

# CCChat Documentation

Leonid Immel and Oliver Schneider

17. März 2018

## Inhaltsverzeichnis

<b>1</b>	<b>Node Modules (Serverside)</b>	<b>2</b>
<b>2</b>	<b>Vendor Libraries (Clientside)</b>	<b>2</b>
<b>3</b>	<b>Project Structure</b>	<b>2</b>
<b>4</b>	<b>Documentation</b>	<b>3</b>
<b>5</b>	<b>Formatting</b>	<b>3</b>
<b>6</b>	<b>SSL</b>	<b>3</b>
6.1	Create certificate . . . . .	3
6.2	Allow Certificate in browser for Dev . . . . .	4
6.2.1	Chromium / Chrome 64 . . . . .	4
6.2.2	Firefox 59 . . . . .	4
<b>7</b>	<b>Logger</b>	<b>5</b>
<b>8</b>	<b>Error Handling</b>	<b>5</b>
<b>9</b>	<b>Useful websites</b>	<b>5</b>

## 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.
jQuery	Easier handling of javascript

## 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 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 little longer and find a better name.

## 5 Formatting

We use the Eclipse JS Formatter standard for our code.

## 6 SSL

### 6.1 Create certificate

---

```
openssl genrsa -out client-key.pem 2048
openssl req -new -key client-key.pem -out client.csr
openssl x509 -req -in client.csr -signkey client-key.pem -out client-cert.pem
```

---

## 6.2 Allow Certificate in browser for Dev

### 6.2.1 Chromium / Chrome 64

Option 1:

1. Go to Settings. Hit the „Advanced“ button on bottom of the page
2. Click the „Manage certificates“ button in the „Privacy and security“ Tab
3. Click the „Authorities“ Tab
4. Click on „Import“
5. Navigate to the cert file within your app source code folder.
6. Click open and allow what you want to allow

Option 2:

1. Open the side explicitly with the url „https://localhost:3000“
2. Click the advanced Button
3. Allow the certificate
4. Continue with Finish (see a little further down)
5. follow option 2 until point 3.
6. find your certificate in the list of authorities.
7. click expand on the certificate
8. click the 3 dots that appear under the certificate
9. click „edit“
10. allow what you want to allow

### 6.2.2 Firefox 59

1. Open the side explicitly with the url „https://localhost:3000“
2. Click the „Advanced“ Button
3. Click „Add Exception“
4. Click „Confirm Security Exception“

## 7 **Logger**

## 8 **Error Handling**

## 9 **Useful websites**

- [Simple chat with socket.io](#)
- [Handlebars](#)
- [Combine Bootstrap and ReactJS](#)
- [Mongoose](#)
- [Example of secure chat in production](#)
- [Add a SSL-Cert in Chrome](#)