CrowdShelf server

Server for the CrowdShelf apps, written with NodeJS and Express.

Workflow

We're using JIRA for issue tracking. We try to follow Chrockfords style guide (http://javascript.crockford.com/code.html) for our code.

Do you want to contribute? Do it like this:

1. Pick a task to do on JIRA, or just something you want implemented yourself.
2. On JIRA, assign the task to yourself and move it to "In progress".
3. Create a new branch from master with a name that starts with the JIRA task name, and is followed by a short description. For example cs-32-user-management-api.
4. Work, work. Commit often. Try to follow good commit practice (http://chris.beams.io/posts/git-commit/). You can push your branch to GitHub if you want.
5. When you're done, push your latest changes and create a pull request. Wait for someone other than yourself to give feedback etc. Work out any differences of opinion, and the pull request will be merged and deployed.

Any questions? Feel free to ask!

Installation

Get a local server by doing the following:

1. Clone the repository with git@github.com:CrowdShelf/server.git
2. cd into the folder
3. Run npm install to install dependencies
4. Set up Mongodb (https://www.mongodb.org/) locally and export an environment variable MONGODB that defines its URL.
5. You'll also need a SMTP-server. You can set this up locally or wherever you want, then export the following environment variables: EMAIL_HOST, EMAIL_PORT, EMAIL_ADDRESS, EMAIL_USER and EMAIL_PASS.
6. Run npm start and you'll be up and running on port 3000, or an environment defined as PORT.

Project structure

```
server\
  controllers\ - Controllers for handling requsts
  models\ - Models that handle database-operations
  helpers\ - Helper-services with methods used throughout the solution
  index.js - Starts a server
  server.js - Sets the server up, defines port etc.
  router.js - Sets the routing up by setting allowed headers etc. and calling API setup-methods
```

Deployment

The server is currently running on a droplet on Digital Ocean. It has CI with the master-branch. You'll find it on crowdshelf.xyz/api.

There's also CI on the dev-branch, which is the latest version of the API. You'll find it on crowdshelf-dev.herokuapp.com/api

Docker

A Docker-image is avaiable on the Docker Hub (https://hub.docker.com/r/crowdshelf/server/). You can get it with docker pull crowdshelf/server, and run it with
docker run --net host -e "EMAIL_HOST=yourhost" -e "EMAIL_HOST=yourport" -e "EMAIL_USER=youruser" -e "EMAIL_PASS=yourpass" -e "EMAIL_ADDRESS=adress to send from" crowdshelf/server. The -e-parameter sets the environmental variables for your e-mail serivce. This will expose the ports 3000 and 27017 locally, which means you can reach the API under localhost:3000/api.

You can also build the image yourself with docker build -t crowdshelf/server .

Data model and API

Latest API definition can be found on our servers:

- Latest - dev (http://crowdshelf-dev.herokuapp.com/api)
- Stable - master (http://crowdshelf.xyz/api)

Licence