CCChat Documentation

Leonid Immel and Oliver Schneider

17. März 2018

Inhaltsverzeichnis

1 Node Modules (Serverside)

Module	Benefit
express	Web application framework that includes several useful features for web applications.
passport	Authentication middleware for session management.
socket.io	Realtime event-based bidirectional communication.
winston	Logger.
body-parser	Handles http post requests. Extracts body of http to req.body.
serve-favicon	Handles the requests from browsers for the favicon and caches the favicon.
handlebars	Builds semantic templates
mongoose	For MongoDB modeling
mocha	Testframework

2 Vendor Libraries (Clientside)

Library	Benefit
bootstrap	Responsive website programming for mobile-first implementation.
reactjs	Helps to create user interfaces

3 Project Structure

−− app

   −− public (all static served files)

      −− js

      −− css

      −− html

      −− libs

–– views (handlebar views)

–– models (handles data, logic and storage of the app)

–– controllers (defines the routes within the app)

–– middleware (processes incoming requests before they are passed to the routes (loginauthentication))

–– helpers (functionality and code shared in the whole app (regex- or renderoperations))

–– tests (all tests for the application)

4   Rules in documentation and formatting

4.1   Documentation

modules The functionality of modules is described within a comment on top of the js-file.

functions Functions with complex implementations or unclear names (should never happen!) need a commentation
Simpler functions are not commented to produce more readable code

variables Variables require clear and good naming, never comment a variable unless absolutely necessary.
You should think a littel longer and find a better name.

5   Logger

6   Error Handling

7   Useful websites

- Simple chat with socket.io
- Handlebars
- Combine Bootstrap and ReactJS

- Mongoose

- Example of secure chat in production