SystemSeed

WHO - Step by step

Developer codebase documentation

Written by
Rodrigo Aguilera, Kate Marshalkina

Published: [10/02/2022]

Technical architecture overview

A high level view of how the app is built and the technologies used for it.

Distribution architecture

Step By Step app is developed as a Drupal distribution (so called “parent”) that can be installed and customized by individual countries (“child”). When installing their own copy of the app, countries benefit from the existing Step By Step features and updates while being able to customise the app to their own needs on functional and content levels.

Drupal 9

The CMS is a central part for the information architecture of the app providing the different content types, the user management and backend administration for the different roles that interact with the app.

Anu LMS

Systemseed open source LMS product built on top of Drupal providing both the entities for data storage plus interfaces for editing them and also the progressively decoupled React application that will be displayed on the frontend for the users to view the content of the different sessions.

PWA

The application is built with the idea of being able to use it offline so it relies on the PWA module for Drupal and the offline features of Anu to allow the user to use the app with no connectivity.

DevOps

When hosted on SystemSeed hosting, Step By Step gets first class DevOps setup with code quality checks, automated tests and security updates.

As with many modern Drupal applications the dependencies are managed by Composer. This provides a structure for the code that separates the core, the contributed modules and the custom code.

The code is structured to use Docker not only for local development but also the same docker containers are going to be used to run the application in a production environment.

To wrap the complexity of using containers and many common tasks the command line tool “make” is leveraged. Since make is already installed in many operating systems we keep the requirements for running the app at a minimum.

Key application features are covered by end-to-end tests written in the Codeception framework.

Local installation

Requirements

- Composer
- PHP 7.4+
- Read access to https://github.com/systemseed/who-stepbystep

For more details on technical requirements, check Drupal 9 system requirements.

Installing from Composer template

This is the quickest way to setup Step By Step project.

Run the command below to create a new Composer project on top of Step By Step installation profile:

```
composer create-project systemseed/who-stepbystep-project stepbystep
--stability=dev --repository="{\"url\":
\"git@github.com:systemseed/who-stepbystep-project.git\", \"type\":
\"vcs\"}"
```

During installation, you may be asked for a GitHub auth token. Follow Composer instructions to generate GitHub token and provide it when asked by Composer installer.

Install your Step By Step site like any other Drupal website using the install.php script or drush.

Example Drush command:

```
drush site:install stepbystep --account-name=sbssuperadmin
--site-name="Step by Step"  --yes
```

Full instructions and alternatives ways to install Step By Step app can be found in the repository documentation.

Demo video: https://www.youtube.com/watch?v=_fON9KKhjcY

Recommended approach to customizations

The best way to customize the app is to create new Drupal modules and use Drupal / Anu LMS APIs to override or extend existing functionality.

To customize the theme, create a new theme based on the existing `material_sbs` theme.

Editing source code of Anu LMS or SbS modules directly is highly discouraged as it will break compatibility and long term maintainability of the app.

If you found a bug in the SBS installation profile, please open pull request in https://github.com/systemseed/who-stepbystep

Anu LMS customizations

One central part of the app is the Anu LMS that is part of the codebase as a contrib Drupal module and then it is personalized for the SbS project in the `sbs_application` and `sbs_sessions` modules.

Note that the concept of “course” in Anu LMS is renamed to “session” in SbS but the internal naming is still “course”.

Initial steps

When the users register for the app and after the verification process they are welcomed with a page that explains the first steps, although the the content is configurable by an editor here:
https://whostepbystep.org/admin/content/questionnaires/first-welcome-page
And a very similar page for when they come back to the app but have not completed all initial steps
https://whostepbystep.org/admin/content/questionnaires/welcome-back-page

The participants continue the journey filling some questionnaires configured in the following config page.
https://whostepbystep.org/admin/content/questionnaires/order
The last questionnaire has a score associated with each answer and based on the final score of the whole questionnaire a different page is displayed for the user.
The score levels can be edited at:
https://whostepbystep.org/node/100/edit-score-levels
And the content for the pages based on the previous score levels

https://whostepbystep.org/node/100/edit-score-level-pages

The logic for the order of the questionnaires and the score levels can be found in `SbsQuestionnaireSubscriber.php`.

Certain levels will also trigger the app to offer an E-helper to the participant. The levels and the content of the popup can edited here https://whostepbystep.org/admin/content/questionnaires/ehelper

The logic for when to display a popup to request an E-helper can be found at `SbsEhelpers.php`.

Character storylines

After completing the questionnaires the participants must choose a character that identifies with them and that choice will determine the sessions that will be available for them.

It can be later changed in the profile section if a participant decides the character chosen is not a fit for them.

This functionality is split between the `anu_lms_storyline` (generic functionality) module and the `sbs_storyline` module (specific to SbS).

E-helper request

When the participants request an E-helper they are presented with another questionnaire that requests some additional information about them. This request is stored in the system for a coordinator to see and later assign an E-helper to a participant.

The questionnaire that will contain the questions can be configured here https://whostepbystep.org/admin/content/questionnaires/ehelper

Activities

During the journey of going through the sessions the participants are presented with activities. There are two kinds of activities, audio and checklist. Once the activities are discovered in a

session they are available in the toolbox (https://whostepbystep.org/toolbox) so participants can go back to them later.

The code for the activities can be found on the `sbs_activities` module including the frontend react application that displays the activities themselves.

Roles

- Developer - this is the super admin, a developer has access to everything
- Manager - has rights to add/manage different admin users
- Content Editor - has rights to edit and create all content in the app
- Coordinator - has access to the coordinator dashboard and can assign E-helpers to participants
- E-helper - has rights to see the data of participants they have been assigned by coordinators
- Translator - has rights to translate content on all pages of application
- Authenticated user: a participant that fills questionnaires and reads through the sessions
- Anonymous user: only able to register or trigger a password reset

Note that only participants are required to go through the initial steps and therefore content editors, coordinators and E-helpers won't have access to a character storyline or the toolbox unless they choose a character storyline in their profile.

E-helper assignment

The coordinator has a dashboard where information about the recently registered users can be seen.
https://whostepbystep.org/admin/people/unassigned
From there a coordinator can view the profile of a participant and assign an E-helper.

It is only possible to have one E-helper assigned at a time but the storage is ready for assigning multiple E-helpers if needed.

The code for assigning and de-assigning can be found in `sbs_ehelpers` module.

Participant notes

Coordinators and E-helpers can share arbitrary notes on the participant profile to track any kind of information to review later.
The code can be found in `sbs_user_notes` module.

Questionnaires

To fulfill the ability of content editors creating forms with a limited set of questions the questionnaire module was created, the machine name of the module is `webform_content` since it relies heavily on the contrib webform module.

A questionnaire is a node that references paragraphs entities, when the node is added/edited the `webform_content` module transforms that information into a webform entity that is going to be used to display the form to the participant.
https://docs.google.com/document/d/1dCnCw7gsjDhbhf-NWMpZarmN006o_Kg6jtFVq1vh49E/edit#heading=h.2zzabikol0mb
The code can be found in `webform_content` module.

Setup SMS provider for OTP

Users have the option to register with an email or a phone number. To validate that the phone number is valid the app sends a one-time password during registration or password reset.
The app at https://whostepbystep.org/ is configured to use the Twilio account of SystemSeed.
In order to you this functionality, you need to configure your own SMS provider at /admin/config/smsframework/gateways/twilio

Further reading

- Installation profile documentation on Github
- SbS ANU User Manual