BEST

*boosting environmental
and social topic*

BILLBO / TECHNICAL INSIGHTS

HOW TO USE AND SPREAD BILLBO FOR BEST COMMUNITY BUILDING IN SOCIAL PROJECTS, ACTIVITIES, EVENTS + INTERVENTIONS

Contracting Authority: Starkmacher e.V.
Product Owner: Anja Roth
Application Development: Leon Roth

With the support of the
Erasmus+ Programme
of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

TECHNICAL INSIGHTS

Technologies

Billbo's code base utilises state-of-the-art web technologies, libraries and software packages to ensure an easy maintenance and code hygene and also to enable a smooth future feature development process.
The server side code as well as the frontend code relies on Typescript, Microsoft's static typing addition, which gets transpiled to Javascript in the build process of the development server, which is included in the project's code base and also in the production deployment pipeline. This implements the "Javascript everywhere" paradigm envisioned by Node.js's creator Ryan Dahl which results in a unified development experience and makes best use of the developers proficieny with Javascript.

On the server side, Billbo exposes a Node.js REST API which requires authentication via Json Web Tokens (JWT). The API uses the Express library to group REST calls into domain scoped controllers, which handle http requests based on the domain of the requested data. Connection to the database is handled by Mongoose, a package that implements a MongoDB client for a Node.js environment.

The frontend uses React v16+ with it's support for functional components and hooks. React is a frontend library created by Facebook and to this date, is one of the most loved libraries used by frontend developers. The frontend handles the http requests made to Billbo's backend and manages that data along with all state management using Rematch, a package that builds on top of Redux, a design pattern to handle frontend state and data.

Billbo uses Material UI react components to build a seamless and intuitive frontend to display information in a low-threshold, concise way with a beautiful user experience.

The Node.js server and MongoDB run inside of Docker containers to ensure portability and reproducability of the environment regardless of hosting or development choices.

Architecture

Billbo's invite-only user policy is implemented by a multitenancy pattern, which allows for creation and scoping of tenants, each encapsuling a certain use case of Billbo. Users can be part of several tenants and are provided with a tenant switcher in the UI, to quickly switch the context. These tenants can be managed and customized by users with a certain role.

Billbo implements several user roles to handle authentication for their respective data resources and actions permitted on the client. These roles are reflected via an express route decorator that evaluates and authorizes an incoming users roles.

The three roles currently implemented are "admin", "eco" and "user". Admins are technical administrators capable of accessing and changing all tenants. This is a technical role and is not used in day-to-day handling of Billbo. Ecos are designated users in charge of managing tenants and it's users and appearance in the client. Users are capable of accessing the tenants they were invited to and can also customize their own user profile.

Hosting

Billbo's backend infrastructure is hosted in a DigitalOcean droplet, which is a secure virtual machine in the cloud. The droplet can be scaled horizontally and vertically depending on acute requirements, which makes it a very flexible and reliable choice.

Inside of the droplet runs Dokku, a PaaS implementation to handle the management of Docker containers. Billbo is running two Docker containers, one for the Node.js server and one for the Mongo Database. Dokku provides a rich CLI for handling container lifecycle commands and deployment actions.

Billbo's UI is hosted on Netlify, a static site cloud host with high uptime.

DX (Developer experience)

Since Billbo uses Docker, a local version of Billbo can be started with minimal setup, creating a pleasant developer experience. Docker containers encapsule software requirements and environments needed to run a specified set of instructions. These are platform-agnostic, so the setup can run regardless of the developers operating system of choice. Starting the backend docker containers on a local machine will automatically spin up a development Node.js server. The development server for the frontend can be started after installing it's package dependencies via npm, Node.js's package managing tool.

Changes to the code base of Billbo are handled with Github, the leading code versioning platform, designed to make code collaboration as seamless as possible. Access to the repository on Github is granted while onboarding the project.

How to

The deployment process of Billbo can vary depending on whether the user chooses to deploy it with Docker or with MongoDB and NodeJS. Here are the general steps to get Billbo up and running:

1. Purchase and prepare a server and a domain
2. Install software: MongoDB + NodeJS or Docker
3. Start database, backend and frontend

After completing these steps, Billbo should be up and running on the server, and can be accessed through the domain name. Note that the exact commands and steps may vary depending on the specific deployment environment and configuration.

MIT License

BEST

boosting environmental and social topics

STARK macher

KU LEUVEN

AMU Portugal

AIPEC

With the support of the Erasmus+ Programme of the European Union

The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.