

Technical Documentation

This document details the technical info on the code base for storyweaver.

| Author | Version/Date | Change |
|--------------------|---------------------------|--|
| Manoj Sukhavasi | V0.1 11th May 2018 | Initializing the document |
| Chiradeep Majumder | V1.0 4th July 2018 | Update the document with Dev support tasks |

[Introduction](#)

[How to add a new Language](#)

[Banner Management](#)

[Create](#)

[Update](#)

[List Management](#)

[Create](#)

[Blog related Support tasks](#)

[Blog creation](#)

[Blog support task](#)

[Localization text updates](#)

[Updating in Rails \(spp\)](#)

[Updating in React \(sw-js\)](#)

[How Search works](#)

[Story Editor](#)

Introduction

The core part of the website is built upon Ruby on Rails(RoR). We also use ReactJS for front-end to render the views. RoR provides API's which React will call to render the views. We

use three different databases - Postgresql, Elasticsearch and couchbase. Postgresql is a traditional database used for relations and models in Rails. We use elasticsearch for search purposes and couchbase to handle cache and user sessions. React renders most of the pages, but the profile dashboards and editor are still rendered by Rails.

How to add a new Language

Most of the fonts we use are from the Google Noto fonts.

1. In the `app/assets/stylesheets/language_fonts.css.scss`, add the new class and the font family.
2. In the `app/assets/stylesheets/fonts.css.scss.erb` add the url to access the respective font.
3. Then add the font file in the `app/assets/fonts` folder.
4. Add a new record in LanguageFont table with the script name and font name.
5. Then you can add the language on the Content Manager Dashboard on the website.
 - a. While adding the language another text field would be available to add translated name of the language

Banner Management

Create

1. We need to first have a designed banner in image format (jpeg/png) offline
2. The image needs to be put in the folder `app/assets/images/`
3. In the rails console and update the following:

Table name: banners

```
#
# id :integer not null, primary key
# name :string(255)
# is_active :boolean default(FALSE)
# link_path :string(255)
# position :integer
```

Update

1. For disabling an existing banner from showing up set

```
Is_active = FALSE
```

List Management

Create

1. Prerequisite is to have CSV file containing the mapping of lists and stories as follows :
Name , <Name of List>
Creator Org , <Org name>
Category , <Category name>
Description , <List description>
<story id> , <Synopsis/How to use>
2. Use `spp/lib/tasks/seedling_lists.rake` to parse the CSV and create/update lists

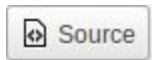
Blog related Support tasks

Blog creation

1. From the CM dashboard, click on “Blog dashboard” > “Published posts” tab and then on “Write new post”
2. Add title and content
3. Change status to published/scheduled
4. Click on Create blog post

Blog support task

1. If the blog is part of contest, there might be a need to add a contest link in the blogpost
2. For this, edit the blog > Change the editor to raw HTML by clicking on source



3. Update the required link for user to enter the contest
 - a. Sample HTML update `<p dir="ltr"></p>`

Localization text updates

Updating in Rails (spp)

1. Prerequisite is to have the translated text available along with the English keyword
2. For the English keyword existing in the `en.yml` file a corresponding entry should be made in the local language `.yml` file.
3. For example for having "Story Title" with a localized version the `en.yml` file and the `hi.yml` (Hindi localization file) files contains the following :

- a. `story_title: "Story Title"`
- b. `story_title: "कहानी का शीर्षक"`
- 4. To match these the keys in the yml file we will call this format "`<%= t "story_title" %>`" in the two yml files
- 5. If the key value pair does not exist, add it to both `en.yml` and `hi.yml` files and the respective key should be in the corresponding HTML page.

Updating in React (sw-js)

1. We use 'react-polyglot' [<https://www.npmjs.com/package/react-polyglot>] for handling translations in react pages.
2. Corresponding files are located in `src/i18n.en.json` (corresponding hindi translations are in `src/i18n.hi.json`)

How Search works

We use Elasticsearch engine for our search purposes. We use [Searchkick](#) gem to interface with the the Elasticsearch. Our Search classes can be found at lib folder.

Story Editor

We use [Wysihtml](#) editor for our story editor. We had modified some parts of wysihtml to suit it to the storyweaver.