder.

|  |  |
| --- | --- |
| Open the ‘**assets**’ folder. | A screenshot of a computer  Description automatically generated |
| Open the ‘**data**’ folder. | A screenshot of a computer  Description automatically generated |
| Within the data folder, you will find the ‘**config.json**’ file. This is the main structure setup of the website.  The content for each section is stored in  ‘**s1.json, s2.json, s3.json, s4.json, s5.json** and **references.json**’ files. | A screenshot of a computer  Description automatically generated |

The JSON files are structured as follows:

|  |  |  |
| --- | --- | --- |
| **JSON File** | **Section Title** | **Abbreviation** |
| s1.json | 1. Prevention, Screening and Diagnosis | psd |
| s2.json | 2. Lifestyle Management for People With Diabetes | lmpd |
| s3.json | 3. Pharmacological Management of Diabetes | pmd |
| s4.json | 4. Preventing and Managing Complications of Diabetes | pmcd |
| s5.json | 5. Patient Engagement in Diabetes Management | ropdm |

All other content e.g. ‘Terms and Conditions’, ‘Instructions for Offline Access’ and ‘Disclaimer’, can be edited directly in the HTML files.

## JSON File Setup

Once you have opened the file, you need to update ‘**s1.json**’ file with VSCode, go to View > Command Palette (or press Ctrl+Shift+P on a PC or Command+Shift+P on a Mac), which will bring up the following dialog box so you can start the JSON editor session.

A screenshot of a computer

Description automatically generated with medium confidence

The JSON editor will appear on the right-hand side, which will be split into two areas:

* **Menu** – this drives the side and main menu titles.
* **Content** – this drives the page content­.

A screenshot of a phone

Description automatically generated

## Editing JSON Content

To find the section of content you want to edit:

* Open the app locally with VSCode
* Find the page you want to edit and make a note of the final term in the URL
* The final term in the URL is the same as the page id in the JSON file
  + In the example below, please note that the final term in the URL is **/psd\_c1/psd\_c1p1**

A screenshot of a computer

Description automatically generated

* **/psd\_c1/psd\_c1p1** matches the JSON file below:
  + **/psd\_** = The first part ‘psd’ is the main section abbreviation it belongs to
  + **/psd\_c1/** = The section abbreviation + the chapter number of that section
  + **/psd\_c1p1** = The section abbreviation + chapter number + page number

A screenshot of a computer

Description automatically generated

## JSON Editing Tips and Tricks

### Figures

|  |  |
| --- | --- |
| The images and videos used within the site have divs wrapped around them with classes attached that control their styles.  Example CSS used:   * image-wrapper * video-wrapper | A screenshot of a computer program  Description automatically generated  A graph of diabetes type one  Description automatically generated  A person wearing glasses and a suit  Description automatically generated |

### Callouts

|  |  |
| --- | --- |
| The app contains callout blocks of text that are wrapped with a div, like this example.  Example CSS used:  topic-question-callout | A black background with white text  Description automatically generated  A screenshot of a phone  Description automatically generated |

### Making Body Content Easier to View/Edit for Significant Edits

|  |  |
| --- | --- |
| Find the page you need to edit and copy the whole block of text from the content block (e.g. see highlighted text. | A screenshot of a computer screen  Description automatically generated with medium confidence |
| Open a new HTML file.  Paste copied content into the new HTML file, select all, right click to open a submenu and select ‘format document’ (or press Option +Shift+F on a Mac or Alt+Shift+F on a PC). | A screenshot of a computer  Description automatically generated |
| The content will now be easier to view/edit.  HTML tags are used throughout – most of the content is built using p, ul, li and divs. | A screenshot of a computer  Description automatically generated |
| Once all edits are completed, select the content you want to combine.  Open Command Palette by going to View > Command Palette (or pressing Ctrl+Shift+P on a PC or Command+Shift+P on a Mac).  In the popup, type and select ‘Join lines’. | A screenshot of a computer  Description automatically generated |
| Content converts from multiple lines into a single line, which makes it better for copying into the JSON file. | A screenshot of a computer  Description automatically generated |
| Copy the updated content from the HTML file and paste it back into the content area of the JSON file.  Then click on the .JSON file and save it; the green circle at the end of file name indicates that the file has been updated. | A screenshot of a computer program  Description automatically generated with medium confidence |

## Viewing Updates on the App

It will be likely that the content will be cached; therefore, to view your updates to images or to text, hold down ‘shift’ and click the ‘reload’ button on the browser.

A screenshot of a computer

Description automatically generated

## Overwriting Images If Required

Files in the root level of the ‘assets’ folder within the source code are the images or graphics that are used globally within the site.

Images used in the content are stored in the relevant course folder in the ‘assets’ folder. Each image is systematically labelled with each element of the file name separated by a dash. The elements of the file name are defined in the following table.

|  |  |  |
| --- | --- | --- |
| **Chapter Number** | **Page Number** | **Graphic on Page Number** |
| c1 | p1 | fig-01 |

As an example:

|  |  |
| --- | --- |
| This is the first figure  (**fig-01**) on the first page (**p1**) of the first chapter (**c1** i.e. Diabetes Overview), which is within the ‘Prevention, Screening and Diagnosis’ (**psd**) section | A graph of diabetes type one  Description automatically generated |
| Therefore, this image sits within assets/**psd** and is labelled **c1**-**p1**-**fig-01**.png | A screenshot of a computer  Description automatically generated |

