

Babble - Project Requirements

Functionality

- Web-based responsive chat
- One room
- Content open to everyone
- User can type a new message and submit to room
- New messages appear at the bottom
- User can delete his own messages
- User is defined by email address
- Avatars should be displayed automatically for email address
- First load should display a modal popup to collect email address
- User can discard and continue as anonymous
- Non-persistent (no db - all messages erased after server restart)
- Client-side persistence:
 - Typed text in new message textarea
 - Email address / preference to be anonymous
- Chat-room stats:
 - Number of connected users
 - Number of messages

Implementation

- Server in NodeJS
- Real-time updates with long polling: <http://book.mixu.net/node/ch3.html>
- Automatic avatars with Gravatar: <https://en.gravatar.com/site/implement/profiles/json/>
 - MD5 hash should be implemented on the server
- Client-side persistence with LocalStorage:
https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API
 - **exact** (one) LocalStorage key: `babble` with these properties
 - `currentMessage:String`
 - `userInfo:Object`
 - `name:String`
 - `email:String`
- No JS libraries (e.g. jQuery, Angular, etc.)
- No CSS frameworks (e.g. Bootstrap, etc.)
- No IDs should be used for CSS styling
- Exactly **one** JS global variable allowed: `Babble`
- **All** class names should adhere to the SUIT CSS naming convention:
<https://github.com/suitcss/suit/blob/master/doc/naming-conventions.md>

Client-Server API (RESTful, see [link](#))

- All requests and responses are in JSON (not url-encoded)

- GET /messages?counter=XX (like /poll in the article)
- POST /messages (new message)
 {name:String, email:String, message:String, timestamp:Number(ms)} (request body)
- DELETE /messages/:id (delete one message)
- GET /stats (get statistics)
 {users:Number, messages:Number} (response body)
- Server should handle these errors and send appropriate status code:
 - 404 - for non-existent URLs (not found)
 - e.g. GET /moshe
 - 400 - when the sent data is bad (bad request)
 - e.g. GET /messages?shuki=5
 - e.g. GET /messages?counter=hello
 - 405 - when the HTTP method is bad for certain URL (method not allowed)
 - e.g. POST /stats
 - 204 - for OPTIONS request (sometimes sent automatically by the browser)

Client API

- You should implement these methods:
- `Babble.register(userInfo:Object)`
- `Babble.getMessages(counter:Number, callback:Function)`
- `Babble.postMessage(message:Object, callback:Function)`
- `Babble.deleteMessage(id:String, callback:Function)`
- `Babble.getStats(callback:Function)`
- See unit tests for more info

Server API

- You should implement a module to handle messages: `server/messages-util.js`
- `messages.addMessage(message:Object) : Number(id)`
- `messages.getMessages(counter:Number) : Array(messages)`
- `messages.deleteMessage(id:String)`
- You should use this module in your server code to separate concerns from the client-server API handling logic in `server/main.js`
- You can add additional modules if needed to handle additional concerns

Unit Tests

- Your code should pass unit tests, on the client and server.
- Tests will test the above API
- Use the provided tests in your code, **as is, don't alter**
- Add additional tests for additional code you write, but don't change the above API (you can add more methods if you need)
- In addition to [Mocha](#) tests will use [Chai](#) (assertion) and [Sinon](#) (spies).
- Client tests: https://hu-mar-17.slack.com/files/krulik/F6B4QNWBA/test_client_test.js.js
- Server tests: https://hu-mar-17.slack.com/files/krulik/F6BRMDR60/test_server_test.js.js

Desktop Design

- PNG: <https://drive.google.com/open?id=0B42daK8eCfktZzJvdXNBQWVsSVE>
- PSD:
<https://drive.google.com/open?id=0B42daK8eCfktalo3aUtmN1RDNE5Wa1JCcG93SzRGZGM1TElr>
- Fluid: takes full width and height regardless of screen size
- Only scroll is in the messages area (vertical)
- New message text area is stick to bottom
- New message text area enlarges (grows from bottom to top) as needed to accommodate the text inside until a max of 300px and then an inner scrollbar appears
- Message width shrink-wraps to text width with a max width of 600px
- Focused/hovered message has a gray background and the delete button appears
- Chat-room name is aligned left
- Chat-room stats are aligned right
- Anonymous user design -
<https://drive.google.com/open?id=0B42daK8eCfktZzJvdXNBQWVsSVE>
- User register modal design -
<https://drive.google.com/open?id=0B42daK8eCfktVmRvNzdYYVVkREE>

Mobile Design

- PSD:
<https://drive.google.com/open?id=0B42daK8eCfktZTBTWWJHSjN4QXISQWtKS05qYkFpeTNLdktB>
- Mobile breakpoints (mobile first)
 - 320 (0-320)
 - 638 (320-638, 638-?)

HTML5 Semantics/Usability/Accessibility

- Page/tab title should be "Vegans - Babble"
- URL should be <http://username.github.io/babble/#/vegans>
- New message should be a textarea element
- Submit image should be a button element with an aria-label attribute
- textarea and submit button should be inside an HTML form with relevant action which will be submitted upon button click/enter
- Logo is an img
- Header text is an h1
- Logo and h1 is in a header element
- Left area (green) is an empty nav element
- Right area is inside a main element
- Message list is an ordered list
- User images should have an empty alt
- User names should be a cite element
- Message time should be a time element with correct datetime attribute (UNIX time)

- Each message should be focusable with the tab key
- Delete button for each message should be focusable with the tab key
- Delete icon should be a button element with an `aria-label` attribute
- Chat-room stats should be inside a definition list (with the terms hidden):
<http://devdocs.io/html/element/dl>
- Hidden content should be hidden in an accessible way:
 - <http://a11yproject.com/posts/how-to-hide-content/>
 - <http://webaim.org/techniques/css/invisiblecontent/>

Standards and Validation

- HTML5 doctype + UTF8 charset
(https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5/Introduction_to_HTML5)
- Valid HTML5 (zero errors: <https://validator.w3.org/>)
- Accessible (zero errors: <http://wave.webaim.org/>)
- Logical heading structure (zero outline orphans: <https://validator.w3.org/> (show outline))
- No JS errors (DevTools console) (**without** global onerror handler)
- Should pass all unit tests - see relevant section
- 80/100 or higher Google PageSpeed score
(<https://developers.google.com/speed/pagespeed/insights/>)
- Mobile optimized
 - Viewport meta tag present with width=device-width
(https://developer.mozilla.org/en/docs/Mozilla/Mobile/Viewport_meta_tag)
 - Initial scale 100% (no automatic zoom out)
 - **Don't** disable pinch zoom
 - No horizontal scroll on **any** screen size

Submission

- Github username in [Contacts List](#) should be present and valid (attention: case **sensitive**)
- **Exact** repository name: babble
- Folder structure
 - /babble
 - /client
 - /styles
 - main.css
 - /images
 - ...
 - /scripts
 - main.js
 - index.html
 - /server
 - main.js
 - messages-util.js
 - /test
 - /client

- test.js
 - index.html
 - /server
 - test.js
- *NOTE: a code similarity checker will be used*
- Should be able to start your project with `npm start`
 - The server will start on port 9000
 - The client will start on port 8080
- Should be able to test your project with `npm test`
 - Server tests will be run by mocha
 - Client tests will be available in `http://localhost:8081/test/client/`