Evaweb

Technical Documentation

Authored by Philipp Fehr and Lukas Bischof, 2020

Index

1 Abstract

EvaWeB is a survey platform with features like live evaluation of question groups as well as survey creation and easy access. This document should enable you to setup and maintain the project easily, besides helping you understand the code.

2 Structure

2.1 Language

The project is a Ruby application using the web framework Ruby on Rails. Rails is designed to to be easy, maintainable without the need of configuring a lot.

2.2 Components

The app mainly consists of a ”customer” / user part, consisting of the survey form and an administrator part, consisting of an admin dashboard and a live dashboard for the real-time evaluation of survey results.

The main app is written without any third party tools and can be found in the `app` directory. It consists of backend code and frontend code, which is located in the `app/webpack` directory.

The admin dashboard uses an adapted version of the the gem ”administrate” from Thoughtbot, which can be found under `https://github.com/orsa-scholis/administrate`. It is combined with the code of the live dashboard, which can be found alongside with the main app code Furthermore, the admin adjustements are namespaced under `admin` and can be found under `app/controllers/admin`, `app/dashboards`, `app/fields`, `app/views/admin` and `app/webpack/{javascript,stylesheets}/administrate`.

3 Development

3.1 Prerequisites

1. Ruby installed, with bundler. Required version: 2.6.3. It's best to install Ruby with rbenv or a similar tool
2. Postgres installed and a server instance listening
3. Optionally: A Redis server, if you want to replicate "real" production behaviour

3.2 Setup

1. Clone the repository using `git clone` or download with GitHub download button
2. Switch working directory: `cd evaweb`
3. Run `bin/setup`
4. Configure application by copying example file and filling in required values in `config/application.yml`

A possible sample configuration might look something like this:

```
SECRET_KEY_BASE: 2ad2be6b13a1e4094af9b16199729a91fbf
DEVISE_SECRET_KEY: 2ad2be6b13a1e4094af9b16199729a91fb
DEVISE_PEPPER: 2ad2be6b13a1e4094af9b16199729a91fbf3
BITLY_ACCESS_TOKEN: adf23jkl234ij3hj64j356nj4lk356h
APP_HOST: localhost
```

Listing 1: application.yml

3.3 Working

Required processes:

1. Rails Server: `bin/rails server`
2. Optionally: Webpack Dev Server `bin/webpack-dev-server` for hot reloading and continuous builds
3. Optionally: Sidekiq: `bin/bundle exec sidekiq -C config/sidekiq.yml` to have a production queue worker

Using a tool like Foreman or Procodile to start all services at once is also possible. Processes are specified in Procfile.
Before running a new database variation, call `bin/rails db:migrate`.
The tool is tested using RSpec and coverage is set to 100%, counting unit tests.