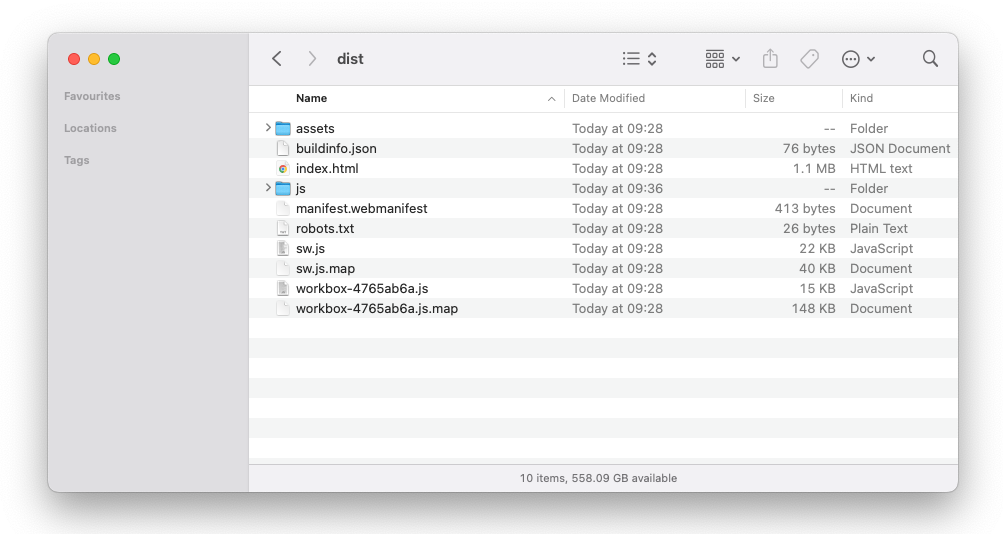
Please note that the width of figures in the .eps file should fit within the artboard area of 670px width and there is no height restriction. Please keep in mind that the artboard area must sit as close as it can to the top and bottom of the graphic for when you export them as PNG files.

# Compiling the Compass Learning App

You have now created a [locally running app](#_Running_the_App) and [updated](#_Updating_the_App) it, the next step is to compile the updated app so that it is ready to be deployed onto a webserver or distribution network.

To compile the updated app (Nurse or Doctor), you need to complete the following steps:

* Open the terminal or command line and navigate to the app folder using the ‘**cd**’ command
* Within the same terminal or command line window, type ‘**npm run build**’, which will create a distribution/production version of the PWA ready to be added to any webserver so it can be accessed by end users
* The contents of the ‘**dist**’ folder will look like the screenshot below



* The ‘**assets**’ folder contains all images, videos and data that are shown within the app
* The ‘**js**’folder contains a javascript module required to view the pdf within the app
* The ‘**index.html**’ file contains all the static JavaScript, CSS and HTML code that is needed for app to function

# Deploying and Maintaining the Compass Learning App

Once you have compiled the app and have produced a ‘**dist**’ folder, please copy all contents of this folder to the desired webserver folder that will be hosting the app.

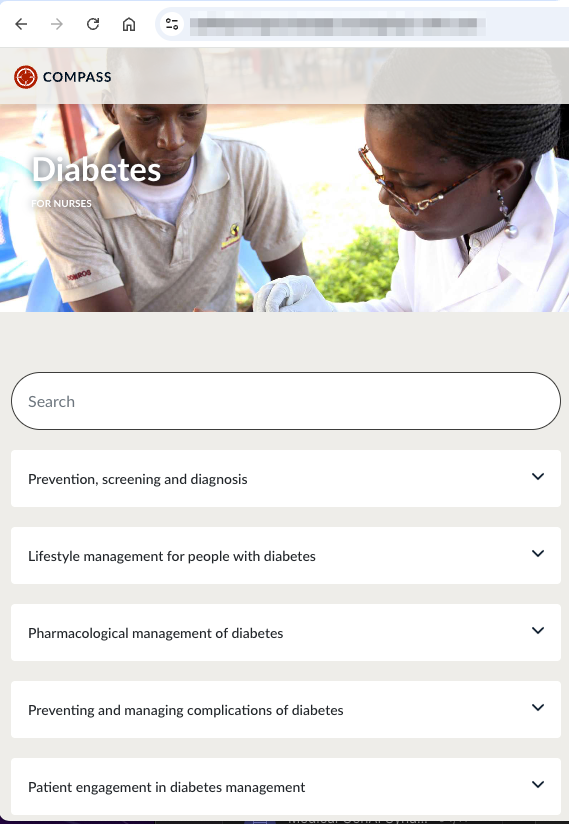
* The ‘index html’ file should be the default document to point to when a webmaster is setting up the domain
* The webserver needs to support the following mime types, the definitions of which can be found here at [developer.mozilla.org:](https://developer.mozilla.org/en-US/docs/Web/HTTP/MIME_types/Common_types)

|  |  |
| --- | --- |
| .png | .html |
| .jpg | .css |
| .mp4 | .svg |
| .mjs | .json |
| .js | .webmanifest |
| .txt | .map |

* When a user visits this new webpage, the app should begin to function and the contents should be viewable
* Important note for the webserver setup:
  + A new mime type will need to be added to the webserver to allow parts of the application to be displayed. The webmaster who is controlling the server setup needs to add the following:

|  |  |  |
| --- | --- | --- |
| **Extension** | **Kind of Document** | **MIME Type** |
| .mjs | Javascript module | text/javascript |

* You will know that the app is working when you see a similar screen in your browser to the one below when you visit the new domain name:



Maintenance: although it is not essential, we recommend a quick review of the app once every 6–12 months to check for any bugs.

# More Information

Please email [contact@worlddiabetesfoundation.org](mailto:contact@worlddiabetesfoundation.org) for more information or if you have any questions.